

# Set Theory for Matching in Multiple Data Sets

Richard G. Clegg — richard@richardclegg.org

February 18, 2005

## Abstract

This paper describes a method for analysing problems where it is of interest to match data items between multiple data sets. The research was motivated by problems in road traffic engineering where it was sought to answer questions about the number of vehicles seen at several survey sites during licence plate surveys. However, the work in this paper should be applicable to any problems where it is of interest to match data across multiple data sets. A framework is detailed for describing such problems using basic concepts from set theory. This framework is used to provide corrections for two types of errors which may arise in matching data items across multiple data sets. The corrections are tested in simulation.

## 1 Introduction

This paper describes a framework for the analysis of matches across multiple data sets in the presence of errors. The sort of problems considered are those where it is required to identify the same item of data in multiple data files. This work uses set theory to provide a framework to investigate this type of problem and answer questions of the type “How many data items appear in each of these six files?” The framework becomes useful in analysis when there are potential errors in the matching process. Two distinct sources of error are considered: firstly, errors caused by observations being insufficient to distinguish two data items which are different; and secondly, errors caused by recording problems which mean that data items which are the same are recorded as being different. An algorithm to correct the sources of error is described. This is implemented in software and tested on example data sets.

The problem under investigation arose in the field of traffic engineering while investigating vehicle licence plate surveys. Examples throughout the paper are given as examples using licence plate surveys to aid the readers understanding. However, the method produced is extremely general and the author is very interested in finding researchers who might be interested in applying this research to their own data sets.

## 1.1 Description of the Problem

Consider data sets which come from observations of unique individuals (the phrase individual need not refer to people but can be thought of as any object which can be distinguished from other similar objects in some way). This paper addresses questions about the number of individuals who appear in all of the data sets. If the data is perfectly recorded and the individuals can all be differentiated then there is no problem to be solved. However, errors may occur in a several different ways.

The first way that errors may occur is when the data collected is insufficient to distinguish different individuals. Two or more individuals who are distinct appear to be the same individual. This can lead to the false belief that an individual has appeared in all the data sites because two or more individuals have been confused by the recording process.

The second way that errors may occur is if observations are misrecorded in the data. This could lead to an individual who appears in all the data sets not being counted since at one (or more) sites the individual has been misrecorded.

The problem addressed by this paper is to firstly describe a framework which uses set theory to describe such problems and secondly to use this framework to correct the number of matches seen in the data when errors of the two types above are present.

## 1.2 Background in Road Traffic Engineering

The work throughout this paper is explained in the context of licence plate data surveys. Indeed, the examples are given in terms of a specific format of British licence plates used between 1983 to mid 2001. However, it should be reemphasised that the work described is general and, with the assumptions which will be described, is suitable for application in any situation where it is required to match individual observations in more than two data sets in the presence of errors. The context of licence plates is given here because that is the field in which the work originally arose and because a concrete example will help with explanations.

The problem under investigation arose in the context of road traffic engineering. It is common in this area to perform licence plate surveys in order to trace a vehicle's route through the road network. It is also common, for reasons of efficiency, only to record partial licence plate data — that is, to record only a fraction of the characters on the plate. This, unfortunately, leads to the problem of “false matches”. Consider the two licence plates A123XYZ and A123ABC. If only the initial letter and the digits are recorded then these two vehicles cannot be distinguished. At first this might not appear to be a serious objection. After all, the chances of any two vehicles having the same partial licence plate are relatively small. However, consider that if two licence plate surveys each have one thousand vehicles then this is one million pairs of vehicles. The chances of one or more of these pairs having the same partial licence plates becomes extremely high. In real data sets the number of false matches can easily exceed

the number of genuine matches.

The problem is somewhat complicated by the combinatorics of the situation. Consider recordings of licence plates made at three sites. Taking three identical partial licence plates, one from each site, then this could represent any of five situations: the same vehicle seen at each of three sites; three different vehicles, one at each site; one vehicle at site one and another at sites two and three; one vehicle at site two and another at sites one and three; or one vehicle at site three and another at sites one and two. Call these possibilities *types of match*. When four or five sites are under considerations then even evaluating how many types of match there are becomes a significant problem.

The problem of identifying false matches in licence plate survey data is something of a classic problem in road traffic engineering. However, it has only been considered with respect to matching vehicles between two sites at a time. An early approach for two sites using a simple probabilistic correction is given by [4]. The problem is dealt with by [5] in a number of ways including the possibility of picking matches between any of a number of site pairs around a junction. A graphical procedure based upon analysing journey time between two sites is given in [7]. Further research in the area including a maximum likelihood estimator based upon assumptions about travel time distributions are given in [8] and [6]. All of these methods in the literature deal with only matches between site pairs although in some the site pairs are selected from several sites.

The method described here offers now new insight for matches between pairs of sites and, if matches are only sought between two sites (even if there are several data sets) then one of the methods in the other papers is likely to be more appropriate.

### 1.3 Notation Used in this Paper

Throughout this paper bold lower case ( $\mathbf{x}$ ) is used to indicate a tuple (ordered set) and subscripted lower case will be used to indicate elements of that tuple, for example,  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is an n-tuple. Upper case ( $M$ ) is used to indicate a set and bold upper case ( $\mathbf{M}$ ) is used to indicate a set of sets. The notation  $\#$  is used to indicate the cardinality (number of members) of a set, for example  $\#\mathbf{M}$ . Caligraphic lettering ( $\mathcal{S}$ ) is used to indicate higher order entities such as sets of tuples or sets of sets of sets.

The following specific notation is used.

- $n$  — the number of sites under investigation.
- $S_i$  — the set of observations at site  $i$ .
- $\mathbf{x} = (x_1, \dots, x_n)$  — an n-tuple of observations (see Definition 2.1). Note that  $\mathbf{y}$  and  $\mathbf{z}$  will also be used for n-tuples.
- $\mathcal{S}$  — the set of all possible n-tuples of observations in the data  $S_1, S_2, \dots, S_n$  (see Definition 2.1).

- $\mathcal{M}_n$  — the set of all possible partitions of the first  $n$  integers (see Definition 2.3).
- $C(\mathbf{x})$  — a member of  $\mathcal{M}_n$  representing the *type of match* of  $\mathbf{x}$  (see Definition 2.4).
- $\mathbf{A}_n$  — the set  $\{\{1, 2, \dots, n\}\}$  representing the same observation across all  $n$  sites (see Definition 2.5).
- $\mathbf{x}^*$  — a *partial observation* derived from  $\mathbf{x}$  (see Definition 3.1).
- $\mathcal{S}^*$  — the set of all partial observations in the data (see Definition 3.1).
- $x(\mathbf{x}, \mathbf{M})$  — the *exact match function* which is one if an  $n$ -tuple of observations  $\mathbf{x}$  is a match of type  $\mathbf{M}$  (see Definition 3.3).
- $X(\mathbf{M})$  — the *exact match function* which counts the number of matches of type  $\mathbf{M}$  in  $\mathcal{S}$  (see Definition 3.3).
- $r(\mathbf{x}, \mathbf{M})$  — the *relaxed match function* which is analogous to  $x(\mathbf{x}, \mathbf{M})$  (see Definition 3.4).
- $R(\mathbf{M})$  — the *relaxed match function* which is analogous to  $X(\mathbf{M})$  (see Definition 3.4).
- $p(i)$  — the probability of  $i$  distinct individuals appearing the same in partial data (see Definition 3.6).

