A SET THEORETIC FRAMEWORK FOR ENUMERATING MATCHES IN SURVEYS AND ITS APPLICATION TO REDUCING INACCURACIES IN VEHICLE ROADSIDE SURVEYS

Richard G. Clegg,

Department of Mathematics, University of York,

York, YO10 5DD, United Kingdom

Email: richard@manor.york.ac.uk

Abstract

This paper describes a method for enumerating the ways in which combinations of vehicles can be observed at different survey points. The framework described is quite general and can be applied to a variety of problems where matches are to be found in data surveyed at a number of locations (or at a single location over a number of days). As an example, the framework is applied to the problem of false matches in licence plate survey data.

In this paper, a method for representing the possible *types of match* is outlined using set theory. The phrase *types of match* will be defined and formalised in this paper. A method for quickly calculating $\mathcal{M}_n$, the set of all types of match over n survey sites, is described and it is shown that the number of types of match can be simply calculated. The method is applied to the problem of correcting survey data for false matches using a simple probabalistic method. An algorithm is developed for correcting false matches over multiple survey sites and its use is demonstrated with simulation results.

1 Introduction

In the analysis of roadside survey data, it is often desirable to analyse matches between several data sets simultaneously. For example, we might wish to answer questions of the general type "How many drivers are seen at point A, point B and point C?" or "How many vehicles are seen on all five survey days?" This paper attempts to create a general framework for the analysis of matching between data from more than two surveys. The framework is then applied to the specific case of false matching in partial licence plate surveys (that is non-matches which are mistaken for matches because only part of the licence plate is observed). It should be stressed throughout that the framework outlined is applicable to any data series where matches are sought between two or more distinct data sets.

Licence plate surveys are commonly used in the study of traffic systems, particularly when measurements of the same vehicle are required more than one point (for example, calculating travel times or the routes of vehicles). Although automated techniques are

becoming more common (GPS, toll-tags and automated recognition cameras) the manual licence plate survey remains an important tool for the road transport engineer. If a road with a high volume of traffic is being surveyed then it often the case that only part of the licence plate is recorded. When this is the case, the possibility of spurious matches occurs. To take an example, standard British licence plates used to be of the following form: single letter, three digits, three letters: e.g. `A123BCD` [1]. If a surveyor only recorded the first letter and three digits, then a vehicle `A123ABC` would not be distinguished from a vehicle `A123XYZ` since the disambiguating information (the final three letters) would not be recorded.

While the chances of such a false match are low, quite often the combinatorics of the problem means that the actual recorded number of false matches remains high. To mathematicians, this is familiar as the celebrated *Birthday Paradox*. The Birthday Paradox asks the question "How many people must we have in a room before we might expect that two share the same birthday?" Intuitively, we might expect this to be quite a high number (since it is unlikely that any two people share a birthday). However, the number of pairs of people in a room goes up with the square of the number of people in the room $(n^2 - n)/2$. If we made the assumption that the chance of two randomly selected people sharing a birthday is one in 365 then we only need twenty three people in the room before it becomes likely (probability above 50%) that two will share a birthday. Combinations in multiple point surveys work similarly. If we had two survey sites, each with one thousand observations then this is one million pairs of observations. If the chances of a false match in a given pair are only one in a ten thousand, we will still get (on average) one hundred false matches. This could well be larger than the actual number of genuine matches in the data set and will certainly be a significant bias.

This paper attempts to provide a sound theoretic backing (using the well-known framework of set theory) to matching problems across multiple data sites. In section two, a general background of matching problems in licence plate data is given to put the problem into context within the transport field. In section three, the concept of *types of match* is formalised using the standard set theoretic concept of an equivalence class. In section four, a simple method is given for constructing the set $\mathcal{M}_n$, the set of every possible type of match across n survey sites. In section five, partial ordering is introduced to apply the problem to false matches due to incomplete observations. In section six, an algorithm is given for correcting false matches using the framework developed in sections three, four and five. Finally, in section seven, computational results are given on artificially generated survey data. To save space, some proofs are omitted from this paper. These are available from the author on request.

2 THE FALSE MATCH PROBLEM IN LICENCE PLATE DATA AT MULTIPLE SITES

It is often the case that on-street traffic surveys collect partial vehicle licence plate information [2]. This information can then be used to reconstruct travel times and to

[1] This form will be used throughout the paper, however, it must be stressed that this method would work with partial observations of any type given the assumptions stated in the paper.

[2] The reason for collecting partial rather than full licence plate information is that the recording and transcription of the data is often done manually and time constraints would preclude recording a full

infer route information about drivers. In partial plate data, however, problems can occur from *false matches* as discussed above. Of course, false matches could also occur through recording or transcription errors. While this paper will not discuss these problems, it is in principle possible to extend this framework to cover recording and transcription errors.

In the case of two survey sites and no recording or transcription errors the situation is relatively clear. If our data shows that a match occurs between two observations (one from each site) then, this must mean that either the same vehicle has been observed at both, or that two different vehicles have been observed which happened to have the same partial licence plate. At multiple sites the situation is much more complex. An apparent match at four survey points may be any of the following: a true match (the same vehicle seen at all four points); a different vehicle at each of the four points which (by coincidence) have the same partial plate; a vehicle at survey point one and two which has the same partial plate as a second vehicle at survey points three and four; or any other of fifteen total possibilities. The problem becomes more difficult as the number of sites increases. Indeed it is not immediately clear how to enumerate the number of ways in which a match as described above can occur over multiple data sites.

