Assessing Deduplication and Data Linkage Quality: What to Measure?

http://datamining.anu.edu.au/linkage.html

Peter Christen[*] and Karl Goiser

Department of Computer Science,
Australian National University,
Canberra ACT 0200, Australia
{peter.christen,karl.goiser}@anu.edu.au

Abstract. Deduplicating one data set or linking several data sets are increasingly important tasks in the data preparation steps of many data mining projects. The aim of such linkages is to match all records relating to the same entity. Research interest in this area has increased in recent years, with techniques originating from statistics, machine learning, information retrieval, and database research being combined and applied to improve the linkage quality, as well as to increase performance and efficiency when deduplicating or linking very large data sets. Different measures have been used to characterise the quality of data linkage algorithms. This paper presents an overview of the issues involved in measuring deduplication and data linkage quality, and it is shown that measures in the space of record pair comparisons can produce deceptive accuracy results. Various measures are discussed and recommendations are given on how to assess deduplication and data linkage quality.

Keywords: data or record linkage, data integration and matching, deduplication, data mining pre-processing, quality measures.

1 Introduction

With many businesses, government organisations and research projects collecting massive amounts of data, data mining has in recent years attracted interest both from academia and industry. While there is much ongoing research in data mining algorithms and techniques, it is well known that a large proportion of the time and effort in real-world data mining projects is spent understanding the data to be analysed, as well as in the data preparation and pre-processing steps (which may well dominate the actual data mining activity). An increasingly important task in data pre-processing is detecting and removing duplicate records that relate to the same entity within one data set. Similarly, linking or matching records relating to the same entity from several data sets is often required, as information from multiple sources needs to be integrated, combined or linked in

[*] Corresponding author

order to allow more detailed data analysis or mining. The aim of such linkages is to match all records relating to the same entity, such as a patient, a customer, a business, a consumer product, or a genome sequence.

Deduplication and data linkage can be used to improve data quality and integrity, to allow re-use of existing data sources for new studies, and to reduce costs and efforts in data acquisition. In the health sector, for example, deduplication and data linkage have traditionally been used for cleaning and compiling data sets for longitudinal or other epidemiological studies [23]. Linked data might contain information that is needed to improve health policies, and which traditionally has been collected with time consuming and expensive survey methods. Statistical agencies routinely link census data [18, 37] for further analysis. Businesses often deduplicate and link their data sets to compile mailing lists, while within taxation offices and departments of social security, data linkage and deduplication can be used to identify people who register for benefits multiple times or who work and collect unemployment benefits. Another application of current interest is the use of data linkage in crime and terror detection. Security agencies and crime investigators increasingly rely on the ability to quickly access files for a particular individual, which may help to prevent crimes by early intervention.

The problem of finding similar entities doesn't only apply to records which refer to persons. In bioinformatics, data linkage helps to find genome sequences in large data collections that are similar to a new, unknown sequence at hand. Increasingly important is the removal of duplicates in the results returned by Web search engines and automatic text indexing systems, where copies of documents – for example bibliographic citations – have to be identified and filtered out before being presented to the user. Comparing consumer products from different online stores is another application of growing interest. As product descriptions are often slightly different, comparing them becomes difficult.

If unique entity identifiers (or keys) are available in all the data sets to be linked, then the problem of linking at the entity level becomes trivial: a simple database join is all that is required. However, in most cases no unique keys are shared by all of the data sets, and more sophisticated data linkage techniques need to be applied. An overview of such techniques is presented in Section 2. The notation used in this paper, and a problem analysis are discussed in Section 3, before a description of various quality measures is given in Section 4. A real-world example is used in Section 5 to illustrate the effects of applying different quality measures. Finally, several recommendations are given in Section 6, and the paper is concluded with a short summary in Section 7.

2 Data Linkage Techniques

Computer-assisted data linkage goes back as far as the 1950s. At that time, most linkage projects were based on *ad hoc* heuristic methods. The basic ideas of probabilistic data linkage were introduced by Newcombe and Kennedy [30] in 1962, and the theoretical statistical foundation was provided by Fellegi and Sunter [16] in 1969. Similar techniques have independently been developed in the 1970s by computer scientists in the area of document indexing and retrieval [13]. However, until recently few cross-references could be found between the statistical and the computer science community.

As most real-world data collections contain noisy, incomplete and incorrectly formatted information, data cleaning and standardisation are important pre-processing steps for successful deduplication and data linkage, and before data can be loaded into data warehouses or used for further analysis [33]. Data may be recorded or captured in various, possibly obsolete, formats and data items may be missing, out of date, or contain errors. Names and addresses can change over time, and names are often reported differently by the same person depending upon the organisation they are in contact with. Additionally, many proper names have different written forms, for example *'Gail'* and *'Gayle'*. The main tasks of data cleaning and standardisation are the conversion of the raw input data into well defined, consistent forms, and the resolution of inconsistencies [7, 9].

