Detecting changes in time series of network graphs using minimum mean squared error and cumulative summation

B. Pincombe[1]

(Received 28 July 2006; revised 3 August 2007)

Abstract

Through characterising a computer network as a time series of graphs, with IP addresses on the vertices and edges weighted by the number of packets transmitted, we apply graph distance metrics to arrive at a measure of the distance between the network at different times. Two computationally simple methods of detecting change points in a one dimensional time series of this distance data are proposed. These techniques are cumulative summation and minimum mean squared error. This offers a very space efficient method of detecting change points as only the time series of graph distances and the network graph for the last time slice need be kept. The two techniques are compared on a dataset containing 102 consecutive working days

See http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/62 for this article, © Austral. Mathematical Soc. 2007. Published October 7, 2007. ISSN 1446-8735

of TCP/IP traffic collected from five probes on the enterprise network of an organisation with many tens of thousands of employees. Network managers identified three highly anomalous days, one a change point associated with the introduction of a web based personnel management system. Computationally simpler graph distance metrics are shown to yield better results than their more complicated counterparts when coupled with either change point detection technique.

Contents

1 Introduction

Graphical representations are widely used in computer vision and pattern recognition for character recognition [12], three dimensional object recognition [21], fingerprint classification [15], video indexing [18] and image registration [3]. Unknown objects are represented as graphs and compared to known models stored in a database, transforming a difficult recognition problem into a more tractable graph matching problem [2]. The broad literature available on graph distance metrics has been successfully transitioned to areas as diverse as text data mining [4] and detection of change in computer networks [5, 6, 19]. This article builds on the use of graph distance metrics to detect change in computer networks to consider the problem of detecting changes in the rate of change in computer networks.

Computer networks are constantly in flux but the patterns of change can be quite stable. Subtle network faults or gradual variations in usage can alter these patterns of change. Automated tip-offs advising system administrators of the time at which a suspected change in the rate of change occurred can be useful so long as they have low false alarm rates, high detection rates, minimal time lags, low system impact and low processing and memory requirements. The exact level and priority order of each of these requirements varies from system to system; for example, in some critical systems a high false alarm rate may be acceptable if needed to gain a high detection rate.

In this article, communications networks are represented as a time series of network graphs. The graph distances between sequential graphs are calculated using ten commonly used graph distance metrics: weight [19], Maximum Common Subgraph weight, MCS vertex, MCS edge [19, 6], edit [2], median edit [5], spectral, modality [13], diameter [8] and entropy distances. This radically reduces the amount of information that needs to be stored, as only the number representing the distance from the last graph to the present one and the graphs needed to calculate the next graph distance need be retained. These time series of graph distances are then tested for serial correlation of error terms to see if they meet the definition of a mean-shift series. If they do, Minimum Mean Squared Error and cumulative summation are used recursively to find change points. This is done off-line and over the entire time series in order to show that this method works even with long time series. In practice, the change point detection techniques would be applied on-line over a window of points much shorter than the entire series considered in the example thus increasing the chances there is only one change in the interval considered and improving the accuracy.

The example used is the problem that initially motivated this work. A total of 102 consecutive working days of TCP/IP traffic was collected from five probes on an enterprise network servicing several tens of thousands of employees. Network administrators identified changes in patterns of change beginning on days 22, 64 and 89. Prior to doing this they had seen a plot of the day-to-day distances based on the median edit graph distance metric and it is important to note that these days are outstanding peaks on that plot (see Figure 1). The introduction of a web based personnel management system and its growing use across the organisation was suggested as a reason for the alteration in change patterns detected from day 64 but no reasons were given for the choice of the other two days. This example still allows demonstration that the techniques described in this article can work in a timely manner on a large real-world dataset containing multiple change points, producing a memory efficient representation of day-to-day changes and results approximating those of human operators.

The errors in the time series produced by MCS weight were definitely too correlated to enable the application of either cumulative summation or MMSE and those in the weight and MCS edge generated time series were probably too correlated and so were excluded. Of the remaining time series the one generated by the diameter distance metric produced the best combination of actual detections and false alarms. None of the change points were detected through use of MMSE on time series generated using the spectral, entropy and modality distance metrics, but nor were any false alarms created. When MMSE was used on the MCS vertex generated time series two of the three change points were detected with five false alarms, one of which was next to the other change point. For the edit distance time series MMSE detected one change point and produced five false alarms, one of which was the neighbour of a change point. Use on the diameter distance time series produced two detections and two false alarms. Both change point detection methods

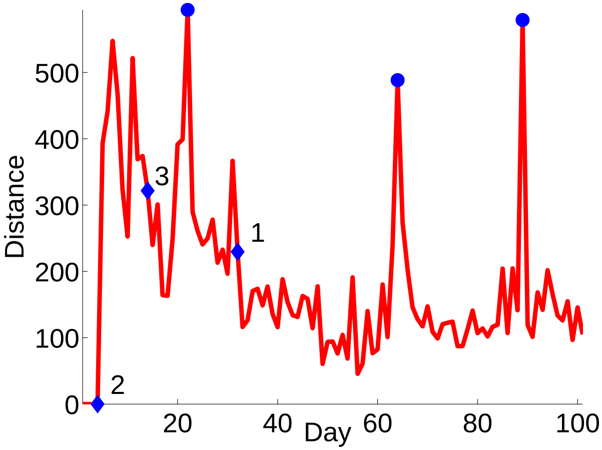

Figure 1: Median edit distance time series with change points detected using MMSE with $\alpha = \{0.01, 0.02, 0.03, 0.04, 0.05\}$.

