AMSI
AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

National Collaboration in the Mathematical Sciences

Random Graph Prototypes for Noisy Biometric Graphs

Adrian Hecker

April 30, 2013

1 Introduction

Riesen and Bunke (2010) demonstrated that the dissimilarity representation approach proposed by Pekalska and Duin (2005) outperforms traditional pattern recognition systems when applied to the domain of noisy graphs [1]. In particular, they had great success with the task of classifying fingerprints into their basic types (ie arch, whorl or loop). The dissimilarity representation is a method of embedding a graph to an n-dimensional vector space. This method helps overcome the complexity of working with objects in the graph domain, and allows the use of many well defined mathematical operations in the vector space. The embedding procedure of Pekalska and Duin uses a graph distance measure against a set of n user-defined prototype graphs. By means of this method, the graph is then represented in a vector space where each axis is associated with the input graph's distance to each prototype graph. We refer to this space as the dissimilarity space. Riesen and Bunke undertook their thesis using only a metric distance measure for the embedding procedure, the graph edit distance. Riesen and Bunke also demonstrated that a well defined set of prototypes is necessary for successful matching.

In this report, the following questions are addressed:

1. Do randomly generated prototypes distinguish between genuine and imposter matches in the noisy biometric graph domain?
2. Does a non-metric graph distance measure outperform the graph edit distance measure used by Riesen and Bunke?

Postal Address: 111 Barry Street
c/- The University of Melbourne
Victoria 3010 Australia

Email: enquiries@amsi.org.au
Phone: +61 3 8344 1777
Fax: +61 3 9349 4106
Web: www.amsi.org.au

3. Is there an optimum size for the set of prototype graphs?

2 Method

2.1 Biometric Graph Data

The fingerprints used in this project were obtained from the publicly accessible FVC2002 [2] database of fingerprint images. For this work, the DB1 database was used. This database contains 110 fingerprints, and 8 samples of each fingerprint (880 fingerprint images in total). Graph representations of the fingerprint images have been obtained in previous research completed at RMIT University [3]. The minutiae extracted from the fingerprints represents a node. An edge exists between two nodes if there is a ridge physically connecting the corresponding minutiae. The graphs were obtained using manual and automatic extraction methods. The graph obtained is undirected and nodes are labeled, where the labeling represents the (x,y) co-ordinate of the node.

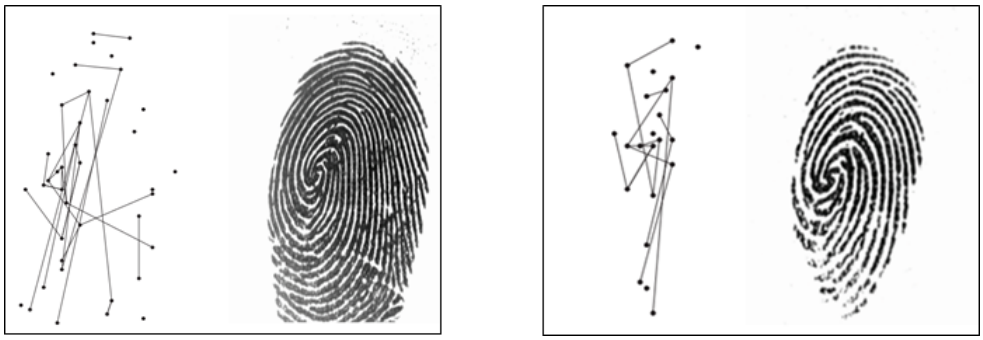

Figure 1: Examples of graphs extracted from fingerprint images.

2.2 Random Graph Generation

One of the first tasks was to generate a set of graphs for use as the prototype set. The distribution of the number of nodes, edges and (x,y) co-ordinates of the graphs obtained from the FVC2002 DB1 set of fingerprints was tested and was found to be approximately normally distributed. A randomly generated graph is then built with the following algorithm.

1. Input mean and standard deviation scores for the number of nodes, number of edges and the (x,y) co-ordinates.
2. Select a random number of nodes from the distribution of the number of nodes.
3. Randomly assign (x,y) co-ordinates to each node, taking co-ordinates from their respective distributions.
4. Select a random number of edges from the distribution of the number of nodes.
5. Assign the edges to the nodes randomly.

A set of 1000 random graphs for use as prototypes was generated using this method.

