Quality and Complexity Measures for Data Linkage and Deduplication

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 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 linking or deduplicating very large data sets. Different measures have been used to characterise the quality and complexity of data linkage algorithms, and several new metrics have been proposed. An overview of the issues involved in measuring data linkage and deduplication quality and complexity is presented in this chapter. 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 data linkage and deduplication quality and complexity.

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

1 Introduction

With many businesses, government organisations and research projects collecting massive amounts of data, the techniques collectively known as data mining have 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 preprocessing steps (which may dominate the actual data mining activity) [34]. It is generally accepted [14] that about 20% to 30% of the time and effort in a data mining project is used for data understanding, and about 50% to 70% for data preparation.

* Corresponding author

An increasingly important task in the data preprocessing step of many data mining projects 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 order to allow more detailed data analysis or mining. The aim of such linkages is to match all records related to the same entity, such as a patient, a customer, a business, a consumer product, or a genome sequence.

Data linkage and deduplication 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, 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. Data linkage can also help to enrich data that is used for pattern detection in data mining systems. Businesses routinely 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 money. 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 does not only apply to records which refer to persons. In bioinformatics, data linkage can help to find genome sequences in a large data collection 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. Finding and 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 *join* operation in *SQL* or its equivalent in other data management systems is all that is required. However, in most cases no unique keys are shared by all of the data sets, and more sophisticated linkage techniques need to be applied. These different techniques can be broadly classified into *deterministic* or rules-based approaches, and *probabilistic* approaches, as discussed in Section 2. The notation and problem analysis are then presented in Section 3, before an overview of the various quality measures used to assess data linkage techniques is given in Section 4. When linking large data sets, it is normally not feasible to compare all possible record pairs due to the resulting computational complexity, and special blocking, sorting or indexing techniques have to be applied. Several recently proposed complexity measures, and the influence of blocking techniques

upon quality measures, are discussed in Section 5. A real-world example is used in Section 6 to illustrate the effects of using different quality and complexity measures. Finally, the issues involved in quality measures in data linkage and deduplication are discussed, and a series of recommendations is given in Section 7 on how to assess the quality and complexity of data linkage and deduplication algorithms and techniques, before this chapter is concluded with a short summary in Section 8.

2 Data Linkage Techniques

Data linkage and deduplication techniques have traditionally been used in the health sector for cleaning and compiling data sets for longitudinal or other epidemiological studies [25], and in statistics for linking census and related data [19, 41]. 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 [31] in 1962, and the theoretical statistical foundation was provided by Fellegi and Sunter [17] in 1969.

Similar techniques have independently been developed 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. While statisticians and epidemiologists speak of *record* or *data linkage* [17], the computer science and database communities often refer to the same process as *data* or *field matching*, *data scrubbing*, *data cleaning* [18, 35], *data cleansing* [28], *preprocessing*, *duplicate detection* [5], *entity uncertainty* or as the *object identity problem*. In commercial processing of customer databases or business mailing lists, data linkage is sometimes called *merge/purge processing* [23], *data integration* [11], *list washing* or *ETL* (extraction, transformation and loading).

2.1 Data Linkage Process

A general schematic outline of the data linkage process is given in Figure 1. As most real-world data collections contain noisy, incomplete and incorrectly formatted information, data cleaning and standardisation are important preprocessing steps for successful data linkage, and before data can be loaded into data warehouses or used for further analysis [35]. Data may be recorded or captured in various, possibly obsolete, formats and data items may be missing, out of date, or contain errors. The cleaning and standardisation of names and addresses is especially important, to make sure that no misleading or redundant information is introduced (e.g. duplicate records). Names are often reported differently by the same person depending upon the organisation they are in contact with, resulting in missing middle names, initials-only, or even swapped name parts. Additionally, while for many regular words there is only one correct spelling, there are often different written forms of proper names, for example *'Gail'* and

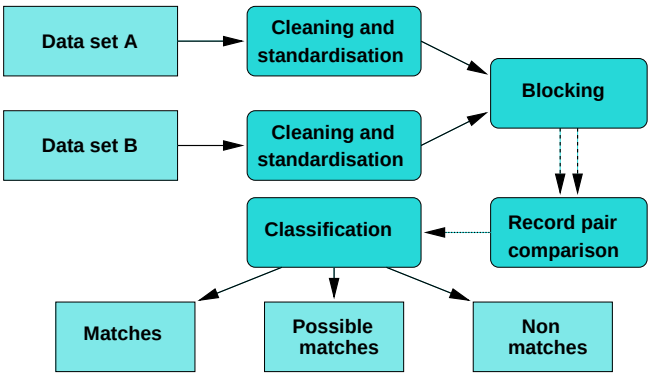

Fig. 1. General linkage process. The output of the blocking step are record pairs, and the output of the comparison step are numerical vectors with matching weights

'Gayle'. The main task of data cleaning and standardisation is the conversion of the raw input data into well defined, consistent forms and the resolution of inconsistencies in the way information is represented or encoded [9, 10].