If two data sets **A** and **B** are to be linked, the number of possible record pairs equals the product of the size of the two data sets $|\mathbf{A}| \times |\mathbf{B}|$. Similarly, when deduplicating a data set **A** the number of possible record pairs is $|\mathbf{A}| \times (|\mathbf{A}| - 1)/2$. The performance bottleneck in a data linkage or deduplication system is usually the expensive detailed comparison of fields (or attributes) between pairs of records [1], making it unfeasible to compare all record pairs when the data sets are large. For example, linking two data sets with $100,000$ records each would result in ten billion possible record pair comparisons. On the other hand, the maximum number of truly matched record pairs that are possible corresponds to the number of records in the smaller data set (assuming a record can only be linked to one other record). For deduplication, the number of duplicate records will be smaller than the number of records in the data set. The number of potential matches increases linearly when linking larger data sets, while the computational efforts increase quadratically.

To reduce the large number of possible record pair comparisons, data linkage systems therefore employ blocking [1, 16, 37], sorting [22], filtering [20], clustering [27], or indexing [1, 5] techniques. Collectively known as *blocking*, these techniques aim at cheaply removing pairs of records that are obviously not matches. It is important, however, that no potential match is removed by blocking.

All record pairs produced in the blocking process are compared using a variety of field (or attribute) comparison functions, each applied to one or a combination of record attributes. These functions can be as simple as an exact string or a numerical comparison, can take into account typographical errors, or be as complex as a distance comparison based on look-up tables of geographic locations (longitude and latitude). Each comparison returns a numerical value, often positive for agreeing values and negative for disagreeing values. For each compared record pair a *weight vector* is formed containing all the values calculated by the different field comparison functions. These weight vectors are then used to classify record pairs into *matches*, *non-matches*, and *possible matches* (depending upon the decision model used). In the following sections the various techniques employed for data linkage are discussed in more detail.

2.1 Deterministic Linkage

Deterministic linkage techniques can be applied if unique entity identifiers (or keys) are available in all the data sets to be linked, or a combination of attributes can be used to create a *linkage key*, which is then used to match records that have the same key value. Such linkage systems can be developed based on standard *SQL* queries. However, they only achieve good linkage results if the entity identifiers or linkage keys are of high quality. This means they have to be precise, stable over time, highly available, and robust with regard to errors (for example, include a check digit for detecting invalid or corrupted values).

Alternatively, a set of (often very complex) rules can be used to classify pairs of records. Such rule-based systems can be more flexible than using a simple linkage key, but their development is labour intensive and highly dependent upon the data sets to be linked. The person or team developing such rules not only needs to be proficient with the rule system, but also with the data to be deduplicated or linked. In practise, therefore, deterministic rule based systems are limited to ad-hoc linkages of smaller data sets. In a recent study [19], an iterative deterministic linkage system was compared with the commercial probabilistic system *AutoMatch* [25], and empirical results showed that the probabilistic approach achieved better linkages.

2.2 Probabilistic Linkage

As common unique entity identifiers are rarely available in all data sets to be linked, the linkage process must be based on the existing common attributes. These normally include person identifiers (like names and dates of birth), demographic information (like addresses) and other data specific information (like medical details, or customer information). These attributes can contain typographical errors, they can be coded differently, and parts can be out-of-date or even be missing.

In the traditional probabilistic linkage approach [16, 37], pairs of records are classified as matches if their common attributes predominantly agree, or as non-matches if they predominantly disagree. If two data sets $\mathbf{A}$ and $\mathbf{B}$ are to be linked, the set of record pairs $\mathbf{A} \times \mathbf{B} = \{(a, b);\ a \ \varepsilon\ \mathbf{A},\ b \ \varepsilon\ \mathbf{B}\}$ is the union of the two disjoint sets of true matches M and true non-matches U.

$$M = \{(a, b);\ a = b,\ a \ \varepsilon\ \mathbf{A},\ b \ \varepsilon\ \mathbf{B}\} \tag{1}$$

$$U = \{(a, b);\ a \neq b,\ a \ \varepsilon\ \mathbf{A},\ b \ \varepsilon\ \mathbf{B}\} \tag{2}$$

Fellegi and Sunter [16] considered ratios of probabilities of the form

$$R = \frac{P(\gamma \ \varepsilon\ \mathit{\Gamma} | M)}{P(\gamma \ \varepsilon\ \mathit{\Gamma} | U)}, \tag{3}$$

where γ is an arbitrary agreement pattern in a comparison space $\mathit{\Gamma}$. For example, $\mathit{\Gamma}$ might consist of six patterns representing simple agreement or disagreement on given name, surname, date of birth, street address, suburb and postcode.

Alternatively, some of the γ might additionally consider typographical errors, or account for the relative frequency with which specific values occur. For example, a surname value *'Miller'* is much more common in many western countries than a value *'Dijkstra'*, resulting in a smaller agreement value. The ratio R, or any monotonically increasing function of it (such as its logarithm) is referred to as a *matching weight*. A decision rule is then given by

if $R > t_{upper}$, then designate a record pair as *match*,
if $t_{lower} \leq R \leq t_{upper}$, then designate a record pair as *possible match*,
if $R < t_{lower}$, then designate a record pair as *non-match*.

The thresholds t_{lower} and t_{upper} are determined by a-priori error bounds on false matches and false non-matches. If $\gamma \; \varepsilon \; \Gamma$ for a certain record pair mainly consists of agreements, then the ratio R would be large and thus the pair would more likely be designated as a match. On the other hand for a $\gamma \; \varepsilon \; \Gamma$ that primarily consists of disagreements the ratio R would be small.