A number of researchers have approached the false matching problem for licence plates. [Hauer, 1979] provides an early approach for two sites. [Maher, 1985] describes several methods including the possibility of two point matches between vehicles observed at pairs of sites selected from several survey sites (for example entering and leaving a cross-roads). [Watling and Maher, 1988] gives a graphical procedure which is highly recommended for visualising matches between two sites. In fact, this procedure is a good starting point for any analysis of journey time data whether it contains false matches or not. [Watling and Maher, 1992] and [Watling, 1994] describe further refinements. However, all of these methods concentrate on matches between pairs of sites and the majority of them also assume that journey time information can be used to aid in finding false matches, which is not the case if, for example, we are interested in correcting false matches at the same site over different days. The method described in this paper concentrates on matches between observations at more than two sites, particularly where journey time information is not available or cannot be used.

3 EQUIVALENCE CLASSES FOR TYPES OF MATCH

Before examining the types of match we must first define exactly what we mean when we say that matches are of the same type. Assume that we have a set of n survey sites $\mathbf{S} = \{S_1 \ldots S_n\}$. At each site S_i we have a set of unique observations [3] $L(S_i)$. For the moment, the assumption will be made that these observations include enough information to uniquely identify an individual. Later we will introduce a censoring function which represents partial observations. We will refer to an n-tuple (or n dimensional vector) of observations over $\mathbf{S}$ with the notation $\mathbf{x}_n(\mathbf{S}) = (x_1, x_2, \ldots, x_n)$ and the set of all such n-tuples over $\mathbf{S}$ as $L_n(\mathbf{S})$.

---

plate.

[3]The requirement that the observations are unique is necessary in set theory. However, if we wish to include the possibility that our observations are not unique at a single site, we could simply tag each observation with a number, say the order in which the observation was made, and not use that in further comparisons.

We can say that $L_n(\mathbf{S})$ is the Cartesian product (or product set) of the sets of observations from each site. That is $L_n(\mathbf{S}) = L(S_1) \times L(S_2) \times \ldots \times L(S_n) = \prod_{i=1}^{n} L(S_i)$

Take the following observation n-tuples made at three sites:

$$\mathbf{x}_3(\mathbf{S}) = (\texttt{A123XYZ}, \texttt{B256ABC}, \texttt{B256ABC})$$
$$\mathbf{y}_3(\mathbf{S}) = (\texttt{A123XYZ}, \texttt{A123XYZ}, \texttt{B256ABC})$$
$$\mathbf{z}_3(\mathbf{S}) = (\texttt{C789ABC}, \texttt{A543OPQ}, \texttt{A543OPQ})$$

It is clear that in some sense $\mathbf{x}_3(\mathbf{S})$ and $\mathbf{y}_3(\mathbf{S})$ are not the same type of match whereas $\mathbf{x}_3(\mathbf{S})$ and $\mathbf{z}_3(\mathbf{S})$ are the same type of match. We would therefore like to express the notion that two n-tuples of observations are the same *type of match* if a match between two sites in the first n-tuple is also a match between the same two sites in the second n-tuple and if two observations in the first n-tuple do not match then they also do not match in the second. Formally we express this notion using the concept of an equivalence class (see, amongst others [Halmos, 1970]). Very loosely speaking, an equivalence class is a generalisation of the familiar concept of equality.

Definition 3.1. $\mathbf{x}_n(\mathbf{S}) \sim \mathbf{y}_n(\mathbf{S})$ iff $x_i = x_j \Leftrightarrow y_i = y_j \quad \forall i, j \in \mathbb{N} : 1 \leq i, j \leq n$ [4]

It can be trivially proved that this obeys the necessary conditions for an equivalence relation (reflexive, symmetric and transitive). Thus, from our example above, we can now say: $\mathbf{x}_3(\mathbf{S}) \sim \mathbf{z}_3(\mathbf{S})$ since the second and third elements of $\mathbf{x}_3$ are equal but not the first and the same is true of $\mathbf{z}_3$. We can also say $\mathbf{x}_3(\mathbf{S}) \not\sim \mathbf{y}_3(\mathbf{S})$ with similar reasoning. This has formalised the earlier notion of two n-tuples of observations being the same *type of match*. Two n-tuples, can be said to be the same type of match if they are part of the same equivalence class.

If we can create a set containing exactly one representative from each of these equivalence classes, then this set will have one representative for each type of match. Such a set is known as a *transversal*. Let $\mathcal{M}_n$ be a transversal of the equivalence relation defined in 3.1. Let $\mathbf{x}_n^{\mathcal{M}} = (x_1, x_2, \ldots, x_n)$ be an n-tuple which is a member of $\mathcal{M}_n$. If we can construct such a set $\mathcal{M}_n$, then we have a set of all the different possible types of matches which can occur over n survey sites.

Definition 3.2. $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ iff:

$$x_i = \begin{cases} 1 & i = 1 \\ x_j \quad \text{or} \quad 1 + \max(x_j) & j < i \end{cases}$$

This is most easily understood as the following procedure:
a) Label the first vehicle with a 1.
b) Label subsequent vehicles in turn either with the number of a previous vehicle which they match or with the next available integer if they match no previous vehicles.

[4] Note that for simplicity, such limits on indices will be omitted in future definitions where they can be trivially inferred by the reader.

It is not difficult to prove that:

$\forall \mathbf{x}_n(\mathbf{S}) \exists \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{x}_n(\mathbf{S}) \sim \mathbf{y}_n^{\mathcal{M}}$

(that is to say any n-tuple of observations is equivalent to a member of $\mathcal{M}_n$) and also that:

$\forall \mathbf{x}_n^{\mathcal{M}}, \mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n \quad \mathbf{x}_n^{\mathcal{M}} \sim \mathbf{y}_n^{\mathcal{M}} \Rightarrow \mathbf{x}_n^{\mathcal{M}} = \mathbf{y}_n^{\mathcal{M}}$

(in other words $\mathcal{M}_n$ contains no duplicates — if two n-tuples are members of $\mathcal{M}_n$ and are equivalent then they must be the same member). Thus $\mathcal{M}_n$ is a *transversal* and has exactly one representative of each type of match. It is worth emphasising that these types of matches would apply to any observations where we can define an equality relation between observations — the framework would be just as applicable to discrete sets of colours or shapes as it is to vehicle number plates.