If two data sets are to be linked, the number of possible comparisons equals the product of the number of records in the two data sets. The performance bottleneck in a data linkage system is usually the expensive evaluation of the similarity measures between pairs of records [2]. It is therefore computationally not feasible to consider all 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 true 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). Thus, the space of potential matches becomes sparser when linking larger data sets, while the computational efforts increase exponentially. To reduce the large amount of possible record pair comparisons, traditional data linkage techniques [17, 41] work in a blocking fashion, i.e. they use one or a combination of record attributes to split the data sets into blocks. Only records having the same value in such a *blocking variable* are then compared (as they will be in the same block). This technique becomes problematic if a value in a blocking variable is recorded wrongly, as the corresponding record is then inserted into a different block. To overcome this problem, several passes (iterations) with different blocking variables are normally performed.

While the aim of blocking is to reduce the number of comparisons made as much as possible (by eliminating comparisons between records that obviously are not matches), it is important that no potential match is overlooked because of the blocking process. There is a trade-off between the reduction in number of record pair comparisons and the number of missed true matches [2]. An alternative to standard blocking is the *sorted neighbourhood* [24] approach, where

records are sorted according to the values of the blocking variable, then a sliding window is moved over the sorted records, and comparisons are performed between the records within the window. Newer experimental approaches based on approximate q-gram indices [2, 7] or high-dimensional clustering [29] are current research topics. The effects of blocking upon the quality and complexity of the data linkage process are discussed in more details in Section 5.

Each record pair produced in the blocking process is compared using a variety of field comparison functions, each applied to one or a combination of record attributes. These functions can be as simple as a numerical or an exact string 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 function returns a numerical weight, often a positive weight for agreeing values and a negative weight for disagreeing values. For each record pair a *weight vector* is formed containing all the weights 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.

2.2 Deterministic Linkage

Deterministic linkage techniques can be applied if unique entity identifiers are available in all the data sets to be linked. Alternatively, a combination of attributes can be used to create a *linkage key* which is then used to match records that have the same linkage key value. Such linkage systems can be developed using 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, robust (for example include a check digit for detecting invalid or corrupted values), stable over time, and highly available.

Alternatively, a set of (often very complex) rules can be used to classify pairs of records as matches or as non-matches. Such rules can be more flexible than using a linkage key, but their development is labour intensive and highly dependent on 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 set(s) to be linked or deduplicated. In practise, therefore, deterministic rule based systems are limited to ad-hoc linkages of smaller data sets. In a recent study [20] an iterative deterministic linkage system has been compared with the commercial probabilistic system *AutoMatch* [27]. Empirical results showed that the probabilistic approach resulted in better linkage quality.

2.3 Probabilistic Linkage

As common unique entity identifiers (or keys) are rarely available in all data sets to be linked, the linkage process must be based on existing common attributes, for example 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, parts can be out-of-date or even be missing.

In the traditional probabilistic linkage approach [17, 41], 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 (or files) **A** and **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

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

of true matches, and

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

of true non-matches. Fellegi and Sunter [17] considered ratios of probabilities of the form

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

where γ is an arbitrary agreement pattern in a comparison space Γ. For example, Γ 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. In theory, the person undertaking this clerical review has access to additional data (or may be able to seek it out) which enables them to resolve the linkage status. In practice, often no additional data is available and the clerical review process becomes one of applying experience, common sense or human intuition to the decision based on available data. As shown in an early study [39] comparing a computer based probabilistic linkage system with a fully manual linkage of

health records, the computer based approach resulted in more reliable, consistent and more cost effective linkage results.

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 administrative 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.

2.4 Modern Approaches

Improvements [42] upon the classical probabilistic linkage [17] approach include the application of the expectation-maximisation (EM) algorithm for improved parameter estimation [43], the use of approximate string comparisons [33] to calculate partial agreement weights when attribute values have typographical errors, and the application of Bayesian networks [44]. A system that is capable of linking very large data sets with hundreds of millions of records is presented in [45]. It is based on special sorting, preprocessing and indexing techniques and assumes that the smaller of two data sets fits into the main memory of a large compute server.

In recent years, researchers have 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 linkage or deduplication status) is available.

One approach based on ideas from information retrieval is to represent records as document vectors and compute the *cosine distance* [11] between such vectors. Another possibility is to use an *SQL* like language [18] 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 [26]. The authors also describe the precision-recall trade-off (see Section 4 below), where choosing a higher recall results in lower precision (more non-duplicates being classified as duplicates), or vice versa.

A hybrid system is described in [15] which utilises both supervised and unsupervised machine learning techniques in the data linkage process, and introduces metrics for determining the quality of these techniques. The authors find that machine learning techniques outperform probabilistic techniques, and provide a lower proportion of possible matching pairs. In order to overcome the problem of the lack of availability of training data in real-world data sets, they propose a hybrid technique where class assignments are made to a sample of the data through unsupervised clustering, and the resulting data is then used as a training set for a supervised classifier (specifically, a decision tree or an instance-based classifier).

The authors of [38] apply active learning to the problem of lack of training instances in real-world data. Put simply, by repeatedly providing an example which is representative of part of the unclassified data set for clerical review, then using that manually classified result to add to the training set of a committee of classifiers, they found that review of less than 100 training examples provided better results than from 7,000 randomly selected reviews. A similar approach is presented in [40], 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 [29] 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, allowing user control over the trade-off between accuracy and the amount of clerical review needed. Methods based on nearest neighbours are explored by [8], with the idea to capture local structural properties instead of a single global distance approach.