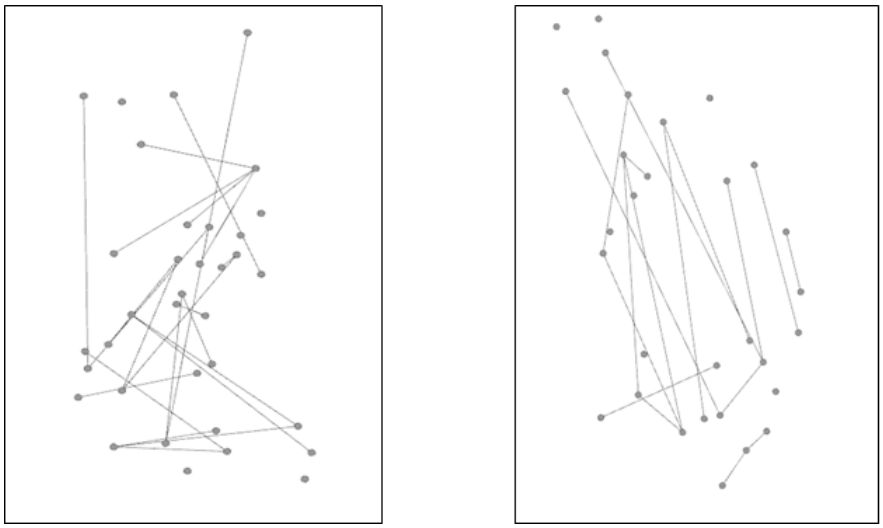

Figure 2: A comparison between a randomly generated graph, and a graph extracted from a fingerprint image. The left graph is the random graph 927, the right graph is taken from subject 55, sample 2.

2.3 Graph Dissimilarity Measures

The graph edit distance is a function that assigns a cost to the number of distortions required to turn one graph into another graph. A larger graph edit distance suggests greater dissimilarity between two graphs, and vice versa. In this project, a suboptimal algorithm proposed by Riesen and Bunke [4] is used. This algorithm is also used to

obtain the maximum common subgraph between two input graphs.

Figure 3: An example of a graph edit path between two graphs, g_1 and g_2.

The non-metric distance measure used is the square root distance. This measure of distance has been shown to outperform metric distance measures when used to match fingerprint graphs [3]. This function assigns a number $[0, 1]$ according to the number of matched nodes, edges or a combination of both between two graphs. A score of 1 suggests complete dissimilarity between two graphs and a score of 0 suggests two graphs are identical. Formally, the square root distance is defined as follows:

Given two graphs g and g', the square root distance is:

$$d^{SQRT} = 1 - \frac{s(g, g')}{\sqrt{|g||g'|}}$$

Where $s(g, g')$ is the number of matched nodes, edges or a combination of both. $|g|$ and $|g'|$ is simply the number of nodes, edges or both of g and g'. The number of matched nodes and/or edges is obtained from the maximum common subgraph.

2.4 Graph Embedding Procedure

The graph embedding procedure is formally defined as follows:

Assume a graph domain $\mathcal{G}$, and set of prototype graphs $\mathcal{P} = \{p_1, \ldots, p_n\}$ with n graphs is given, the mappping:

$$\varphi_n^{\mathcal{P}} : \mathcal{G} \to \mathbb{R}^n$$

is defined as the function:

$$\varphi_n^{\mathcal{P}}(g) = (d(g, p_1), \ldots, d(g, p_n)),$$

where $d(g, p_i)$ is any graph dissimilarity measure between graph g and the i-th prototype graph.

In this case, the prototype set used is a set of graphs generated randomly. By means of this definition, a vector space where each axis is associated with a prototype graph $p_i \in \mathcal{P}$ and the coordinates of the embedded graph are the distances of g to the prototypes of $\mathcal{P}$. Graph embeddings were carried out using the graph edit distance, and square root distance using nodes, edges and a combination of both. This means the graphs were embedded into four different dissimilarity spaces.

2.5 Dissimilarity Space Distance Measures

After two graphs g and g' have been mapped to the dissimilarity space, the distance between the embedded graphs can be found using any vector space measure. In this project, the following distance measures were used: Euclidean distance (d^{EUC}), vector angular distance (d^{VAD}), Chebyshev distance (d^{CSD}) and the Canberra distance (d^{CAD}). Assume two graphs, g and g', have been embedded into the dissimilarity space. Writing $\varphi_n^{\mathcal{P}}(g)$ as $\varphi(g)$ for simplicity, the distance measures are defined as follows:

$$d^{EUC}(\varphi(g), \varphi(g')) = \left(\sum_{i=1}^{n} (d(g, p_i) - d(g', p_i))^2 \right)^{\frac{1}{2}}$$

$$d^{VAD}(\varphi(g), \varphi(g')) = \frac{1}{\pi} \cos^{-1} \left(\frac{\langle \varphi(g), \varphi(g') \rangle}{\|\varphi(g)\| \|\varphi(g')\|} \right)$$

$$d^{CSD}(\varphi(g), \varphi(g')) = \max_i (|d(g, p_i) - d(g', p_i)|)$$

$$d^{CAD}(\varphi(g), \varphi(g')) = \sum_{i=1}^{n} \frac{|d(g, p_i) - d(g', p_i)|}{|d(g, p_i)| + |d(g', p_i)|}$$