## 2 Defining “Type of Match”

Consider observations taken at  $n$  sites. The term site will be used throughout this paper to refer to a place and time where data has been collected. However, this could represent data being collected at a single geographical location on  $n$  different occasions or any of a number of other possibilities.

Let  $S_i$  be the set of observations made at the  $i$ th site. So, for example, if  $n = 3$  then a very small data set might be

$$\begin{aligned} S_1 &= \{\text{A123XYZ}, \text{C789ABC}\} \\ S_2 &= \{\text{A123XYZ}, \text{A123XDR}, \text{D555SDD}\} \\ S_3 &= \{\text{C789ABC}, \text{A123XYZ}\}. \end{aligned}$$

Note that formally in set theory it is usually required that all members of a set are distinct. If this requirement is a problem for the particular sets of observations being considered then the individual observations could be tagged with an identifying label which was ignored when comparing them. This minor technicality does not affect anything which follows.

When considering matches between sites, the obvious unit to consider is an  $n$ -tuple of observations with one observation taken from each site.

**Definition 2.1.** An n-tuple of observations  $\mathbf{x} = (x_1, \dots, x_n)$  is an n-tuple where  $x_i \in S_i$  for all  $i$ . Let  $\mathcal{S}$  be the set of all such n-tuples which is given by

$$\mathcal{S} = S_1 \times S_2 \cdots \times S_n.$$

Continuing the previous example

$$\begin{aligned} \mathcal{S} = \{ & (\text{A123XYZ}, \text{A123XYZ}, \text{C789ABC}), \\ & (\text{A123XYZ}, \text{A123XYZ}, \text{A123XYZ}), \\ & \dots (\text{C789ABC}, \text{D555SDD}, \text{A123XYZ}) \}, \end{aligned}$$

where  $\mathcal{S}$  has  $2.3.2 = 12$  members each of which is an triple of observations.

The next step is to consider which elements within an n-tuple are equal. Those n-tuples  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{S}$  where  $x_i = x_j$  for all  $i, j$  are clearly the n-tuples most of interest since those are the ones which represent the same individual being observed at all sites. In the example under consideration, this would be the triple  $(\text{A123XYZ}, \text{A123XYZ}, \text{A123XYZ})$ . Intuitively, it is clear that the triples  $(\text{A123XYZ}, \text{A123XDR}, \text{A123XYZ})$  and  $(\text{C789ABC}, \text{A123XDR}, \text{C789ABC})$  are structurally similar in some way (they match at sites one and three) and both are structurally different to  $(\text{A123XYZ}, \text{A123XYZ}, \text{C789ABC})$ . Two n-tuples of observations  $\mathbf{x} = (x_1, \dots, x_n)$  and  $\mathbf{y} = (y_1, \dots, y_n)$  are the same type of match if whenever two elements of  $\mathbf{x}$  are equal the same two elements of  $\mathbf{y}$  are equal and vice versa.

**Definition 2.2.** Let  $\mathbf{x}$  and  $\mathbf{y}$  be two n-tuples of observations.

$$\begin{aligned} \mathbf{x} \sim \mathbf{y} \text{ if and only if } & (x_i = x_j) \Leftrightarrow (y_i = y_j) \\ & \text{for all } i, j \in \{1, 2, \dots, n\}. \end{aligned}$$

If  $\mathbf{x} \sim \mathbf{y}$  then the two n-tuples are said to be the same *type of match*.

Note that, for simplicity, limits on indices  $(i, j \in \{1, 2, \dots, n\})$  will be dropped in future definitions where they are obvious.

This definition of  $\sim$  meets the requirements of an equivalence relation in set theory.

## 2.1 Defining $\mathcal{M}_n$ , the Set of All Types of Match

In Definition 2.2 an equivalence relation was used to define when two n-tuples are the same type of match. In this section, the number of possible types of match will be calculated. To do this, it is necessary to introduce an object which can represent each type of match. This is achieved using partitions of the integers.

**Definition 2.3.** The set  $\mathcal{M}_n$  is the set of all possible partitions of the first  $n$  integers where a partition is defined as a set of disjoint sets, the members of which are the integers from one to  $n$ .

For  $n = 1$  only the partition  $\{\{1\}\}$  is in  $\mathcal{M}_1$ . For  $n = 2$ , two possible partitions are available  $\{\{1, 2\}\}$  and  $\{\{1\}, \{2\}\}$ . For  $n = 3$ , five partitions are available. The enumeration of  $\#\mathcal{M}_n$  is well understood and uses the Bell numbers [2]. The sequence of the Bell numbers begins 1, 2, 5, 15, 52, 203, 877, 4140, 21147. If  $\mathbf{M} = \{M_1, \dots, M_m\}$  is one of the partitions in  $\mathcal{M}_n$  the sets  $M_i$  are sometimes referred to as blocks.

**Definition 2.4.** The *type of match* of an  $n$ -tuple of observations  $\mathbf{x} = (x_1, \dots, x_n)$  is given by  $C(\mathbf{x}) \in \mathcal{M}_n$  where  $C(\mathbf{x}) = \mathbf{M} = \{M_1, \dots, M_m\}$  is the partition of the first  $n$  integers which satisfies  $(x_i = x_j) \Leftrightarrow i, j \in M_k$  for some  $k \in \{1, 2, \dots, m\}$ . That is,  $\mathbf{M}$  is the partition chosen such that any two site indices are in the same block within  $\mathbf{M}$  if and only if the observations in  $\mathbf{x}$  at those sites are equal.

It is clear that  $C(\mathbf{x})$  is uniquely specified by this definition. To continue with the earlier example, if  $\mathbf{x} = (\mathbf{A123XYZ}, \mathbf{A123XYZ}, \mathbf{C789ABC})$  then  $C(\mathbf{x}) = \{\{1, 2\}\{3\}\}$  and if  $\mathbf{x} = (\mathbf{A123XYZ}, \mathbf{A123XYZ}, \mathbf{A123XYZ})$  then  $C(\mathbf{x}) = \{\{1, 2, 3\}\}$ .

To be sure that  $C(\mathbf{x})$  can be used to represent types of match it must be shown that it is consistent with the definition of  $\sim$  already given. This is equivalent to proving the following theorem.