The class of possible matches are those record pairs for which human oversight, also known as *clerical review*, is needed to decide their final linkage status. While in the past (when smaller data sets were linked, for example for epidemiological survey studies) clerical review was practically manageable in a reasonable amount of time, linking today's large data collections – with millions of records – make this process impossible, as tens or even hundreds of thousands of record pairs will be put aside for review. Clearly, what is needed are more accurate and automated decision models that will reduce – or even eliminate – the amount of clerical review needed, while keeping a high linkage quality. Such approaches are presented in the following section.

2.3 Modern Approaches

Improvements [38] upon the classical probabilistic linkage [16] approach include the application of the expectation-maximisation (EM) algorithm for improved parameter estimation [39], the use of approximate string comparisons [32] to calculate partial agreement weights when attribute values have typographical errors, and the application of Bayesian networks [40].

In recent years, researchers have also started to explore the use of techniques originating in machine learning, data mining, information retrieval and database research to improve the linkage process. Most of these approaches are based on supervised learning techniques and assume that training data (i.e. record pairs with known deduplication or linkage status) is available.

One approach based on ideas from information retrieval is to represent records as document vectors and compute the *cosine distance* [10] between such vectors. Another possibility is to use an *SQL* like language [17] that allows approximate joins and cluster building of similar records, as well as decision functions that decide if two records represent the same entity. A generic knowledge-based framework based on rules and an expert system is presented in [24], and a hybrid system which utilises both unsupervised and supervised machine learning techniques is described in [14]. That paper also introduces metrics for determining the quality of these techniques. The authors find that machine learning outperforms probabilistic techniques, and provides a lower proportion of possible matches.

The authors of [35] apply active learning to the problem of lack of training instances in real-world data. Their system presents a representative (difficult to classify) example to a user for manual classification. They report that manually classifying less than 100 training examples provided better results than a fully supervised approach that used 7,000 randomly selected examples. A similar approach is presented in [36], where a committee of decision trees is used to learn mapping rules (i.e. rules describing linkages).

High-dimensional overlapping clustering (as alternative to traditional blocking) is used by [27] in order to reduce the number of record pair comparisons to be made, while [21] explore the use of simple k-means clustering together with a user tunable fuzzy region for the class of possible matches. Methods based on nearest neighbours are explored by [6], with the idea to capture local structural properties instead of a single global distance approach. An unsupervised approach based on graphical models [34] aims to use the structural information available in the data to build hierarchical probabilistic models. Results which are better than the ones achieved by supervised techniques are presented.

Another approach is to train distance measures used for approximate string comparisons. [3] presents a framework for improving duplicate detection using trainable measures of textual similarity. The authors argue that both at the character and word level there are differences in importance of certain character or word modifications, and accurate similarity computations require adapting string similarity metrics for all attributes in a data set with respect to the particular data domain. Related approaches are presented in [5, 12, 29, 41], with [29] using support vector machines for the binary classification task of record pairs. As shown in [12], combining different learned string comparison methods can result in improved linkage classification. An overview of other methods – including statistical outlier identification, pattern matching, and association rules based approaches – is given in [26].

3 Notation and Problem Analysis

The notation used in this paper is presented here. It follows the traditional data linkage literature [16, 37, 38]. The number of elements in a set $\mathbf{X}$ is denoted $|\mathbf{X}|$. A general linkage situation is assumed, where the aim is to link two sets of entities. For example, the first set could be patients of a hospital, and the second set people who had a car accident. Some of the car accidents resulted in people being admitted into the hospital, some did not. The two sets of entities are denoted as $\mathbf{A}_e$ and $\mathbf{B}_e$. $\mathbf{M}_e = \mathbf{A}_e \cap \mathbf{B}_e$ is the intersection set of matched entities that appear in both $\mathbf{A}_e$ and $\mathbf{B}_e$, and $\mathbf{U}_e = (\mathbf{A}_e \cup \mathbf{B}_e) \setminus \mathbf{M}_e$ is the set of non-matched entities that appear in either $\mathbf{A}_e$ or $\mathbf{B}_e$, but not in both. This space of entities is illustrated in Figure 1, and called the *entity space*.

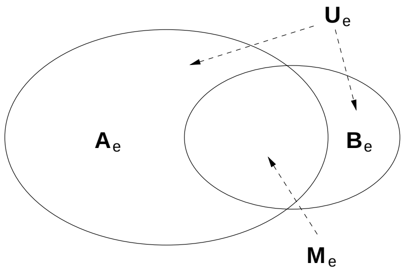

Fig. 1. General linkage situation with two sets of entities $\mathbf{A}_e$ and $\mathbf{B}_e$, their intersection $\mathbf{M}_e$ (the entities that appear in both sets), and the set $\mathbf{U}_e$ which contains the entities that appear in either $\mathbf{A}_e$ or $\mathbf{B}_e$, but not in both

The maximum possible number of matched entities corresponds to the size of the smaller set of $\mathbf{A}_e$ or $\mathbf{B}_e$. This is the situation when the smaller set is a proper subset of the larger one, which also results in the minimum number of non-matched entities. The minimum number of matched entities is zero, which is the situation when no entities appear in both sets. The maximum number of non-matched entities in this situation corresponds to the sum of the entities in both sets. The following equations show this in a more formal way.

$$0 \leq |\mathbf{M}_e| \leq min(|\mathbf{A}_e|, |\mathbf{B}_e|) \quad (4)$$

$$abs(|\mathbf{A}_e| - |\mathbf{B}_e|) \leq |\mathbf{U}_e| \leq |\mathbf{A}_e| + |\mathbf{B}_e| \quad (5)$$

In a simple example, assume the set $\mathbf{A}_e$ contains 5 million entities (e.g. hospital patients), and set $\mathbf{B}_e$ contains 1 million entities (e.g. people involved in car accidents), with 700,000 entities present in both sets (i.e. $|\mathbf{M}_e| = 700,000$). The number of non-matched entities in this situation is $|\mathbf{U}_e| = 4,600,000$, which is the sum of the entities in both sets (6 millions) minus twice the number of matched entities (as they appear in both sets $\mathbf{A}_e$ and $\mathbf{B}_e$). This simple example will be used as a running example in the discussion below.