performed poorly on the computationally intensive spectral and modality distance generated time series but that generated by the (approximately half as) computationally intensive diameter distance time series produced the best results. There is a greater degree of time averaging in the distance metrics that performed poorly. It could be that crisp change in the mean is required for techniques like MMSE and cumulative summation that are built on the assumption that the change point is a shift of the mean. Testing on simulated networks with known types of change over set regions is necessary before the trade off between false alarm rate and accuracy can be accurately estimated. This work can be extended to an on-line environment.

2 Constructing the time series of graph distances

The communications network for each day is characterised as a graph $G = (V, E)$ containing a finite set of vertices V and edges E. Vertices represent IP addresses. Edges, $(u, v) \in E$, are defined by the pair of vertices, u and v, that they join. Edges represent packet transmission between vertices, so $(u, v) \in E$ is an ordered pair, and the edges are, therefore, directed. Two vertices $u, v \in V$ are considered to be adjacent, $u \leftrightarrow v$, if there is an edge defined in terms of u and v; that is, u and v are joined by an edge. Vertices, edges and their combinations associated with a graph, G, are referred to as elements. All ten distance measures are metrics. Therefore, the distance between two graphs is a positive real number, $d(G, H) \in \mathbb{R}^+$; the zero distance is equivalent to graph isomorphism, all distance measures are symmetric, $d(G, H) = d(H, G)$, and they satisfy the triangle inequality, $d(G, F) \leq d(G, H) + d(H, F)$. The weight values, w_V and w_E, assigned to elements of the graph $G = (V, E, w_V, w_E)$ are symbolic for vertex weights, that is $w_V : V \to L_V$ where L_V are unique one-to-one labels for each $v \in V$, and numerical values based on the number of packets sent for edge weights with the weight $w_E : E \to \mathbb{R}^+$. All graphs have a unique one-to-one symbolic value for each vertex weight and are thus considered to be labelled. The number of vertices in $G = (V, E)$ is denoted by $|V|$ and the number of edges by $|E|$.

Several of the graph topology distance measures rely on identification and comparison of the elements in common between graphs by finding the Maximum Common Subgraph (MCS) between graph pairs [19]. A subgraph of $G = (V_G, E_G, w_V^G, w_E^G)$ is a graph $S = (V_S, E_S, w_V^S, w_E^S)$ where $V_S \subseteq V_G$ and $E_S \subseteq E_G \cap (V_S \times V_S)$. The vertex weight w_V^S of S is w_V^G restricted to V_S and the edge weight w_E^S of S is w_E^G restricted to E_S. The maximum common subgraph F of G and H, $F = \text{mcs}(G, H)$, is the common subgraph with the most vertices, that is, there is no other common subgraph K of G and H, with more vertices than F.

The **Weight** distance [19] between two graphs is

$$d(G,H) = |E_G \cup E_H|^{-1} \sum_{u,v \in V} \frac{|w_E^G(u,v) - w_E^H(u,v)|}{\max\{w_E^G(u,v), w_E^H(u,v)\}}\,, \tag{1}$$

where $w_E^{(\cdot)}(u,v)$ is the weight of the edge joining u and v; and $d(G,H)$ is the distance between graphs G and H [19]. The **MCS Weight** distance [19, 6] simplifies calculations by only considering those edges appearing in the MCS and is

$$d(G,H) = |E_G \cap E_H|^{-1} \sum_{u,v \in V} \frac{|w_E^G(u,v) - w_E^H(u,v)|}{\max\{w_E^G(u,v), w_E^H(u,v)\}}\,, \tag{2}$$

where $w_E^{(\cdot)}(u,v)$ is the weight of the edge joining u and v; and $d(G,H)$ is the distance between graphs G and H [19]. The **MCS Edge** distance [19, 6] is

$$d(G,H) = 1 - \frac{|\mathrm{mcs}(E_G, E_H)|}{\max\{|E_G|, |E_H|\}}\,, \tag{3}$$

where $|\mathrm{mcs}(E_G, E_H)|$ is the number of edges in the maximum common subgraph of G and H and $\max\{|E_G|, |E_H|\}$ is the maximum of the number of edges in either G or H [19]. This metric will always be in the interval $[0,1]$ with proximity to 0 indicating maximal similarity. The **MCS Vertex** distance [19, 6] is also in the interval $[0,1]$ with proximity to 0 indicating greater similarity and is

$$d(G,H) = 1 - \frac{|\mathrm{mcs}(V_G, V_H)|}{\max\{|V_G|, |V_H|\}}\,, \tag{4}$$

