
Quality and Complexity Measures for Data Linkage and Deduplication

Peter Christen and Karl Goiser

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

Summary. 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 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 quality results. Various measures are discussed and recommendations are given on how to assess data linkage and deduplication quality and complexity.

Key words: data or record linkage, data integration and matching, deduplication, data mining pre-processing, quality and 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 from both industry and academia. 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) [38]. It is generally accepted [43] 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.

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 and aggregate all records relating to the same entity, such as a patient, a customer, a business, a product description, 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. They can also help to enrich data that is used for pattern detection in data mining systems. In the health sector, for example, linked data might contain information that is needed to improve health policies [2, 8, 28], and which traditionally has been collected with time consuming and expensive survey methods. Statistical agencies routinely link census data for further analysis [22, 49], and businesses often deduplicate and link their data sets to compile mailing lists. Within taxation offices and departments of social security, data linkage and deduplication can be used to identify people who register for assistance 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 can help 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, linking 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 the data sets, and more sophisticated linkage techniques need to be applied. These techniques can be broadly classified into *deterministic*, *probabilistic*, and modern approaches, as discussed in Sect. 2. The notation and problem analysis are presented in Sect. 3, and an overview of the various quality measures used to assess data linkage and deduplication techniques is given in Sect. 4. When linking large data sets, it is normally not feasible to compare all possible record pairs due to the resulting computational complexity, so special *blocking* techniques have to be applied. Several recently proposed complexity measures, and the influence of blocking upon quality

measurements, are discussed in Sect. 5. A real-world example is used in Sect. 6 to illustrate the effects of different quality and complexity measures. The issues involved in measuring quality in data linkage and deduplication are discussed and a series of recommendations is given in Sect. 7. Finally, the chapter is concluded with a short summary in Sect. 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 [2, 8, 28], and in statistics for linking census and related data [22, 49]. 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 [35] in 1962, and the mathematical foundation was provided by Fellegi and Sunter [20] in 1969.

Similar techniques were independently developed by computer scientists in the area of document indexing and retrieval [17]. 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* [20], the computer science and database communities often refer to the same process as *data* or *field matching*, *data scrubbing*, *data cleaning* [21, 39], *data cleansing* [30], *preprocessing*, *duplicate detection* [7], *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* [26], *data integration* [14], *list washing* or *ETL* (extraction, transformation and loading).

2.1 The Data Linkage Process

A general schematic outline of the data linkage process is given in Fig. 1. As most real-world data collections contain noisy, incomplete and incorrectly formatted information, data cleaning and standardisation are important pre-processing steps for successful data linkage, or before data can be loaded into data warehouses or used for further analysis [39]. 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,

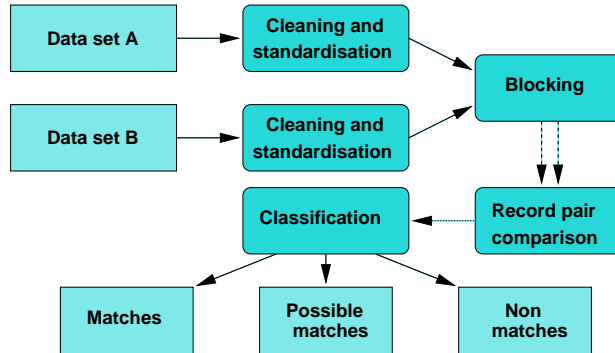


Fig. 1. General linkage process. The output of the blocking step are record pairs, and the output of the comparison step are vectors with numerical matching weights

for example ‘*Gail*’ and ‘*Gayle*’. The main task of data cleaning and standardisation is the conversion of the raw input data into well defined, consistent forms, as well as the resolution of inconsistencies in the way information is represented and encoded [12, 13].

If two data sets, \mathbf{A} and \mathbf{B} , are to be linked, potentially each record from \mathbf{A} has to be compared with all records from \mathbf{B} . The number of possible record pair comparisons thus equals the product of the size of the two data sets, $|\mathbf{A}| \times |\mathbf{B}|$. Similarly, when deduplicating one data set, \mathbf{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 [3], making it unfeasible to compare all pairs when the data sets are large. For example, linking two data sets with 100,000 records each would result in 10^{10} (ten billion) record pair comparisons. On the other hand, the maximum number of true matches that are possible corresponds to the number of records in the smaller data set (assuming a record in \mathbf{A} can only be linked to a maximum of one record in \mathbf{B} , and vice versa). Therefore, the number of potential matches increases linearly when linking larger data sets, while the computational efforts increase quadratically. The situation is the same for deduplication, where the number of duplicate records is always less than the number of records in a data set.

To reduce the large amount of possible record pair comparisons, traditional data linkage techniques [20, 49] employ *blocking*, i.e. they use one or a combination of record attributes (called the *blocking variable*) to split the data sets into blocks. All records having the same value in the blocking variable will be put into the same block, and only records within a block will be compared. This technique becomes problematic if a value in the blocking variable is recorded wrongly, as a potentially matching record may be inserted into a different block, prohibiting the possibility of a match. To overcome this problem, several passes (iterations) with different blocking variables are normally performed.

