Applying Disclosure Control to Temporal Data

Anna Manning and Mark Elliot

Dept. of Computer Science, University of Manchester, M13 9PL, UK
anna@cs.man.ac.uk

Cathie Marsh Centre for Census and Survey Research, University of Manchester, M13 9PL, UK
mark.elliot@man.ac.uk

Abstract

An important aspect of disclosure control is the isolation and control of individual-level records that have a high probability of being identified (as their contents, or variables. are unusual) - consider, for example, a sixteen-year-old widow. However, many such datasets contain temporal information relating specifically to an individual which needs to be addressed when making such decisions. An unusual temporal sequence of events pertaining to an individual has the potential to lead to disclosure but is missed if the analysis is restricted to a single period in time.

Algorithms have been specifically designed to conduct a comprehensive search for "risky" records in survey-type data in a single period of time. This paper examines the extension of one of these for finding temporal sequences which may pose a disclosure risk to individuals in a datasets. Preliminary results indicates that this methodology has the potential to enhance disclosure control techniques significantly.

1. Introduction

The analysis of data collected over time (temporal data) has the potential to reveal the dynamic behaviour of an individual or group and is an important means of collecting information. The value of knowledge captured in temporal data is reflected in the growing interest in temporal data mining, the goal of which is to discover new, implicit, relationships and patterns over time. It is a rapidly growing area of research that brings together many disciplines including statistics, temporal pattern recognition and high-performance computing [Garofalakis et al. 1999, Srikant et al. 1996, Yan et al. 2003, Zaki et al. 2001]. Such techniques can reveal sensitive information about individuals and groups within temporal data and it is therefore important that careful consideration is given when releasing such data.

Temporal data can be discrete or continuous. The term *time series* is often used in statistics to refer to one (or few), but usually long, series of continuous, real-valued elements. *Longitudinal studies*, which are addressed here, are generally repeated measurements where the repetition is taken in time and are usually concerned with many but short series, often augmented in groups. They also differ from traditional time series analysis because the measurements are regarded as a sample from some underlying population.

Disclosure limitation in longitudinal linked data in a well-recognized problem [Adowd and Woodcock 2001] and one that can occur at individual-level, at employer level, at household level etc. depending on which groups are represented in the data. In this paper we concentrate on individual-level risk and focus on the identification of records which are particularly risky rather than on the methods used to mask them. Examples are drawn from the UK Labour Force Survey[1] which is a longitudinal linked data set containing individual-level (and household level) observations.

Some records have a high probability of being identified as their contents, or attributes, are unusual and therefore have the potential to be recognized spontaneously - such records are referred to as *special uniques* [Elliot 2000]. A sequential algorithm, *SUDA (Special Unique Detection Algorithm)* [Elliot et al. 2002] has been developed to locate special unique records from a given period in time by first identifying all record-level unique attribute patterns (up to a user-specified maximum size) and, secondly, by grading the risk of each record by considering the number and distribution of unique

[1] Office for National Statistics Labour Market Division, Labour Force Survey Five-Quarter Longitudinal Dataset, June 1999 - August 2000 [computer file]. Colchester, Essex: UK Data Archive [distributor], 21 June 2001. SN: 4304.

patterns that it contains. The SUDA algorithm has been shown to be efficient at picking out records with a high level of risk. In this paper the extension of SUDA to longitudinal data is explored and demonstrated.

2. SUDA

SUDA has been designed to provide a comprehensive search for special unique records in a given dataset. A two-stage approach is employed: firstly, all unique patterns (up to a user-specified size) are located at record level and, secondly, the size and distribution of unique patterns within each record is used to grade its 'riskiness'. In order to streamline the search SUDA has been structured around the observation that 'Every superset of a unique pattern is itself unique' (as every unique pattern is bounded by the size of its subsets). Only unique patterns without any unique subsets, *minimal unique patterns*, are considered in order to avoid the use of redundant information and to keep the classification process as focused as possible; the smaller the number of attributes contained in a unique pattern the more 'risky' it is considered to be and therefore it is important to know if a unique pattern is minimal or not. SUDA has been developed for discrete data (both numerical attributes and numerically coded categorical data) but can also accept continuous data if it is transformed into a discrete form beforehand (via multiplication by factors of 10 and/or rounding up).

