On the Approximate Communal Fraud Scoring of Credit Applications

Clifton Phua[1], Ross Gayler[2], Vincent Lee[1] and Kate Smith[1]

[1]Clayton School of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800, Australia

[2]Baycorp Advantage, Level 12, 628 Bourke Street, Melbourne, VIC 3000, Australia

{clifton.phua, vincent.lee, kate.smith}@infotech.monash.edu, ross.gayler@baycorpadvantage.com

ABSTRACT

This paper describes a technique to generate numeric suspicion scores on credit applications based on implicit links to each other, and over time and space. Its contributions include pair-wise communal scoring of identifier attributes for applications, definition of categories of suspiciousness for application-pairs, smoothed *k*-wise scoring of multiple linked application-pairs, and the incorporation of temporal and spatial weights. With fixed parameters, results on a moderate-sized synthetic data set illustrate the potential strengths (handles implicit links, categories, relative time and space) and expose the weaknesses (parameter tuning and scalability issues) of our technique. In the near future, our attention will be on the linking and empirical scoring of a few million real applications with different parameter values.

Keywords

Credit application fraud detection, communal scoring, multi-attribute directed graph, dynamic application data streams, black and white lists, anomaly detection, exponential smoothing, temporal and spatial weights, and adversarial data mining

1. INTRODUCTION

Annually, credit bureaus collect millions of enquiries relating to credit applications. In Australia, credit card and personal loan applications have increased significantly, and around half a million credit bureau enquiries are made per month (Baycorp, 2005). Each credit application contains sparse identity attributes such as personal names, addresses, telephone numbers, driver licence numbers (or social security number), dates of birth, other personal identifiers, and these are potentially available to the credit bureau (if local privacy laws permit it).

Application fraud, a manifestation of identity crime, is present when application form(s) contain plausible and synthetic identity information (identity fraud), or real but stolen identity information (identity theft). In developed countries, the monetary cost of application fraud and identity crime is often estimated to be in the billions of dollars. By performing better once-off assessments in the first stage of the credit life cycle, some transactional fraud can also be prevented.

Typical commercial techniques involve the use of attribute verification rules using lookup tables, and pair-wise matching rules of credit application and credit history data. Rule-based approaches can be weak against increasingly common fraudster-tampered applications (Oscherwitz, 2005) which have valid attributes and no credit history. Other techniques include known fraud matching using black lists, and supervised modelling/classification using labelled data. Often, these labelled data approaches are operationally inefficient and ineffective (Phua *et al*, 2005).

Our novel approach is designed to generate links and score incoming current/new applications on demand and focuses at the level of each pair of linked applications (application-pair). It compresses multiple identifier attributes to a single attribute vector representation of each link/non-link (Section 2.1). The approach distinguishes between three different categories of links: black list, white list, and anomalous links, which will result in different weights and scores for every application-pair (Section 2.2). It scores multiple linked applications, gradually diminishes the impact of prior linked applications, and presents the decision thresholds (Section 2.3). The approach accounts for the temporal and spatial effects by applying weights to each linked application-pair's communal score computation (Section 2.4).

Section 3 discusses the simulated data generation and experiments with the training and scoring phases. Section 4 explores the descriptive graph and predictive scores with examples. Section 5 compares and contrasts research in other related application areas and Section 6 concludes the paper.

2. COMMUNAL SCORING

Graph theory is the established academic field which studies graph properties. Connections between credit applications can be denoted in a graph-theoretic notation (Wasserman and Faust, 1994). The simple directed graph (digraph), G_d can be described mathematically by two sets as $G_d = \langle V, E \rangle$. Let V represent a set of g vertices or nodes (applications for credit), where $V = \{v_1, v_2, ..., v_g\}$. Let E represent a set of h directed links or edges (relationships between applications based on shared identity information), where $E = \{e_1, e_2, ..., e_h\}$ with $g(g-1)$ maximum directed links. Let v_i, v_j each represent a vertice labelled with a set of N attributes, where $v_i = \{a_{i1}, a_{i2}, ..., a_{iN}\}$. In this paper, we analyse the direct

link between every ordered application-pair, where $e_{ij} = \langle v_i, v_j \rangle$ with $v_i \rightarrow v_j$, $\forall i, j$.

2.1 Pair-wise Matching

The ultimate purpose is to derive an accurate suspicion score for all incoming current/new applications in real-time. Given that, our design of pair-wise matching for dynamic applications has to be effective and efficient. For every current application v_i which arrives into the application fraud system, it can be pair-wise matched against all previous/ existing scored applications within a window w. Note that w can be an applications window (e.g. for the previous thirty thousand applications) or a time window (e.g. within the last thirty days).

In Table 1 below, all applications are primarily sorted in descending order by *date_received* to capture the applications' arrival sequence. There is a training/initialisation phase with a fixed/tuned set of parameters to ensure that the scoring/testing phase will be effective. After the training phase, the initialised applications are also secondarily sorted in ascending order by *suspicion_score*, where the least suspicious applications are removed for the scoring phase to maximise efficiency. For example, within the window, "legal-1" is compared to all the previously trained applications below it (1), and then "legal-2" is compared to "legal-1" (2) and to all other applications below it (1).

Table 1: Pair-wise matching design from the table point-of-view.

	rec_id	date_received	. . .	suspicion_score
Before Scoring	legal-6	31/12/2004	. . .	
	legal-5	30/12/2004	. . .	
	legal-4	30/12/2004	. . .	
	legal-3	29/12/2004	. . .	
	legal-2	28/12/2004	. . .	
	legal-1	28/12/2004	. .	
After Training	:	:	:	:
	legal-iv	14/02/2004	. . .	0.055
	legal-iii	6/02/2004	. . .	0
	fraud-i	26/01/2004	. . .	0.9
	legal-ii	26/01/2004	. . .	1.55
	legal-i	25/01/2004	. . .	0

2

1

The attribute vector y between v_i and v_j which represents the relationship for an application-pair is defined as:

$$y[v_i, v_j] = \{y_1[a_{i1}, a_{j1}], y_2[a_{i2}, a_{j2}], ..., y_N[a_{iN}, a_{jN}]\}, \forall i, j \quad ,$$

where y contains the individual suspicion scores of each pair-wise attribute and N is the number of attributes. The Boolean suspicion score of each pair-wise attribute y_k is determined either by exact matching (e.g. if $a_{ik} = a_{jk}$, then $y_k = 1$, else $y_k = 0$) or by fuzzy matching using string similarity metrics (e.g. if $similarity(a_{ik}, a_{jk}) \geq T_{similarity}$, then $y_k = 1$, else $y_k = 0$) of the same attribute, and the maximum possible score of an application-pair is $\sum_{k=1}^{N} y_k = N$. Note that y_k can also be determined by the matching of different but similar attributes (e.g. $y_k[a_{ik}, a_{jl}]$, where l is another attribute) and is case sensitive.