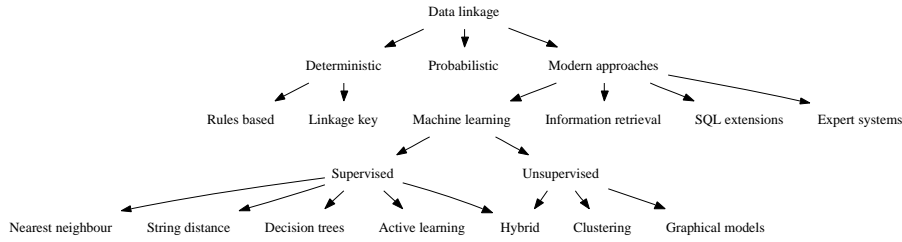


Fig. 2. Taxonomy of data linkage techniques, with a focus on modern approaches

While the aim of blocking is to reduce the number of record pair comparisons made as much as possible (by eliminating pairs of records that obviously are not matches), it is important that no potential match is overlooked because of the blocking process. An alternative to standard blocking is the *sorted neighbourhood* [27] 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 [3, 10] or high-dimensional overlapping clustering [32] are current research topics. The effects of blocking upon the quality and complexity of the data linkage process are discussed in Sect. 5.

The record pairs not removed by the blocking process are compared by applying a variety of comparison functions to one or more – or a combination of – attributes of the records. These functions can be as simple as a numerical or an exact string comparison, can take into account typographical errors [37], 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 vector is formed containing all the values calculated by the different comparison functions. These vectors are then used to classify record pairs into *matches*, *non-matches*, and *possible matches* (depending upon the decision model used). Figure 2 shows a taxonomy of the various techniques employed for data linkage. They are discussed in more detail in the following sections.

2.2 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* [2] which is then employed 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. Extra robustness for identifiers can be obtained

by including a check digit for detecting invalid or corrupted values. A recent study [2] showed how different linkage keys can affect the outcome of studies that use linked data, and that comparisons between linked data sets that were created using different linkage keys should be regarded very cautiously.

Alternatively, a set of (often very complex) rules can be used to classify pairs of records. Such *rules 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 data to be deduplicated or linked, but also with the rules system. In practise, therefore, deterministic rules based systems are limited to ad-hoc linkages of smaller data sets. In a recent study [23], an iterative deterministic linkage system was compared with the commercial probabilistic system *AutoMatch* [31], and the presented results showed that the probabilistic approach achieved better linkage quality.

2.3 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, parts can be out-of-date or swapped, or they can be missing.

In the traditional probabilistic linkage approach [20, 49], 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) \mathbf{A} and \mathbf{B} are to be linked, the set of record pairs

$$\mathbf{A} \times \mathbf{B} = \{(a, b); a \in \mathbf{A}, b \in \mathbf{B}\}$$

is the union of the two disjoint sets

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

of true matches, and

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

of true non-matches. Fellegi and Sunter [20] considered ratios of probabilities of the form

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

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, locality

and postcode. Alternatively, some of the γ might additionally consider typographical errors [37], 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 for ‘*Miller*’. 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 <i>match</i> ,
if $t_{lower} \leq R \leq t_{upper}$, then	designate a record pair as <i>possible match</i> ,
if $R < t_{lower}$, then	designate a record pair as <i>non-match</i> .

The thresholds t_{lower} and t_{upper} are determined by a-priori error bounds on false matches and false non-matches. If $\gamma \in \Gamma$ 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 $\gamma \in \Gamma$ 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, it is assumed that the person undertaking this clerical review has access to additional data (or may be able to seek it out) which enables her or him to resolve the linkage status. In practice, however, often no additional data is available and the clerical review process becomes one of applying experience, common sense or human intuition to make the decision. As shown in an early study [44] 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 results.

In the past, generally only small data sets were linked (for example for epidemiological survey studies), and clerical review was manageable in a reasonable amount of time. However, with today’s large administrative data collections with millions of records, this process becomes impossible. In these cases, even a very small percentage being passed for clerical review will result in hundreds of thousands of record pairs. 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. Developments towards this ideal are presented in the following section.

2.4 Modern Approaches

Improvements [48] upon the classical probabilistic linkage [20] approach include the application of the expectation-maximisation (EM) algorithm for improved parameter estimation [46], the use of approximate string comparisons [37] to calculate partial agreement weights when attribute values have typographical errors, and the application of Bayesian networks [47]. A system that is capable of extracting probable matches from very large data sets with

hundreds of millions of records is presented in [50]. 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 computing 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. A taxonomy is shown in Fig. 2. Many of these approaches are based on supervised learning techniques and assume that training data is available (i.e. record pairs with known linkage or deduplication status).

An *information retrieval* based approach is to represent records as document vectors and compute the *cosine distance* [14] between such vectors. Another possibility is to use an *SQL* like language [21] that allows approximate joins and cluster building of similar records, as well as decision functions that determine if two records represent the same entity. A generic knowledge-based framework based on rules and an *expert system* is presented in [29]. The authors also describe the precision-recall trade-off (which will be discussed in Sect. 4), where choosing a higher recall results in lower precision (more non-matches being classified as matches), and vice versa.