where $|\mathrm{mcs}(V_G, V_H)|$ is the number of vertices in the maximum common subgraph of G and H and $\max(|V_G|, |V_H|)$ is the maximum of the number of vertices in either G or H [19]. The **Graph Edit** distance [17, 2, 19, 5, 6] between graphs G and H is calculated by evaluating the sequence of edit operations required to make graph G isomorphic to graph H using the formula

$$d(G,H) = |V_G| + |V_H| - 2|V_G \cap V_H| + |E_G| + |E_H| - 2|E_G \cap E_H|\,, \tag{5}$$

where E_G and V_G are the edges and vertices of graph G and E_H, and V_H are the edges and vertices of graph H [17]. The computational complexity of this measure is reduced by assuming unique labeling of the nodes in the graph [5].

The median graph $\bar{G}$ of a sequence of n graphs $S = (G_1, \ldots, G_n)$ minimises the sum of distances between itself and the members of S for a particular distance metric [5]. The median graph depends on the distance metric, $d(G_i, G_j)$, chosen but the general formula for the median graph is

$$\bar{G} = \arg\min_{G \in S} \sum_{i=1}^{n} d(G, G_i)\,. \tag{6}$$

The graph edit distance metric, Equation (5), is used both to construct $\bar{G}$ and to calculate the distance from $\bar{G}$ to other graphs [5]. The median graph $\tilde{G}_n$ is calculated from a sequence of graphs $(G_{n-L+1}, \ldots, G_n)$ in window of length L. This window length is arbitrarily chosen to be five in accordance with Dickinson et al. [5]. The **median graph edit** distance to the next graph, $d(\tilde{G}_n, G_{n+1})$, is then calculated using graph edit distance and is referred to herein as *median graph distance*. The distance between $\tilde{G}_n$ and G_{n+1} is classified as abnormal if

$$d(\tilde{G}_n, G_{n+1}) \geq \alpha\phi\,, \tag{7}$$

where $\alpha = 2.5$ is a parameter set following Dickinson et al. [5], and the average deviation of the graphs in the window, $(G_{n-L+1}, \ldots, G_n)$, from the median graph, $\tilde{G}_n$, is

$$\phi = \frac{1}{L} \sum_{i=n-L+1}^{n} d(\tilde{G}_n, G_i)\,. \tag{8}$$

The **Modality** distance [13] between graphs G and H is the absolute value of the difference between the Perron vectors of these graphs:

$$d(G, H) = \|\pi(G) - \pi(H)\|\,, \tag{9}$$

where $\pi(G)$ and $\pi(H)$ are the Perron vectors of graphs G and H respectively. The Perron vector $\pi_{m\times 1}$ satisfies

$$A\pi = \rho\pi\,, \quad \pi > 0\,, \quad \sum_{i=1}^{m} \pi_i = 0\,, \tag{10}$$

where $A_{m\times m}$ is the non-negative irreducible adjacency matrix $A_{m\times m}$ with spectral radius ρ. The **Diameter** distance [8] between graphs G and H is the difference in the average longest shortest paths for each graph and is

$$d(G,H) = \left|\sum_{v\in V_H} \operatorname{maxd}(H,v) - \sum_{v\in V_G} \operatorname{maxd}(G,v)\right|\,, \tag{11}$$

where $\operatorname{maxd}(G,v)$ is the distance to the vertex in G farthest away from v, via the shortest path. The **Entropy** distance between graphs G and H is the following difference between entropy-like values:

$$d(G,H) = E(H) - E(G) = -\sum_{e\in E_H} \left(\tilde{w}_e^H - \ln \tilde{w}_e^H\right) + \sum_{e\in E_G} \left(\tilde{w}_e^G - \ln \tilde{w}_e^G\right), \tag{12}$$

where $\tilde{w}_e^* = w_e^* / \sum_{e\in E_*} w_e^*$ is the normalized weight for edge e. The **Spectral** distance [10] between graphs G and H is calculated by using the k largest positive eigenvalues of the Laplacian:

$$d(G,H) = \begin{cases} \sqrt{\left[\sum_{i=1}^{k} (\lambda_i - \mu_i)^2\right] \left[\sum_{i=1}^{h} \lambda_i^2\right]^{-1}}\,, & \sum_{i=1}^{h} \lambda_i^2 \le \sum_{j=1}^{h} \mu_j^2\,, \\ \sqrt{\left[\sum_{i=1}^{k} (\lambda_i - \mu_i)^2\right] \left[\sum_{j=1}^{h} \mu_j^2\right]^{-1}}\,, & \sum_{i=1}^{h} \lambda_i^2 > \sum_{j=1}^{h} \mu_j^2\,, \end{cases} \tag{13}$$

where λ_i represents the eigenvalues of the Laplace matrix for graph G and μ_i represents the eigenvalues of the Laplace matrix for graph H.

Distance measurements based on graph edit distance [2] and Maximum Common Subgraph (MCS) based distances [9, 14] are thought to be more error tolerant [2].