**Theorem 2.1.** For  $n$ -tuples of observations  $\mathbf{x}$  and  $\mathbf{y}$  then

$$C(\mathbf{x}) = C(\mathbf{y}) \text{ if and only if } \mathbf{x} \sim \mathbf{y}.$$

*Proof.* It must be shown that  $\mathbf{x} \sim \mathbf{y} \Rightarrow C(\mathbf{x}) = C(\mathbf{y})$  and also that  $C(\mathbf{x}) = C(\mathbf{y}) \Rightarrow \mathbf{x} \sim \mathbf{y}$ .

Let  $\mathbf{M}_x = C(\mathbf{x})$  and  $\mathbf{M}_y = C(\mathbf{y})$ . Assume that  $\mathbf{x} \sim \mathbf{y}$  and therefore that  $(x_i = x_j) \Leftrightarrow (y_i = y_j)$  (from Definition 2.2). If  $i, j$  are in the same set in  $\mathbf{M}_x$  then  $x_i = x_j$  (from Definition 2.4). Therefore it must be the case that  $y_i = y_j$  and hence  $i, j$  are in the same set in  $\mathbf{M}_y$ . By similar reasoning, if  $i, j$  are not in the same set in  $\mathbf{M}_x$  then they cannot be in the same set in  $\mathbf{M}_y$ . Hence the assumption  $\mathbf{x} \sim \mathbf{y}$  implies  $\mathbf{M}_x = \mathbf{M}_y$ .

Assume that  $\mathbf{M}_x = \mathbf{M}_y$ . If  $x_i = x_j$  then  $i, j$  are in the same set in  $\mathbf{M}_x$  (which equals  $\mathbf{M}_y$ ) and therefore  $y_i = y_j$ . If  $x_i \neq x_j$  then  $i, j$  cannot be in the same set in  $\mathbf{M}_x$  (which equals  $\mathbf{M}_y$ ) and hence  $y_i \neq y_j$ . This means that  $x_i = x_j \Leftrightarrow y_i = y_j$ . Since  $x_i = x_j \Leftrightarrow y_i = y_j$  then  $\mathbf{x} \sim \mathbf{y}$  from Definition 2.2.  $\square$

In terms of  $\mathcal{M}_n$ , the partition most of interest is the partition with all elements in the same block,  $\{\{1, 2, \dots, n\}\}$ . This represents the case when all the observations are equal (in other words, an  $n$ -tuple where the same individual has been seen at all  $n$  sites).

**Definition 2.5.** Let  $\mathbf{A}_n \in \mathcal{M}_n$  represent a the type of match where all observations are equal. Therefore,  $\mathbf{A}_n = \{\{1, 2, \dots, n\}\}$ .

### 3 The Problem of False Matches

So far it has been assumed that all observations made have made with no error and that it is an easy matter to see when two individual elements of an n-tuple are the same. In this section, this assumption is relaxed in a certain way by introducing the notion of a *partial observation*. In the traffic engineering example previously, this is equivalent to reading only part of a licence plate as discussed in the introduction. So, for example, given an n-tuple

$$\mathbf{x} = (\text{A123XYZ}, \text{A123XDR}, \text{C789ABC}),$$

then the partial observations could be,

$$\mathbf{x}^* = (\text{A123}, \text{A123}, \text{C789}),$$

if it were chosen to observe only the initial letter and three numbers. The process of taking only a partial observation has changed the type of match of  $\mathbf{x}$  so that  $\mathbf{x} \not\sim \mathbf{x}^*$  and  $C(\mathbf{x}) \neq C(\mathbf{x}^*)$ .

The important thing to note about the partial observations is that they capture only part of the information necessary to distinguish two observations. In other words, two individuals who would appear distinct if full information would be available may appear to be the same if only partial observations are available. However, a single individual observed at two sites can never appear to be two distinct individuals due to a partial observation. This is captured in the following definition.

**Definition 3.1.** Let  $\mathbf{x} = (x_1, \dots, x_n)$  be an n-tuple of observations. If  $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$  is the n-tuple of *partial observations* derived from  $\mathbf{x}$ . then the following relation holds,

$$(y_i = y_j) \Rightarrow (y_i^* = y_j^*).$$

This star notation will also be used to distinguish the set of all possible partial observations in the data  $\mathcal{S}^*$  and, in general, to distinguish functions which apply to partial data rather than the full data.

It is worth noting that this definition is very similar to that for the original equivalence relation  $\sim$  but with the implication only in one direction.

#### 3.1 A Partial Ordering for False Matches

In order to address the problems created by the partial observations, it is necessary to understand in which ways the type of match of observations can change when only partial observations are considered. This is done using the set theoretic idea of a partial ordering. (The clash of names between partial observation and partial ordering is unfortunate although the two concepts are related in this paper, the similarity in names is purely a coincidence.)

**Definition 3.2.** For two partitions,  $\mathbf{M} = \{M_1, \dots, M_m\} \in \mathcal{M}_n$  and  $\mathbf{M}' = \{M'_1, \dots, M'_{m'}\} \in \mathcal{M}_n$ , then a partial ordering  $\succsim$  is given by,

$$\mathbf{M} \succsim \mathbf{M}' \text{ if and only if } (i, j \in M_k) \Rightarrow (i, j \in M'_l),$$

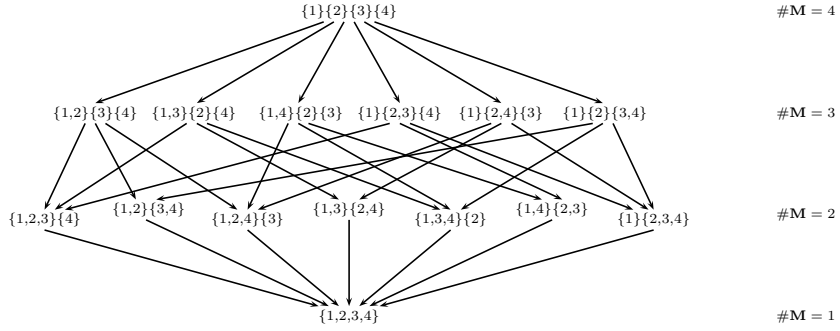


Figure 1: Hasse diagram for  $\mathcal{M}_4$ .

for some  $k$  and  $l$ . Put more simply,  $\mathbf{M} \succsim \mathbf{M}'$  if whenever  $i$  and  $j$  are in the same set within  $\mathbf{M}$  then they are also in the same set within  $\mathbf{M}'$ .

The symbol  $\succ$  will be used to mean *strictly succeeds*. That is  $\mathbf{x} \succ \mathbf{y}$  means  $\mathbf{x} \succsim \mathbf{y}$  and  $\mathbf{x} \not\sim \mathbf{y}$ . The symbol  $\succ\prec$  will be used to mean *immediate successor* that is, if  $\mathbf{x} \succ\prec \mathbf{z}$  then  $\mathbf{x} \succ \mathbf{z}$  but there is no  $\mathbf{y}$  such that  $\mathbf{x} \succ \mathbf{y} \succ \mathbf{z}$ . The symbols  $\succ$ ,  $\succsim$  and  $\prec\prec$  will have their obvious meanings.