Graphical models [36] is an approach which aims to use the structural information available in the data to build hierarchical probabilistic graphical models, an unsupervised technique not requiring any training data. The authors present results which are better than results achieved by supervised techniques.

Another approach is to train distance measures used for approximate string comparisons. The authors of [4] present a framework for improving duplicate detection using trainable measures of textual similarity. They argue that both at the character and word level there are differences in importance of certain character or word modifications (like inserts, deletes and transpositions), and accurate similarity computations require adapting string similarity metrics for all attributes in a data set with respect to the particular data domain. They present two learnable string similarity measures, the first based on edit distance (and better suitable for shorter strings) and the second based on a support vector machine (more appropriate for attributes that contain longer strings). Their results on various data sets show that learned edit distance resulted in improved precision and recall results. Very similar approaches are presented in [7, 30, 46, 47], with [30] 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 [28].

Different measures for the quality of the achieved linkages and the complexity of the presented algorithms have been used in these recent publications. An overview of these measures is given in Sections 4 and 5. In the following section the notation and problem analysis is presented first, and a simple illustrative example is given.

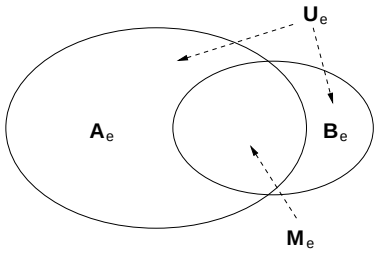

Fig. 2. General linkage situation with two sets of entities $\mathbf{A}_e$ and $\mathbf{B}_e$, their intersection $\mathbf{M}_e$ (the entities which are appear in both sets), and the set $\mathbf{U}_e$ which contains the entities that appear in either $\mathbf{A}_e$ or $\mathbf{B}_e$, but not in both

3 Notation and Problem Analysis

In this section the standard notation as given in the traditional data linkage literature [17, 41, 42] is followed. The number of elements in a set $\mathbf{X}$ will be denoted by $|\mathbf{X}|$. A general linkage situation, where the aim is to link two sets of entities, is assumed. 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. Therefore, people may appear in both sets. 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 2, and called the *entity space*.

The maximum 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 possible. 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:

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

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

In a simple example, assume the set $\mathbf{A}_e$ contains 5 million entities (hospital patients), and set $\mathbf{B}_e$ contains 1 million entities (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.

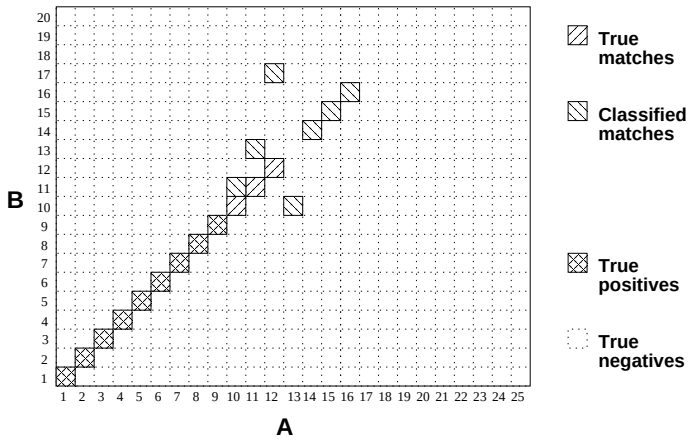

Fig. 3. General record pair comparison space with 25 records in data set **A** arbitrarily sorted on the horizontal axis and 20 records in data set **B** arbitrarily sorted on the vertical axis. The full rectangular area corresponds to all possible record pair comparisons. Assume that record pairs $(A1, B1)$, $(A2, B2)$ 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)

The entities in $\mathbf{A}_e$ and $\mathbf{B}_e$ are now stored in two data sets (or databases or files), denoted by **A** and **B**, such that there is exactly one record in **A** for each entity in $\mathbf{A}_e$ (i.e. the data set contains no duplicate records), and each record in **A** corresponds to an entity in $\mathbf{A}_e$. The same holds for $\mathbf{B}_e$ and **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 [17, 41].

It is assumed that no blocking or indexing (as discussed in Section 2.1) is being applied, and that all possible pairs of records are being 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 deduplication of a single data set **A** the number of record pair comparisons equals $|\mathbf{A}| \times (|\mathbf{A}| - 1)$, as each record in the data set will be compared to all others, but not to itself. The space of record pair comparisons is illustrated in Figure 3 and called the *comparison space*.

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). It is assumed here that the linkage algorithm does not classify record pairs as possible matches (as discussed in Section 2.3).

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)

Both records of a true matched pair correspond to the same entity in $\mathbf{M}_e$. Unmatched 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 a pair corresponding to different entities in $\mathbf{M}_e$. As each record corresponds to exactly one entity, a record in $\mathbf{A}$ can only be matched to a maximum of one record in $\mathbf{B}$, and vice versa.