To account for the complex nature of identifier attributes, the weighted suspicion score of y_k assigns the highest weights to permanent attributes, followed by stable attributes, and transient attributes (IDAnalytics, 2004), so that the maximum possible score of an application-pair is now $\sum_{k=1}^{N} y_k = 1$. Note that the identifiers/string attributes are heuristically weighted.

2.2 Approximate Pair-wise Score

The purpose of this section is to define the categories of suspiciousness/risk for the dynamic and complex nature of application-pairs. A linked application-pair is first associated either with a black list or white list or as anomalous, and then the suspicion score is computed.

2.2.1 Black List

If (v_j is a known fraud and $\sum_{k=1}^{N} y_k \geq T_{fraud}$), then $v_i \rightarrow v_j$ and $w_{communal_{ij}} = 1$ and $\frac{S(v_j)}{E_O(v_j)} = 1$,

where T_{fraud} is the threshold for linking v_i to v_j for a black list score and $0 \leq T_{fraud} \leq 1$. $w_{communal_{ij}}$ is the communal link weight derived from pair-wise matching of identifier/string attributes (note that numerical attributes can be treated as identifier/string attributes) and $0 \leq w_{communal_{ij}} \leq 1$. $\frac{S(v_j)}{E_O(v_j)}$ is the average suspicion score of each previous application and $0 \leq \frac{S(v_j)}{E_O(v_j)} \leq 1$. $S(v_j)$ is the total/combined/final suspicion

score of previous applications which v_i links to . $E_O(v_j)$ is the number of outgoing links from v_j .

$$\text{If } v_i \rightarrow v_j \text{ and } E_I(v_j) \geq T_{E_I}, \text{ then } \frac{S(v_j)}{E_O(v_j)} = 1,$$

where $E_I(v_j)$ is the number of incoming links into v_j, T_{E_I} is the threshold for the acceptable number of incoming links.

2.2.2 White List

$$\text{If } y \in \Re, \text{ then } v_i \rightarrow v_j$$

$$\text{and } w_{communal_{ij}} = w_{normal} * \sum_{k=1}^{N} y_k ,$$

where $\Re = [R_1, R_2, ..., R_N]$ is a set of one or more relationships defined as normal, and $w_{normal} \equiv [w_{R_1}, w_{R_2}, ..., w_{R_N}]$ where w_{R_k} is the corresponding weight of R_k and $0.5 \leq w_{R_k} \leq 1$. Note that $\Re$ and w_{normal} are sorted in ascending order of w_{R_k} .

2.2.3 Anomalous Application-Pairs

Our definition of anomalies refers to linked applications which are not in the black and white lists (as opposed to the widely accepted definition that anomalies are deviants from the white list).

$$\text{If } y \notin \Re \text{ and } \sum_{k=1}^{N} y_k \geq T_{anomalous}, \text{ then } v_i \rightarrow v_j$$

$$\text{and } w_{communal_{ij}} = \sum_{k=1}^{N} y_k ,$$

where $T_{anomalous}$ is the threshold for linking v_i to v_j for an anomalous score and $0 \leq T_{anomalous} \leq 1$.

2.2.4 Summary of Application-Pair Categories

Table 2 on the right describes the score range, advantage, disadvantage, and secondary use of each application-pair category. The main advantage of a black list is the feedback of fraudulent applications to the system to stop subsequent similar applications from getting approved. However, there is usually a time delay to flag particular applications as fraudulent, and during this delay, such similar applications will go unnoticed (Phua *et al*, 2005). To prevent fraudsters from tampering and attempting to defeat our communal scoring technique, in addition, there should be frequent supervised modelling on the recent known frauds and non-frauds relationships y to recalibrate the attribute, normal, temporal, and spatial weights.

The white list accounts for linked applications submitted by real identities. It defines the rational applicant behaviour (e.g. application-pair submitted by the same identity with address changes) and the normal social relationships (e.g. application paired with another family member's, housemate's, colleague's, neighbour's, or friend's application) (IDAnalytics, 2004). The white list differentiates similar/linked applications as normal or anomalous. However, the fraudster can also create masqueraded "normal" applications to delay the detection time. It seems reasonable that if $y \in \Re$ and with minimal fraudster activity, this information can be used to exploit pre-existing social networks for marketing of credit products by targeting customers with the strongest influence.

Linked pairs of anomalous applications will have lower scores than those in the black list and generally higher scores than those in the white list. It reveals abnormal relationships between applications which could be indicative of fraud, but also of data entry errors either by the employee or customer. However, the effective prioritisation of anomalous applications is dependent on accurate pair-wise attribute weights which have to be truly reflective of fraud.

Unlinked application pairs are the result of too few attribute matches (below a set threshold), or no attribute matches at all. There is a strong possibility that there will be fraudulent applications which seem to be standalone and this communal scoring technique will not be able to detect them.

Table 2: Properties of black and white lists, anomalous and unlinked applications.

Application-Pairs	**Black List**	**White List**	**Anomalous Links**	**Unlinked**
Score range	Highest	Lowest	Medium to high	None
Main advantage	Reasonably accurate	Filters normal relations	Finds irregular relations	Contains weak or no relations
Main disadvantage	Time delay	Prone to manipulation	Reliant on attribute weights	Unlinked (once-off) fraud
Secondary use other than real-time fraud detection	Update attribute, normal, temporal, spatial weights	Viral marketing	Error detection	-

Note that this communal scoring technique can also work without a black list (with no known frauds), so it becomes semi-supervised (or commonly known as anomaly detection system).