Records for the entities in $\mathbf{A}_e$ and $\mathbf{B}_e$ are now stored in two data sets (or databases or files), denoted by $\mathbf{A}$ and $\mathbf{B}$, such that there is exactly one record in $\mathbf{A}$ for each entity in $\mathbf{A}_e$ (i.e. the data set contains no duplicate records), and each record in $\mathbf{A}$ corresponds to an entity in $\mathbf{A}_e$. The same holds for $\mathbf{B}_e$ and $\mathbf{B}$. The aim of a data linkage process is to classify pairs of records as matches or non-matches in the product space $\mathbf{A} \times \mathbf{B} = M \cup U$ of true matches M and true non-matches U [16, 37] as given in Equations 1 and 2.

It is assumed that no blocking (as discussed in Section 2) is applied, and that all possible pairs of records are compared. The total number of comparisons equals $|\mathbf{A}| \times |\mathbf{B}|$, which is much larger than the number of entities available in $\mathbf{A}_e$ and $\mathbf{B}_e$ together. In case of the deduplication of a single data set $\mathbf{A}$, the number of record pair comparisons equals $|\mathbf{A}| \times (|\mathbf{A}| - 1)/2$, as each record in the data set must be compared with all others, but not to itself. The space of record pair comparisons is illustrated in Figure 2 and called the *comparison space*.

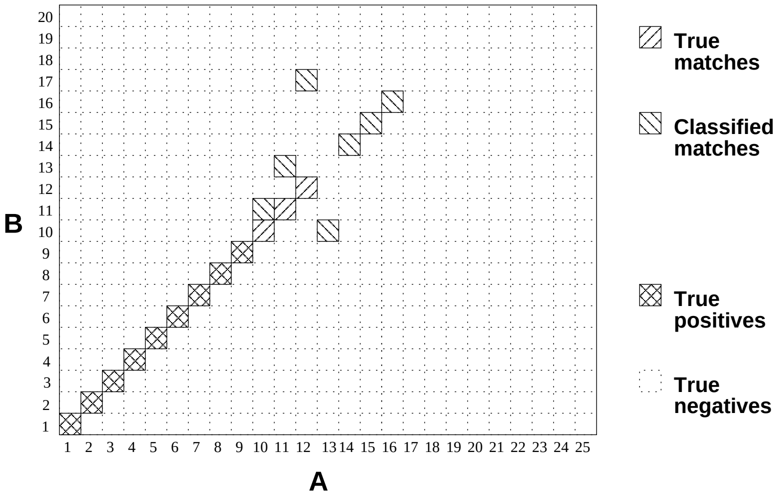

Fig. 2. General record pair comparison space with 25 records in data set **A** arbitrarily numbered on the horizontal axis and 20 records in data set **B** arbitrarily numbered on the vertical axis. The full rectangular area corresponds to all possible record pair comparisons. Assume that record pairs $(A1, B1)$, $(A2, B2)$ up to $(A12, B12)$ are true matches. The linkage algorithm has wrongly classified $(A10, B11)$, $(A11, B13)$, $(A12, B17)$, $(A13, B10)$, $(A14, B14)$, $(A15, B15)$, and $(A16, B16)$ as matches (false positives), but missed $(A10, B10)$, $(A11, B11)$, and $(A12, B12)$ (false negatives)

For the simple example given earlier, the comparison space consists of $|\mathbf{A}| \times |\mathbf{B}| = 5,000,000 \times 1,000,000 = 5 \times 10^{12}$ record pairs, with $|M| = 700,000$ and $|U| = 5 \times 10^{12} - 700,000 = 4.9999993 \times 10^{12}$ record pairs.

A linkage algorithm compares pairs of records and classifies them into $\tilde{M}$ (record pairs considered to be a match by the algorithm) and $\tilde{U}$ (record pairs considered to be a non-match). To keep this analysis simple, it is assumed here that the linkage algorithm does not classify record pairs as possible matches (as discussed in Section 2.2). Both records of a truly matched pair correspond to the same entity in $\mathbf{M}_e$. Un-matched record pairs, on the other hand, correspond to different entities in $\mathbf{A}_e$ and $\mathbf{B}_e$, with the possibility of both records of such a pair corresponding to different entities in $\mathbf{M}_e$. As each record relates to exactly one entity, and there are no duplicates in the data sets, a record in **A** can only be correctly matched to a maximum of one record in **B**, and vice versa. For each record pair, the binary classification into $\tilde{M}$ and $\tilde{U}$ results in one of four possible outcomes [15] as shown in Table 1. As can be seen, $M = TP + FN$, $U = TN + FP$, $\tilde{M} = TP + FP$, and $\tilde{U} = TN + FN$.