For each record pair, the binary classification into $\tilde{M}$ and $\tilde{U}$ results in one of four different outcomes [16] as illustrated in the confusion matrix in Table 1. True matched record pairs from M that are classified as matches (into $\tilde{M}$) are called *true positives* (TP). True non-matched record pairs from U that are classified as non-matches (into $\tilde{U}$) are called *true negatives* (TN). True matched record pairs from M that are classified as non-matches (into $\tilde{U}$) are called *false negatives* (FN), and true non-matched record pairs from U that are classified as matches (into $\tilde{M}$) are called *false positives* (FP). As illustrated, $M = TP + FN$, $U = TN + FP$, $\tilde{M} = TP + FP$, and $\tilde{U} = TN + FN$.

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

The maximum number of true 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:

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

This is also illustrated in Figure 3. Therefore, any quality measure used in data linkage or deduplication that uses the number of TN will result in deceptive results, as will be shown in Sections 4 and 6.

The analysis so far was done under the assumption of no duplicate records in the data sets **A** and **B**, which resulted in a record in one data set being matched to a maximum of one record in another data set (often called *one-to-one* assignment restriction [3]). In practise, however, *one-to-many* and *many-to-many* linkages or deduplications are common. Examples include longitudinal studies of administrative health data, where several records might correspond to a certain patient over time (this happens when data sets have not been deduplicated properly), or business mailing lists where several records might relate to the same customer. While the above analysis would become more complicated, the issue of having a very large number of TN will still 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 (in practise often only a small number of *best* matches for each record are of interest).

In the following section the different quality measures that have been used for assessing data linkage algorithms [4, 8, 15, 29, 38, 40, 47] are presented. Various publications have used measures that include the number of TN, which can lead to deceptive results.

4 Quality Measures

Given that data linkage is a classification problem, various quality measures are available to the data linkage researcher and practitioner [16]. With many recent approaches being based on supervised learning, no clerical review process (i.e. no possible matches) is assumed and the classification problem becomes a binary classification, with record pairs being classified as either matches or non-matches. One issue with many algorithms is the setting of a threshold which influences the classifier performance. In order to determine which ones to select for a particular problem, comparative evaluations must be sourced or conducted. An obvious, much used, and strongly underpinned methodology for doing this involves the use of statistical techniques. Salzberg [37] describes this issue in terms of data mining and the use of machine learning algorithms, and points out several pitfalls which can lead to misleading results, but offers a solution to overcome them. This issue of classifier comparison is discussed in more details, before the different quality measures are presented in Section 4.2

4.1 On Comparing Classifiers

When different classifiers are compared on the same problem class, care has to be taken to make sure that the achieved quality results are statistically valid and not just an artifact of the comparison procedure. One pitfall in particular, the *multiplicity effect* [37], means that, when comparing algorithms on the

same data, because of the lack of independence of the data, the chances of erroneously achieving significance on a single test increases, so the level below which significance of the statistical p-value is accepted must be adjusted down (a conservative adjustment used in the statistics community is known as Bonferroni correction). In an example [37], if 154 variations (i.e. combinations of parameter settings) of a test algorithm are used, there is a 99.96% chance that one of the variations will be incorrectly significant at the 0.05 level. Multiple independent researchers using the same data sets (e.g. community repositories like the UCI machine learning repository [6]) can suffer from this problem as well. Tuning – the process of adjusting an algorithm's parameters in an attempt to increase the quality of the classification – is subject to the same issue if the data for tuning and testing are the same.

Salzberg's [37] recommended solution for the above is to use k-fold cross validation (k-times hold out one k'th of the data for testing), and to also hold out a portion of the training data for tuning. Also, since the lack of independence rules out the use of the t-test, he suggests the use of the binomial test or an analysis of variance (ANOVA) of distinct random samples.

While the aim of this chapter is not to compare the performance of classifiers for data linkage, it is nevertheless important for both researchers and practitioners working in this area to be aware of the issues discussed in this section.

4.2 Quality Measures used for Data Linkage

In this section, different measures [16] that have been used for assessing the quality of data linkage algorithms [5] are presented, and using the simple example from Section 3, it is shown how the results can be deceptive. The assumption is that a supervised data linkage algorithm is being used to classify record pairs as matches and non-matches, resulting in a confusion matrix of classified record pairs as shown in Table 1. As discussed in Section 2.3, the linkage algorithm is assumed to have a single threshold parameter t 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 the threshold 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 being reported in empirical studies). Alternatively, quality measures can be visualised in a graph over a range of threshold values, with the threshold normally being plotted along the horizontal axis, as illustrated by the examples in Section 6. The following list presents the commonly used quality measures.

- **Accuracy** is measured as $acc = \frac{TP+TN}{TP+FP+TN+FN}$. It is a widely used measure and mainly suitable for balanced classification problems. As this measure includes the number of TN, it is affected by their large number (i.e. the number of TN will dominate the formula). The calculated accuracy values

will be too high (for example, simply classifying all compared record pairs as non-matches will result in a very high accuracy value). Accuracy is therefore not a good quality measure for data linkage and should not be used.