Essentially, the Euclidean distance is a direct 'ruler' measure of how far each point is in the metric space. A high Euclidean distance suggests greater dissimilarity between the two embedded graphs, and vice-versa. The vector angular distance assigns a number

$[0, 1]$ according to the angle between the two vectors. A $180°/\pi$ angle maps to 1, and a 0 angle maps to 0. The Chebychev distance disregards all other axis' with the exception of the axis with the greatest distance between them. The Canberra distance is a metric distance measure that has been shown to be useful in measuring data scattered around the origin [5].

2.6 Implementation

The entirety of the project was undertaken using code written in R. 200 graphs were selected randomly from the set of random graphs generated, making for a prototype set size of 200. 25 fingerprints, and four samples of each fingerprint were chosen arbitrarily from the set of FVC2002 fingerprint graphs. A total of 100 graphs were embedded to the dissimilarity space using the graph edit distance, and square root distance. Scores for matching fingerprints were obtained by measuring the distance between fingerprint graphs obtained from the same subject. Scores for imposters/non-matching fingerprints were obtained by measuring the distance of each fingerprint against a fingerprint taken from another finger. 75 scores were obtained for matches and non-matches.

3 Results

For a successful separation between genuine and imposter matches, we require an obvious difference between the scores taken from matches and non-matches. The scores for matches should be positively skewed, and non-matches negatively skewed. A summary of the mean scores and their standard deviations is given below. Each cell gives the mean score, ± the standard deviation. The square root distance using a combination of nodes and edges offered the best performance, so only these scores have been included.

Table 1: Distance Scores - Prototype Set Size 200

Graph Edit Distance

	Genuine	**Imposters**
d^{EUC}	219.72±158.2	272.72±187.89
d^{VAD}	0.0087±0.0038	0.0103±0.0045
d^{CSD}	20.676±11.43	24.365±13.583
d^{CAD}	0.7898±0.576	0.9914±0.6873

Square Root Distance

	Genuine	**Imposters**
d^{EUC}	0.2691±0.0447	0.2786±0.0348
d^{VAD}	0.0062±0.0011	0.0064±0.0007
d^{CSD}	0.0707±0.0172	0.0714±0.0169
d^{CAD}	0.1082±0.0228	0.1276±0.0151

No combination of graph dissimilarity measure and dissimilarity space measure offers a clear separation between genuine and imposter matches. The results suggest the graph edit distance combined with the Euclidean distance and the vector angular distance offer the best separation. As can be seen from figures 4 and 5 below, the scores are more skewed to the left for matches. However, there is still no clear difference between the two sets of scores and thus this method would not be reliable for the task of separating between matches and non-matches. Reducing the size of the prototype set only offered a poorer separation between the match and imposter scores, with the separation between matches and imposters only getting worse the smaller the prototype set size. Distance scores for the prototype set size of 100 are included below.

Table 2: Distance Scores - Prototype Set Size 100

Graph Edit Distance

	Genuine	**Imposters**
d^{EUC}	155.25±111.76	225.97±151.94
d^{VAD}	0.0086±0.0038	0.0116±0.0051
d^{CSD}	20.185±11.497	27.609±15.561
d^{CAD}	0.5556±0.4052	0.8241±0.551

Square Root Distance

	Genuine	**Imposters**
d^{EUC}	0.1863±0.032	0.2018±0.0286
d^{VAD}	0.0061±0.001	0.0064±0.0009
d^{CSD}	0.0597±0.013	0.0609±0.0107
d^{CAD}	0.0757±0.015	0.0912±0.0145