A popular approach [6, 10, 15, 34, 51, 52] is to learn distance measures that are used for approximate string comparisons. The authors of [6] 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, substitutions, and transpositions), and accurate similarity computations require adapting string similarity metrics 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. Similar approaches are presented in [10, 51, 52]. [34] uses support vector machines for of classifying record pairs. As shown in [15], combining different learned string comparison methods can result in improved linkage classification.

The authors of [42] use *active learning* to address the problem of lack of training data. Their approach involves repeatedly (i) selecting an example that a vote of classifiers disagree on the most, (ii) manually classifying it, then (iii) adding it to the training data and (iv) re-training the classifiers. The key idea is to use human input only where the classifiers could not provide a clear result. It was found that less than 100 examples selected in this manner provide better results than the random selection of 7,000 examples. A similar approach is presented in [45], where a committee of *decision trees* is used to learn mapping rules (i.e. rules describing linkages).

A *hybrid system* is described in [18] 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).

High-dimensional overlapping *clustering* is used in [32] as an alternative to traditional blocking in order to reduce the number of record pair comparisons to be made, while in [25] the use of simple k-means clustering together with a user-tunable fuzzy region for the class of possible matches is explored, thus allowing control over the trade-off between accuracy and the amount of clerical review needed. Methods based on *nearest neighbours* are explored in [11], with the idea being to capture local structural properties instead of a single global distance approach. *Graphical models* [40] are another unsupervised technique not requiring training data. This approach aims to use the structural information available in the data to build hierarchical probabilistic graphical models. Results are presented that are better than those achieved by supervised techniques.

An overview of other methods (including statistical outlier identification, clustering, pattern matching, and association rules) is given in [30].

Different measures for the quality of the achieved linkages and the complexity of the presented algorithms have been used in many recent publications. An overview of these measures is given in Sects. 4 and 5.

3 Notation and Problem Analysis

The notation used in this chapter follows the traditional data linkage literature [20, 48, 49]. 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 have resulted in people being admitted into the hospital. 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. The space described by the above is illustrated in Fig. 3 and termed *entity space*.

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. In this situation the number of non-matched entities corresponds to the sum of the entities in both sets. The following equations show this in a formal way:

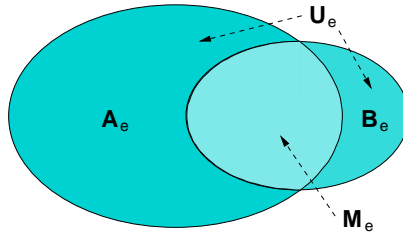


Fig. 3. General linkage situation with two sets of entities A_e and B_e , their intersection M_e (entities that appear in both sets), and the set U_e (entities that appear in either A_e or B_e , but not in both)

$$0 \leq |M_e| \leq \min(|A_e|, |B_e|) \quad (4)$$

$$\text{abs}(|A_e| - |B_e|) \leq |U_e| \leq |A_e| + |B_e|. \quad (5)$$

Example 1. Assume the set A_e contains 5 million entities (e.g. hospital patients), and set B_e contains 1 million entities (e.g. people involved in car accidents), with 700,000 entities present in both sets (i.e. $|M_e| = 700,000$). The number of non-matched entities in this situation is $|U_e| = 4,600,000$, which is the sum of the entities in both sets (6 million) minus twice the number of matched entities (as they appear in both sets A_e and B_e).

Records which refer to the entities in A_e and 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 A_e (i.e. the data set contains no duplicate records), and each record in A corresponds to an entity in A_e . The same holds for 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 $A \times B = M \cup U$ of true matches and true non-matches [20, 49], as defined in (1) and (2).

It is assumed that no blocking or indexing (as discussed in Sect. 2.1) is applied, and that all pairs of records are compared. The total number of comparisons equals $|A| \times |B|$, which is much larger than the number of entities available in A_e and B_e together. In the case of the deduplication of a single data set A , the number of record pair comparisons equals $|A| \times (|A| - 1)/2$, 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 Fig. 4 and called the *comparison space*.

Example 2. For Example 1 given above, the comparison space consists of $|A| \times |B| = 5,000,000 \times 1,000,000 = 5 \times 10^{12}$ 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 record pairs 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

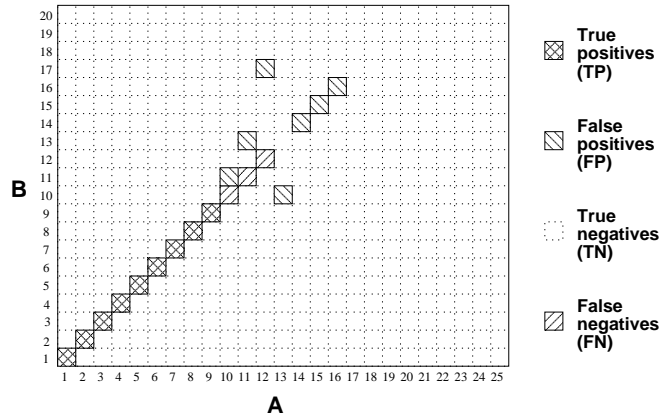


Fig. 4. Record pair comparison space with 25 records in data set **A** arbitrarily arranged on the horizontal axis and 20 records in data set **B** arbitrarily arranged on the vertical axis. The full rectangular area corresponds to all possible record pair comparisons. Assume that record pairs (A_1, B_1) , (A_2, B_2) up to (A_{12}, B_{12}) are true matches. The linkage algorithm has wrongly classified (A_{10}, B_{11}) , (A_{11}, B_{13}) , (A_{12}, B_{17}) , (A_{13}, B_{10}) , (A_{14}, B_{14}) , (A_{15}, B_{15}) , and (A_{16}, B_{16}) as matches (false positives), but missed (A_{10}, B_{10}) , (A_{11}, B_{11}) , and (A_{12}, B_{12}) (false negatives)