When assessing the quality of a linkage algorithm, the general interest is in how many truly matched entities and how many truly non-matched entities have been classified correctly as matches and non-matches, respectively. However, the outcome of the classification is measured in the comparison space (as number

Table 1. Confusion matrix of record pair classification

Actual	Classification Match ($\tilde{M}$)	 Non-match ($\tilde{U}$)
Match (M)	True match True positive (TP)	False non-match False negative (FN)
Non-match (U)	False match False positive (FP)	True non-match True negative (TN)

of classified record pairs). While the number of truly matched record pairs is the same as the number of truly matched entities, $|M| = |\mathbf{M}_e|$ (as each truly matched record pair corresponds to one entity), there is however no correspondence between the number of truly non-matched record pairs and non-matched entities. Each non-matched record pair contains two records that correspond to two different entities, and so it not possible to easily calculate a number of non-matched entities.

The maximum number of truly matched entities is given by Equation 4. From this follows the maximum number of record pairs a linkage algorithm should classify as matches is $|\tilde{M}| \leq |\mathbf{M}_e| \leq min(|\mathbf{A}_e|, |\mathbf{B}_e|)$. As the number of classified matches $\tilde{M} = TP + FP$, it follows that $|TP + FP| \leq |\mathbf{M}_e|$. And with $M = TP + FN$, it also follows that both the numbers of FP and FN will be small compared to the number of TN, and they will not be influenced by the multiplicative increase between the entity and the comparison space. The number of TN will dominate, however, as, in the comparison space, the following equation holds:

$$|TN| = |\mathbf{A}| \times |\mathbf{B}| - |TP| - |FN| - |FP|. \tag{6}$$

This is also illustrated in Figure 2. Therefore, any quality measure used in deduplication or data linkage that uses the number of TN will give deceptive results, as will be illustrated and discussed further in Sections 4 and 5.

The above discussion assumes no duplicates in the data sets **A** and **B**. Thus, a record in one data set can only be matched to a maximum of one record in the other data set (often called *one-to-one* assignment restriction). In practise, however, *one-to-many* and *many-to-many* linkages or deduplications are possible. Examples include longitudinal studies of administrative health data, where several records might correspond to a certain patient over time, or business mailing lists where several records can relate to the same customer (this happens when data sets have not been properly deduplicated). While the above analysis would become more complicated, the issue of having a very large number of TN stills hold in one-to-many and many-to-many linkage situations, as the number of matches for a single record will be small compared to the full number of record pair comparisons.

Table 2. Quality measures used in recent deduplication and data linkage publications

Measure	Formula / Description	Used in
Accuracy	$acc = \frac{TP+TN}{TP+FP+TN+FN}$	[21, 35, 36]
Precision	$prec = \frac{TP}{TP+FP}$	[1, 2, 10, 11, 14, 27]
Recall	$rec = \frac{TP}{TP+FN}$	[1, 11, 14, 21, 27]
F-measure	$f{-}measure = 2(\frac{prec \times rec}{prec+rec})$	[1, 11, 27]
False positive rate	$fpr = \frac{FP}{TN+FP}$	[2]
Precision-Recall graph	Plot precision on vertical and recall on horizontal axis	[3, 6, 28]

4 Quality Measures

Given that deduplication and data linkage are classification problems, various quality measures are available to the data linkage researcher and practitioner [15]. With many recent approaches being based on supervised learning, no clerical review process (i.e. no possible matches) is often assumed and the problem becomes a binary classification, with record pairs being classified as either matches or non-matches, as shown in Table 1. A summary of the quality measures used in recent publications is given in Table 2 (a more detailed discussion can be found in [8]).

As presented in Section 2.2, a linkage algorithm is assumed to have a threshold parameter t (with no possible matches $t_{lower} = t_{upper}$), which determines the cut-off between classifying record pairs as matches (with matching weight $R \geq t$) or as non-matches ($R < t$). Increasing the value of t results in an increased number of TN and FP and in a reduction in the number of TP and FN, while lowering t reduces the number of TN and FP and increases the number of TP and FN. Most of the quality measures presented here can be calculated for different values of such a threshold (often only the quality measure values for an optimal threshold are reported in empirical studies). Alternatively, quality measures can be visualised in a graph over a range of threshold values, as illustrated by the examples in Section 5.

Taking the example from Section 3, assume that for a given threshold a linkage algorithm has classified $|\tilde{M}| = 900,000$ record pairs as matches and the rest ($|\tilde{U}| = 5 \times 10^{12} - 900,000$) as non-matches. Of these $900,000$ classified matches $650,000$ were true matches (TP), and $250,000$ were false matches (FP). The number of false non-matched record pairs (FN) was $50,000$, and the number of true non-matched record pairs (TN) was $5 \times 10^{12} - 950,000$. When looking at the entity space, the number of non-matched entities is $4,600,000 - 250,000 = 4,350,000$. Table 3 shows the resulting quality measures for this example in both the comparison and the entity spaces, and as discussed, any measure that includes the number of TN depends upon whether entities or record pairs are counted. As can be seen, the results for accuracy and the false positive rate

Table 3. Quality results for the simple example

Measure	Entity space	Comparison space
Accuracy	94.340%	99.999994%
Precision	72.222%	72.222000%
Recall	92.857%	92.857000%
F-measure	81.250%	81.250000%
False positive rate	5.435%	0.000005%

all show misleading results when based on record pairs (i.e. measured in the comparison space). This issue will be illustrated further in Sections 5 and 6.