The ten graph distance metrics discussed above each produce a time series of graph distances between successive time slices of the network. In the example used to demonstrate the methods proposed, these time slices are working days but other, possibly non-constant, divisions are possible. What remains to be done is the detection of change points in these time series.

3 Change point detection methods

Cumulative Summation and Minimum Mean Squared Error assume that the time series adheres to a mean-shift model, that is, a time series with an independent error structure. Mathematically, this means that the time series $S = X_1, X_2, \ldots, X_M$ is made up of the individual observations $X_i = \mu_i + \epsilon_i$ where μ_i are the mean values of the process and ϵ_i are the independently and identically distributed (iid) random variations on that mean. The ongoing disturbances, ϵ_i, are not correlated to one and other. This is not as restrictive as the assumption of no serial correlation that must be met to use some other change point detection techniques. A mean-shift causes serial correlation but does not violate the assumption of independent errors. It is possible to test to see if the independent errors assumption has been violated by directly measuring the autocorrelations or by examining the variance ratios. For the most part $\mu_i = \mu_{i+1}$. Occasionally, the mean shifts and $\mu_i \neq \mu_{i+1}$. In this case the ith value in the time series is called a change point as the mean of the underlying process has changed.

The original article on cumulative summation [16] suggested that change points could be detected by moving through the time series and at each point using a sequential probability ratio test. In this article, change point detection is approached as a problem of off-line hypothesis testing where the time series $S = X_1, X_2, \ldots, X_M$ is viewed as a sequence of observed random variables with conditional density $p_\theta(X_k \mid X_{k-1}, \ldots, X_1)$ [1]. The null hypothesis, H_0, is that there is no change and the alternative hypothesis, H_1, is that change occurs at time t_c. Formally,

$$\begin{aligned} H_0 &: \text{for all } 1 \leq k \leq M : \theta = \theta_0 \\ H_1 &: \text{there exists } t_1 \leq t_c \leq t_M \text{ such that} \\ & \text{for all } t_1 \leq t_k \leq t_{c-1}\,, \quad \theta = \theta_0\,, \\ & \text{for all } t_c \leq t_k \leq t_M\,, \quad \theta = \theta_1\,, \end{aligned} \tag{14}$$

where θ_0 and θ_1 are constant conditional density parameters. The requirements under such a situation are to maximise the power (the chance of rejecting H_0 when it should be rejected), or recall, while minimising Type II errors (the chance of rejecting H_0 when it should not be rejected), or false alarms. Although it is not done in this article, this can be framed as an on-line change detection problem where the time series $S = X_1, X_2, \ldots, X_M$ is again viewed as a sequence of observed random variables with conditional density $p_\theta(X_k \mid X_{k-1}, \ldots, X_1)$ [1]. Prior to the change point at time t_c, $\theta = \theta_0$ and after the change point $\theta = \theta_1$. The change point detection method should aim to detect this change as quickly as possible both to alert users and to minimise the chance of another change presenting itself before the first change is detected. Cumulative summation and MMSE are both able to operate as stopping rules in an on-line framework.

3.1 Minimum mean squared error

MMSE [1] sequentially splits the time series $S = X_1, X_2, \ldots, X_M$ into two segments $M - 1$ times, so S is split into $S_1 = X_1$ and $S_2 = X_2, X_3, \ldots, X_M$, then into $S_1 = X_1, X_2$ and $S_2 = X_3, X_4, \ldots, X_M$ etcetera until it is split into $S_1 = X_1, X_2, \ldots, X_{M-1}$ and $S_2 = X_M$. For each of these $M - 1$ splits the Mean Squared Error is calculated

$$\text{MSE}(m) = \sum_{i=1}^{m} \left(X_i - \bar{X}_L\right)^2 + \sum_{i=m+1}^{M} \left(X_i - \bar{X}_R\right)^2, \tag{15}$$

where $\bar{X}_L = \big(\sum_{i=1}^{m} X_i\big)/m$ is the mean of the series $(X_1, \ldots, X_m)$ and $\bar{X}_R = \big(\sum_{i=m+1}^{M} X_i\big)/(M - m)$ is the mean of the remainder of the series. This is a measure of how well the data fits the means of the two segments. The value of m yielding the Minimum MSE is the best estimator of the last point before the change, so the point $m + 1$ estimates the first point after the change. To determine confidence in whether the MMSE value calculated is indicative of a candidate change point we use bootstrapping to obtain an estimate of the error by randomly re-sampling the original dataset. This creates pseudo-replicate datasets to which the MMSE procedure is applied to see how likely it is that the candidate change point was detected by chance. For the number of bootstrap samples specified a bootstrap sample is generated by randomly reordering the entries in the data set. In each case the MSE for the bootstrap sample is found when that sample is split at the point producing the minimum MSE for the actual data. Every time the MSE for the bootstrap sample is larger than the actual MSE at the split point our confidence that the split point is a candidate change point increases. The confidence level is calculated by dividing the number of times the MSE for the bootstrap sample is larger than the MSE for the original data set by the number of bootstrap samples.