It can trivially be shown that this relation meets the conditions to be a partial ordering on the set  $\mathcal{M}_n$ . This partial ordering relates to partial observations via the following theorem.

**Theorem 3.1.** *Let  $\mathbf{x}$  be an  $n$ -tuple of observations.*

$$C(\mathbf{x}^*) \prec\prec C(\mathbf{x}).$$

*Proof.* Let  $\mathbf{M} = (M_1, \dots, M_m) = C(\mathbf{x})$  and  $\mathbf{M}' = (M'_1, \dots, M'_{m'}) = C(\mathbf{x}^*)$ . The theorem follows trivially from the relation given in 3.2. If  $i, j \in M_k$  for some  $k$  then  $x_i = x_j$  and hence  $x_i^* = x_j^*$  which in turn implies,  $i, j \in M'_l$  for some  $l$ . Therefore,  $(i, j \in M_k) \Rightarrow (i, j \in M'_l)$  which is Definition 3.2.  $\square$

This theorem shows that if only partial observations are recorded then the type of match of an  $n$ -tuple of observations can only change in certain ways. This can be visualised using a Hasse diagram. A Hasse diagram is constructed by plotting a partially ordered set  $S$  graphically in such a way that for all  $\mathbf{x}, \mathbf{y} \in S$  if  $\mathbf{x} \prec \mathbf{y}$  then  $\mathbf{x}$  is physically lower on the diagram than  $\mathbf{y}$ . An arrow is drawn in a Hasse diagram from  $x$  to  $y$  if  $x \succ\prec y$ . Figure 1 shows the Hasse diagram of  $\mathcal{M}_4$  with the partial ordering given by Definition 3.2.

### 3.2 Defining Some Counting Functions

The next step in this procedure is to define some functions to count the number of tuples which have a given type of match.

**Definition 3.3.** Let  $\mathbf{x}$  be an  $n$ -tuple of observations and  $\mathbf{M} \in \mathcal{M}_n$  be a type of match. The *exact matching function* for an observation  $\mathbf{y}$  is defined by,

$$x(\mathbf{x}, \mathbf{M}) = \begin{cases} 1 & \text{if and only if } C(\mathbf{x}) = \mathbf{M} \\ 0 & \text{otherwise .} \end{cases}$$

The *exact matching function* for  $\mathcal{S}$  is given by,

$$X(\mathbf{M}) = \sum_{\mathbf{x} \in \mathcal{S}} x(\mathbf{x}, \mathbf{M}).$$

An obvious extension to this is to allow a function which counts not just the given type  $\mathbf{M}$  but all types which are of a preceding type according to the relation  $\preceq$ .

**Definition 3.4.** Let  $\mathbf{x}$  be an  $n$ -tuple of observations and  $\mathbf{M} \in \mathcal{M}_n$  be a type of match. The *relaxed matching function* for an observation is defined by,

$$r(\mathbf{x}, \mathbf{M}) = \begin{cases} 1 & \text{if and only if } C(\mathbf{x}) \preceq \mathbf{M} \\ 0 & \text{otherwise .} \end{cases}$$

Equivalently,

$$r(\mathbf{y}, \mathbf{M}) = \sum_{\mathbf{M}' \preceq \mathbf{M}} x(\mathbf{x}, \mathbf{M}').$$

The *relaxed matching function* for  $\mathcal{S}$  is given by

$$R(\mathbf{M}) = \sum_{\mathbf{x} \in \mathcal{S}} r(\mathbf{x}, \mathbf{M}).$$

Equivalently,

$$R(\mathbf{M}) = \sum_{\mathbf{M}' \preceq \mathbf{M}} X(\mathbf{M}').$$

It is important to note the implication that if  $\mathbf{M} = \{\{1, 2, \dots, n\}\}$  then  $r(\mathbf{x}, \mathbf{M}) = x(\mathbf{x}, \mathbf{M})$  and  $R(\mathbf{M}) = X(\mathbf{M})$  since there are no  $\mathbf{M}' \in \mathcal{M}_n$  such that  $\mathbf{M}' \prec \mathbf{M}$  in this case.

The problem of finding the number of  $n$ -tuples which are observations of the same individual at all of the  $n$  sites can now be thought of as the problem of evaluating  $X(\mathbf{A}_n)$ . In the case of partial observations, however, only observations of  $\mathcal{S}^*$  are available not  $\mathcal{S}$ .

Some important relations can be defined between these counting functions which will allow a solution to the false matching problem.

**Lemma 3.2.** *Any exact matching function can be expressed in terms of relaxed matching functions and “lower” exact matching functions.*

$$X(\mathbf{M}) = R(\mathbf{M}) - \sum_{\mathbf{M}' \prec \mathbf{M}} X(\mathbf{M}').$$

*Proof.* This follows trivially from Definition 3.4.  $\square$

This lemma can be applied recursively so that for any  $\mathbf{M}$ ,  $X(\mathbf{M})$  can be expressed as a function of the  $R(\mathbf{M}')$  for all  $\mathbf{M}' \lesssim \mathbf{M}$ . This can be thought of as being a version of the inclusion/exclusion principle for partitions of the integers under this partial ordering.

It is now useful to consider matches in subsets of the original set of  $n$  sites.

**Definition 3.5.** Let  $M = \{m_1, \dots, m_k\}$  be a set of  $k$  (where  $k \geq 1$ ) distinct integers, such that all  $m_i \in \{1, 2, \dots, n\}$ . In other words,  $M$  is some subset of the original  $n$  sites. Let  $\mathcal{S}'$  be the set of tuples of observations formed by the cartesian product,

$$\mathcal{S}' = S_{m_1} \times S_{m_2} \times \dots \times S_{m_k}.$$

That is,  $\mathcal{S}'$  is the set of tuples of observations over some subset of the original  $n$  sites. Then define,

$$T(M) = X(\mathbf{A}_k),$$

where the exact match  $X(\mathbf{A}_k)$  is in this case over the tuples in  $\mathcal{S}'$  rather than the  $n$ -tuples in  $\mathcal{S}$ . In other words,  $T(M)$  is the number of individuals seen at all sites in the set  $M$ .

The problem of evaluating  $T(M)$  is either the same as the original problem of evaluating  $X(\mathbf{A}_n)$  (if  $M = \{1, 2, \dots, n\}$ ) or it is a sub-problem of finding matches in a smaller number of sites. Because of the way that matching has been defined, if the set  $M$  has a single member, then  $T(\{i\}) = \#S_i$  (because, for a single site,  $\mathbf{A}_1 = \{\{1\}\}$  and every 1-tuple has this match type).