(as discussed in Sect. 2.3). Where a record pair comparison in \tilde{M} is actually a match (a truly matched record pair), both of its records will refer to the same entity in \mathbf{M}_e . Records in 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 it is assumed there are no duplicates in the data sets, a record in data set **A** can only be matched to a maximum of one record in data set **B**, and vice versa.

Given the binary classification into \tilde{M} and \tilde{U} , and knowing the true classification of a record pair comparison, an assignment to one of four categories can be made [19]. This is illustrated in the confusion matrix in Table 1. Truly matched record pairs from M that are classified as matches (into \tilde{M}) are called *true positives* (TP). Truly non-matched record pairs from U that are classified as non-matches (into \tilde{U}) are called *true negatives* (TN). Truly matched record pairs from M that are classified as non-matches (into \tilde{U}) are called *false negatives* (FN), and truly 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 truly matched entities and how many truly non-matched entities have been classified correctly as matches and non-matches, respectively. However, as the record pair comparisons occur in the comparison space, the results of measurements are also bound to this space. While the number of

Table 1. Confusion matrix of record pair classification

Actual	Classification	
	Match (\tilde{M})	Non-match (\tilde{U})
Match (M)	True matches	False non-matches
	True positives (TP)	False negatives (FN)
Non-match (U)	False matches	True non-matches
	False positives (FP)	True negatives (TN)

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 pair contains two records that correspond to two different entities, and each un-matched entity can be part of many record pairs. It is thus more difficult than it would first seem to decide on a proper value for the number of non-matched entities.

If no duplicates are assumed in the data sets \mathbf{A} and \mathbf{B} , then the maximum number of truly matched entities is given by (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 $(|TP| + |FP|) \leq |\mathbf{M}_e|$. 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 quadratic increase between the entity and the comparison space. The number of TN will dominate (as illustrated in Fig. 4), because in the comparison space the following equation holds:

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

Therefore (assuming no duplicates in the data sets) any quality measure used in data linkage or deduplication that uses the number of TN will give deceptive results, as will be shown in Sects. 4 and 6.

In reality, data sets are known to contain duplicate records, in which case a *one-to-one* assignment restriction [5] can be applied if there is only interest in the best match for each record. On the other hand, *one-to-many* and *many-to-many* linkages or deduplications are also 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). In such cases, a linkage algorithm may classify more record pairs as matches than there are entities (or records in a data set). The inequality $|\tilde{M}| \leq |\mathbf{M}_e|$ is not valid anymore in this context. The number of matches for a single record, however, will be small compared to the total number of record pair comparisons, as in practise often only a small number of best matches for each record are of interest. While a simple analysis

as done above would not be possible, the issue of having a very large number of TN still holds in one-to-many and many-to-many linkage situations.

In the following section the different quality measures that have been used for assessing data linkage algorithms [4, 6, 11, 18, 32, 42, 45, 52] are presented. Various publications have used measures that include the number of TN, which leads to deceptive results.

4 Quality Measures

Given that data linkage and deduplication are classification problems, various quality measures are available to the data linkage researcher and practitioner [19]. 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. One issue with many algorithms is the setting of a threshold value which determines the classifier performance. In order to select a threshold 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. In [41] this issue is described in terms of data mining and the use of machine learning algorithms. Several pitfalls are pointed out which can lead to misleading results, and a solution to overcome them is offered. This issue of classifier comparison is discussed in more detail first, before the different quality measures are presented in Sect. 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* [41], 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 correction used in the statistics community known as the *Bonferroni* adjustment). In an example [41], 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 [36]) 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.

A recommended solution [41] 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, it is suggested in [41] to use the binomial test or the 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.

4.2 Quality Measures used for Data Linkage and Deduplication

In this section, different measures [19] that have been used for assessing the quality of data linkage algorithms [7] are presented. Using the simple example from Sect. 3, it is shown how the calculated results can be deceptive for some measures. The assumption is that a data linkage technique is used that classifies record pairs as matches and non-matches, and that the true matches and true non-matches are known, resulting in a confusion matrix of classified record pairs as shown in Table 1. The linkage classifier is assumed to have a single 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 can result in an increased number of TN and FN and in a reduction in the number of TP and FP, while lowering t can reduce the number of TN and FN and increase the number of TP and FP. 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 example in Sect. 6. The following list presents the commonly used quality measures, as well as a number of other popular measures used for binary classification problems (citations given refer to data linkage or deduplication publications that have used these measures in recent years).