To give an example, it is now possible to express our three earlier n-tuples in terms of equivalent members of this matching class.

$$\mathbf{x}_3(\mathbf{S}) = (\texttt{A123XYZ}, \texttt{B256ABC}, \texttt{B256ABC}) \sim (1, 2, 2)$$
$$\mathbf{y}_3(\mathbf{S}) = (\texttt{A123XYZ}, \texttt{A123XYZ}, \texttt{B256ABC}) \sim (1, 1, 2)$$
$$\mathbf{z}_3(\mathbf{S}) = (\texttt{C789ABC}, \texttt{A543OPQ}, \texttt{A543OPQ}) \sim (1, 2, 2)$$

This further formalises the notion of *type of match*. It is now possible to represent the type of match of any n-tuple of observations over n sites by saying that its type is the member of $\mathcal{M}_n$ to which it is equivalent. That is, the type of match of a given n-tuple $\mathbf{x}_n(\mathbf{S})$ is $\mathbf{y}_n^{\mathcal{M}}$ where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}_n(\mathbf{S})$.

Define the height of a type of match $\mathbf{x}_n^{\mathcal{M}}$ as $H(\mathbf{x}_n^{\mathcal{M}}) = \max(x_i)$. It should be clear from the definition that the height of a type of match is the number of different observations which are in the n-tuple (the number of unique vehicles observed) — for example, a match of type $(1, 2, 1, 3)$ has a height of three and contains observations of three unique vehicles.

4 CONSTRUCTING THE SET OF EVERY TYPE OF MATCH

Having defined $\mathcal{M}_n$, the set of all possible types of match over n observation sites, it will now be useful to create a rule for constructing the set $\mathcal{M}_n$. The set $\mathcal{M}_{n+1}$ can be easily constructed from the set $\mathcal{M}_n$ using definition 3.2. Given $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$ then we can construct $\mathbf{y}_1, \mathbf{y}_2 ... \in \mathcal{M}_{n+1}$ from $\mathbf{x}_n^{\mathcal{M}}$ by adding an $n+1$th element to the n-tuple. From 3.2 we can see that y_{n+1} (the $n+1$th element of $\mathbf{y}_1$) can take any integer value from 1 to $H(\mathbf{x}_n^{\mathcal{M}}) + 1$.

To construct $\mathcal{M}_{n+1}$ from $\mathcal{M}_n$:
a) Take each element of $\mathcal{M}_n$ in turn.
b) To each n-tuple $\mathbf{x}_n^{\mathcal{M}}$ construct a vector by adding the integers from 1 to $H(\mathbf{x}_n^{\mathcal{M}}) + 1$ as the $n+1$th element of the n-tuple.
c) These vectors (n+1-tuples) together form the set $\mathcal{M}_{n+1}$.

Therefore, given that $\mathcal{M}_1 = (1)$ we can easily construct computationally $\mathcal{M}_n$ by building up $\mathcal{M}_2$, $\mathcal{M}_3$ and so on. This process is illustrated in figure 1.

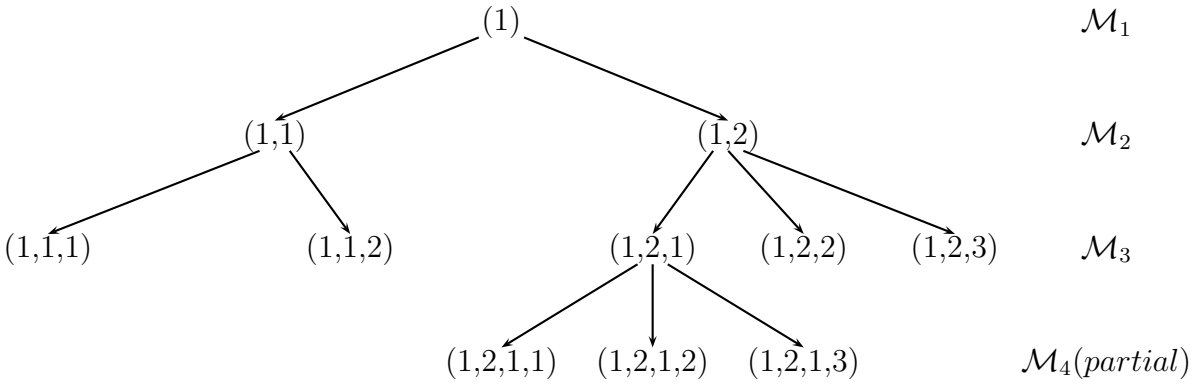

Figure 1: Creation of $\mathcal{M}_{n+1}$ from $\mathcal{M}_n$.

It can be simply shown that there is a one-to-one correspondence between the members of $\mathcal{M}_n$ and the partitions of the set of the first n natural numbers. A partition of a set is the division of that set into subsets. For example, we may partition the first three numbers in any of the following five ways: $\{1,2,3\}$, $\{1,2\}\{3\}$, $\{1,3\}\{2\}$, $\{2,3\}\{1\}$ and $\{1\}\{2\}\{3\}$. The one-to-one correspondence is easily seen by the following mapping between members of $\mathcal{M}_n$ and the partitions given. For a given $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n$, take the numbers from 1 to n. Two numbers are part of the same set if and only if the elements in $\mathbf{x}_n^{\mathcal{M}}$ with those numbers are equal. So, for example, $\mathbf{x}_n^{\mathcal{M}} = (1,2,1,1,3)$ maps to $\{1,3,4\}\{2\}\{5\}$. This mapping can be trivially shown to be one-to-one. Thus there are as many elements of M_n as there are ways to partition the first n natural numbers. Therefore we can count the members of $\mathcal{M}_n$ using Stirling numbers (see [Biggs, 1961] for more information on Stirling numbers).