- **Precision** is measured as $prec = \frac{TP}{TP+FP}$ and is also called *positive predictor value*. It is the proportion of classified matches that are true matches, and is widely used in the information retrieval field [1] in combination with the *recall* measure for visualisation in precision-recall graphs.
- **Recall** is measured as $rec = \frac{TP}{TP+FN}$ (true positive rate). Also known as *sensitivity*, it is the proportion of actual matches that have been classified correctly. Sensitivity is a common measure in epidemiological studies [48].
- **Precision-recall graph** is created by plotting precision values on the vertical axis and recall values on the horizontal axis. In information retrieval [1], the graph is normally plotted for 11 standardised recall values at $0.0, 0.1, \ldots, 1.0$, and is interpolated if a certain recall value is not available. In data linkage, a varying threshold can be used. There is a trade-off between precision and recall, in that high precision can normally only be achieved at the cost of low recall values, and vice-versa.
- **Precision-recall break-even point** is the value where precision becomes equal to recall, i.e. $\frac{TP}{TP+FP} = \frac{TP}{TP+FN}$. At this point, positive and negative classifications are made at the same rate. This measure is a single number.
- **F-measure** (or *F-score*) is the harmonic mean of precision and recall and is calculated as $f\!-\!meas = 2(\frac{prec \times rec}{prec+rec})$. It will have a high value only when both precision and recall have high values, and can be seen as a way to find the best compromise between precision and recall [1].
- **Maximum F-measure** (which is also called *F1 score*) is the maximum value of the F-measure over a varying threshold.
- **Specificity** (which is the *true negative rate*) is calculated as $spec = \frac{TN}{TN+FP}$. This measure is used frequently in epidemiological studies [48]. As it includes the number of TN, it suffers from the same problem as accuracy, and should not be used for measuring the quality of data linkage algorithms.
- **False positive rate** is measured as $fpr = \frac{FP}{TN+FP}$. Note that $fpr = (1 - spec)$. As the number of TN is included in this measure, it suffers from the same problem as accuracy and specificity, and should not be used.
- **ROC curve** (Receiver operating characteristic curve) [16] is plotted as the true positive rate (which is the recall) on the vertical axis against the false positive rate on the horizontal axis for a varying threshold. While ROC curves are being promoted to be robust against skewed class distributions [16], the problem when using them in data linkage is the number of TN, which only appears in the false positive rate. This rate will be calculated too low, resulting in too optimistic ROC curves, as shown in the examples in Section 6.

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).

Table 2. Quality results for the given example

Measure	Entity space	Comparison space
Accuracy	94.340%	99.999994%
Precision	72.222%	72.222%
Recall	92.857%	92.857%
F-measure	81.250%	81.250%
Specificity	94.565%	99.999995%
False positive rate	5.435%	0.000005%

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 2 shows the resulting quality measures for this example in both the comparison and the entity spaces. As discussed, any measure that includes the number of TN depends upon weather entities or record pairs are counted. As can be seen, the results for accuracy, specificity and the false positive rate all show misleading results when based on record pairs (i.e. measured in the comparison space). This issue will be illustrated and discussed further in Sections 6 and 7.

The authors of a recent publication [5] discuss the issue of evaluating data linkage and deduplication systems. They advocate the use of precision-recall graphs over the use of single number measures like accuracy or maximum F-measure, on the grounds that such single number measures assume that an optimal threshold value has been found. A single number can also hide the fact that one classifier might perform better for lower threshold values, while another has improved performance for higher thresholds.

While all quality measures presented so far assume a binary classification without clerical review, a new measure has been proposed recently [22] that aims to quantify the proportion of possible matches within a traditional probabilistic linkage system (which classifies record pairs into matches, non-matches and possible matches, as discussed in Section 2.3). The authors propose the measure $pp = \frac{N_{P,M}+N_{P,U}}{TP+FP+TN+FN}$, where $N_{P,M}$ is the number of true matches that have been classified as possible matches, and $N_{P,U}$ is the number of true non-matches that have been classified as possible matches. This measure quantifies the percentage of record pairs that are being classified as possible matches, and therefore needing manual clerical review. Low pp values are desirable, as they correspond to less manual clerical review.

5 Blocking and Complexity Measures

The assumption in the analysis and discussion of quality measures given so far was that all possible record pairs are being compared. The number of com-

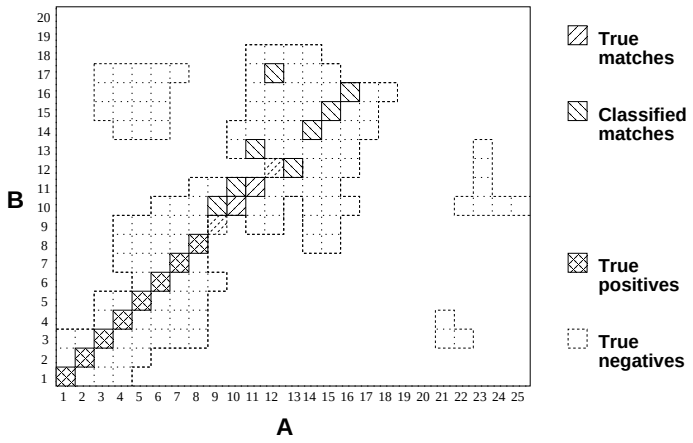

Fig. 4. Version of Figure 3 in a blocked linkage space. The empty space are record pairs which were removed by blocking. Besides many non matches, the blocking process has also removed the true matched record pairs $(A9, B9)$ and $(A12, B12)$, and then wrongly classified the pairs $(A9, B10)$ and $(A12, B17)$ as matches

parisons in this situation equals the product of the number of records in the two data sets, $|\mathbf{A}| \times |\mathbf{B}|$. As discussed earlier, this is computationally feasible only for small data sets. In practise, blocking, filtering, indexing, searching, or sorting algorithms [2, 9, 15, 21, 23] are used to reduce the number of record pair comparisons as discussed in Section 2.1. The aim of such algorithms is to cheaply remove as many record pairs from the set of non-matches U that are obvious non-matches, without removing any record pairs from the set of matches M. Two complexity measures that quantify the efficiency and quality of such blocking methods have recently been proposed [15]:

- **Reduction ratio** is measured as $rr = 1 - \frac{N_b}{|\mathbf{A}| \times |\mathbf{B}|}$, with $N_b \leq |\mathbf{A}| \times |\mathbf{B}|$ being the number of record pairs produced by a blocking algorithm (i.e. the number of record pairs not removed by blocking). The reduction ratio measures the relative reduction of the comparison space, but without taking into account the quality of the reduction, i.e. how many record pairs from U and how many from M are removed by the blocking process.
- **Pairs completeness** is measured as $pc = \frac{N_m}{|M|}$ with N_m being the number of correctly classified true matched record pairs in the blocked comparison space, and $|M|$ the total number of true matches as defined in Section 3. Pairs completeness can be seen as being analogous to recall.

There is a trade-off between the reduction ratio and pairs completeness (similar to the precision-recall trade-off). As no blocking algorithm is perfect and will thus remove record pairs from M, the blocking process will affect both true matches and true non-matches. All quality measures presented in Section 4 will therefore be influenced by blocking.

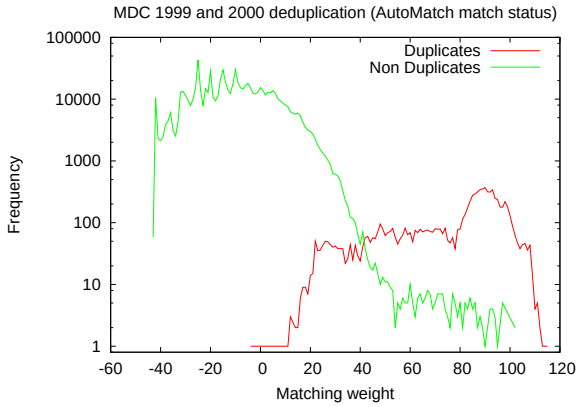

Fig. 5. The density plot of the matching weights for a real-world administrative health data set. This plot is based on record pair comparison weights in a blocked comparison space. The smallest weight is -43, the highest 115. Note that the vertical axis with frequency counts is on a log scale

6 Experimental Examples

In this section the previously discussed issues on quality and complexity measures are illustrated using a real-world administrative health data set, the *New South Wales Midwives Data Collection* (MDC) [32]. 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 using the commercial probabilistic linkage system *AutoMatch* [27]. According to this deduplication, the data contains 166,555 unique mothers, with 158,081 having one, 8,295 having two, 176 having three, and 3 having four records in this data set. Of these last three mothers, two gave birth to twins twice in the two years 1999 and 2000, while one mother had a triplet and a single birth. The *AutoMatch* deduplication decision (which included clerical review) was used as the true match (or deduplication) status.

A deduplication was then performed using the *Febrl* (Freely extensible biomedical record linkage) [9] data linkage system. Fourteen attributes in the MDC were compared using various comparison functions (like exact and approximate string, and date of birth comparisons), and the resulting fourteen numerical weights were summed into a matching weight R as discussed in Section 2.3. The resulting density plot is shown in Figure 5. As can be seen, 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 the two years data set with 175,211 records would result in $175,211 \times 175,210 = 30,698,719,310$ record pairs, which is infeasible to process even with today's powerful computers. Standard blocking as implemented in *Febrl* was used to reduce the number of comparisons, resulting in 759,773 record pair comparisons. The reduction ratio in this case was therefore

$$rr = 1.0 - \frac{759,773}{30,698,719,310} = 1.0 - 2.4749 \times 10^{-5} = 0.999975.$$

This corresponds to only around 0.0025% of all record pairs in the full comparison space not being removed by the blocking process. The total number of true 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. The resulting pairs completeness value therefore was

$$pc = \frac{8,808}{8,841} = 0.99626,$$

which corresponds to more than 99.6% of all the true duplicates being included in the blocked comparison space and classified the same by both *AutoMatch* and *Febrl*.

The quality measures discussed in Section 4 applied to this real-world deduplication procedure are shown in Figure 6 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 was simulated by assuming that the record pairs removed by blocking were normally distributed with matching weights between -43 and -10. The number of TN was therefore different between the blocked and the full comparison spaces. As can be seen, the precision-recall graph is not affected by the blocking process, and the F-measure is only differs slightly. All other measures, however, resulted in graphs of different shape. The large number of TN compared to the number of TP resulted in the specificity measure being very similar to the accuracy measure. Interestingly, the ROC curve, being promoted as robust with regard to skewed classification problems, resulted in the least illustrative graph, especially for the full comparison space, making it not very useful for data linkage and deduplication.

7 Discussion and Recommendations

Primarily, the measurement of quality in data linkage and deduplication involves either absolute or relative results (for example, either technique X had an accuracy of 93% or technique X performed better than technique Y on all data examined). In order for the practitioner or researcher to make informed choices, the results of experiments must be comparable, or the techniques must be repeatable so comparisons between techniques can be made.