Clearly, all possible attribute sets must be considered in order for an exhaustive search for minimal unique patterns to be conducted. However, to minimize computation time, SUDA selects all attribute sets with the same prefix[2] in succession so that any extensions of a unique prefix at a given record are ignored without reverting to stored information. For example, given four attributes, labelled a_1, a_2, a_3, a_4, the sets with prefix a_2 are: $\{a_2\}$, $\{a_2,a_3\}$, $\{a_2,a_3,a_4\}$, $\{a_2,a_4\}$. If attribute a_2 was found to be unique for record R then attribute sets $\{a_2,a_3\}$, $\{a_2,a_3,a_4\}$ and $\{a_2,a_4\}$ could be ignored for that record. This has the effect of reducing the number of records that need to be considered for each attribute set while at the same time minimizing memory usage.

The process of identifying minimal unique patterns is conducted by partitioning the dataset according to the value of each of the attributes in a given attribute set. For example, if the attribute set was (AGE, SEX) the dataset would be divided into groups of records containing attributes with values such as (AGE=20, SEX=male), (AGE=40, SEX=female) etc. Any group containing only one record represents a unique and a check for minimal uniqueness would then be made. The minimal uniqueness of a unique pattern X of size n (where $n \geq 2$) is determined by confirming that all subsets of X of size $n-1$ are non-unique. The partitioning method of SUDA has the effect of minimizing the amount of data storage that is necessary to identify minimal uniques by localizing the required information. In addition, the generation of attribute sets according to their prefixes allows this partitioning procedure to be undertaken efficiently and without redundant sorting [Elliot et al. 2002].

3. How can SUDA be applied to longitudinal data?

As described above, SUDA is currently designed for application to data collected in a single time period. Figure 1 illustrates the nature of the data. Each individual is represented by a record (row) in the dataset containing a fixed number (c) of attributes. The i^{th} record is represented by $(a_{1i}, a_{2i}, \ldots, a_{ci})$ and there are a total of r records altogether.

a_{11}	a_{21}	...	a_{c1}
a_{12}	a_{22}	...	a_{c2}
...	...	...	...
a_{1r}	a_{2r}	...	a_{cr}

Figure 1: Data from a single time period

[2] In general, for an attribute set A containing c attributes $a_1,\ldots a_c$, a prefix of size P of A where $1 \leq P \leq c$ contains the first P attributes $(a_1,\ldots,a_P)$ of A.

Figure 2 illustrates the nature of longitudinal data, with one set of r records of a fixed length of c attributes for t time periods. Each attribute is represented by a_{ijk} where i=attribute number, j= record number and k=time period.

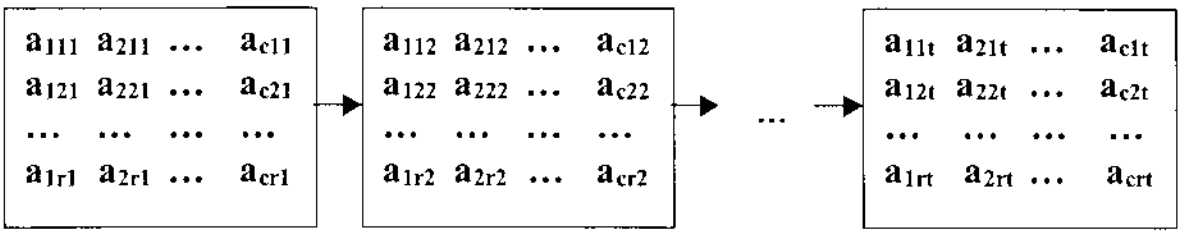

Figure 2: Data from t time periods

A sequential pattern pertaining to record i, attribute j is represented as follows:

$a_{ij1} \rightarrow a_{ij2} \ldots \rightarrow a_{ijt}$ (Sequence 1)

This sequence is unique if any of its subsequences are unique. Sequential patterns can include more than one attribute. For example, at sequential pattern pertaining to record i, attributes j and k is represented as follows:

$(a_{ij1}, a_{kj1}) \rightarrow (a_{ij2}, a_{kj2}) \rightarrow \ldots \rightarrow (a_{ijt}, a_{kjt})$ (Sequence 2)

How can the disclosure risk of a sequential pattern be assessed? The most logical method of extending SUDA would be to compare a given sequential pattern with every other sequence of the same kind in the data in order to determine whether it is unique or not. The SUDA approach is advantageous as it identifies subsequences that are minimally unique and it is possible to judge how risky a record is based on this information; the smaller the subsequence the more risky the entire sequential pattern is likely to be.

In order for SUDA to be applied to sequential patterns consideration needs to be given to the data input. As discussed above, SUDA accepts data in the form shown in Figure 1. It is straightforward to organize data in this way for a sequential patterns consisting of a single attribute. For example, if an attribute such as marital status is captured by the i^{th} attribute in each record then the corresponding values of this attribute from all records in all time periods can be gathered together, with one sequential pattern per individual – i.e. SUDA would be applied to the data in Figure 3.