2.3 Smoothed *K*-wise Scoring Function

Our scoring approach for *k*-pairs (multiple-pairs) of linked credit applications:

$$S(v_i) = \sum_{v_j \in M(v_i)} \left[w_{ij} + \frac{S(v_j)}{E_O(v_j)} \right],$$

where $S(v_i)$ is the total suspicion score of the current application and $S(v_i) \geq 0$, $M(v_i)$ is the set of k applications which v_i links to, w_{ij} is the link weight between v_i and v_j and $0 \leq w_{ij} \leq 1$. Therefore, a high suspicion score $S(v_i)$ roughly captures the strong connection between v_i and v_j (high link weight), and/or quality of v_j (high average suspicion scores), and/or quantity of v_j (number of connected applications).

Exponential smoothing gradually discounts the effects of previous suspicion scores:

$$S(v_i) = \sum_{v_j \in M(v_i)} \left[(1-\alpha) * w_{ij} + \alpha * \frac{S(v_j)}{E_O(v_j)} \right],$$

where α is the smoothing factor and $0 \leq \alpha \leq 1$.

The decision thresholds can be defined as:

If $0 < S(v_i) \leq T_{lower}$, then current application v_i is unusual

If $T_{lower} < S(v_i) < T_{upper}$, then v_i is suspicious

If $S(v_i) \geq T_{upper}$, then investigate v_i

If the investigated v_i is fraudulent, provide feedback to the fraud detection system.

2.4 Communal, Temporal, Spatial Weights

The modified link weight $\widetilde{w}_{ij}$ between v_i and v_j is defined as:

$$\widetilde{w}_{ij} = w_{communal_{ij}} * w_{temporal_{ij}} * w_{spatial_{ij}},$$

where $w_{temporal_{ij}}, \forall i, j$ is the link weight derived from pair-wise time difference of *date_received* and $0.5 \leq w_{temporal_{ij}} \leq 1$, $w_{spatial_{ij}}, \forall i, j$ is the link weight derived from pair-wise geographical distance of *postcode* and $0.5 \leq w_{spatial_{ij}} \leq 1$. Therefore, this modified link weight $\widetilde{w}_{ij}$ roughly captures the relative strength of the communal links over time and space and $0 \leq \widetilde{w}_{ij} \leq 1$. Note that if $w_{temporal_{ij}}$ and $w_{spatial_{ij}}$ is going to have any effect, an application-pair has to be linked ($w_{communal_{ij}} > 0$).

A small time difference means that v_i is new relative to v_j, but more importantly it implies that there is not enough time to get feedback or reveal the true state of v_j. A large geographical distance means that v_i is far relative to v_j, and it suggests that the sphere of influence of v_j has changed. The hypothesis is that a linked application-pair is most suspicious if there is no time difference and there is the maximum geographic distance between v_i and v_j. Therefore, $w_{temporal_{ij}} = w_{spatial_{ij}} = 1$. On the other hand, a linked application-pair is least suspicious if it has the largest time difference specified and there is no geographic distance between v_i and v_j. Therefore, $w_{temporal_{ij}} = w_{spatial_{ij}} = 0.5$. Note that within the credit application context, *date_received*, being system-generated, and *postcode*, being the essential geographical area of contact (e.g. card destination or card collection), will be less prone to significant external manipulation than other identifier attributes.

3. EXPERIMENTS

All experiments are performed on a single Pentium IV 3.0GHz, 2Gb RAM workstation, running on Windows XP platform. The communal scoring software is written in Visual Basic and C# .NET and the synthetic credit application data is stored in Microsoft Access.

3.1 Synthetic Data

3.1.1 Justification for Use of Synthetic Data

There are few published research studies which explicitly analyse real personal identifiers for fraud. It is probably ironic that privacy and confidentiality concerns restrict them from being used in the raw form, while encryption or removal of these key attributes undermines the full capability of the data mining-based fraud detection system (therefore results will be undermined and organisations are reluctant to provide data). In general, the fraud analytics business is also competitive (publication of results on real data can be time-consuming, and unrewarding as fraudsters and competitors get more knowledgeable).

Synthetic data allows the fraud detection system designer to test the system and/or study effects of data set size, population drift, concept drift, adversarial countermeasures, and data entry error rates on performance measures in a controlled environment (where actual class labels are known).

3.1.2 Generation of Synthetic Data

The FEBRL data generator (Christen, 2005) is primarily for matching all structured records related to the same entity based on common attributes (record linkage/de-duplication). First, original records are randomly created from identifier/string attributes from

frequency look-up tables and identifier/numerical/date attributes from specified ranges.

Second, duplicate records are generated based on selected original records with the following additional parameters: total duplicates, maximum duplicates per chosen original record, and probability distribution of how many duplicates are being created based on one original record.

Third, errors are introduced in duplicates based on user-defined probabilities (e.g. common misspellings, insert, delete, substitute, transpose and swap adjacent attribute values).

The version 0.2 generator has been modified to accommodate our idea that both fraudsters, and normal individuals and their social networks will submit similar applications. The key difference is that fraudsters will purposely reuse some successful information (uniformly distributed number of duplicates), while normal people will unknowingly send in additional related legal applications (poisson distributed number of duplicates). Error probability rates are the same for fraudulent and legitimate duplicates (some are set at 0 and the rest range from 0.005 to 0.04). This generator does not have the capability to create normal communities yet.

3.1.3 Details of Generated Synthetic Data

There are close to 20 attributes (see Appendix C) and some of them are: *rec_id* (primary key label), *date_received*, *given_name* and *surname* (personal name), *street_number*, *current_address*, *previous_address*, *suburb*, *postcode*, and *state* (geographical location), *home_phone* and *mobile_phone* (contact phone numbers), *driver_licence* and *date_of_birth* (id numbers). The data does not contain title, gender, email address, internal protocol address, card type, or approved status.

There are 52,750 applications which span the entire year 2004. Given a maximum of 10 duplicates per original record, 4,700 are fraudulent (about 400 a month) and they reflect 3,000 regular frauds (2,700 duplicates), 600 occasional frauds (540 duplicates), 600 seasonal frauds (540 duplicates) which happen in late March, early April and December, and 500 once-off frauds (no duplicates). 48,000 (about 4,000 a month) are legal applications, and 50 are hand-crafted applications with both fraudulent and legal examples.