Let $S(n,k)$ be the number of members of $\mathcal{M}_n$ with height k (where $1 \leq k \leq n$). Clearly $S(n,1) = 1$ — the only member with height 1 is $(1,1,\ldots,1)$. Also $S(n,n) = 1$ — the only member with height n is $(1,2,\ldots,n)$.

We can also show that $S(n,k) = S(n-1,k-1) + kS(n-1,k)$. A sketch of such a proof is that using our constructive method above, every member of $\mathcal{M}_{n-1}$ with height $k-1$ will construct one member of $\mathcal{M}_n$ with height k and every member with height k will construct k members with height k (by adding the final elements from 1...k). From this recursive formula we can calculate the number of elements in $\mathcal{M}_n$ of a given height and, by summing, the number of elements in $\mathcal{M}_n$.

5 INTRODUCING FALSE MATCHING INTO THE FRAMEWORK

In order to introduce false matches into this framework, it is necessary to introduce two things: the idea of false matches caused by part of the data being unobserved and the idea from set theory of a partial ordering.

To include partial observations in the framework, the notion of a censoring function is introduced — this function is to simulate the observation of only part of a unique identifier (in this case, the recording of only part of a unique licence plate).

Definition 5.1. A censoring function acts on an n-tuple of observations and produces another n-tuple. If observations at two sites within the n-tuple are seen to be the same,

then they will remain the same in the n-tuple produced by the censoring function. Conversely, however, if two observations are not the same then the censoring function may cause them to become the same.

This is equivalent to the common-sense notion that two vehicles which have the same licence plate will never appear to be different if we correctly record only part of their plates. However, two vehicles with different licence plates may appear to be the same if we correctly record only part of their plates. The censoring function is introduced with the notation $C(\mathbf{x}_n(\mathbf{S}))$ meaning the censored n-tuple of observations produced by partial observation of the uncensored n-tuple $\mathbf{x}_n(\mathbf{S})$. It is clear that the censored n-tuple may not be in the same matching class as the uncensored n-tuple. However, it is also clear that only certain transitions are possible (since it is not possible for a censoring function to make observations appear different). It is therefore important to investigate in which ways n-tuples can move between matching classes because of a censoring function. (This is the same as investigating in which ways an observation can appear to be a different type of match if we only observe part of the data).

To investigate this problem it is necessary to introduce the concept of a *partial ordering* of a set. The partial ordering can be very loosely thought of as an extension of the notion of less than and greater than (a partial ordering also allows two elements to be incomparable). See [Halmos, 1970] amongst others for more details. A partial ordering can be induced on the set $\mathcal{M}_n$ as follows:

Definition 5.2. $\mathbf{x}_n^{\mathcal{M}} \succsim \mathbf{y}_n^{\mathcal{M}}$ iff $x_i = x_j \Rightarrow y_i = y_j$

In words, an n-tuple is a successor of (loosely, greater than) or equal to a second n-tuple if and only if, wherever two elements of the first are equal, the same two elements of the second are equal. For example: $(1, 2, 1) \succsim (1, 1, 1)$. To be a partial ordering, the relation must be reflexive, anti-symmetric and transitive. These properties are easily shown. Note that this definition is the same as that for the equivalence relation in definition 3.1 except that the implication only goes one way. This partial ordering defines exactly how the matching class of an n-tuple of observations may move into a different matching class if only partial observations are taken.

Theorem 5.1. *If* $\mathbf{y}_n^{\mathcal{M}} \sim \mathbf{x}_n(\mathbf{S})$ *and* $\mathbf{x}_n^{\mathcal{M}} \sim C(\mathbf{x}_n(\mathbf{S}))$ *then* $\mathbf{y}_n^{\mathcal{M}} \succsim \mathbf{x}_n^{\mathcal{M}}$.

In words, the match type of the censored data is a predecessor of or equal to the match type of the uncensored (complete) data using the partial ordering defined above.

This can be seen by comparing the above definition of a censoring function and our definition of a partial ordering. Having defined a partial ordering, the set $\mathcal{M}_n$ can be visualised using a Hasse diagram. A Hasse diagram is a depiction of a finite partially ordered set where the elements are represented by points in a plane and a directed arrow from element x to element y indicates that $x \prec\prec y$. (x immediately precedes y — that is $\not\exists z : x \prec z \prec y$). The diagram of $\mathcal{M}_n$ has discrete levels defined by $H(\mathbf{x}_n^{\mathcal{M}})$ and will have singular upper and lower levels defined by $\mathbf{x}_n^{\mathcal{M}} = (1, 2, ...n)$ and $\mathbf{x}_n^{\mathcal{M}} = (1, 1...1)$ respectively. The Hasse diagram for $\mathcal{M}_4$ is shown in figure 2. Note that, from theorem 5.1, if we take a complete or uncensored observation, then censoring it (taking only partial