**Lemma 3.3.** *The relaxed matching function  $R(\mathbf{M})$  where  $\mathbf{M} = \{M_1, \dots, M_m\} \in \mathcal{M}_n$  can be expressed as a product of exact matches over subsets of sites using the expression,*

$$R(\mathbf{M}) = \prod_{i=1}^m T(M_i).$$

*Proof.* Let  $\mathbf{x}$  be some  $n$ -tuple of observations in  $\mathcal{S}$ . From Definition 3.4 then,

$$r(\mathbf{x}, \mathbf{M}) = \begin{cases} 1 & \text{if for all } i, j, k \text{ then} \\ & (i, j \in M_k) \Rightarrow (x_i = x_j) \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$R(\mathbf{M}) = \sum_{\mathbf{x} \in \mathcal{S}} r(\mathbf{x}, \mathbf{M}) = \#\{\mathbf{x} \in \mathcal{S} : (i, j \in M_k) \Rightarrow (x_i = x_j) \text{ for all } i, j, k\}.$$

Since  $\mathcal{S}$  is a Cartesian product, it can be seen that those members of  $\mathcal{S}$  which meet the set condition are exactly those which would be picked out by having all members the same for the sites defined by the blocks in  $\mathbf{M}$  which leads to the conclusion that the right hand side is  $\prod_{i=1}^m T(M_i)$  as required.  $\square$

This lemma allows the problem of evaluating a relaxed match function to a problem related to the original problem of evaluating  $X(\mathbf{A}_n)$  but on some subset of sites.

**Definition 3.6.** The probability  $p(i)$  is the probability that  $i$  individuals observed at  $i$  different sites who are distinct in the full data appear to all be identical in the partial data. By convention  $p(1) = 1$ .

This can be thought of simply as the probability of a false match between  $i$  individuals. In the case of licence plate analysis it is the probability that  $i$  randomly chosen different licence plates have the same partial plate. Note that this definition implicitly puts some restriction on the types of data which can be analysed. For example, it has been assumed that  $p(i)$  does not depend on the individual sites at which the observations are made. This does not imply that the distribution of individuals at each site has to be identical. It is likely that these assumptions could be weakened with a different formulation though at the price of complicating the analysis. Measuring the value of  $p(i)$  in real-life may be complex and the details of how this is done for a particular data set are probably not of general interest. An obvious approach is available if data sets are available where it is known a priori that no genuine matches should occur. For more details of the problem in the context of licence plate surveys see [3, Chapter 5]. For the purposes of this paper, it will simply be assumed that  $p(i)$  is independent of sites and can be calculated in some manner from the data collected or other knowledge about the system under study.

**Lemma 3.4.** An unbiased estimator  $\hat{t}$  for  $X(\mathbf{A}_n)$  is given by,

$$\hat{t} = X^*(\mathbf{A}_n) - \sum_{\mathbf{M} \succ \mathbf{A}_n} p(\#\mathbf{M})X(\mathbf{M}).$$

*Proof.* The quantity  $X^*(\mathbf{A}_n)$  is equal to  $X(\mathbf{A}_n)$  plus all those n-tuples which have moved into  $X^*(\mathbf{A}_n)$  because of a false match which arose due to the partial observation. The sum on the right hand side of the equation relates to the ways in which this can occur.

Formally, the  $X^*(\mathbf{A}_n)$  must be reduced by removing those n-tuples which were not matches of type  $\mathbf{A}_n$  but became so due to the partial observation. For each type of match, in  $\mathcal{M}_n$  apart from  $\mathbf{A}_n$  then the set  $\{\mathbf{y} \in \mathcal{S} : C(\mathbf{y}^*) = \mathbf{A}_n, C(\mathbf{y}) = \mathbf{M}\}$  is the set of n-tuples in the data  $\mathcal{S}$  which are a match of type  $\mathbf{M}$  in the complete data but appear to be a match of type  $\mathbf{A}_n$  in the partial data  $\mathcal{S}^*$ . Writing this as an equation,

$$\hat{t} = X^*(\mathbf{A}_n) - \sum_{\mathbf{M} \succ \mathbf{A}_n} \mathbb{E}[\#\{\mathbf{y} \in \mathcal{S} : C(\mathbf{y}^*) = \mathbf{A}_n, C(\mathbf{y}) = \mathbf{M}\}].$$

For an n-tuple of observations which is of type  $\mathbf{M}$  then the number of distinct individuals in this n-tuple must be  $\#\mathbf{M}$ . Therefore,

$$\mathbb{P}[C(\mathbf{y}^*) = \mathbf{A}_n | C(\mathbf{y}) = \mathbf{M}] = p(\#\mathbf{M}).$$

Bayes theorem gives,

$$\begin{aligned} & \mathbb{P}[C(\mathbf{y}^*) = \mathbf{A}_n, C(\mathbf{y}) = \mathbf{M}] \\ &= p(\#\mathbf{M})\mathbb{P}[C(\mathbf{y}) = \mathbf{M}] \\ &= \frac{p(\#\mathbf{M})X(\mathbf{M})}{\#\mathcal{S}}. \end{aligned}$$

Hence, the expected number of false matches arising from each type of match can be given by,

$$\begin{aligned} & \mathbb{E}[\#\{\mathbf{y} \in \mathcal{S} : C(\mathbf{y}^*) = \mathbf{A}_n, C(\mathbf{y}) = \mathbf{M}\}] \\ &= \#\mathcal{S}\mathbb{P}[C(\mathbf{y}^*) = \mathbf{A}_n, C(\mathbf{y}) = \mathbf{M}] p(\#\mathbf{M})X(\mathbf{M}), \end{aligned}$$

and the lemma follows immediately.  $\square$

From this lemma, it is possible to get an estimate of  $X(\mathbf{A}_n)$  if all  $p(i)$  are known and if the  $X(\mathbf{M})$  can somehow be estimated. Note that  $X^*(\mathbf{A}_n)$  is measured on the partial data and can therefore be directly calculated from the data.

### 3.3 An Algorithm to Solve the Problem

It is not immediately obvious, but the three lemmas, 3.2, 3.3 and 3.4 allow a solution to the problem of false matches. Firstly, 3.4 allows an estimate for  $X(\mathbf{A}_n)$  to be found, if the values of  $X(\mathbf{M})$  are known for all  $\mathbf{M} \in \mathcal{M}_n : \mathbf{M} \neq \mathbf{A}_n$  (the  $p(i)$  are assumed to be known and  $X^*(\mathbf{A}_n)$  can be measured from the data).

Lemma 3.2 allows any exact match to be expressed as a sum of relaxed matches and Lemma 3.3 allows any relaxed match to be expressed as a product of  $T(M)$  where  $M$  is some subset of sites. If  $M$  is the complete set of sites then this quantity is  $X(\mathbf{A}_n)$  which is the quantity sought. Otherwise, the quantity  $T(M)$  is a match across a smaller number of sites and can be thought of as a simpler subproblem of the original problem. Hence, an equation can be created with terms which are either  $X(\mathbf{A}_n)$  or terms in  $T(M)$  where  $M$  is a strict subset. The subproblems of solving for all the matches must be solved first and this is done recursively. When the match is only over one site, the problem is trivial. Java code which solves this problem can be found online at:

<http://www.richardclegg.org/matching>.