3.2 Training and Scoring Phases

In the experiment, all attributes are included for calculating $w_{communal_{ij}}$ except *rec_id* (all unique), date_received (for calculating $w_{temporal_{ij}}$), postcode (for calculating $w_{spatial_{ij}}$) *street_number*, *suburb*, and *state* (too dense). Also, 3 additional attributes are created to store *suspicion_score*, *outlinks*, and *inlinks*.

Table 3 across lists the parameters and basic measurements of 10,000 trained applications from 01/01/2004 to 10/03/2004 (inclusive of 1,488 applications have non-zero scores) and 42,750 scored applications from 10/03/2004 to 31/12/2005 (exclusive of the 1,488 trained applications). The scoring phase retains trained applications which are suspicious, investigated, fraudulent, and within a window. Although parameters can be varied to fine-tune them, they are fixed for this experiment. The parameters which will have a significant impact on the suspicion scores are g, w, T_{E_I}, and α. If normalised *Levenshtein* similarity (Chapman, 2005) and $T_{similarity} \geq 0.8$, then both attributes are a match. The experiment also cross-matches *given_name* with *surname*, *current_address* with *previous_address*, and *home_phone* with *mobile_phone* numbers. If T_{fraud} and $T_{anomalous}$ have 3 or more attributes which match, then both applications are linked. In this experiment, there are 126 normal relationships defined in $\Re$.

The results show a significant change of link activity in scoring phase over training phase and it is due to seasonal frauds. This is observed in the increase of links h, mean nodal degree $\bar{d}$, and average suspicion score $\frac{\sum_{i=1}^{N} S(v_i)}{g}$. The graph density Δ is higher for training than scoring phase. As evidenced with the computation time in Table 3, larger g and/or w (necessary for the training phase) will cause the technique to run into scalability problems.

Table 3: Parameters and some basic measurements.

	Training	Scoring
g	10,000	44,238
w	10,000	10,000
$T_{similarity}$	0.8	0.8
T_{E_I}	10	10
$T_{fraud} = T_{anomalous}$	3	3
$\Re_N$	126	126
α	0.8	0.8
h	2,917	17,868
$computation_{time}$	2.7 secs/app 452 mins	6.3 secs/app 4,688 mins
$\bar{d}_I = \bar{d}_O = \frac{h}{g}$	0.292	0.418
$\Delta = \frac{h}{g(g-1)}$	0.000029	0.00001
$\frac{\sum_{i=1}^{N} S(v_i)}{g}$	0.116	0.197

4. DISCUSSION

The descriptive directed graphs in Appendices A and B allow the analyst to visually inspect/explore the subgraphs or "communities of interest" (Cortes *et al*, 2003). The predictive suspicion scores in Appendix C allow the analyst to rank the most recent applications

and be aware of subgraphs with $S(v_i) > T_{lower}$ and investigate applications with $S(v_i) \geq T_{upper}$.

Appendix A presents the compressed hierarchical subgraph structures of linked applications after training. The synthetic data generation process causes all the 27 different types of subgraphs to be disconnected although this is unlikely amongst real applications.

Appendix B (drilled down from Appendix A) displays 50 hand-crafted applications which are separated into 7 subgraphs with vertice and link labels. Each subgraph illustrates a different type of linked applications:

- Subgraph (i) consists of known fraud applications and subsequent applications which link to them.
- Subgraph (ii) is made up of linked applications which have frequent address changes by the same identity.
- In addition to subgraph (ii), subgraph (iii) includes linked applications submitted by an identity's social network.
- Subgraph (iv) consists of linked applications with data entry errors which can also turn out to be frauds.
- Subgraph (v) illustrates synthetic applications which "mix and match" attributes from a few other previous applications.
- Subgraph (vi) have all exact applications to demonstrate the effects of temporal and spatial weights.
- Subgraph (vii) show the effects of α (exponential smoothing).

The link weights for each graphical link are $w_{communal_{ij}}$, and if the link belongs to the black list or white list, its description will be appended. The newest vertices are on top (with no incoming links and highest outgoing links in the subgraph) and the oldest ones are at the bottom (with the highest incoming links and no outgoing links in the subgraph). The 3 applications with dotted links and outside the groups are from the scoring phase. As 2 of are received in December and 1 in June, $w = 10{,}000$ is not large enough to link the 3 applications to their groups. To overcome this scalability issue, in descending priority, the following can be implemented: explore if temporal and spatial weights increase predictive power, reduce long loops and excessive access to data, convert code to C or Java scripts and place data in text files, and use of parallel and/or distributed computing (our future work).

Appendix C shows the subgraphs and their application predictive scores, outlinks, and inlinks.

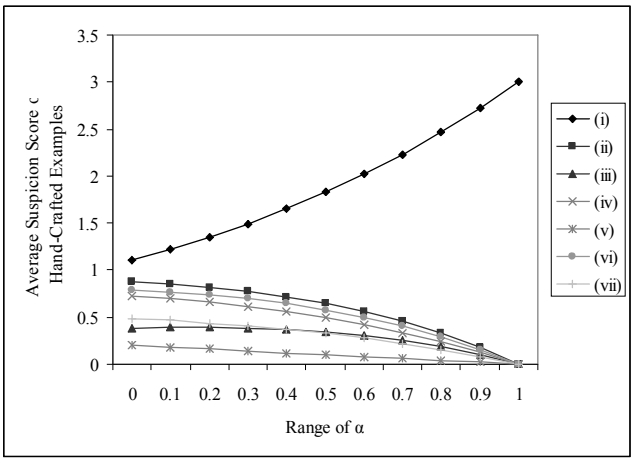


Figure 1: Effects of α on average suspicion score of 7 subgraphs.