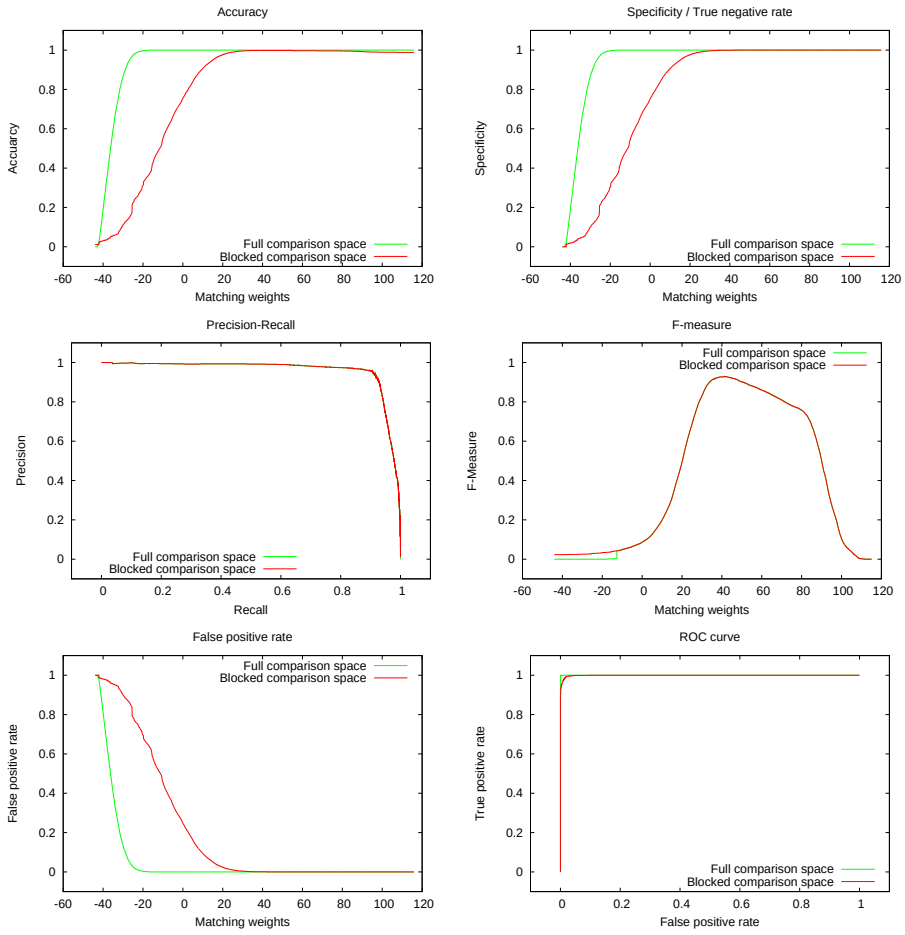

Fig. 6. Quality measurements of a real-world administrative health data set. The full comparison space (30, 698, 719, 310 record pairs) was simulated by assuming that the record pairs removed by blocking were normally distributed with matching weights between -43 and -10. Note that the precision-recall graph does not change at all, and the F-measure graphs does change only slight. Accuracy and specificity are almost the same as both are dominated by the large number of true negatives. The ROC curve is the least illustrative graphs, which is again due to the large number of true negatives

It is known, however, that the quality of techniques vary depending on the nature of the data sets the techniques are applied to [4, 37]. Whether producing absolute or comparable results, it is thus necessary for the experiments to be conducted using the same data. Therefore, results should be produced from data sets which are available to researchers and practitioners in the field. However, this does not preclude research on private data sets. The applicability of a technique

to a type of data set may be of interest, but the results produced are not beneficial for evaluating relative quality of techniques.

Of course, for researchers to compare techniques against earlier ones, either absolute results must be available, or the earlier techniques must be repeatable for comparison to occur. Ultimately, and ideally, a suite of data sets should be collected and made publicly available for this process, and they should encapsulate as much variation in types of data as feasible.

Recommendations for the various steps of a data linkage process are given in the following sections. Their aim is to provide both the researcher and practitioner with guidelines on how to perform empirical studies on different linkage algorithms or production linkage projects, as well as on how to properly assess and describe the outcome of such linkages or deduplications.

7.1 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, specificity, false positive rates, and thus ROC curves) 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 also be reported.

When comparing techniques, precision-versus-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 [5].

7.2 Blocking

As described above, the aim of blocking is to cheaply remove obvious non-matches before the more detailed, expensive record pair comparisons are made. Working perfectly, blocking will 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 it removes record pairs from the set of true matches (that is, resulting in a pairs completeness $pc < 1.0$), 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, then, would be to publish the blocking measures, reduction ratio and pairs completeness, 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.

7.3 Complexity

The overall complexity of a linkage technique is fundamentally important due to the potential size of the data sets it could be applied to: when sizes are in the millions or even billions, techniques which are $O(n^2)$ become problematic and those of higher complexity cannot even be contemplated. While blocking can provide improvements, complexity is still important. For example, if linkage is attempted on a real-time data stream, a complex algorithm may require faster hardware, more optimisation, or replacement.