- **Accuracy** [18, 25, 42, 45, 53] 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 when used in the comparison space (i.e. $|TN|$ will dominate the formula). The calculated accuracy values will be too high. For example, erroneously classifying all matches as non-matches will still result in a very high accuracy value. Accuracy is therefore not a good quality measure for data linkage and deduplication, and should not be used.
- **Precision** [4, 14, 32] is measured as $prec = \frac{|TP|}{|TP|+|FP|}$ and is also called *positive predictive value* [8]. It is the proportion of classified matches that are true matches, and is widely used in information retrieval [1] in combination with the *recall* measure for visualisation in *precision-recall graphs*.

- **Recall** [25, 32, 53] is measured as $rec = \frac{|TP|}{|TP|+|FN|}$ (true positive rate). Also known as *sensitivity* (commonly used in epidemiological studies [53]), it is the proportion of actual matches that have been classified correctly.
- **Precision-recall graph** [6, 11, 16, 33] is created by plotting precision values on the vertical and recall values on the horizontal axis. In information retrieval [1], the graph is normally plotted for eleven standardised recall values at 0.0, 0.1, ..., 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 lower recall values, and vice versa [29].
- **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 misclassifications are made at the same rate, i.e. $|FP| = |FN|$. This measure is a single number.
- **F-measure** [16, 32] (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** is the maximum value of the F-measure over a varying threshold. This measure is a single number.
- **Specificity** [53] (which is the *true negative rate*) is calculated as $spec = \frac{|TN|}{|TN|+|FP|}$. This measure is used frequently in epidemiological studies [53]. As it includes the number of TN, it suffers from the same problem as accuracy, and should not be used for data linkage and deduplication.
- **False positive rate** [4, 27] is measured as $fpr = \frac{|FP|}{|TN|+|FP|}$. Note that $fpr = (1 - spec)$. As this measure includes the number of TN, it suffers from the same problem as accuracy and specificity, and should not be used.
- **ROC curve** (Receiver operating characteristic curve) 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 [19], 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.
- **AUC** (Area under ROC curve) is a single numerical measure between 0.5 and 1 (as the ROC curve is always plotted in the unit square, with a random classifier having an AUC value of 0.5), with larger values indicating better classifier performance. The AUC has the statistical property of being equivalent to the statistical *Wilcoxon* test [19], and is also closely related to the *Gini* coefficient.

Example 3. Continuing the example from Sect. 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

Table 2. Quality measure results for Example 3

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 %
Specificity	94.565 %	99.999995 %
False positive rate	5.435 %	0.000005 %

900,000 classified matches 650,000 were true matches (TP), and 250,000 were false matches (FP). The number of falsely non-matched record pairs (FN) was 50,000, and the number of truly 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 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 Sects. 6 and 7.

The authors of a recent publication [7] 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.

In [8] a method is described which aims at estimating the positive predictive value (precision) under the assumption that there can only be one-to-one matches (i.e. a record can only be involved in one match). Using combinatorial probabilities the number of FP is estimated, allowing quantification of the linkage quality without training data or a *gold standard* data set.

While all quality measures presented so far assume a binary classification without clerical review, a new measure has been proposed recently [25] 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 Sect. 2.3). The measure $pp = \frac{N_{P,M} + N_{P,U}}{|TP| + |FP| + |TN| + |FN|}$ is proposed, 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 proportion of record pairs that are classified as possible matches, and therefore needing manual clerical review. Low pp values are desirable, as they correspond to less manual clerical review.

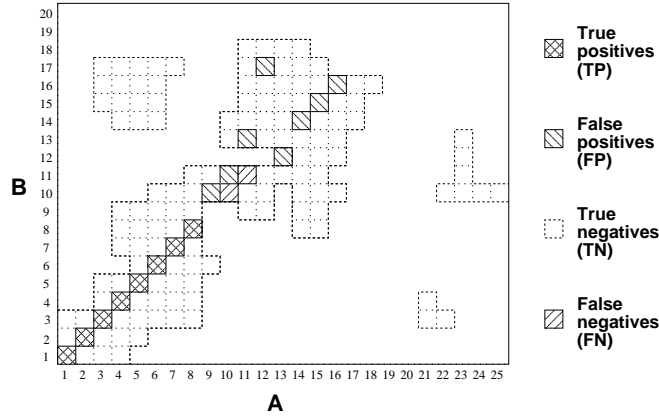


Fig. 5. Version of Fig. 4 in a blocked comparison space. The empty space are the record pairs which were removed by blocking. Besides many non-matches, the blocking process has also removed the truly matched record pairs $(A9, B9)$ and $(A12, B12)$, and then the linkage algorithm has wrongly classified the pairs $(A9, B10)$ and $(A12, B17)$ as matches

5 Blocking and Complexity Measures

An assumption in the analysis and discussion of quality measures given so far has been that all record pairs are compared. The number of comparisons in this situation equals $|\mathbf{A}| \times |\mathbf{B}|$, which is computationally feasible only for small data sets. In practise, blocking [3, 20, 49], sorting [27], filtering [24], clustering [32], or indexing [3, 10] techniques are used to reduce the number of record pair comparisons (as discussed in Sect. 2.1). Collectively known as *blocking*, these techniques aim at cheaply removing as many record pairs as possible from the set of non-matches U that are obvious non-matches, without removing any pairs from the set of matches M . Two complexity measures that quantify the efficiency and quality of such blocking methods have recently been proposed [18] (citations given refer to data linkage or deduplication publications that have used these measures):

- **Reduction ratio** [3, 18, 24] 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, 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 blocking.
- **Pairs completeness** [3, 18, 24] is measured as $pc = \frac{N_m}{|M|}$ with $N_m \leq |M|$ being the number of correctly classified truly matched record pairs in the blocked comparison space, and $|M|$ the total number of true matches as defined in Sect. 3. Pairs completeness can be seen as being analogous to recall.