Figure 1 above shows that as α increases, all the average scores will decrease except for subgraph (i) (known fraud related applications) which will increase steadily. The scores in Appendix C are obtained by heuristically setting $\alpha = 0.4$ where the average score of subgraph (i) is between 1.5 and 2. If the thresholds are set as $T_{lower} = 0.8$ and $T_{upper} = 1$, out of 50 applications, 13 will be investigated (strong fraud symptoms) and 3 are considered suspicious (some fraud symptoms). The investigations will be prioritised on linked applications from subgraphs (i), (ii), (vi), (iv), and (vii) as they are either connected to known frauds (i), have too many address changes (ii) or similar applications (vii) or large geographical distances (vi) within a short time frame, or data entry errors/fraud (iv). Despite having the most number of linked applications, subgraph (iii) will not be investigated (although one of them is suspicious) as it has legitimate links defined by the white list, therefore scores are lower. It is interesting to note that the synthetic fraud in subgraph (v) cannot be detected at an early stage. However to counter that, a more sophisticated credit application fraud detection system can be augmented by an attribute-value temporal/spike/correlation analysis technique (our future work).

5. RELATED WORK

There is no academic research, to the best of our knowledge (Phua *et al*, 2005); into the scoring of dynamic credit applications which accounts for its sparse-identifiers, communal, temporal, and spatial aspects. However, there are other related and established application fields in multi-attribute pair-wise matching (e.g. record linkage/de-duplication detection), and single-attribute communal scoring/directed graphs with explicit links (e.g. telecommunications fraud detection, terrorist detection, social network analysis, and webpage ranking). Below are representative work in these areas and some explanation on how it is related (or not) to the work in this paper:

Bilenko *et al* (2003) applies a series of character-based and token-based string similarity metrics to identifying approximately duplicate database records from multiple sources. They propose treating each record as a set of fields and then measuring the

average similarity across these fields. By doing so, the record's similarity is represented with a feature vector (represented by our attribute vector y). They also propose adapting string similarity metrics' edit operation's cost to each attribute (not implemented as it is computationally too expensive for large data sets). The authors applied their techniques to comparatively small data sets and claims good results on some data sets.

Cortes *et al* (2003) illustrates computational methods to large dynamic graphs of entity-pair interactions for telecommunications fraud detection. To prevent the management and storage of many graphs from different time steps, it exponentially smooths the previous and current graphs (our approach is to smooth linked nodes/applications) and uses a continually updated top-*k* link set for each node/telephone account to monitor suspicious calling patterns (after scoring incoming applications, they will not be updated unless when they have been flagged as fraudulent). The authors dealt with hundreds of millions of nodes and billions of links per day.

Macskassy and Provost (2005) also use suspicion scoring to detect malicious individuals and their associates. Their relational classification and collective inference estimates suspicion as the weighted sum of connected individuals (very similar to our smoothed scoring function except our total suspicion score for an application is not scaled to between 0 and 1, and our approach requires no algorithm iteration as it takes into account the temporal sequence of applications). Their algorithm is tested on data sets created by a terrorist-world simulator and concludes that good rankings are generated even with a small number of known labels and moderate noise.

Kubica *et al* (2003) examines graphs and distance measures for predicting future links between friends. It uses temporal weights to exponentially smooth the effects of older links (different to our temporal weights which reflect the importance for days difference in each application-pair, and also different to our exponential smoothing of current and previous scores), and typical weights to reflect the quality/importance of link types (similar to our $w_{communal_{ij}}$). Their approach was evaluated on five link data sets together with five other competing algorithms, and their proposed algorithm was either the best performer or close to the best performer.

Brin and Page (1998) and Kleinberg (1999) utilise the web link structure to determine the importance of webpages. *PageRank* (Brin and Page, 1998) uses a hyperlink to a webpage as a popularity vote, is defined recursively (not possible if applications need to be processed in real-time), and depends on quantity and *PageRank* webpages' quality of incoming links (regarding this aspect, our approaches are very similar except we measure suspiciousness of the current/most recent application based on outgoing links). *PageRank* is an essential part of Google's search engine. *HITS* (Kleinberg, 1999) recursively seeks to find static webpages which are authorities (provides good information and so many webpages link to it) and hubs (links to many authorities) to locate similar topic groups (we also penalise current application scores heavily if the incoming links threshold T_{E_I} is exceeded for any previous applications which the current one links to).

6. CONCLUSION

We have discussed our communal scoring technique, performed preliminary experiments on simulated data, briefly discussed the results and related work. In the near future, our attention will be on the empirical linking and scoring of a few million real applications with different parameter values.

ACKNOWLEDGMENTS

This research is financially supported by the Australian Research Council under Linkage Grant Number LP0454077. Special thanks to the developers of FEBRL/DBGen data set generator and yEd for their useful software.

REFERENCES

Baycorp Advantage. (2005). Zero-Interest Credit Cards Cause Record Growth In Card Applications.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P. & Fienberg, S. (2003). Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems* **18**(5): pp16-23.

Christen, P. (2005). Probabilistic Data Generation for Deduplication and Data Linkage. *Proc. of the Sixth International Conference on Intelligent Data Engineering and Automated Learning*.

Chapman, S. (2005). Simmetrics – Open Source Similarity Measure Library. Accessed from: http://sourceforge.net/projects/simmetrics/. Accessed in April 2005.

Cortes, C., Pregibon, D. & Volinsky, C. (2003). Computational Methods for Dynamic Graphs, *Journal of Computational and Graphical Statistics* **12**: pp950-970.

ID Analytics. (2004). Identity 2004: The Identity Risk Management Conference.

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM* **46**(5): pp604-632.

Kubica, J., Moore, A., Cohn, D. & Schneider, J. (2003) Finding Underlying Connections: A Fast Graph-Based Method for Link Analysis and Collaboration Queries. *Proc. of the International Conference on Machine Learning*: pp392-399.

Macskassy, S. & Provost, F. (2005). Suspicion scoring based on Guilt-by-Association, Collective Inference, and Focused Data Access. *Proc. of the International Conference on Intelligence Analysis*.