If the confidence level is sufficient to be satisfied that there is a candidate change point, then it is stored. The time series is broken at this candidate change point and the analysis repeated for each segment, thus recursively yielding lower level candidate change points. The level of the procedure is incremented before each recursive call for the segments. This procedure stops when the confidence in the candidate change point being significantly different to random variation drops below a critical, user set, threshold (often $\alpha = 0.01$ or $\alpha = 0.05$).

After all possible candidate change points are found in this manner, their confidence levels are re-estimated. The confidence in some points may fall and these are eliminated. The remaining points are considered to be change points.

3.2 Cumulative summation

Cumulative summation [16, 1, 22, 20] calculates the mean $\bar{X}$ over the series $S = X_1, X_2, \ldots, X_M$ and forms the series of the cumulative summation of the differences from this mean $C = (s_0, s_1, \ldots, s_M)$ where $s_0 = 0$ and $s_k = s_{k-1} + X_k - \bar{X}$. To determine whether a change point occurs in the series the difference, $\triangle C$, between the maximum, $\max_{i=1,\ldots,M} C$, and minimum, $\min_{i=1,\ldots,M} C$, values of the set C is calculated and compared numerous times to similar values from many bootstrap samples, $\triangle C_b^k$, where $k = 1, \ldots, B$ is one of the B bootstrap samples, using the function $D = \sum_{i=1}^{B} d_i$ where

$$d_i = \begin{cases} 1\,, & \triangle C \geq \triangle C_b^k\,, \\ 0\,, & \triangle C < \triangle C_b^k\,. \end{cases} \qquad (16)$$

The confidence level that a change has occurred is D/B. If a change is detected at a confidence level acceptable to the user, the time slice at which the change is considered to have occurred is $\{k : |s_k| = \max_{i=1,\ldots,M} |C|\}$. That is, the time slice at which the cumulative sum of the differences from the mean is furthest from zero. The time series is then split at this point and the same process is applied recursively until no change point is found.

4 Results

An application of these techniques to the example that motivated this work is described here in detail. The data set consists of 102 consecutive working days of TCP/IP traffic collected from five probes on a large enterprise network. Although the IP addresses were recorded in the original data set, the set used in this example only looks at communications between different subnets to keep the memory requirements down. Using this data, aberrant patterns of change were identified by network administrators on days 22, 64 and 89. Only on day 64 was a reason suggested, the introduction of a Web Based Personnel

Table 1: Processing times, in seconds, for graph distance metrics.

Metric	Time
Edit	0.16
MCS weight	0.18
MCS vertex	0.17
MCS edge	0.17
Spectral	4.32
Diameter	2.96
Entropy	0.47
Weight	0.42
Modality	4.94
Median Edit	1.17

Management System. Prior to identifying these days as dates of change in the rate of change the network administrators saw the median edit distance generated plot of day to day distances shown in Figure 1. The days they chose as change points also happen to be the days with outstanding maxima on this plot. As these maxima are not necessarily related to a change in the rate of change there is some question as to the accuracy of the ground truth used in this assessment.

Questions about the veracity of the identified change points do not affect the saliency of the time taken to process all 102 single-day graph into a time series of graph distances, as given in Table 1 for all ten graph distance metrics. This performance was achieved with Java code running on an unloaded 3.00 GHz Pentium 4 PC with 1 GB of RAM using the Windows XP operating system. Nor does it alter the fact that just under 80 MB of data was reduced to somewhere in the order of a kilobyte of data (although the exact figure varied slightly depending on the distance metric used).

Change point detection using seven of the ten graph distance metrics coupled with both cumulative summation and MMSE is demonstrated on a data set consisting of 102 consecutive working days of TCP/IP traffic collected

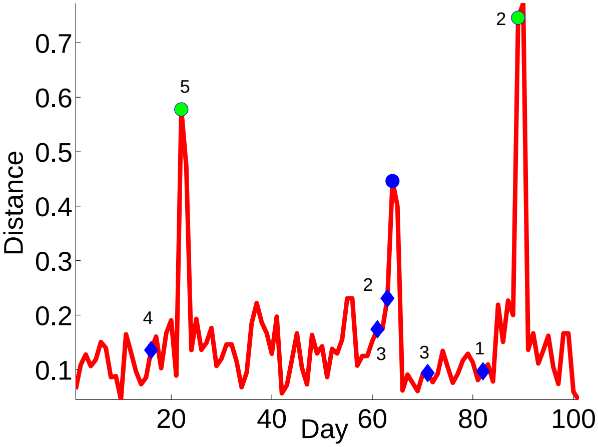

Figure 2: MCS vertex distance time series with change points detected using MMSE with $\alpha = \{0.03, 0.04, 0.05\}$.

from five probes on a large enterprise network. Aberrant patterns of change were identified by network administrators on days 22, 64 and 89. Only on day 64 was a reason suggested, the introduction of a Web Based Personnel Management System. Tests of serial correlation showed that the errors in the time series produced by MCS Weight were too correlated to enable the application of either cumulative summation or MMSE. Furthermore, the two-sided null hypothesis of no first-order autocorrelation in the residuals was not rejected at the 5% significance level by the Durbin–Watson test [7, 11] for the time series based on the MCS edge and weight distance metrics. It is therefore highly likely that the MMSE and cumulative summation approaches are not applicable to the time series produced by these distance metrics in this example. They are also excluded from consideration in this case.