As data linkage, being an important step in the data mining process, is a field rooted in practice, the practicality of a technique's implementation and use on very large data sets should be indicated. Thus, at least, the reporting of the complexity of a technique in $O()$ terms should always be made. The reporting of other usage, such as disk space and memory size, could also be beneficial.

8 Conclusions

Data linkage and deduplication are important steps in the pre-processing phase of many data mining projects, and also important for improving data quality before data is loaded into data warehouses. Different data linkage techniques have been presented and the issues involved in measuring both the quality and complexity of linkage algorithms have been discussed. It is recommended that the quality be measured using the precision-recall or F-measure graphs rather than single numerical values, and that quality 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 report measures that quantify the effects of 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.

References

1. Baeza-Yates, R.A. and , Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Boston, 1999.
2. 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.

3. D.P. Bertsekas, *Auction Algorithms for Network Flow Problems: A Tutorial Introduction*, Computational Optimization and Applications, Vol. 1, pp. 7–66, 1992.
4. 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.
5. 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.
6. Blake, C.L. and Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
7. 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.
8. Chaudhuri, S., Ganti, V. and Motwani, R.: Robust identification of fuzzy duplicates. Proceedings of the 21st international conference on data engineering, Tokyo, April 2005.
9. 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.
10. 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. Available online at: `http://www.biomedcentral.com/1472-6947/2/9/`
11. Cohen, W.W.: Integration of heterogeneous databases without common domains using queries based on textual similarity. Proceedings of SIGMOD, Seattle, 1998.
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. Shearer, C.: The CRISP-DM Model: The new blueprint for data mining. Journal of Data Warehousing, vol. 5, no. 4, pp. 13–22, Fall 2000.
15. 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.
16. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers, HP Labs Tech Report HPL-2003-4, HP Laboratories, Palo Alto, March 2004.
17. Fellegi, I. and Sunter, A.: A theory for record linkage. Journal of the American Statistical Society, December 1969.
18. Galhardas, H., Florescu, D., Shasha, D. and Simon, E.: An Extensible Framework for Data Cleaning. Proceedings of the Inter. Conference on Data Engineering, 2000.
19. Gill, L.: Methods for Automatic Record Matching and Linking and their use in National Statistics. National Statistics Methodology Series No. 25, London, 2001.
20. Gomatam, S., Carter, R., Ariet, M. and Mitchell G.: An empirical comparison of record linkage procedures. Statistics in Medicine, vol. 21, no. 10, pp. 1485–1496, May 2002.
21. Gu, L. and Baxter, R.: Adaptive filtering for efficient record linkage. SIAM international conference on data mining, Orlando, Florida, April 2004.
22. Gu, L. and Baxter, R.: Decision models for record linkage. Proceedings of the 3rd Australasian data mining conference, pp. 241–254, Cairns, December 2004.

23. Hernandez, M.A. and Stolfo, S.J.: The merge/purge problem for large databases. Proceedings of the ACM SIGMOD conference, May 1995.
24. 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.
25. 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.
26. 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.
27. *AutoStan and AutoMatch, User's Manuals*, MatchWare Technologies, Kennebunk, Maine, 1998.
28. Maletic, J.I. and Marcus, A.: Data Cleansing: Beyond Integrity Analysis. Proceedings of the Conference on Information Quality (IQ2000), Boston, October 2000.
29. 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.
30. 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.
31. 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.
32. Centre for Epidemiology and Research, NSW Department of Health. New South Wales Mothers and Babies 2001. NSW Public Health Bull 2002; 13(S-4).
33. 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.
34. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann Publishers, Inc., 1999.
35. Rahm, E. and Do, H.H.: Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, 2000.
36. 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.
37. Salzberg, S.: On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 317–328, 1997.
38. Sarawagi, S. and Bhamidipaty, A.: Interactive deduplication using active learning. Proceedings of the 8th ACM SIGKDD conference, Edmonton, July 2002.
39. Smith, M.E. and Newcombe, H.B.: Accuracies of Computer versus Manual Linkages of Routine Health Records. Methods of Information in Medicine, vol. 18, no. 2, pp. 89–97, April 1979.
40. 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.
41. W.E. Winkler and Y. Thibaudeau, *An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 U.S. Decennial Census*, Research Report RR91/09, US Bureau of the Census, 1991.
42. Winkler, W.E.: The State of Record Linkage and Current Research Problems. RR 1999-04, US Bureau of the Census, 1999.
43. 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.

44. Winkler, W.E.: Methods for Record Linkage and Bayesian Networks. RR 2002-05, US Bureau of the Census, 2002.
45. Yancey, W.E.: BigMatch: A Program for Extracting Probable Matches from a Large File for Record Linkage. RR 2002-01, US Bureau of the Census, March 2002.
46. Yancey, W.E.: An adaptive string comparator for record linkage RR 2004-02, US Bureau of the Census, February 2004.
47. Zhu, J.J., and Ungar, L.H.: String edit analysis for merging databases. KDD-2000 workshop on text mining, held at the 6th ACM SIGKDD conference, Boston, August 2000.
48. Zingmond, D.S., Ye, Z., Ettner, S.L. and Liu, H.: Linking hospital discharge and death records – accuracy and sources of bias. Journal of Clinical Epidemiology, vol. 57, pp. 21–29, 2004.