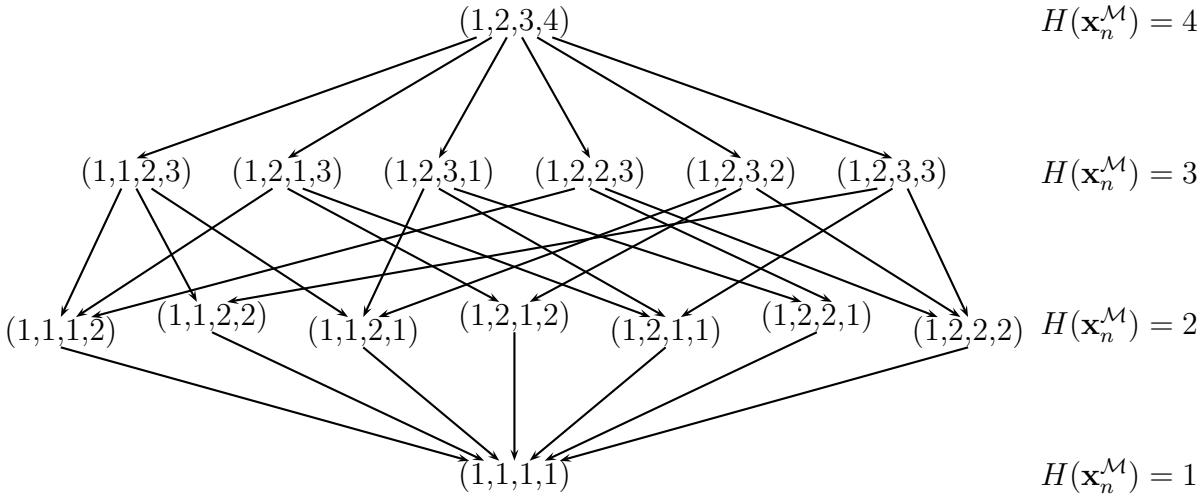

Figure 2: Hasse diagram for $\mathcal{M}_4$.

observations) can move the observation to any matching class which can be reached by moving along one or more arrows (or, obviously, the censoring can leave the observation in the same class).

In order to count matches in the particular classes it is useful to define the exact matching function $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S})$ (which counts the number of matches in a data set over the sites $\mathbf{S}$ which are in matching class $\mathbf{y}_n^{\mathcal{M}}$) and the relaxed matching function $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S})$ (which counts the number of matches in $\mathbf{S}$ which are in the matching class $\mathbf{y}_n^{\mathcal{M}}$ and all its predecessors in $\mathcal{M}_n$).

Definition 5.3. The exact matching function $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S}))$ for a single observation is defined as:

$$X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S})) = \begin{cases} 1 & \text{iff} \quad \mathbf{x}_n(\mathbf{S}) \sim \mathbf{y}_n^{\mathcal{M}} \\ 0 & \text{otherwise} \end{cases}$$

Definition 5.4. Defined over a set of observations $L_n(\mathbf{S})$ then $X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S})$, the exact matching function is the number of n-tuples in the set of observations which are in a particular type of match $\mathbf{y}_n^{\mathcal{M}}$.

$$X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S}) = \sum_{\mathbf{x}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x})$$

where the sum is over all $\mathbf{x} \in \mathbf{S}$.

Similarly we wish to define a relaxed matching function $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S}))$ for a single observation.

Definition 5.5. The relaxed matching function $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S}))$ is defined as:

$$R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S})) = \begin{cases} 1 & \text{iff} \quad \mathbf{y}_n^{\mathcal{M}} \succeq \mathcal{M}_n(\mathbf{x}_n(\mathbf{S})) \\ 0 & \text{otherwise} \end{cases}$$

And as before we wish to extend this to the set of n-tuples of observations over the sites **S**.

Definition 5.6. Defined over the n sites in **S** then $R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S})$, the relaxed matching function is the number of n-tuples in the set of all possible observations $L_n(\mathbf{S})$ for which equation 5.5 is equal to one. Alternatively:

$$R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S}) = \sum_{\mathbf{x}_n(\mathbf{S})} R(\mathbf{y}_n^{\mathcal{M}}, \mathbf{x}_n(\mathbf{S}))$$

where the sum is over all $\mathbf{x}_n(\mathbf{S}) \in L_n(\mathbf{S})$.

Finally, in order to estimate the number of false matches we need an estimate of the value of $p(i)$ which is defined as the probability that i randomly chosen items which are not equal in the uncensored data are all equal in the censored data. (For convenience define $p(1) = 1$). This can be estimated in real roadside surveys by considering the frequency distribution of licence plates.

6 SOLVING THE FALSE MATCH PROBLEM

The original problem was to count the number of vehicles which are seen at all sites. This is the problem of enumerating the number of n-tuples of observations which are in matching class $(1, 1, \ldots, 1)$. This will be referred to as $\mathcal{M}_n(\mathcal{T})$ — a true match across all sites. At the same time, it is useful to introduce the notation $\mathcal{M}_n(\mathcal{F})$ to indicate matching class $(1, 2, \ldots, n)$ which corresponds to a completely false match, n different vehicles, one observed at each site. In the notation of the previous section this means evaluating $X(\mathcal{M}_n(\mathcal{T}), \mathbf{S})$. However, only $X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S}))$ (the number of matches in the censored data) can be directly calculated since the original assumption was that only partial (censored) data was collected.

Clearly it is the case that:

$$R(\mathcal{M}_n(\mathcal{F}), \mathbf{S}) = \prod_{i=1} L(S_i) \tag{1}$$

since every member of $\mathcal{M}_n$ is either $\mathcal{M}_n(\mathcal{F})$ or a predecessor of it. (This can be confirmed by reviewing figure 2).

Computationally it is too much effort to evaluate each n-tuple and find out which exact and relaxed matching class it is part of. Consider that if there are six sites and 100 observations in each then there are 100^6 possible n-tuples to consider. This is not an unreasonable size of problem to consider in vehicle surveys. It is not as difficult to calculate how many n-tuples are in class $\mathcal{M}_n(\mathcal{T})$ since the matching procedure can stop whenever a non-match is found (and it is assumed that matches are rarer than non-matches). Thus it is necessary to estimate $X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S})$ in terms of some set of $X_m(\mathcal{T})$ on subsets of sites. The following three lemmas will help us to do this. The proofs of these lemmas are not given in this paper but are available from the author on request.