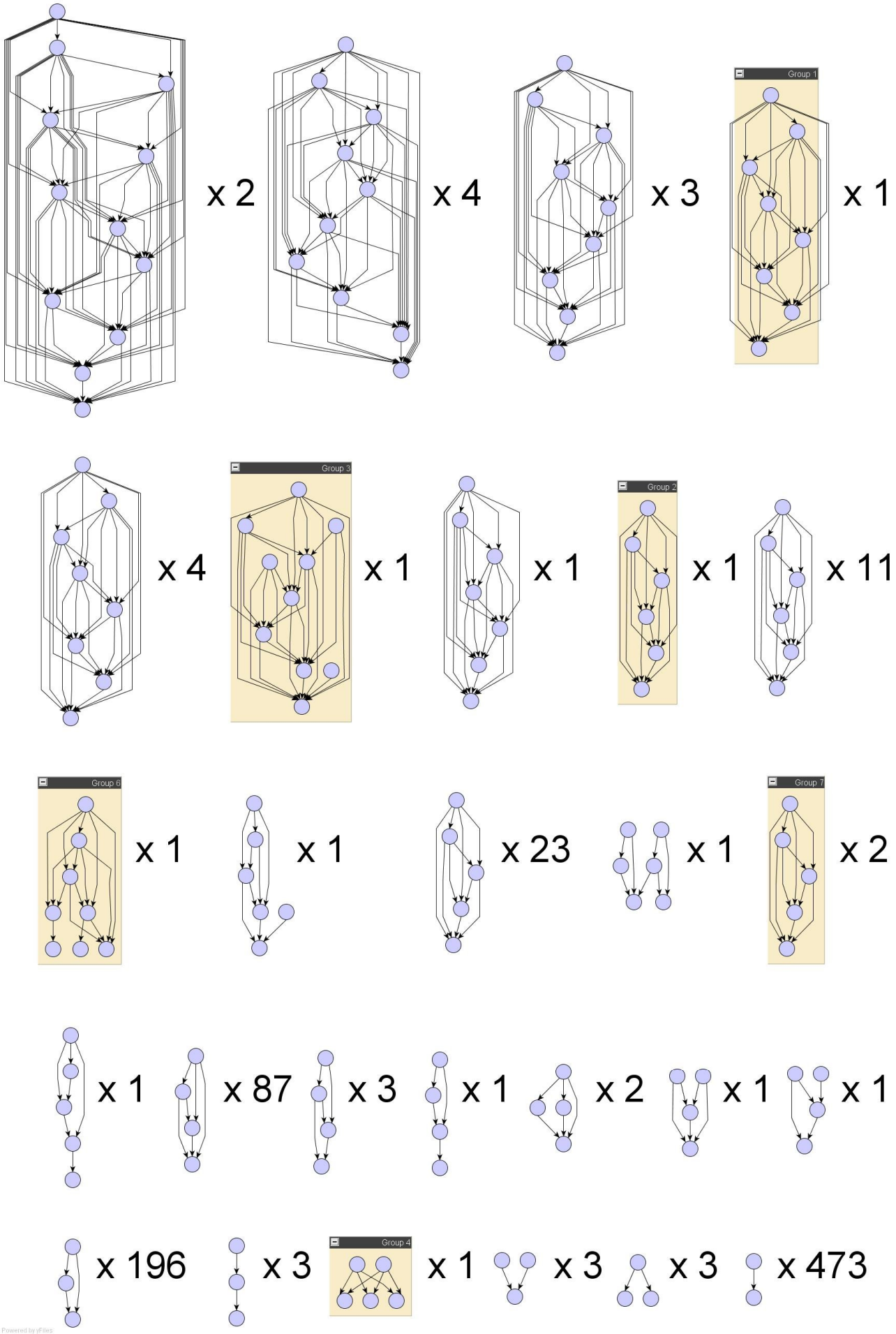
Oscherwitz, T. (2005). Synthetic Identity Fraud: Unseen Identity Challenge. *Bank Security News* **3**(7).

Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. *Stanford Digital Library Project Technical Report*.

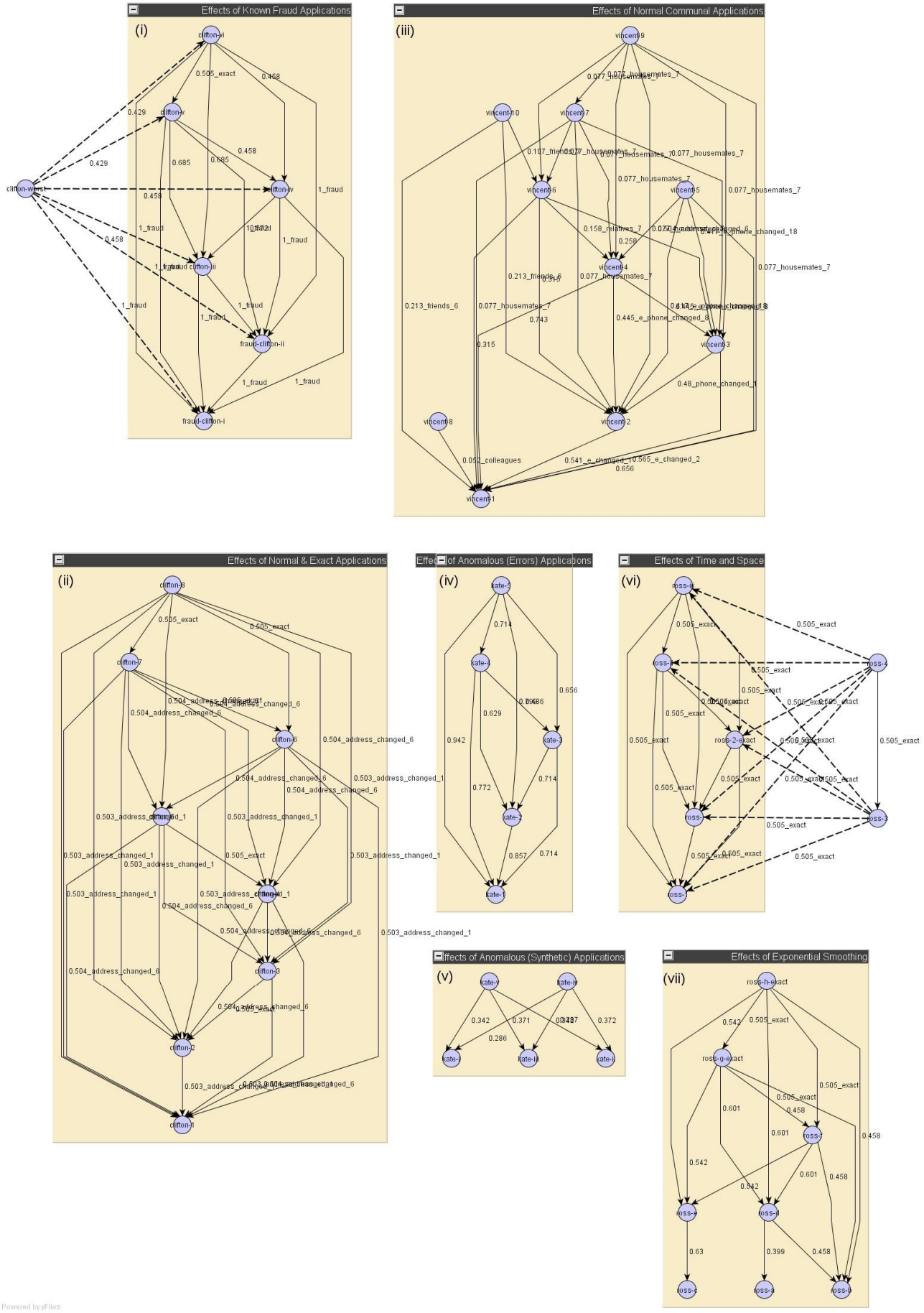
Phua, C., Lee, V., Smith, K. & Gayler, R. (2005). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *Artificial Intelligence Review*, submitted.