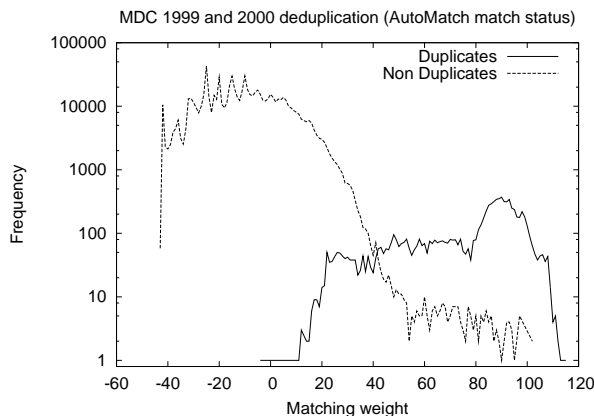


Fig. 6. The histogram plot of the matching weights for a real-world administrative health data set. This plot is based on record pair comparisons in a blocked comparison space. The lowest matching weight is -43 (disagreement on all comparisons), and the highest 115 (agreement on all comparisons). Note that the vertical axis with frequency counts is on a logarithmic scale

There is a trade-off between the reduction ratio and pairs completeness [3] (i.e. between number of removed record pairs and the number of missed true matches). 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 Sect. 4 will therefore be influenced by blocking.

6 Illustrative Example

In this section the previously discussed issues of quality and complexity measures are illustrated using a real-world administrative health data set, the *New South Wales Midwives Data Collection* (MDC) [9]. 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 linkage system *AutoMatch* [31]. 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 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 was used as the true match (or deduplication) status.

A deduplication was then performed using the *Febrl* (Freely extensible biomedical record linkage) [12] 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 numerical values were summed into a matching weight R (as discussed in Sect. 2.3) ranging from -43 (disagreement on all fourteen comparisons) to 115 (agreement on all comparisons). As can be seen in Fig. 6, 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 to process even with today’s powerful computers. Standard blocking was used to reduce the number of comparisons, resulting in 759,773 record pair comparisons (corresponding to each record being compared to around 4 other records). The reduction ratio in this case was therefore

$$rr = 1.0 - \frac{759,773}{15,349,359,655} = 1.0 - 4.9499 \times 10^{-5} = 0.99995.$$

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. 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 as duplicates by both *AutoMatch* and *Febrl*.

The quality measures discussed in Sect. 4 applied to this real-world deduplication are shown in Fig. 7 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 comparison 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). 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 graph differs only 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 [19], resulted in the least illustrative graph, especially for the full comparison space, making it not very useful for data linkage and deduplication.