Lemma 6.1. *The number of exact matches of a given type in a data set is equal to the number of relaxed matches minus the number of exact matches of all predecessor types.*

$$X(\mathbf{x}_n^{\mathcal{M}}, S_n) = R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S}) - \sum_{\mathbf{y}_n^{\mathcal{M}}} X(\mathbf{y}_n^{\mathcal{M}}, \mathbf{S})$$

where $\mathbf{y}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{y}_n^{\mathcal{M}} \prec \mathbf{x}_n^{\mathcal{M}}$.

An intuitive way to see this lemma is that an exact match, is a relaxed match of the same class, minus those exact matches of predecessor classes.

It is worth noting a trivial corollary of this:

Corollary 6.2.

$$R(\mathcal{M}_n(\mathcal{T})) = X(\mathcal{M}_n(\mathcal{T})) \tag{2}$$

which follows obviously from lemma 6.1 since there are no $\mathbf{x}_n^{\mathcal{M}} \prec \mathcal{M}_n(\mathcal{T})$ and therefore the subtracted term in the lemma vanishes.

Lemma 6.3.

$$R(\mathbf{x}_n^{\mathcal{M}}, S_n) = \prod_{i=1}^{H(\mathbf{x}_n^{\mathcal{M}})} X(\mathcal{M}_{m(i)}(\mathcal{T}), Y_i)$$

where $m(i) = \#Y_i$ *and* Y_i *is an m-tuple constructed from the original n-tuple of sites* $\mathbf{S}$ *using the matching class* $\mathbf{x}_n^{\mathcal{M}}$. Y_i *is constructed such that* $Y_i = (S_{j_1}, S_{j_2}, \ldots, S_{j_m})$ *where* $\{j_1, \ldots, j_{m(i)}\}$ *is the set of all indices of the n-tuple* $\mathbf{x}_n^{\mathcal{M}}$ *such that* $x_{j_k} = i$.

An example may help understand how Y_i is constructed. If $\mathbf{x}_6^{\mathcal{M}} = (1, 1, 2, 3, 2, 1)$ then $Y_1 = (S_1, S_2, S_6)$, $Y_2 = (S_3, S_5)$ and $Y_3 = (S_4)$.

Again, the proof of this lemma is not stated here. It can be thought of as breaking down a relaxed match into the component exact true matches on subsets of $\mathbf{S}$, which are necessary conditions for a set of observations to be a relaxed match of the given type.

It is worth noting a trivial corollary of this.

Corollary 6.4.

$$R(\mathcal{M}_n(\mathcal{F}), \mathbf{S}) = \prod_{i=1}^{n} \#L(S_i)$$

Proof. This should be obvious since for $\mathcal{M}_n(\mathcal{F}) = (1, 2, \ldots, n)$ each set of sites Y_i consists of exactly one site i. Since $\mathcal{M}_1 = (1)$ then there is only one matching class for each of the sites and $X(M_{S_i}(\mathcal{T}), \mathbf{y}_1 S_i) = \#L(S_i)$. □

Lemma 6.5.

$$X(\widehat{\mathcal{M}_n(\mathcal{T})}, \mathbf{S}) = X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S})) - \sum_{\mathbf{x}_n^{\mathcal{M}}} X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S}) p(H(\mathbf{x}_n^{\mathcal{M}}))$$

where $\mathbf{x}_n^{\mathcal{M}} \in \mathcal{M}_n : \mathbf{x} \succ \mathcal{M}_n(\mathcal{T})$

Proof. This comes from our observation that $H(\mathbf{x}_n^{\mathcal{M}})$ is the number of separate objects which are in a match of type $\mathbf{x}_n^{\mathcal{M}}$. The probability that an n-tuple of observations which is a match of type $\mathbf{x}_n^{\mathcal{M}}$ in the uncensored data is a match of type $\mathcal{M}_n(\mathcal{T})$ in the censored data is therefore $H(\mathbf{x}_n^{\mathcal{M}})$. It follows that an estimation of the number of n-tuples of observations which are matches of type $\mathbf{x}_n^{\mathcal{M}}$ in the uncensored data but appear to be matches of type $\mathcal{M}_n(\mathcal{T})$ in the censored data is given by:

$$X(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S})p(H(\mathbf{x}_n^{\mathcal{M}})) \tag{3}$$

Lemma 6.5 follows from the observation that $X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S}))$ is the sum of the exact match, $X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S}))$ and the total contributions of the type described in equation 3 for all $\mathbf{x}_n^{\mathcal{M}}$ apart from $\mathcal{M}_n(\mathcal{T})$. □

It is not immediately obvious, but from the above lemmas 6.1, 6.3 and 6.5 a procedure can be created to estimate $X(\mathcal{M}_n(\mathcal{T}), S)$ — the number of true matches in a set of observations $L_n(\mathbf{S})$.

Lemma 6.3 allows estimation of $X(\mathcal{M}_n(\mathcal{T}), \mathbf{S})$ from $X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S}))$ (which can be measured directly since it is measured on the censored data) and $X(\mathbf{x}, \mathbf{S})$ if it is known for all $\mathbf{x} \in \mathcal{M}_n : \mathbf{x} \prec \mathcal{M}_n(\mathcal{T})$ assuming that $p(i)$ is also known for any i. Thus the number of true matches can be estimated from the number of exact matches in all other matching classes.

From lemma 6.1 these matches can be calculated exactly if the number of relaxed matches $R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S})$ is known and also the number of exact matches in all successor matching classes is known.

From lemma 6.3 we can calculate the number of relaxed matches of a particular order if we know the number of exact true matches in a subset of sites. The value of $R(\mathcal{M}_n(\mathcal{F}), \mathbf{S})$ is given by equation 1. From 6.2, $R(\mathcal{M}_n(\mathcal{T}), \mathbf{S}) = X(\mathcal{M}_n(\mathcal{T}), \mathbf{S})$, which is the quantity we are trying to find. For all other values of $\mathbf{x}_n^{\mathcal{M}}$, 6.3 allows us to find $R(\mathbf{x}_n^{\mathcal{M}}, \mathbf{S})$ in terms of $X(\mathcal{M}_m(\mathcal{T}), \mathbf{S}_m)$ where $m < n$ and $\mathbf{S}_m$ is a subset of the sites in S. Thus, we can solve our problem in terms of a problem with a reduced number of sites. This procedure can be followed recursively until the number of sites is 1 when the problem becomes trivial. (With one site, $X(\mathcal{M}_1(\mathcal{T})) = X(\mathcal{M}_1(\mathcal{F})) = L(\mathbf{S}_1)$.)