The authors of [4] discuss the topic of evaluating deduplication and data linkage systems. They advocate the use of precision-recall graphs over the use of single value measures like accuracy or maximum F-measure, on the grounds that such single value measures assume that an optimal threshold has been found. A single value can also hide the fact that one classifier might perform better for lower threshold values, while another better for higher thresholds.

5 Experimental Examples

In this section the previously discussed issues on quality measures are illustrated using a real-world administrative health data set, the *New South Wales Midwives Data Collection* (MDC) [31]. 175, 211 records from the years 1999 and 2000 were extracted, containing names, addresses and dates of birth of mothers giving birth in these two years. This data set has previously been deduplicated (and manually clerically reviewed) using the commercial probabilistic data linkage system *AutoMatch* [25]. According to this deduplication, the data set contains 166, 555 unique mothers, with 158, 081 having one, 8, 295 having two, 176 having three, and 3 having four records (births). The *AutoMatch* deduplication decision was used as the true match (or deduplication) status for this example

A deduplication was then performed using the *Febrl* (Freely extensible biomedical record linkage) [7] data linkage system. Fourteen attributes in the MDC were compared using various comparison functions (like exact and approximate string comparisons), and the resulting comparison values were summed into a matching weight (as discussed in Section 2.2) ranging from -43 (disagreement on all fourteen comparisons) to 115 (agreement on all comparisons). As can be seen in the density plot in Figure 3, almost all true matches (record pairs classified as true duplicates) have positive matching weights, while the majority of non-matches have negative weights. There are, however, non-matches with rather large positive matching weights, which is due to the differences in calculating the weights between *AutoMatch* and *Febrl*.

The full comparison space for this data set with 175, 211 records would result in $175,211 \times 175,210/2 = 15,349,359,655$ record pairs, which is infeasible

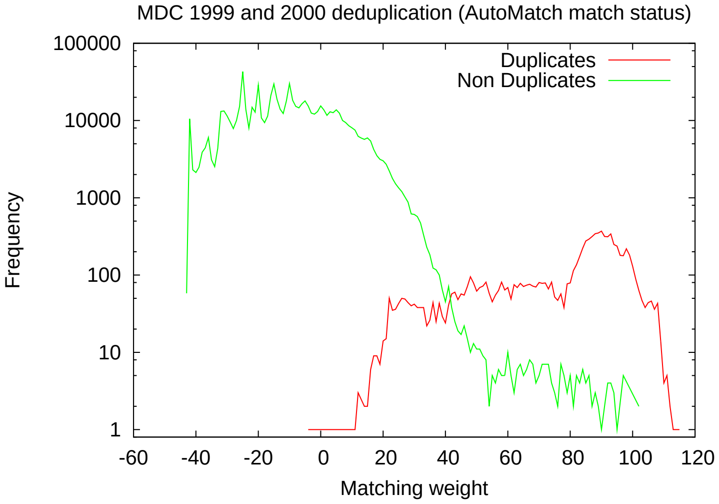

Fig. 3. The density plot of the matching weights for a real-world administrative health data set. This plot is based on record pair comparison weights in a blocked comparison space. The lowest weight is -43 (disagreement on all comparisons), and the highest 115 (agreement on all comparisons). Note that the vertical axis with frequency counts is on a logarithmic scale

to process even with today's powerful computers. Standard blocking was used to reduce the number of comparisons, resulting in $759,773$ record pairs (this corresponds to only around 0.005% of all record pairs in the full comparison space). The total number of truly classified matches (duplicates) was $8,841$ (for all the duplicates as described above), with $8,808$ of the $759,773$ record pairs in the blocked comparison space corresponding to true duplicates (thus, 33 true matches were removed by blocking).

The quality measures discussed in Section 4 applied to this real-world deduplication procedure are shown in Figure 4 for a varying threshold $-43 \leq t \leq 115$. The aim of this figure is to illustrate how the different measures look for a deduplication example taken from the real world. The measurements were done in the blocked comparisons space as described above. The full comparison space ($15,349,359,655$ record pairs) was simulated by assuming that blocking removed mainly record pairs with negative comparison weights (normally distributed between -43 and -10). As discussed previously, this resulted in different numbers of TN between the blocked and the (simulated) full comparison spaces. As can be seen, the precision-recall graph is not affected by the blocking process, and the F-measure differs only slightly. The two other measures, however, resulted in graphs of different shape.