In the figures in the results section the x-axis represents the number of the day, and the y-axis represents the distance score. The days on which the human experts consulted thought the changes occurred on are shown as circles. If the change point detection algorithm detects these days as change

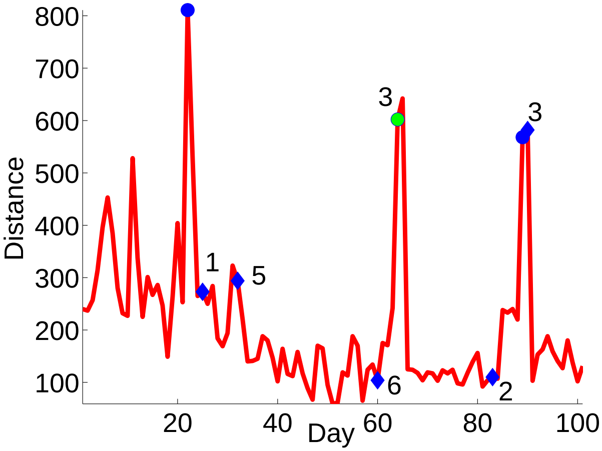

Figure 3: Edit distance time series with change points detected using MMSE with $\alpha = 0.05$.

points they are shaded with the level of the detection appearing next to them, otherwise they are hollow. Days on which the change point detection algorithm produced false alarms are shown as unfilled diamonds.

For the seven time series to which MMSE was applied, the spectral, entropy and modality distance metrics do not result in any detections of change points, nor produce any false alarms for $\alpha = 0.01$, $\alpha = 0.02$, $\alpha = 0.03$, $\alpha = 0.04$ and $\alpha = 0.05$.

As seen in Figure 1 the median graph time series produces three false alarms at all values of α tried. The first of these false alarms is a technical artifact of the five point windowing used to construct the median edit distance time series as the first five points are arbitrarily set to zero.

For $\alpha = 0.01$ and $\alpha = 0.02$, MMSE applied to the MCS vertex time series detects no change points and yields no false alarms. When the 3%, 4% and 5% confidence intervals are considered, the situation shown in Figure 2 occurs with two accurate detections and five false alarms.

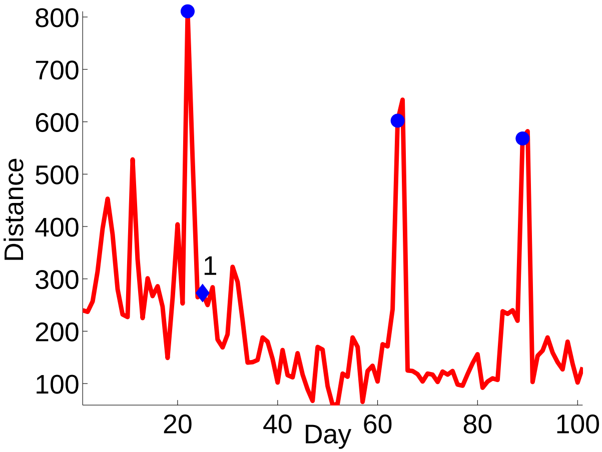

Figure 4: Edit distance time series with change points detected using MMSE with $\alpha = \{0.01, 0.02, 0.03, 0.04\}$.

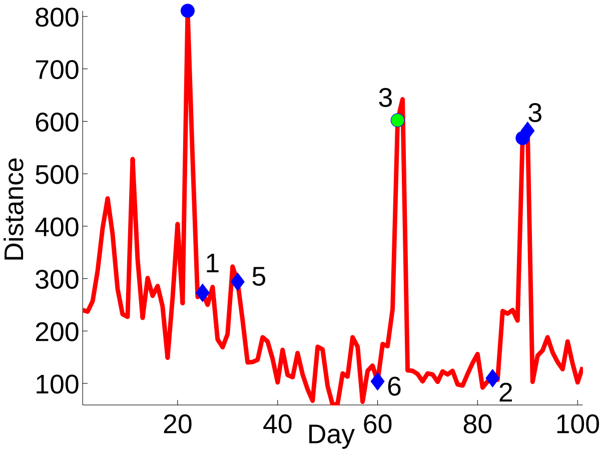

Figure 5: Diameter distance time series with change points detected using MMSE with $\alpha = \{0.04, 0.05\}$.