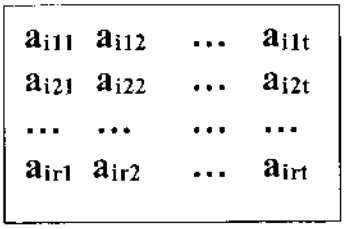

Figure 3: Sequential sequences for the i^{th} attribute

Applying SUDA in this way would find all unique subsequences whose components are not necessary consecutive. It would be possible to adapt the search so that only consecutive events are considered. It may also be possible to adapt the search so that larger time windows are considered – for example, it would be possible to consider unique subsequences over a time window of a year from data that is collected every three months.

To find all unique sequential patterns in the form of sequence 2 above, SUDA would be applied to the data in Figure 4.

(a_{j11}, a_{k11})	(a_{j12}, a_{k12})	...	(a_{j1t}, a_{k1t})
(a_{j21}, a_{k21})	(a_{j22}, a_{k22})	...	(a_{j2t}, a_{k2t})
...	...	...	...
(a_{jr1}, a_{kr1})	(a_{jr2}, a_{kr2})	...	(a_{jrt}, a_{krt})

Figure 4: Sequential sequences for the jth and kth attribute

For SUDA to accept this data as input each of the data items in brackets must be treated as one unit. There are three main approaches to this:

(1) SUDA is applied separately to each time period simultaneously (using parallel processing) and the results for each sequence collected together after this procedure is complete.
(2) The values for each set of attributes are mapped to a unique integer representation and SUDA is applied once to the resulting data.
(3) A multivariate version of SUDA is produced so that datasets, such as the one in Figure 4 could be processed immediately.

In this paper method 2 has been used. For example, with attributes AGE (95 possibilities) and SEX (2 possibilities) the combined attribute has $2 \times 95 = 190$ possibilities (one for male or female in each age group). These combined values can be decoded once SUDA has finished.

The algorithm was applied to the Labour Force Survey Five-Quarter Longitudinal Dataset, June 1999 - August 2000 (10, 951 records) and the next section demonstrates the effectiveness of the extended SUDA algorithm, *temporal SUDA*, for locating unusual and potentially risky sequential patterns.

4. Applying temporal SUDA to the LFS – some observations

In order to demonstrate the application of SUDA to temporal data the sequences that were found to be most "risky" (i.e. in terms of the number of unique subsequences that they contain) for different types of attributes are presented as follows:

(1) **MSTATUS - Marital status**: This attribute is clearly constrained in the options available over time – for example, an individual cannot be divorced until s/he has been married. The attribute can take five values as shown in Figure 1. When temporal SUDA was applied to MSTATUS over all five time periods of the LFS data 17 unique sequential patterns were found, containing a total of 38 minimally unique subsequences of size 2 and 9 of size 3 and these are presented in Table 2. It can be seen that temporal SUDA is capable of detecting sequences that show an unusual progression of this attribute over time: for example, the first sequence shows an individual whose marital status changed from "single" to "divorced" over two time periods which is very unexpected as this suggests that they must also have been married during this time. Other sequences suggest errors in the data - for example sequences 3, 11 and 12 all end with "single, never married" even though the beginning of the sequence suggests otherwise. The sequences become less unusual the fewer minimally unique subsequences they contain, suggesting that the SUDA approach is efficient for this problem.

Code	Detail
1	Single, never married
2	Married, living with husband/wife
3	Married, separated from husband/wife
4	Divorced
5	Widowed

Table 1: Attribute values for marital status

Unique subsequences:	Sequential

5. Discussion and further work

Section 4 demonstrates that temporal SUDA can pick out unusual sequential patterns pertaining to an individual which could pose a disclosure risk. When an attribute naturally possesses restrictions on the sequence of values it can take it appears this gives more potential for "risky" sequences as it can be seen immediately that a generally accepted course of events has not occurred. An attribute that is free to take all values appears to present less surprising results.

At present, SUDA is being applied to one sequential pattern at a time. The next step is to develop the algorithm so that it can conduct a search for all minimally unique sequences. Some thought will be required as to how to incorporate the search for minimally unique patterns during one time period with the search for minimally unique sequential patterns over all time periods. Attention also needs to be given to the weight allocated to each of these sequences.