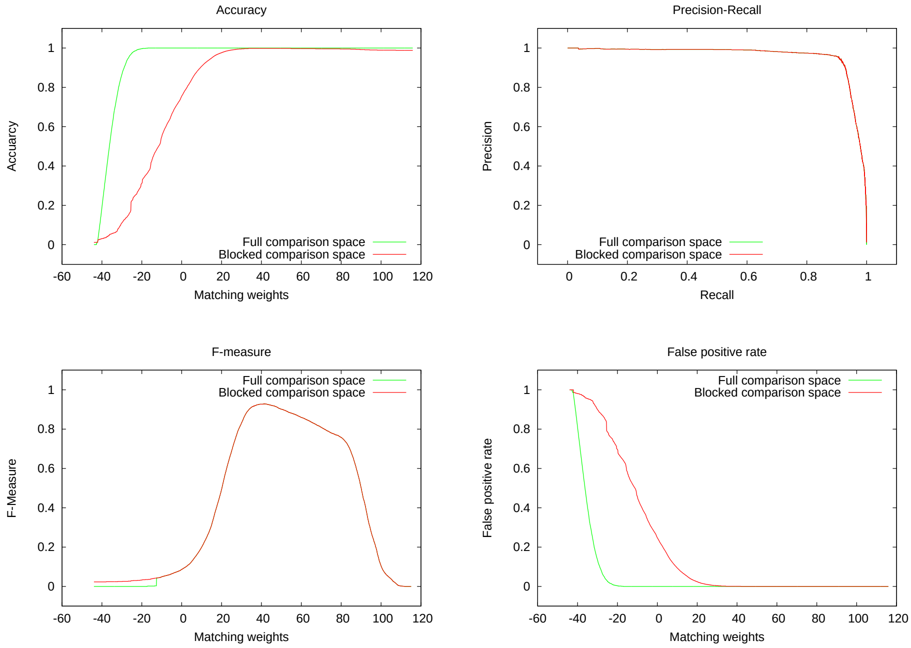

Fig. 4. Quality measurements of a real-world administrative health data set

6 Recommendations

Based on the above discussions, several recommendations for measuring deduplication and data linkage quality can be given. Their aim is to provide both researchers and practitioners with guidelines on how to perform empirical studies on different algorithms, or production deduplication or linkage projects, as well as on how to properly assess and describe the outcome of such linkages.

Record Pair Classification Due to the problem of the number of true negatives in any comparison, quality measures which use that number (for example accuracy or the false positive rate) should not be used. The variation in the quality of a technique against particular types of data means that results should be reported for particular data sets. Also, given that the nature of some data sets may not be known in advance, the average quality across all data sets used in a certain study should be reported. When comparing techniques, precision-recall or F-measure graphs provide an additional dimension to the results. For example, if a small number of highly accurate links is required, the technique with higher precision for low recall would be chosen [4].

Blocking The aim of blocking is to cheaply remove obvious non-matches before the more detailed, expensive record pair comparisons are made. Working perfectly, blocking would only remove record pairs that are true non-matches, thus affecting the number of true negatives, and possibly the number of false positives. To the extent that, in reality, blocking also removes record pairs from the set of true matches, it will also affect the number of true positives and false negatives. Blocking can thus be seen to be a *confounding* factor in quality measurement – the types of blocking procedures and the parameters chosen will potentially affect the results obtained for a given linkage procedure. If computationally feasible, for example in an empirical study using small data sets, it is strongly recommended that all quality measurement results be obtained without the use of blocking. It is recognised that it may not be possible to do this with larger data sets. A compromise would be to publish the blocking approach and resulting number of removed pairs of records, and to make the *blocked* data set available for analysis and comparison by other researchers. At the very least, the blocking procedure and parameters should be specified in a form that can enable other researchers to repeat it.[1]

7 Conclusions

Deduplication and data linkage are important tasks in the pre-processing step of many data mining projects, and also important for improving data quality before data is loaded into data warehouses. An overview of data linkage techniques has been presented, and the issues involved in measuring the quality of deduplication and data linkage algorithms have been discussed. It is recommended that data linkage quality be measured using the precision-recall or F-measure graphs rather than single numerical values, and measures that include the number of true negative matches should not be used due to their large number in the space of record pair comparisons. When publishing empirical studies, researchers should aim to use non-blocked data sets if possible, or otherwise at least detail the blocking approach taken, and report on the number of record pairs being removed by the blocking process.

Acknowledgements

This work is supported by an Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the NSW Department of Health. The authors would like to thank Markus Hegland for insightful discussions.

[1] It is acknowledged that the example given in Section 5 doesn't follow the recommendations presented here. It's aim is only to illustrate the presented issues, not the actual results of this deduplication.