Therefore, if we can estimate $p(n)$ we can solve the problem of estimating $X(\mathcal{M}_n(\mathcal{T}), S)$ by the following procedure.

Step 1: Calculate from our data, $X(\mathcal{M}_n(\mathcal{T}), C(\mathbf{S}))$ for our n sites.

Step 2: Use a computer to expand lemmas 6.1, 6.3 and 6.5 to give us an expression which estimate $X(\mathcal{M}_n(\mathcal{T}), \mathbf{S})$ as shown above.

Step 3: Again using a computer, gather all the terms which are $X(\mathcal{M}_n(\mathcal{T}), \mathbf{S})$ on the left hand side — these terms will be multiplied factors of $p(k)$ where $1 < k \leq n$.

Step 4: We now have an equation for $X(\mathcal{M}_n(\mathcal{T}), S)$ in terms of $p(k)$, $R(\mathcal{M}_n(\mathcal{F}), \mathbf{S})$ (given by equation 1) and $X(\mathcal{M}_m(\mathcal{T}), \mathbf{S_m}$ where $m < n$.

Step 5: Decrease n by 1 and go to step 2 to find the terms $X(\mathcal{M}_m(\mathcal{T}), \mathbf{S_m})$.

7 RESULTS ON SIMULATED DATA

Table 7 shows simulation results for between two and six observation sites. The table is to be interpreted as follows. Num. Veh. refers to the total number of observations at each of the sites (in these simulations, there are the same number of vehicles in each data set). The five columns of the form $1 - n$ refer to the number of vehicles which genuinely went from site one to site n visiting all sites in between. If this column is blank it means that there was no site n. For example, if $1 - 2 = 100$, $1 - 3 = 200$ and $1 - 4$ is blank. This means that 100 vehicles travelled between site one and site two, 200 vehicles travelled between sites one, two and three and there were only three sites. Note that these are cumulative so that if $1 - 2 = 20$ and $1 - 3 = 10$ this means that 30 vehicles in total went from site one to site two and ten of them continued to site three. Thus the first experiment is two sites, 1000 vehicles at each for which there were ten vehicles which were genuinely seen at both sites. Note that in every experiment, the number of different vehicle types was set at 10,000 with a flat distribution (equal numbers of vehicles seen at each site). It should be clear that the desired answer from the correction process is the rightmost figure in these columns.

Each experiment is repeated twenty times with simulated data being generated anew each time. The correction process has no random element and will always give the same result for the same data. The mean raw number of matches is given — this is the total number of n-tuples which were seen to have the same value for each observation at every site (averaged over the twenty simulation runs). Note that, because of the combinatorial nature of the procedure, this could, in principle, be much larger than the number of vehicles in any of the data sets (since it counts any n-tuple). The sample standard deviation (σ) is given for the raw matches. The mean estimated correct number of matches is then given (again averaged over the twenty simulations). The sample standard deviation σ is then given for the ten corrected matches. It is clear that the most important test is that the mean corrected number of matches is as near to correct as possible. However, it should also be kept in mind that in reality, a researcher could only run the matching procedure once on any given set of data — so it is also important that σ is as low as possible. A significant improvement to the method would be to estimate the variance as well as producing an estimate then the researcher could have some idea as to the likely accuracy of the corrected results. It should also be noted that in every experiment, the chances of any given two vehicles being a false match is 1 in 10,000 with a flat distribution (so the chance of three distinct vehicles having the same partial plate is the square of this). In fact this is an extremely pessimistic assumption since four digits of a licence plate would be the least that a partial plate survey was likely to capture (in the UK, one letter and three digits is the most common). A significant weakness of the method is that it requires a good estimate for $p(n)$. (In fact, it is mainly significant for lower values of n with $p(2)$ being the most important).

The first five rows are all results on just two test sites. This procedure is not the ideal one to use for estimates on matches between just two sites and the work of other authors in the field should be used in such a circumstance. However, these results are included here for completeness. In the two site case, the average corrected matches is simple obtained by subtracting $\frac{n^2}{10000}$ from the raw matches (where n is the number of vehicles at each site) — to take an example, in the first experiment, the average number

of raw matches over the ten runs is 111.4. The average number of corrected matches is 100 less than this (11.4). This is close to the correct answer of 10. However, it should be noticed that the σ is high in comparison to the actual answer. In this case, the σ is 8.5 which is of the same order of magnitude as the answer. This is to be expected since we are looking for only 10 true matches in over 110 observed matches. If we increase the number of vehicles to 2000 then, as would be expected, the number of false matches goes up (to approximately 400) and the σ also rises (to almost 20).

The next five rows of results are all over three sites. In the first of these, 10 vehicles travel between all three and all other matches are coincidence. 1000 vehicles are observed at all sites. The mean corrected match across all sites 9.3 is close to the actual answer of 10 and the σ is lower than in the two site case. However, when the same experiment is run with 500 vehicles travelling from sites one to two in addition to 10 vehicles travelling from sites two to three, the σ increases markedly (it almost doubles). In all cases with three sites, the mean is a good estimate and the σ is generally low enough that a good estimate can be expected.