References

Abowd, John M. and Woodcock, Simon D. (2001), 'Disclosure Limitation in Longitudinal Linked Data', in *Confidentiality, Disclosure, and Data Access Theory and Practical Applications for Statistical Agencies*, North Holland, 2001.

Elliot, M. J. (2000), 'A new approach to the measurement of statistical disclosure risk'. *International Journal of Risk Management*, 2(4), 2000.

Elliot, M. J., Manning, A. M. and Ford, R. W. (2002), 'A computational algorithm for handling the special uniques problem', *In International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, Vol. 10, No. 5, pp 493-509.

Garofalakis, Minos N., Rastogi, Rageev and Shim, Kyuseok (1999), 'SPIRIT: Sequential Pattern Mining with Regular Expressions Constraints'. *In Proceedings of Intl. Conf. Very Large Databases (VLDB)*, pp 223-234, , Edinburgh, 1999.

Srikant, Ramakrishnan and Agrawal, Rakesh (1996), 'Mining sequential patterns: Generalizations and perforance improvements'. *In proceedings of the International Conference on Extending Database Technology (EDBT)*, pp 3-17, Avignon, France.

Yan X., Han, J., and Afshar, R. (2003), 'CloSpan: Mining Closed Sequential Patterns in Large Datasets', (to appear) in *SIAM Data mining* 2003.

Zaki, Mohammed Javeed (2001), 'SPADE: An Efficient Algorithm for Mining Frequent Sequences', *Machine Learning* 40: 31-60.

			pattern
Size 1	Size 2	Size 3	
0	4	0	1 4 4 4 4
0	4	0	2 2 4 4 4
0	4	0	5 1 1 1 1
0	4	0	4 4 3 3 3
0	4	0	5 5 5 5 2
0	3	0	1 1 1 2 3
0	2	0	4 4 4 4 3
0	2	0	2 2 5 5 5
0	2	0	2 2 2 2 4
0	2	0	2 2 2 2 5
0	2	0	2 2 1 1 1
0	2	0	2 2 2 1 1
0	1	2	2 2 3 3 2
0	0	4	3 3 2 3 3
0	0	3	2 3 2 2 2
0	1	0	3 4 4 4 4
0	1	0	3 3 4 4 4

Table 2: Unique sequential patterns for MSTATUS

(2) **INECACA – Basic economic activity:** This attribute has little restriction on the changes that may occur and is therefore far less predictable. INECACA consists of 30 categories ranging from "Employee" to "Unemployed")

1048 unique sequences were identified with a total of 4 minimally unique subsequences of size 1, 743 of size 2, 1597 of size 3, 260 of size 4 and 6 of size 5. Most unique sequences belonged to individuals who were in temporary or unskilled work or to those who were not working and moving between states of inactivity. The sequences were not very suprising as those in such groups would be expected to have changeable values for this attribute.

(3) **AGE:** Is entirely predictable over time.

Although this attribute is entirely predictable over time, 18 unique sequences featuring the same age in all 5 time stamps (i.e. over the 15 month period) were identified which was surprising. One possible explanation is that birthdays coincide with the start of data collection for the first time stamp and suggests that Date of Birth could be derived (by considering AGE) from the file. and this information disclosed. For this file, however, Date of birth already appears and there is no risk of disclosing further information.

(4) **AGE and MSTATUS combined.**

When these two attributes are considered together 317 unique sequences were discovered consisting of 70 minimally unique subsequences of size 1, 745 of size 2, 44 of size 3 and 1 of size 4.

Four sequences were found to have unique patterns of size 1 in each of their five time periods – i.e. five sequential subsequences of size 1 each - which is the maximum possible:

(mstatus=2, age=18) -> (mstatus=2, age=18) -> (mstatus=2, age=18) -> (mstatus=2, age=19) -> (mstatus=2, age=19)
(mstatus=2, age=19) -> (mstatus=2, age=19) -> (mstatus=3, age=20) -> (mstatus=3, age=20) -> (mstatus=3, age=20)
(mstatus=5, age=30) -> (mstatus=5, age=31) -> (mstatus=5, age=31) -> (mstatus=5, age=31) -> (mstatus=5, age=31)
(mstatus=5, age=34) -> (mstatus=5, age=34) -> (mstatus=5, age=34) -> (mstatus=5, age=34) -> (mstatus=5, age=35)

The following sequence has 7 unique subsequences and shows someone in their mid-twenties moving from mstatus="divorced" to mstatus="married"

(mstatus=4, age=24) -> (mstatus=4, age=24) -> (mstatus=4, age=24) -> (mstatus=2, age=25) -> (mstatus=2, age=25)