References

1. Baxter, R., Christen, P. and Churches, T.: A Comparison of Fast Blocking Methods for Record Linkage. ACM SIGKDD '03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation, August 27, 2003, Washington, DC, pp. 25-27.
2. Bertolazzi, P., De Santis, L. and Scannapieco, M.: Automated record matching in cooperative information systems. Proceedings of the international workshop on data quality in cooperative information systems, Siena, Italy, January 2003.
3. Bilenko, M. and Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. Proceedings of the 9th ACM SIGKDD conference, Washington DC, August 2003.
4. Bilenko, M. and Mooney, R.J.: On evaluation and training-set construction for duplicate detection. Proceedings of the KDD-2003 workshop on data cleaning, record linkage, and object consolidation, Washington DC, August 2003.
5. Chaudhuri, S., Ganjam, K., Ganti, V. and Motwani, R.: Robust and efficient fuzzy match for online data cleaning. Proceedings of the 2003 ACM SIGMOD International Conference on on Management of Data, San Diego, USA, 2003, pp. 313-324.
6. Chaudhuri, S., Ganti, V. and Motwani, R.: Robust identification of fuzzy duplicates. Proceedings of the 21st international conference on data engineering, Tokyo, April 2005.
7. Christen, P., Churches, T. and Hegland, M.: Febrl – A parallel open source data linkage system. Proceedings of the 8th PAKDD, Sydney, Springer LNAI 3056, May 2004.
8. Christen, P. and Goiser, K.: Quality and Complexity Measures for Data Linkage and Deduplication. Accepted for Quality Measures in Data Mining, Springer, 2006.
9. Churches, T., Christen, P., Lim, K. and Zhu, J.X.: Preparation of name and address data for record linkage using hidden Markov models. BioMed Central Medical Informatics and Decision Making, Dec. 2002.
10. Cohen, W.W.: Integration of heterogeneous databases without common domains using queries based on textual similarity. Proceedings of SIGMOD, Seattle, 1998.
11. Cohen, W.W. and Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
12. Cohen, W.W., Ravikumar, P. and Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. Proceedings of IJCAI-03 workshop on information integration on the Web (IIWeb-03), pp. 73–78, Acapulco, August 2003.
13. Cooper, W.S. and Maron, M.E.: Foundations of Probabilistic and Utility-Theoretic Indexing. Journal of the ACM , vol. 25, no. 1, pp. 67–80, January 1978.
14. Elfeky, M.G., Verykios, V.S. and Elmagarmid, A.K.: TAILOR: A record linkage toolbox. Proceedings of the ICDE' 2002, San Jose, USA, March 2002.
15. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers, HP Labs Tech Report HPL-2003-4, HP Laboratories, Palo Alto, March 2004.
16. Fellegi, I. and Sunter, A.: A theory for record linkage. Journal of the American Statistical Society, December 1969.
17. Galhardas, H., Florescu, D., Shasha, D. and Simon, E.: An Extensible Framework for Data Cleaning. Proceedings of the Inter. Conference on Data Engineering, 2000.
18. Gill, L.: Methods for Automatic Record Matching and Linking and their use in National Statistics. National Statistics Methodology Series No. 25, London, 2001.
19. Gomatam, S., Carter, R., Ariet, M. and Mitchell G.: An empirical comparison of record linkage procedures. Statistics in Medicine, vol. 21, no. 10, May 2002.

20. Gu, L. and Baxter, R.: Adaptive filtering for efficient record linkage. SIAM international conference on data mining, Orlando, Florida, April 2004.
21. Gu, L. and Baxter, R.: Decision models for record linkage. Proceedings of the 3rd Australasian data mining conference, pp. 241–254, Cairns, December 2004.
22. Hernandez, M.A. and Stolfo, S.J.: Real-world data is dirty: Data cleansing and the merge/purge problem. In Data Mining and Knowledge Discovery 2, Kluwer Academic Publishers, 1998.
23. Kelman, C.W., Bass, A.J. and Holman, C.D.: Research use of linked health data - A best practice protocol. Aust NZ Journal of Public Health, 26:251-255, 2002.
24. Lee, M.L., Ling, T.W. and Low, W.L.: IntelliClean: a knowledge-based intelligent data cleaner. Proceedings of the 6th ACM SIGKDD conference, Boston, 2000.
25. *AutoStan and AutoMatch, User's Manuals*, MatchWare Technologies, 1998.
26. Maletic, J.I. and Marcus, A.: Data Cleansing: Beyond Integrity Analysis. Proceedings of the Conference on Information Quality (IQ2000), Boston, October 2000.
27. McCallum, A., Nigam, K. and Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. Proceedings of the 6th ACM SIGKDD conference, pp. 169–178, Boston, August 2000.
28. Monge, A. and Elkan, C.: The field-matching problem: Algorithm and applications. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, August 1996.
29. Nahm, U.Y, Bilenko M. and Mooney, R.J.: Two approaches to handling noisy variation in text mining. Proceedings of the ICML-2002 workshop on text learning (TextML'2002), pp. 18–27, Sydney, Australia, July 2002.
30. Newcombe, H.B. and Kennedy, J.M.: Record Linkage: Making Maximum Use of the Discriminating Power of Identifying Information. Communications of the ACM, vol. 5, no. 11, 1962.
31. Centre for Epidemiology and Research, NSW Department of Health. New South Wales Mothers and Babies 2001. NSW Public Health Bull 2002; 13(S-4).
32. Porter, E. and Winkler, W.E.: Approximate String Comparison and its Effect on an Advanced Record Linkage System. RR 1997-02, US Bureau of the Census, 1997.
33. Rahm, E. and Do, H.H.: Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, 2000.
34. Ravikumar, P. and Cohen, W.W.: A hierarchical graphical model for record linkage. Proceedings of the 20th conference on uncertainty in artificial intelligence, Banff, Canada, July 2004.
35. Sarawagi, S. and Bhamidipaty, A.: Interactive deduplication using active learning. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
36. Tejada, S., Knoblock, C.A. and Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
37. Winkler, W.E. and Thibaudeau, Y: An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 U.S. Decennial Census. RR 1991-09, US Bureau of the Census, 1991.
38. Winkler, W.E.: The State of Record Linkage and Current Research Problems. RR 1999-04, US Bureau of the Census, 1999.
39. Winkler, W.E.: Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. RR 2000-05, US Bureau of the Census, 2000.
40. Winkler, W.E.: Methods for Record Linkage and Bayesian Networks. RR 2002-05, US Bureau of the Census, 2002.
41. Yancey, W.E.: An adaptive string comparator for record linkage RR 2004-02, US Bureau of the Census, February 2004.