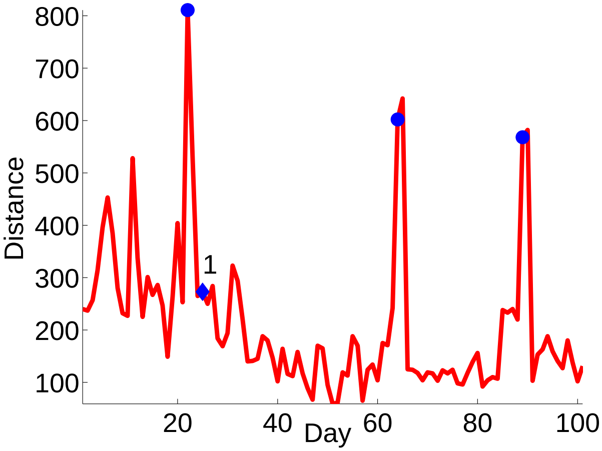

Figure 6: Diameter distance time series with change points detected using MMSE with $\alpha = \{0.01, 0.02, 0.03\}$.

When MMSE is applied to the edit distance time series with 1% to 4% confidence levels (see Figure 4), a single false alarm is produced. If a 5% confidence level is used (see Figure 3), five false alarms and an accurate detection occur. Note that one of the five false alarms is the day before an actual change and another is out by two days. As all changes discussed here are changes perceived by experts in the rate of change of the network, missing these by one or two days could still produce a useful tip-off to those monitoring the network.

Application of MMSE to the time series resulting from application of the diameter distance metric to the graphs for the 102 days gives the results displayed in Figure 5 and Figure 6. Figure 6 shows that using 1%, 2% and 3% confidence levels for MMSE, a single change point is detected without any false alarms. When confidence levels of 4% or 5%, (see Figure 5), are used in MMSE two change points are detected and two false alarms occur.

Using cumulative summation on the time series generated by the MCS

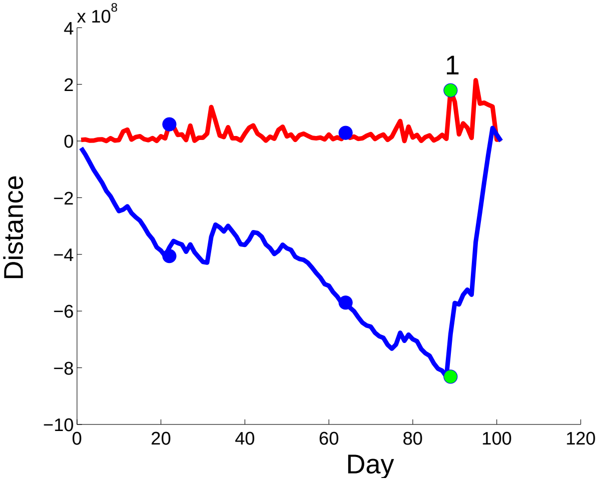

Figure 7: Diameter distance time series with change points detected using cumulative summation with $\alpha = \{0.01, 0.02, 0.03, 0.04, 0.05\}$.

vertex, spectral, entropy and modality distance metrics produces no detections of change points and no false alarms. When cumulative summation is used on the time series generated by the median edit distance, one false alarm and no accurate detections are produced for integer confidence levels from 1% to 4%, and a further two false alarms appear when the confidence level is loosened to 5%. Similarly, the edit distance time series produces a single false alarm and no accurate detections over the range of integer confidence levels from 1% to 5%. Throughout this range of confidence levels the use of cumulative summation on the time series produced by the diameter distance metric resulted in one accurate detection and no false alarms as seen in Figure 7. In this figure the upper line is the time series, the lower line is the cumulative sum, the three dots appearing on both lines show the ground-truth values of the change points and the filled dot on each line is the detected change point.

5 Discussion and conclusion

This article explores detecting a change in the pattern of change in a computer network by representing the computer network as a time series of graphs, using graph distance metrics to turn this into a time series of distances and then using either MMSE or cumulative summation to find change points in these time series. The summarisation of information involved in transforming a time series of graphs to a time series of graph distances greatly reduces the memory requirements. The example used to demonstrate the applicability of this method of detecting changes in the pattern of change is a difficult real world problem. Of most interest are the tractable processing times reported for turning the time series of network graphs into time series of graph distances. Assessment of the accuracy and false alarm rate of the techniques is limited by uncertainty as to whether the human identified change points are the only change points or are indeed change points. All the same, use of a diameter distance generated time series with cumulative summation detected one of the three points identified by systems administrators without producing any false alarms. This situation was the same for low confidence levels when MMSE was used but for higher confidence levels another change point was detected at the cost of two false alarms. For a more accurate assessment of the accuracy and false alarm rate it will be necessary to study simulated data sets where the positions of and types of change in the rate of change are known. For example, I expect that the MCS vertex metric will produce good results when a change in the rate at which vertices start or stop communicating occurs, and the MCS weight distance metric will produce good results when the rate of traffic variation or level of traffic changes its rate of increase or decrease.

The presence of multiple, but strongly correlated, time series of distances based on different distance metrics makes multivariate approaches possible; techniques such using Hotelling's T^2 or principal components analysis could be tried on this problem. As cumulative summation and MMSE are both able to operate as stopping rules in an on-line framework, future work could include applying such a framework to simulated data sets in order to get an estimate of the delay for detection and the mean time between false alarms is for each technique when coupled with each distance metric.