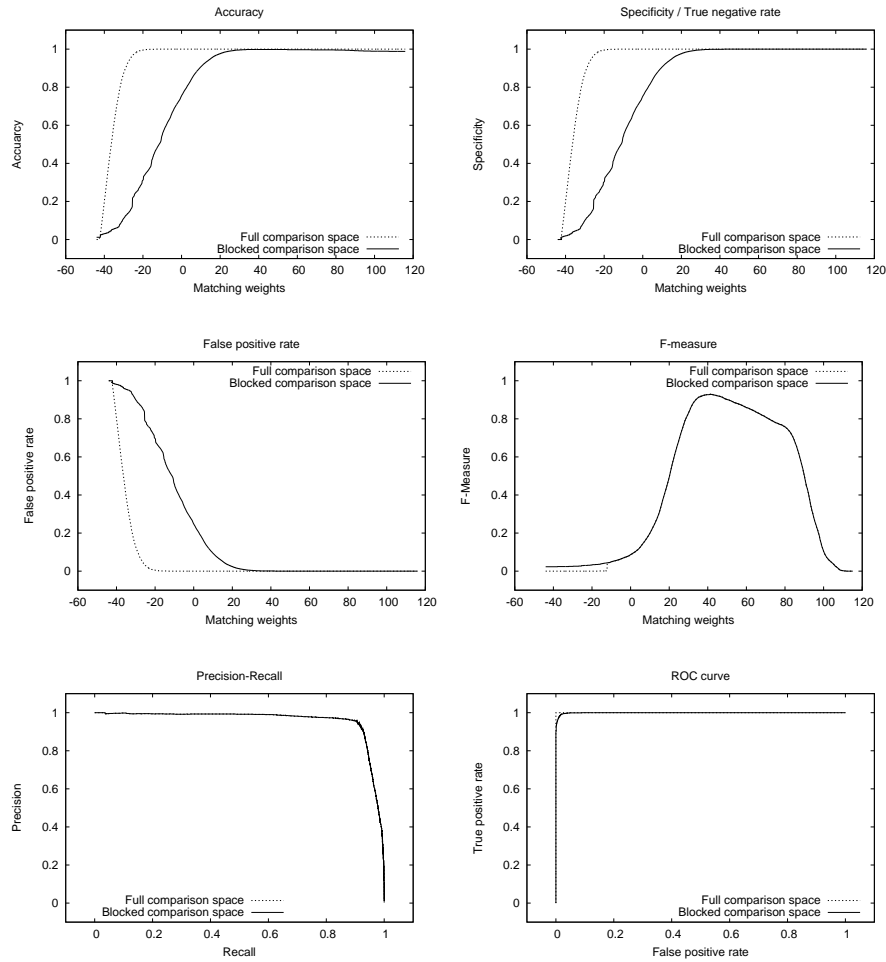


Fig. 7. Quality measurements of a real-world administrative health data set. The full comparison space (15,349,359,655 record pairs) was simulated by assuming that the record pairs removed by blocking were normally distributed with matching weights between -43 and -10 . Note that the precision-recall graph does not change at all, and the change in the F-measure graph is 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 graph, which is again due to the large number of true negatives

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 a 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.

It is known, however, that the quality of techniques vary depending on the nature of the data sets the techniques are applied to [6, 41]. Whether producing absolute or comparable results, it is 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. 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 rate, and thus ROC curve) 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 [7].

7.2 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 (resulting in a pairs completeness $pc < 1$), 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.¹

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. An overview of data linkage techniques has been presented in this chapter, 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 (over a varying threshold) 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.

¹ Note that the example given in Sect. 6 doesn’t follow the recommendations presented here. The aim of the section was to illustrate the presented issues, not the actual results of the deduplication.

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 RA, Ribeiro-Neto B. *Modern information retrieval*. Addison-Wesley Longman Publishing Co., Boston, 1999.
- [2] Bass J. Statistical linkage keys: How effective are they? In *Symposium on Health Data Linkage*, Sydney, 2002. Available online at: <http://www.publichealth.gov.au/symposium.html>.
- [3] Baxter R, Christen P, Churches T. A comparison of fast blocking methods for record linkage. In *Proceedings of ACM SIGKDD workshop on Data Cleaning, Record Linkage and Object Consolidation*, pages 25–27, Washington DC, 2003.
- [4] Bertolazzi P, De Santis L, Scannapieco M. Automated record matching in cooperative information systems. In *Proceedings of the international workshop on data quality in cooperative information systems*, Siena, Italy, 2003.
- [5] Bertsekas DP. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications*, 1:7–66, 1992.
- [6] Bilenko M, Mooney RJ. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of ACM SIGKDD*, pages 39–48, Washington DC, 2003.
- [7] Bilenko M, Mooney RJ. On evaluation and training-set construction for duplicate detection. In *Proceedings of ACM SIGKDD workshop on Data Cleaning, Record Linkage and Object Consolidation*, pages 7–12, Washington DC, 2003.
- [8] Blakely T, Salmond C. Probabilistic record linkage and a method to calculate the positive predictive value. *International Journal of Epidemiology*, 31:6:1246–1252, 2002.
- [9] Centre for Epidemiology and Research, NSW Department of Health. New South Wales mothers and babies 2001. *NSW Public Health Bull*, 13:S-4, 2001.
- [10] Chaudhuri S, Ganjam K, Ganti V, Motwani R. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of ACM SIGMOD*, pages 313–324, San Diego, 2003.
- [11] Chaudhuri S, Ganti V, Motwani R. Robust identification of fuzzy duplicates. In *Proceedings of the 21st international conference on data engineering (ICDE'05)*, pages 865–876, Tokyo, 2005.
- [12] Christen P, Churches T, Hegland M. Febrl – a parallel open source data linkage system. In *Proceedings of the 8th PAKDD, Springer LNAI 3056*, pages 638–647, Sydney, 2004.
- [13] Churches T, Christen P, Lim K, Zhu JX. Preparation of name and address data for record linkage using hidden markov models. *BioMed Central Medical Informatics and Decision Making*, 2(9), 2002. Available online at: <http://www.biomedcentral.com/1472-6947/2/9/>.
- [14] Cohen WW. Integration of heterogeneous databases without common domains using queries based on textual similarity. In *Proceedings of ACM SIGMOD*, pages 201–212, Seattle, 1998.