Furthermore, it can be simply shown that the result of this algorithm produces an unbiased estimate of the number of matches in the full data. If the estimates for the relaxed matches are unbiased, then the final estimator is a sum of unbiased estimators and, hence, must itself be unbiased. The estimators for the relaxed matches are unbiased estimators if the results produced by the subproblem of matching on a reduced set of sites are unbiased. This is certainly true when the set of sites is a single site and, hence, the final estimate produced by the algorithm is, itself, unbiased.

## 4 The Problem of Recording Errors

A problem which is, in some ways, the converse of the previously discussed problem is that of erroneously recorded data. This is, in some ways, a harder problem to deal with correctly than that of false matching previously discussed. Two simplifying assumptions are necessary in order to get an approximate solution.

The first assumption is that, at each site, there is an error rate  $\varepsilon_i$  which is the probability that any single observation at site  $i$  (that is an observation of one individual at that site) is recorded incorrectly. This is obviously a simplification in many situations. For example, in the case of licence plate surveys, a plate which is easily misread at one site (for example because it is partly obscured by dirt) might be easily misread at another site. It is possible that this assumption could be weakened in future work, for example, by using techniques from capture/recapture experiments in biology [1] where trap-shy and trap-happy animals (ones which become more likely or less likely to be recorded having been recorded once) are important experimental effects.

The second simplifying assumption is that a misread observation will never match with another observation. This assumption is not met in real life but the problem is much more difficult if this assumption is not made. The method describe here can, therefore, only be considered as an approximate solution to the problem. Formally, if  $\mathbf{x}$  is an  $n$ -tuple of the true observations (without recording errors) and  $\mathbf{x}'$  is that  $n$ -tuple after possible recording errors have happened then  $x'_i = x'_j \Rightarrow x_j = x_j$ . From Definition 3.2,

$$C(\mathbf{x}') \lesssim C(\mathbf{x}).$$

This can be used to provide a correction in a similar manner to the partial observations problem. Note that the situation is not completely symmetric with the partial observations problem since in that problem, an  $n$ -tuple of type  $C(\mathbf{y})$  to appear in any  $C(\mathbf{y}^*) \lesssim C(\mathbf{y})$  but it is not the case that a misreading can cause an  $n$ -tuple of type  $C(\mathbf{y})$  to appear in any  $C(\mathbf{y}') \lesssim C(\mathbf{y})$ . Note also that if this second assumption that a misreading will never cause a false match is dropped then a switch between any two types of match is possible as a result of misreadings hence the problem is much harder to solve.

Given both assumptions then the problem here is much simpler than the problem of partial matches. Let  $T$  be the total number of  $n$ -tuples of observations which are matches of type  $\mathbf{A}_n$ . Let  $T'$  be the total number of observations which are matches of type  $\mathbf{A}_n$  after recording errors. Then

$$T = T' + \sum_{\mathbf{M}} p(\mathbf{M})T,$$

where the sum is over all  $\mathbf{M} \in \mathcal{M}_n$  which are possible types of match that could be reached by a recording error and  $p(\mathbf{M})$  is the probability that an  $n$ -tuple which is a match of type  $\mathbf{A}_n$  will become a match of type  $\mathbf{M}$  after errors. Therefore this gives the correction,

$$T = \frac{T'}{1 - \sum_{\mathbf{M}} p(\mathbf{M})},$$

which can be applied to the observed number of matches at  $T'$  to get a corrected result.

The sum is, in fact, easily performed. The different types of recording error can be enumerated by considering the probabilities of errors at every combination of sites. For example, consider  $n = 6$ . If there are errors at sites two and four then this would produce a match of type:

$$\mathbf{M} = \{\{1, 3, 5, 6\}, \{2\}, \{4\}\}.$$

The probability  $p(\mathbf{M})$  is then given by the product:

$$(1 - \varepsilon_1)\varepsilon_2(1 - \varepsilon_3)\varepsilon_4(1 - \varepsilon_5)(1 - \varepsilon_6).$$

The correction procedure described can be applied to  $T'$  which is the output of the partial observation correction procedure. This correction is only an approximation due to the simplifying assumptions made. It is tested by simulation in the next section of the paper.

## 5 Simulation Results

A simulation was written to test the algorithm. While the algorithm can, and has, been used on real data [3, Chapter Five], these results are of little use in determining the correctness of the algorithm since the correct answer is not known. The simulation has the following features.

- The number of vehicle types which can be distinguished in the partial data is defined .
- The error rate  $\varepsilon_i$  at each site is defined.
- A number of flows are defined as a set of one or more sites and a number of vehicles.
- For each of these flows, where  $f$  is the flow specified, then repeat the following  $f$  times:
  1. A random number generator with a flat distribution is used to pick a vehicle type (each type has a unique associated partial licence plate).
  2. For each site at which this flow is present a random number is chosen with a flat distribution in  $[0, 1]$ .
  3. If the number chosen is larger than  $\varepsilon_i$  (where  $i$  is the site) then write this partial plate to the file for site  $i$ .
  4. Otherwise assume that a misreading has occurred, choose a new vehicle type and write this new type to the file for site  $i$ .

It can be seen that if a flow of one hundred vehicles is defined, for example, as occurring at sites 1,3 and 5, then one hundred vehicles will be generated and placed in the observation files for sites 1, 3 and 5. This represents a hundred vehicles which should genuinely be seen at those sites. However, the licence plate observations may be misread (with probability  $\varepsilon_i$  at each site). Furthermore, since there is a finite number of vehicle types, then there is a chance of

accidental matches (equivalent to matches in the partial data) occurring. Thus, the reported number of tuples of observations which are identical at sites 1, 3 and 5 may be much higher or lower than one hundred.

There are several possible criticisms of this simulation procedure. Obviously, in reality, a misreading will not generate a completely random new vehicle type but will probably generate one which is somehow close to the original (e.g. misreading just a single letter). As discussed previously, in reality, a vehicle misread at one site is more likely to be misread at another site. The vehicles in each site file are listed in the same order. Obviously, if a human were doing the matching this would give them an advantage in detecting false matches and missed matches due to misreadings. However, the algorithm does not take account of the order of vehicles so this ordering is irrelevant to the answer produced by the algorithm. The algorithm assumes that vehicles are evenly distributed across all vehicle types, however, this is certainly not the case in reality. There is no reason to assume that the algorithm depends on any specific distribution across types, however, it would be useful to follow up this work with tests using a non-uniform distribution.