Wasserman, S. & Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press, New York.

APPENDIX A


x 2
x 4
x 3
Group 1
x 1
x 4
Group 3
x 1
x 1
Group 2
x 1
x 11
Group 6
x 1
x 1
x 23
x 1
Group 7
x 2
x 1
x 87
x 3
x 1
x 2
x 1
x 1
x 196
x 3
Group 4
x 1
x 3
x 3
x 473

APPENDIX B


Effects of Known Fraud Applications
(i)
clifton-vi
clifton-v
clifton-iv
clifton-worst
clifton-iii
clifton-ii
fraud-clifton-ii
fraud-clifton-i
0.505_exact
0.458
0.429
0.685
1_fraud
Effects of Normal Communal Applications
(iii)
vincent9
vincent10
vincent7
vincent6
vincent5
vincent4
vincent3
vincent8
vincent2
vincent1
0.077_housemates_7
0.107_friends_7
0.158_relatives_7
0.213_friends_6
0.445_e_phone_changed_8
0.48_phone_changed_1
0.052_colleagues
0.541_e_changed_1
0.565_e_changed_2
0.656
0.743
0.315
0.258
Effects of Normal & Exact Applications
(ii)
clifton-9
clifton-7
clifton-6
clifton-5
clifton-3
clifton-2
clifton-1
0.505_exact
0.504_address_changed_6
0.503_address_changed_1
Effects of Anomalous (Errors) Applications
(iv)
kate-5
kate-4
kate-3
kate-2
kate-1
0.714
0.656
0.629
0.942
0.772
0.857
Effects of Time and Space
(vi)
ross-vi
ross-v
ross-iv
ross-iii
ross-ii
ross-i
ross-2-exact
0.505_exact
Effects of Anomalous (Synthetic) Applications
(v)
kate-v
kate-iv
kate-iii
kate-ii
kate-i
0.342
0.371
0.372
0.286
Effects of Exponential Smoothing
(vii)
ross-i-exact
ross-g-exact
ross-f
ross-e
ross-d
ross-c
ross-b
ross-a
0.542
0.505_exact
0.601
0.458
0.63
0.399

APPENDIX C[1]