- [15] Cohen WW, Ravikumar P, Fienberg SE. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 workshop on information integration on the Web (IIWeb-03)*, pages 73–78, Acapulco, 2003.
- [16] Cohen WW, Richman J. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of ACM SIGKDD*, pages 475–480, Edmonton, 2002.
- [17] Cooper WS, Maron ME. Foundations of probabilistic and utility-theoretic indexing. *Journal of the ACM*, 25(1):67–80, 1978.
- [18] Elfeky MG, Verykios VS, Elmagarmid AK. TAILOR: A record linkage toolbox. In *Proceedings of ICDE*, pages 17–28, San Jose, 2002.
- [19] Fawcett T. ROC Graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto, 2004.
- [20] Fellegi I, Sunter A. A theory for record linkage. *Journal of the American Statistical Society*, 64(328):1183–1210, 1969.
- [21] Galhardas H, Florescu D, Shasha D, Simon E. An extensible framework for data cleaning. In *Proceedings of ICDE*, page 312, 2000.
- [22] Gill L. Methods for automatic record matching and linking and their use in national statistics. Technical Report National Statistics Methodology Series, no 25, National Statistics, London, 2001.
- [23] Gomatam S, Carter R, Ariet M, Mitchell G. An empirical comparison of record linkage procedures. *Statistics in Medicine*, 21(10):1485–1496, 2002.
- [24] Gu L, Baxter R. Adaptive filtering for efficient record linkage. In *SIAM international conference on data mining*, Orlando, 2004.
- [25] Gu L, Baxter R. Decision models for record linkage. In *Proceedings of the 3rd Australasian data mining conference*, pages 241–254, Cairns, 2004.
- [26] Hernandez MA, Stolfo SJ. The merge/purge problem for large databases. In *Proceedings of ACM SIGMOD*, pages 127–138, San Jose, 1995.
- [27] Hernandez MA, Stolfo SJ. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.
- [28] Kelman CW, Bass AJ, Holman CD. Research use of linked health data – a best practice protocol. *Aust NZ Journal of Public Health*, 26:251–255, 2002.
- [29] Lee ML, Ling TW, Low WL. IntelliClean: a knowledge-based intelligent data cleaner. In *Proceedings of ACM SIGKDD*, pages 290–294, Boston, 2000.
- [30] Maletic JI, Marcus A. Data cleansing: beyond integrity analysis. In *Proceedings of the Conference on Information Quality (IQ2000)*, pages 200–209, Boston, 2000.
- [31] MatchWare Technologies. *AutoStan and AutoMatch, User’s Manuals*. Kennebunk, Maine, 1998.
- [32] McCallum A, Nigam K, Ungar LH. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of ACM SIGKDD*, pages 169–178, Boston, 2000.
- [33] Monge A, Elkan C. The field-matching problem: Algorithm and applications. In *Proceedings of ACM SIGKDD*, pages 267–270, Portland, 1996.
- [34] Nahm UY, Bilenko M, Mooney RJ. Two approaches to handling noisy variation in text mining. In *Proceedings of the ICML-2002 workshop on text learning (TextML’2002)*, pages 18–27, Sydney, 2002.
- [35] Newcombe HB, Kennedy JM. Record linkage: making maximum use of the discriminating power of identifying information. *Communications of the ACM*, 5(11):563–566, 1962.

- [36] Newman DJ, Hettich S, Blake CL, Merz CJ. UCI repository of machine learning databases, 1998.
URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [37] Porter E, Winkler WE. Approximate string comparison and its effect on an advanced record linkage system. Technical Report RR97/02, US Bureau of the Census, 1997.
- [38] Pyle D. *Data preparation for data mining*. Morgan Kaufmann Publishers, San Francisco, 1999.
- [39] Rahm E, Do HH. Data cleaning: problems and current approaches. *IEEE Data Engineering Bulletin*, 23(4):3–13, 2000.
- [40] Ravikumar P, Cohen WW. A hierarchical graphical model for record linkage. In *Proceedings of the 20th conference on uncertainty in artificial intelligence*, pages 454–461, Banff, Canada, 2004.
- [41] Salzberg S. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, 1997.
- [42] Sarawagi S, Bhamidipaty A. Interactive deduplication using active learning. In *Proceedings of ACM SIGKDD*, pages 269–278, Edmonton, 2002.
- [43] Shearer C. The CRISP-DM model: The new blueprint for data mining. *Journal of Data Warehousing*, 5(4):13–22, 2000.
- [44] Smith ME, Newcombe HB. Accuracies of computer versus manual linkages of routine health records. *Methods of Information in Medicine*, 18(2):89–97, 1979.
- [45] Tejada S, Knoblock CA, Minton S. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of ACM SIGKDD*, pages 350–359, Edmonton, 2002.
- [46] Winkler WE. Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. Technical Report RR00/05, US Bureau of the Census, 2000.
- [47] Winkler WE. Methods for record linkage and Bayesian networks. Technical Report RR2002/05, US Bureau of the Census, 2002.
- [48] Winkler WE. Overview of record linkage and current research directions. Technical Report RR2006/02, US Bureau of the Census, 2006.
- [49] Winkler WE, Thibaudeau Y. An application of the Fellegi-Sunter model of record linkage to the 1990 U.S. decennial census. Technical Report RR91/09, US Bureau of the Census, 1991.
- [50] Yancey WE. BigMatch: a program for extracting probable matches from a large file for record linkage. Technical Report RRC2002/01, US Bureau of the Census, 2002.
- [51] Yancey WE. An adaptive string comparator for record linkage. Technical Report RR2004/02, US Bureau of the Census, 2004.
- [52] Zhu JJ, Ungar LH. String edit analysis for merging databases. In *KDD workshop on text mining, held at ACM SIGKDD*, Boston, 2000.
- [53] Zingmond DS, Ye Z, Ettner SL, Liu H. Linking hospital discharge and death records – accuracy and sources of bias. *Journal of Clinical Epidemiology*, 57:21–29, 2004.