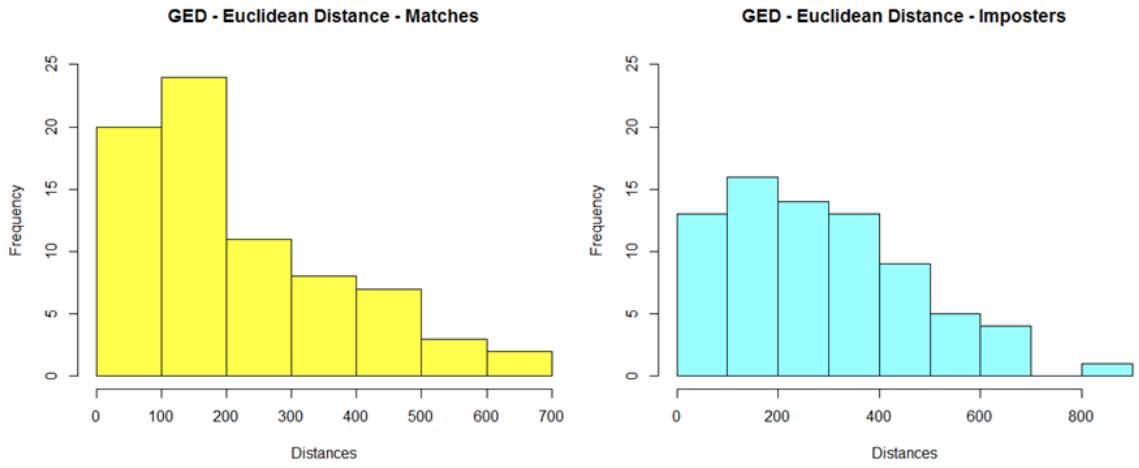

Figure 4: Histogram plots of the distance scores obtained using the graph edit distance and the Euclidean distance. Prototype set size 200.

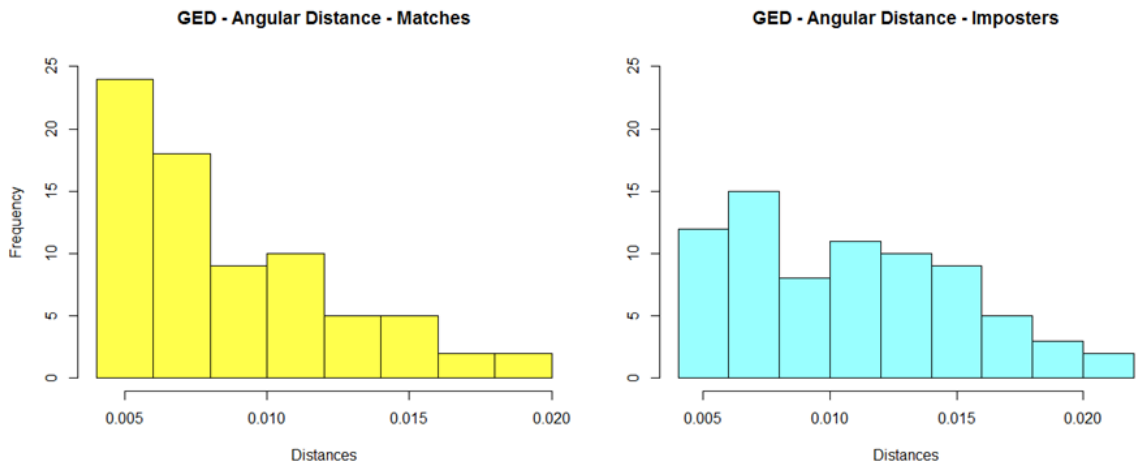

Figure 5: Histogram plots of the distance scores obtained using the graph edit distance and the vector angular distance. Prototype set size 200.

4 Conclusion

The results demonstrate that the metric graph edit distance measure outperforms the non-metric square root distance when embedding biometric graphs to the dissimilarity space, and that the Euclidean distance and the vector angular distance were the best performing measures of distance in the dissimilarity space. The results suggest that larger prototype set offers the best separation between imposter and genuine matches. Larger prototype set sizes, such as 500 to 1000, may offer a good separation between the two scores. This, however would require more computationally efficient graph distance measures. Computing between the distance between two graphs using the graph edit distance/maximum common subgraph algorithm took approximately 20 seconds using an Intel Core 2 Duo 2.13Ghz computer. The embedding quickly becomes very computationally expensive with large prototype graph sets. Embedding biometric graphs using a very large prototype set and a combination of computationally efficient algorithms and/or computers with far higher processing power offers a potential rewarding area for further research.

I would like to thank Dr. Arathi Arakala for her assistance in providing the biometric graph data and R code for the graph distance measures. A very special thank you goes to Prof. Kathy Horadam, her guidance and advice over the summer were a tremendous

support. Finally, I would like to thank AMSI for the funding and opportunity to undertake this project, and CSIRO for hosting the Big Day In.

References

[1] K. Riesen and H. Bunke. Graph classification based on vector space embedding. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(06):1053–1081, 2009.

[2] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. Fvc2002: Second fingerprint verification competition. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 811–814. IEEE, 2002.

[3] KJ Horadam, SA Davis, A. Arakala, and J. Jeffers. Fingerprints as spatial graphs: nodes and edges. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 400–405. IEEE, 2011.

[4] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image and Vision Computing*, 27(7):950–959, 2009.

[5] Syed Masum Emran and Nong Ye. Robustness of canberra metric in computer intrusion detection. In *Proc. IEEE Workshop on Information Assurance and Security, West Point, NY, USA*. Citeseer, 2001.