Acknowledgements: Access to the dataset and the Redback network analysis toolbox was provided by Peter Dickinson and Miro Kraetzl. Miro Kraetzl also provided access to pre-prints of his articles with Wal Wallis and Matt Gaston on the modality [13] and diameter [8] distance metrics. A preprint was also provided by Wayne Taylor [20].

References

[1] M. Basseville and V. I. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993. C459, C460, C462

[2] Bunke, H. and Shearer, K., A graph distance metric based on the maximal common subgraph, *Pattern Recognition Letters*, **19**(3/4), 1998, 255–259. doi:10.1016/S0167-8655(97)00179-7. C452, C456, C458

[3] Christmas, W. J., Kittler, J., and Petrou, M., Structural matching in computer vision using probabilistic relaxation, *IEEE Trans. Pattern Anal. Machine Intell.*, **17**(8), 1995, 749–764. doi:10.1109/34.400565 C451

[4] Dick, S., Meeks, A., Last, M., Bunke, H. and Kandel, A., Data mining in software metrics databases, *Fuzzy Sets and Systems*, **145**(1), 2004, 81–110. doi:10.1016/j.fss.2003.10.006 C452

[5] Dickinson, P. J., Kraetzl, M., Bunke, H., Neuhaus, M. and Dadej, A., Similarity Measures for Hierarchical Representations of Graphs with Unique Node Labels, *International Journal of Pattern Recognition and Artificial Intelligence*, **18**(3), 2004, 425–442. C452, C456, C457

[6] Dickinson, P. J., Bunke, H., Dadej, A. and Kraetzl, M., Matching Graphs with Unique Node Labels, *Pattern Analysis and Applications*, **7**(3), 2004, 243–254. C452, C456

[7] Durbin, J. and Watson, G. S., Testing for Serial Correlation in Least Squares Regression II, *Biometrika*, **38**, 1951, 159–178. C464

[8] Gaston, M. E., Kraetzl, M. and Wallis, W. D., Graph Diameter as a Pseudo-Metric for Change Detection in Dynamic Networks, *Australasian Journal of Combinatorics*, **35**, 299–312. C452, C458, C470

[9] Horaud, R., and Skordas, T., Stereo correspondence through feature grouping and maximal cliques, *IEEE Trans. Pattern Anal. Machine Intell.*, **11**(11), 1989, 1168–1180. doi:10.1109/34.42855 C458

[10] Jakobson, D. and Rivin, I., Extremal metrics on graphs I, *Forum Math*, **14**(1), 2002, 147–163. C458

[11] L. Kanzler. *A Study of the Efficiency of the Foreign Exchange Market through Analysis of Ultra-High Frequency Data.* D.Phil. Thesis, Sub-Faculty of Economics, University of Oxford, 1998. C464

[12] Kaufmann, G. and Bunke, H., Automated reading of cheque amounts, *Pattern Analysis and Applications*, **3**, 2000, 132–141. C451

[13] Kraetzl, M. and Wallis, W. D., Modality Distance between Graphs *Utilitas Mathematica*, **69**, 2006, 97–102. C452, C457, C470

[14] Levinson, R., Pattern associativity and the retrieval of semantic networks, *Comput. Math. Appl.*, **23**, 1992, 573–600. doi:10.1016/0898-1221(92)90125-2 C458

[15] M. Neuhaus, and H. Bunke. A Graph Matching Based Approach to Fingerprint Classification Using Directional Variance. *Lecture Notes in Computer Science*, 3546:191-200, 2005. C451

[16] Page, E. S., Continuous inspection schemes, *Biometrika*, **41**, 1954, 100–115. C459, C462

[17] Sanfeliu, A. and Fu., K., A distance measure between attributed relational graphs for pattern recognition, *IEEE Trans. SMC*, **13**(3), 1983, 353–363. C456, C457

[18] K. Shearer, H. Bunke, S. Venkatesh, and D. Kieronska. Efficient graph matching for video indexing. *Preproceeding GbR97: IAPR Workshop on Graph based Representations.* C451

[19] Shoubridge, P. J., Kraetzl, M., Wallis, W. D. and Bunke, H., Detection of Abnormal Change in a Time Series of Graphs, *Journal of Interconnection Networks*, **3**(1–2), 2002, 85–101. C452, C455, C456

[20] Taylor, W., Change-Point Analysis: A Powerful New Tool For Detecting Changes, *Quality Engineering*, submitted. C462, C470

[21] Wong, E. K., Model matching in robot vision by subgraph isomorphism, *Pattern Recognition*, **25**(3), 1992, 287–304. doi:10.1016/0031-3203(92)90111-U C451

[22] Woodall W. H. and Adams, B. M., The Statistical Design of CUSUM Charts', *Quality Engineering*, **5**(4), 1993, 559–570. C462

Author address

1. **B. Pincombe**, Land Operations Division, Defence Science and Technology Organisation, PO Box 1500, Edinburgh, South Australia 5111, Australia.
mailto:Brandon.Pincombe@dsto.defence.gov.au