## 5.1 Simulation Results for False Match Correction

The following methodology has been used for creating the simulation results described here. Each experiment has a particular set of parameters for the simulation described in the previous section. The parameters are  $N$ , the number of vehicle types which can be distinguished in the corrected data. A realistic value for this in partial licence plate data is around about 10,000. In a large survey  $p(2)$  was measured as  $8.00 \times 10^{-5}$  which would correspond to  $N = 12,500$  if the partial plates had a flat distribution. The majority of experiments will stick to  $N = 10,000$  since this is a realistic number for the scenarios using licence plate data encountered in real life. The  $\epsilon_i$  which are the probabilities of a misrecording of any given observation at site  $i$  and finally, there are the flows to each site (or set of sites). The simulation is run to produce  $n$  data sets where the true number of matches across all sites is known. This data set is analysed to estimate the number of vehicles genuinely observed at all of the  $n$  sites using the correction techniques described above to attempt to correct for the effects of accidental matches in the partial data and also to attempt to correct for the misreading effects. The number of raw matches (before correction) and corrected matches (after correction) is recorded. This procedure is repeated one hundred times to produce a mean, sample variance and standard deviation for the raw and corrected number of matches.

The experiments in this section make the assumption that  $\epsilon_i = 0$  for all sites. That is, all the data is recorded perfectly. The results of the experiments are recorded in Table 1. The table shows, for each experiment, first the expected result (True Result). Secondly, the raw result is given. This is the number of  $n$ -tuples which appear to be a match across all sites if no correction is done. This is given as the mean  $\bar{X}$ , sample variance  $S^2$  and sample standard deviation  $S$  for the set of one hundred results for each experiment. Then the corrected

results are given after the correction algorithm described is applied. Ideally, the corrected results will have  $\bar{X}$  near to the true result and will have  $S^2$  (and hence  $S$ ) low, since if this is high then an individual run of the simulation will produce answers a long way from the correct answer. No experiments have been performed with two sites since the correction in this case is simple and better methods are available in the literature. Other experimental results can be seen in [3, chapter four].

The table below shows the parameters of experiment one. As can be seen, there are three sites. The parameter  $N$  is the number of types of vehicle which are distinct in partial observations. Readings were assumed to be taken without any misrecordings (as they will be done for all the experiments in this chapter). The results of this experiment are good in that the raw number of matches has been corrected down from an average of over 2000 to 999.3 which is very close to the true result of 1000. The variance and standard deviation in the corrected results is relatively low indicating the result should be quite reliable even for a single run.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$		
Flow	1000	1000	1000	1000	1000
Sites	All	1	2	3	1,3

Experiment two (parameters shown below) replicates experiment one but with only half the number of vehicle types. This produces a higher probability of false matches and, consequently,  $\bar{X}$  for the raw results is higher at 3719.8 the variance of the raw results has also risen. Despite this, the mean number of corrected matches is actually closer to correct than in experiment one although the variance has risen.

$N = 5,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$		
Flow	1000	1000	1000	1000	1000
Sites	All	1	2	3	1,3

Experiment three has four sites and a flow of one hundred across all four. However, this flow is obfuscated by large flows between all the combinations of three of the four sites. As can be seen, the raw number of matches is extremely large and varies greatly between runs. While the mean of the corrected results is very close (99.1 rather than 100) it should be noted that the variance is very high indeed. A single reading from one run of such an experiment would be largely valueless since the standard deviation is actually higher than the effect being measured.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$	$\varepsilon_4 = 0$	
Flow	100	1000	1000	1000	1000
Sites	All	1,2,3	1,2,4	1,3,4	2,3,4

Experiment four has six sites and a flow of one hundred across all six. Each site has a flow of one thousand at that site only. The mean number of raw

matches (215) is not high compared with many of the experiments so far performed. However, while the mean number of corrected matches is nearly correct (97.0) the variance is still quite high.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$	$\varepsilon_4 = 0$	$\varepsilon_5 = 0$	$\varepsilon_6 = 0$	
Flow	100	1000	1000	1000	1000	1000	1000
Sites	All	1	2	3	4	5	6

Experiment five repeats this but with only half of the flows at each of the sites. The mean number of raw matches (168) is reduced compared with that of experiment four however, the variance is not. Again the corrected number of matches is nearly the exact right answer but the variance is actually higher than the previous experiment.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$	$\varepsilon_4 = 0$	$\varepsilon_5 = 0$	$\varepsilon_6 = 0$	
Flow	100	1000	1000	1000	1000	1000	1000
Sites	All	1	2	3	4	5	6

Experiment six repeats experiment four again but this time with the single flow which goes to all sites reduced from 100 to 10. The mean raw matches is almost double the correct answer (18.6). The corrected matches is almost exactly right at 10.4 and the variance is about average for the experiments performed here.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$	$\varepsilon_4 = 0$	$\varepsilon_5 = 0$	$\varepsilon_6 = 0$	
Flow	10	1000	1000	1000	1000	1000	1000
Sites	All	1	2	3	4	5	6

Experiment seven has just three sites but with a small flow to all three and a number of large flows which might cause false matches. The mean raw number of matches is nearly seven times the correct answer but the corrected number of matches is almost exactly correct (9.3). The variance is surprisingly low considering this example was created with the idea of causing conflicting flows.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$				
Flow	10	1000	1000	1000	100	100	100
Sites	All	1	2	3	1,2	2,3	1,3

Experiment eight is designed to be an extremely severe test of the algorithm. A relatively small flow to all six sites is mixed with two larger flows going to five of the six sites. This experimental set up was designed with the idea of causing a very large number of false matches in the data. Indeed this is the case the mean raw number of matches is 4483.5, more than 44 times the correct answer of 100. The variance on the results is huge. A single reading from this experiment would tell very little about the true answer. The corrected answer is 130.8 which is rather high. However, the variance of the sample mean should be  $\sigma^2/n$  where  $n$  is the number of samples and  $\sigma^2$  is the variance of the process being sample. From this it is easy to calculate that the variance of the sample mean is 4670 (on the corrected data) after 100 samples. Therefore a reading of 130.8 is still consistent with the idea that the estimator is unbiased.

$N = 10,000$	$\varepsilon_1 = 0$	$\varepsilon_2 = 0$	$\varepsilon_3 = 0$	$\varepsilon_4 = 0$	$\varepsilon_5 = 0$	$\varepsilon_6 = 0$
Flow	100	1000	1000			
Sites	All	1,2,3,4,5	2,3,4,5,6			

Expt. No.	True Result	Raw Results			Corrected Results		
		$\bar{X}$	$S^2$	$S$	$\bar{X}$	$S^2$	$S$
1	1000	2172.9	6352.13	79.7	993.3	3758.7	61.3
2	1000	3719.8	17128.7	130.9	1002.8	6562.5	81.0
3	100	4095.2	66917.4	258.7	99.1	19904.7	141.1
4	100	215.0	3510.8	59.3	97.0	1620.5	40.3
5	100	168.0	3231.1	56.8	97.8	1993.8	44.7
6	10	18.6	59.0	7.7	10.4	28.6	5.3
7	10	67.1	86.9	9.3	9.3	64.7	8.0
8	100	4483.5	1752449.6	1323.8	130.8	467000.9	683.4

Table 1: Experiment results for runs without recording errors.