The next four rows of results are for experiments made over four sites. The first experiment has 100 vehicles which visit all four. The mean corrected match is 104 (very close) and the σ is only 22. It is hard to explain why this σ actually falls in the next experiment when more vehicles are genuinely seen in common between the other sites. This fall in σ is puzzling. In all cases the mean of the predictions is approximately correct (the worst performance being in the case of the fourth experiment when the mean was 106.1 not 100).

The next six rows of results are experiments made over five sites. Again, the mean corrected results are approximately correct. However, in the worst case, the mean is 11 too high and the σ in the results is 46.7 which is comparable to the level of the effect being observed. In this case approximately 120 false matches are being removed each time. However, previous experiments have been able to correct for a greater proportion of false matches with less σ in the result.

The final four rows of results are experiments over six sites. This was the largest number of sites for which it was practical to do runs of twenty or more simulations with the computer power available. Again, the mean corrected estimate of matches was nearly correct in all cases. The worst performance was an estimate of 92.2 (correct result 100). The σ was, however, relatively high. This was a surprise in some cases — particularly the first row of results where the mean number of false matches was only 21.2. In many senses, the worst results was the final one where a σ of 55.0 was given on an corrected prediction of only 101.3.

The time taken to do one run over six sites with one thousand pieces of data on each site was thirty seconds on a Celeron 366 computer running Debian Linux. It is practical (if time consuming) to do experiments on seven sites, even using such comparatively obsolete equipment. However, eight sites or more is probably too computationally expensive for the moment and this is a limitation of the method outlined.

The results given here are certainly consistent with the idea that the method gives an unbiased estimator for the true number of matches. In some experiments, there were

No. Veh.	1 – 2	1 – 3	1 – 4	1 – 5	1 – 6	Av. Raw Matches	σ Raw Matches	Av. Cor. Matches	σ Cor. Matches
1000	10					111.4	8.5	11.4	8.5
2000	10					411.8	19.5	11.8	19.5
1000	100					199.2	12.0	99.2	12.0
1000	200					302.3	7.7	202.3	7.7
1000	500					596.6	12.3	496.7	12.3
1000	0	10				21.9	4.6	9.3	3.3
1000	500	10				73.8	7.5	10.2	6.2
1000	100	100				152.1	8.5	101.9	7.5
1000	500	250				388.3	22.7	253.2	20.1
1000	0	500				667.2	24.9	506.0	22.3
1000	0	0	100			154.6	26.6	104.0	22.6
1000	100	100	100			164.4	11.4	97.7	9.3
500	100	100	100			140.7	19.3	105.8	17.4
1000	500	250	100			207.8	29.7	106.1	23.7
500	10	10	10	10		14.2	2.2	10.5	1.8
1000	10	10	10	10		17.4	4.1	9.4	2.8
500	50	50	50	50		71.3	14.3	47.8	12.3
500	100	100	100	100		151.9	26.9	92.0	22.3
1000	0	0	0	100		177.6	29.9	103.4	22.6
1000	100	100	100	100		222.2	61.5	111.0	46.7
1000	0	0	0	0	10	21.2	13.4	12.3	9.9
500	0	0	0	0	100	152.6	45.5	92.2	37.3
1000	0	0	0	0	100	214.6	58.0	103.5	40.2
1000	100	100	100	100	100	289.8	88.4	101.3	55.0

Table 1: Simulation results — all performed over twenty runs with 10,000 distinct vehicle types.

problems with the standard deviation being higher than would be desirable in real cases. It is important to bear in mind that these were relatively extreme tests of the method since $p(2)$ and $p(3)$ were relatively low and the number of samples given were quite high. Often the method was attempting to predict only ten true matches in a number of observed matches which might be several hundred.

8 CONCLUSIONS

This paper presented a framework for analysis of surveys where matches are required over more than two data collection points. The framework given formalises the concept of a type of match using the concept of the equivalence class. Further a method is given for evaluating $\mathcal{M}_n$ the set of all possible types of match over multiple data sets. An algorithm is given which shows how, computationally, $\mathcal{M}_n$ can be computed and a method is given for enumerating its elements using Stirling numbers.

The framework given is then applied to the problem of false matches — which is put into the language of set theory using the concept of a partial ordering. It is shown how this partial ordering can be used to visualise, by means of a Hasse diagram, the ways in

which false matches can occur in data observed at multiple sites. An algorithm is then given which shows how survey data over multiple sites can be corrected for false matches.

This algorithm was implemented and tested on simulated data. The results show that the estimator seems to be unbiased and in the majority of cases tested the standard deviation on the results is low. The method is suitable for analysis of matches on data between three and seven test sites but becomes too computationally intensive after this point. A significant improvement to the method would be the estimation of a variance as well as a corrected number of matches. A potential weakness of the method is that it relies on good estimates for $p(2)$ and to a lesser extent $p(3)$. However, the author considers the method outlined here to be a reliable way to estimate matches in partial surveys over three or more sites.

References

[Biggs, 1961] Biggs, N. (1961). *Discrete Mathematics*. Oxford Science Publications.

[Halmos, 1970] Halmos, P. (1970). *Naive Set Thoery*. Springer Verlag.

[Hauer, 1979] Hauer, E. (1979). Correction of licence plate surveys for spurious matches. *Transportation Research A*, 13A:71–78.

[Maher, 1985] Maher, M. (1985). The analysis of partial registration-plate data. *Traffic Engineering and Control*, October:495–497.

[Watling, 1994] Watling, D. (1994). Maximum likelihood estimation of an origin-destination matrix from a partial registration plate survey. *Transportation Research B*, 28B(3):289–314.

[Watling and Maher, 1988] Watling, D. and Maher, M. (1988). A graphical procedure for analysing partial registration-plate data. *Transportation Research B*, October.

[Watling and Maher, 1992] Watling, D. and Maher, M. (1992). A statistical procedure for estimating a mean origin-destination matrix from a partial registration plate survey. *Transportation Research B*, 26B(3):171–193.