subgraph	rec_id	date_ received	given_ name	surname	street_ number	current_ address	previous_ address	suburb	postcode	state	home_ phone	mobile_ phone	. . .	driver_ licence_id	date_of_ birth	suspicion_ score	outlinks	inlinks
(i)	clifton-worst	31/12/2004	junwei	pan	3	jean avenue	marsh avenue	cleyton	6000	vic	85777756	707565107	. . .	87870287	2/07/1978	2.714	6	0
(i)	clifton-vi	7/01/2004	chun wei	pin	2	aven jean	marsh ave	creyton	3168	vic	85777756	88080808	. . .	87870287	9/07/1978	2.642	5	0
(i)	clifton-v	6/01/2004	chun wei	pin	2	aven jean	marsh ave	creyton	3168	vic	85777756	88080808	. . .	87870287	9/07/1978	2.266	4	1
(i)	clifton-iv	5/01/2004	junyu	pen	3	jean ave	marsh avenue	cleyton	3168	vic	85777756	88080808	. . .	87870287	3/07/1978	1.849	3	2
(i)	clifton-iii	4/01/2004	junge	pin	3	jean avenue	marsh ave	cleyton	3168	vic	85777756	88080808	. . .	87870287	4/07/1978	1.399	2	3
(i)	fraud-clifton-ii	3/01/2004	junwei	pan	3	jean avenue	marsh avenue	cleyton	3168	vic	85777756	707565107	. . .	87870287	2/07/1978	0.7	1	4
(i)	fraud-clifton-i	2/01/2004	junwei	pan	3	jean avenue	marsh avenue	cleyton	3168	vic	85777756	707565107	. . .	87870287	2/07/1978	0	0	5
(ii)	clifton-8	15/01/2004	clifton	phua	5	atlan street	aven hill	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	1.509	7	0
(ii)	clifton-7	14/01/2004	clifton	phua	5	atlan street	aven hill	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	1.274	6	1
(ii)	clifton-6	13/01/2004	clifton	phua	5	atlan street	aven hill	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	1.041	5	2
(ii)	clifton-5	12/01/2004	clifton	phua	5	aven hill	hill ave	gwen waverley	3150	vic	85775751	711887106	. . .	12775668	1/07/1978	0.811	4	3
(ii)	clifton-4	10/01/2004	clifton	phua	5	aven hill	hill ave	gwen waverley	3150	vic	85775751	711887106	. . .	12775668	1/07/1978	0.583	3	4
(ii)	clifton-3	8/01/2004	clifton	phua	20	hill avenue	nort road	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	0.36	2	5
(ii)	clifton-2	5/01/2004	clifton	phua	20	hill avenue	nort road	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	0.15	1	6
(ii)	clifton-1	1/01/2004	clifton	phua	1474	nort road	juro west	cleyton	3168	vic	85775751	711887106	. . .	12775668	1/07/1978	0	0	7
(iii)	vincent-10	11/01/2004	friend	lim	90	russell street	bour street	melbourne	3000	vic	65876587	78762858	. . .	58668287	13/10/1948	0.279	3	0
(iii)	vincent-9	10/01/2004	neighbour	van basten	1	wellin road	black road	knocks city	3200	vic	67656787	765678677	. . .	78587875	6/05/1980	0.447	6	0
(iii)	vincent-8	9/01/2004	colleague	tan	2	floren avenue	murd road	cleyton	3168	vic	67567658	765876788	. . .	66565657	3/09/1959	0.015	1	0
(iii)	vincent-7	8/01/2004	housemate	jones	1	wellin road	black road	knocks city	3200	vic	67775687	765876587	. . .	78785677	4/07/1958	0.393	5	1
(iii)	vincent-6	7/01/2004	wife	lee	1	wellin road	black road	knocks city	3200	vic	67856287	785662866	. . .	65676588	5/02/1953	0.536	4	3
(iii)	vincent-5	6/01/2004	vincent	lee	1	jeffer road	jane road	cadston	3148	vic	82758656	787567566	. . .	87678857	3/04/1950	0.824	4	0
(iii)	vincent-4	5/01/2004	vincent	lee	1	wellin road	black road	knocks city	3200	vic	82758656	787567566	. . .	87678857	3/04/1950	0.629	3	4
(iii)	vincent-3	4/01/2004	vincent	lee	1	wellin road	black road	knocks city	3200	vic	67856287	786785625	. . .	87678857	3/04/1950	0.378	2	5
(iii)	vincent-2	3/01/2004	vincent	lee	1	wellin road	black road	knocks city	3200	vic	67856287	786785625	. . .	87678857	3/04/1950	0.162	1	7
(iii)	vincent-1	2/01/2004	vincent	lee	1	wellin road	black road	knocks city	3200	vic	67856287	786785625	. . .	87678857	3/04/1950	0	0	9
(iv)	kate-5	5/01/2004	kate	smith	34			cleyton	3168	vic	78765618	676287676	. . .	5827582	12/10/1973	1.242	4	0
(iv)	kate-4	4/01/2004	ronald	hardy	34	sesame street	elmo lane	cleyton	3168	vic	78765618	676287676	. . .	5827582	12/10/1973	0.773	3	1
(iv)	kate-3	3/01/2004	kate	smith	34	sesame street	elmo lane	cleyton	3168	vic	78765618	676287676	. . .	5862776	3/03/1975	0.53	2	2
(iv)	kate-2	2/01/2004	kate	smith	34	sesame street	elmo lane	cleyton	3168	vic	78765618	676287676	. . .	5827582	12/01/1973	0.257	1	3
(iv)	kate-1	1/01/2004	kate	smith	34	sesame street	elmo lane	cleyton	3168	vic	78765618	676287676	. . .	5827582	12/10/1973	0	0	4
(v)	kate-iv	5/01/2004	a	n	31	c	p	k	3168	vic	2	27	. . .	16	2/02/1945	0.299	3	0
(v)	kate-v	5/01/2004	m	h	41	i	j	e	3168	vic	22	7	. . .	6	3/03/1945	0.298	3	0
(v)	kate-i	1/01/2004	a	b	1	c	d	e	3168	vic	2	7	. . .	6	1/01/1945	0	0	2
(v)	kate-ii	1/01/2004	g	h	11	i	j	k	3168	vic	12	17	. . .	16	2/02/1945	0	0	2
(v)	kate-iii	1/01/2004	m	n	21	o	p	q	3168	vic	22	27	. . .	26	3/03/1945	0	0	2
(vi)	ross-4	1/12/2004	A	B	1	C	D	E	3168	vic	1277	2775	. . .	6688	1/08/1960	1.026	6	0
(vi)	ross-3	1/06/2004	A	B	1	C	D	E	3168	vic	1277	2775	. . .	6688	1/08/1960	1.049	5	1
(vi)	ross-iii	4/01/2004	A	B	1	C	D	E	6000	wa	1277	2775	. . .	6688	1/08/1960	1.245	4	0
(vi)	ross-ii	3/01/2004	A	B	1	C	D	E	2000	nsw	1277	2775	. . .	6688	1/08/1960	0.665	3	1
(vi)	ross-2-exact	2/01/2004	A	B	1	C	D	E	3168	vic	1277	2775	. . .	6688	1/08/1960	0.364	2	2
(vi)	ross-i	2/01/2004	A	B	1	C	D	E	3000	vic	1277	2775	. . .	6688	1/08/1960	0.152	1	3
(vi)	ross-1	1/01/2004	A	B	1	C	D	E	3168	vic	1277	2775	. . .	6688	1/08/1960	0	0	4
(vii)	ross-h-exact	8/01/2004	M	T	1	O	V	Q	3168	vic	102	827676657	. . .	106	5/06/1981	1.071	5	0
(vii)	ross-g-exact	7/01/2004	M	T	1	O	V	Q	3168	vic	102	827676657	. . .	106	5/06/1981	0.837	4	1
(vii)	ross-f	6/01/2004	M	T	1	O	V	Q	3168	vic	102	827676657	. . .	106	5/06/1981	0.605	3	2
(vii)	ross-e	5/01/2004	S	T	1	U	V	W	3168	vic	202	827676657	. . .	206	5/06/1981	0.188	1	3
(vii)	ross-d	4/01/2004	M	H	1	O	J	Q	3168	vic	102	8866	. . .	106	5/06/1981	0.256	2	3
(vii)	ross-c	3/01/2004	S	T	1	U	V	W	3168	vic	202	207	. . .	206	1/04/1974	0	0	1
(vii)	ross-b	2/01/2004	M	N	1	O	P	Q	3168	vic	102	107	. . .	106	20/04/1983	0	0	4
(vii)	ross-a	1/01/2004	G	H	1	I	J	K	3168	vic	886	8866	. . .	5772	30/01/1955	0	0	1

[1]Note that all data presented, described, and experimented with in this paper are fictional (except author names) and any similarities to any other actual person are purely coincidental.