## 5.2 Simulation Results for Observation Error Correction

In this section, experiments are run with, in addition to false matching effects, the effects due to recording errors.

Experiment one here is a repeat of experiment one in the previous section but with a recording error of ten percent at each site. The mean raw number of matches is 1711.2 (for the situation without recording errors it was 2162.9). The mean corrected number of matches is almost correct at 993.2 (this would have been only 724.1 before the correction for the error rate was applied). Despite the extra random element due to the recording errors, the variance in the corrected figure has increased little over the case before recording errors were introduced.

$N = 10,000$	$\varepsilon_1 = 0.1$	$\varepsilon_2 = 0.1$	$\varepsilon_3 = 0.1$		
Flow	1000	1000	1000	1000	1000
Sites	All	1	2	3	1,3

Again, this is a repeat of experiment two in the previous section. The mean number of raw matches is high (3065.0) but the mean number of corrected matches is very close to the correct answer (997.5). The variance is higher than the previous experiment but still relatively low and this experiment should provide quite reliable results even for a single run.

$N = 5,000$	$\varepsilon_1 = 0.1$	$\varepsilon_2 = 0.1$	$\varepsilon_3 = 0.1$		
Flow	1000	1000	1000	1000	1000
Sites	All	1	2	3	1,3

In experiment three, variable error rates at each site have been tried with some sites having a twenty percent error rate and some sites having a ten percent

Expt. No.	True Result	Raw Results			Corrected Results		
		$X$	$S^2$	$S$	$X$	$S^2$	$S$
1	1000	1711.2	4386.0	66.2	993.2 (724.1)	4734.1 (2515.9)	68.8 (50.2)
2	1000	3065.0	13914.7	118.0	997.5 (727.2)	9527.8 (5063.5)	97.6 (71.2)
3	100	85.6	126.3	11.2	97.8 (50.7)	274.4 (73.7)	16.6 (8.6)
4	100	2584.1	778537.0	882.3	183.9 (97.7)	1010393.0 (285364.9)	1005.2 (534.2)

Table 2: Experiment results for runs with recording errors.

error rate. The results are, again good as far as the mean goes. Interestingly, the false matches and the matches reduced by recording errors almost cancel out and the mean raw number of matches in this case is almost correct already. After correcting for false matches the mean number of matches drops to half the correct number. Once the correction for recording errors is put back then the mean is 97.8, almost correct. The variance in this example is not too high compared with many of the examples seen.

$N = 10,000$	$\varepsilon_1 = 0.1$	$\varepsilon_2 = 0.2$	$\varepsilon_3 = 0.1$	$\varepsilon_4 = 0.2$	
Flow	100	1000	1000	1000	1000
Sites	All	1	2	3	4

Experiment four is the equivalent of experiment eight but with recording errors added. Recording errors are at a level of ten percent for each site. The mean number of raw matches is large (2584.1) but this should be compared with 4483.5 in the case where there are no recording errors. Clearly the recording errors are having a big effect in this situation. After correcting for all errors the mean number of matches is 183.9 which is considerably over the correct answer of 100. Although the variance in this case is very high (as would be expected), it is uncertain whether the difference in the means is explained purely by this variance. In this case, the estimator is only a first order correction and is not unbiased. Indeed the estimator would be expected to be an overestimate. It is hard to know if the overestimate here is due to estimator bias or is due to the large variance in the results. However, it should be remembered that this experiment was designed to be an extreme test of the method.

$N = 10,000$	$\varepsilon_1 = 0.1$	$\varepsilon_2 = 0.1$	$\varepsilon_3 = 0.1$	$\varepsilon_4 = 0.1$	$\varepsilon_5 = 0.1$	$\varepsilon_6 = 0.1$
Flow	100	1000	1000			
Sites	All	1,2,3,4,5	2,3,4,5,6			

### 5.3 Discussion of Results and Further Work

In general, these experiment results for the false matches alone are very good in that the mean corrected result is very close to the true result in all cases (the worst case being experiment seven which is out by seven percent). These results certainly seem to be in line with the proof that the estimator was unbiased. Of the experiments, some have a very high variance (as seen by comparing the standard deviation to the mean). In fact, only experiments one and two could be expected to produce close answers from a single run. Experiment three is particularly bad in this respect. Experiment eight is worse although that was designed as an artificial situation to test the method to its limits.

The experimental results for the false matches plus error correction are also satisfying. The mean corrected totals are as they should be except for the extreme case of experiment four. While the method was only intended as a first order correction, it seems that this is appropriate for a variety of situations although not all. It is uncertain whether the large error in the corrected mean in experiment four is a result of the bias in the method or simply of the high variance in the results. In any case, the method is still managing to correct a mean of 2584.1 matches down to a mean of 183.9.

The main issue with the solution method as it stands is the high variance on the corrected answer. This could be seen as unavoidable when the high variance on the raw number of matches is considered. Future research could address this in two ways. Firstly, it would be very helpful if an estimate of the variance could be given from a single reading. This would at least allow the experimenter to assign some value to the possible errors on an estimate produced by this method. Secondly, it would be even more helpful to be able to reduce the variance. One possible solution here is to consider the fact that, in many cases, it is possible to rule out the same data item appearing in more than one n-tuple of matches across all sites. This could considerably reduce the number of matches detected in the data. However the formulation of the problem would become much more complex.

## 6 Conclusion

A framework has been introduced for investigating problems which consider matches in multiple data sets. Set theory has been used to map the problem onto that of partitioning the integers. This framework has been used to give algorithms for correcting two different types of error which occur naturally in matching data. Simulation tests show that the algorithm performs well in error correction tasks although the variance in the results can be high.

## References

- [1] D. R. Anderson, K. P. Burnham, and D. L. Otis. *Capture-recapture and removal methods for sampling closed populations*. Los Alamos National Lab-

oratory, 1982.

- [2] N. Biggs. *Discrete Mathematics*. Oxford Science Publications, 1961.
- [3] R. G. Clegg. *Statistics of Dynamic Networks*. PhD thesis, Dept. of Math., Uni. of York., York., 2004. Available online at:  
[www.richardclegg.org/pubs/thesis.pdf](http://www.richardclegg.org/pubs/thesis.pdf).
- [4] E. Hauer. Correction of licence plate surveys for spurious matches. *Transportation Research A*, 13A:71–78, 1979.
- [5] M. J. Maher. The analysis of partial registration-plate data. *Traffic Engineering and Control*, 26(10):495–497, 1985.
- [6] D. P. Watling. Maximum likelihood estimation of an origin-destination matrix from a partial registration plate survey. *Transportation Research B*, 28B(3):289–314, 1994.
- [7] D. P. Watling and M. J. Maher. A graphical procedure for analysing partial registration-plate data. *Traffic Engineering and Control*, 29(10):515–519, 1988.
- [8] D. P. Watling and M. J. Maher. A statistical procedure for estimating a mean origin-destination matrix from a partial registration plate survey. *Transportation Research B*, 26B(3):171–193, 1992.