



Principle Components and Importance Ranking of Distributed Anomalies

KYRRE BEGNUM

MARK BURGESS

Faculty of Engineering, Oslo University College, Norway

mark.burgess@iu.hio.no

Editor: Eamonn Keogh

Abstract. Correlations between locally averaged host observations, at different times and places, hint at information about the associations between the hosts in a network. These smoothed, pseudo-continuous time-series imply relationships with entities in the wider environment. For anomaly detection, mining this information might provide a valuable source of observational experience for determining comparative anomalies or rejecting false anomalies. The difficulties with distributed analysis lie in collating the distributed data and in comparing observables on different hosts, in different frames of reference. In the present work, we examine two methods (Principle Component Analysis and Eigenvector Centrality) that shed light on the usefulness of comparing data destined for different locations in a network.

Keywords: machine learning, anomaly detection

1. Introduction

Data mining for the detection of anomalies in computer systems is a computationally intensive procedure with uncertain rewards. The aim is to determine when a computer experiences a state that is sufficiently unusual as to be noteworthy. What one means by noteworthy is usually left to ad hoc assumption, whereas a strict policy would reduce the uncertainty of the results; this makes the anomaly detection poorly defined in many cases. One computer's anomaly is another another computer's trivia.

In computer science anomaly detection is often associated with network intrusion detection, since network security is an apparent application of the technique. Experimental systems therefore observe and analyse network traffic at a fixed location such as entry point, or gateway, to a Local Area Network (LAN) (Ranum et al., 1997; Snort, <http://www.snort.org>; Han et al., 2002). This choice is based on several assumptions: that anomalies represent threats to the system and that threats to the network come from outside the network rather than within. Both of these assumptions are somewhat flawed (even from a security viewpoint) and alternative schemes for anomaly detection have been suggested that place the hosts rather than the gateways centre stage (Somayaji & Forrest, 2000; Burgess, 2002; Burgess, resubmitted).

Anomaly detection is not just about intrusion detection however. Focusing on computer security misses obvious opportunities for anomaly detection like resource regulation and

diagnostics. The cfengine project has had some success with using statistical anomaly detection for these ends (Burgess, 1993, 1995). In the cfengine view, anomalies are not assumed to be anything other than unusual events that one might wish to respond to in some fashion (Burgess et al., 2001). Moreover, hosts are assumed to have different interpretations and attitudes to such anomalies. Hosts, or end nodes, are in possession of local knowledge about their own private experience and they might have different policies and agendas in relation to anomalous events.

At the hosts themselves, data that go beyond just network arrivals can be sampled directly. The working state of the system (processes, memory usage etc.) can be observed in relation to other environmental stimuli such as console users. Such data are not readily available to a centralized detector, so the distributed host view is richer.

A question that craves an answer, then, is whether centralizing the analysis of network arrivals or distributing the analysis amongst hosts is the best strategy? Amongst the centralizers, there is a tacit belief that centralization allows one to perform collation and comparisons between the traffic to hosts and that these comparisons hold meaningful information that can only be found at a central point. There are clearly disadvantages to centralization: data must be serialized through a bottleneck that throttles the performance of the network and placed great computational demands on a single node (Han et al., 2002). As communication rates increase, this burden of analysis is unlikely to scale. Moreover it has not (to the authors' knowledge) been studied whether there is any information gain in this approach.

One is left with a simple question: suppose one samples the system data at the end-hosts in the network, is any significant statistical information lost about the network as a whole?

In this paper, we attempt to provide an answer to this question for the anomaly analysis method used by the network management agent cfengine (Burgess, 1993, 1995). This method is interesting because it has many desirable properties that relate to efficiency, not only due to its distributed strategy but also in connection with its method of unsupervised learning with lazy evaluation and controlled forgetfulness (unsupervised unlearning).

We begin with a discussion of the method for collecting and characterizing similarities between hosts. The subsequent sections present two methods of analysis for finding trends that can distinguish normal from abnormal. Finally some results are summarized and conclusions drawn.

2. Distributed anomaly detection

The aim of anomaly detection is to characterize what is normal and then classify events that do not fit the normal pattern. As far as distributed anomalies are concerned, there are many reasons why there might be dynamical logical connections between hosts, and therefore changing similarities and differences between hosts at the behavioural level. Network services bind hosts together into groups that makes them collaborate with one another. Similarities in hardware or function can also lead to similarities. Many host behaviours are driven by user interactions that tend to show a strong pattern of intensity that follows the working week (Burgess et al., 2001) and this also creates an implicit channel for likeness.

Thus, what is normal on one machine might be related to what is normal on another, if one understands the relationships between them.

The pressing problem with anomaly detection, either local or distributed, however is that it cannot be calibrated against an absolute scale. Anomalies are by definition *relativistic* events. They must be measured relative to some baseline state. In our case, the baseline state is learned by an unsupervised algorithm.

The lack of an absolute scale makes comparisons between current and normal state difficult, since there is only partial overlap of circumstances at distributed times and locations. It is already known that an adaptive scale is required to baseline the variations encountered over the working week (Burgess, 2002; Burgess, resubmitted).

For any stochastic arrival process, one must calibrate data in relation to learned expectations and the moments of their frequency distributions (Grimmett & Stirzaker, 2001). This comparison conundrum is not made any easier by measuring data at independent locations, where the relative importances of the data are quite different. However, there are technical reasons for wanting to distribute the computation of anomalies. A data mining operation has access to more specific information about the policy at home base, and can take advantage of the large amount of idle CPU time on most distributed hosts (Burgess, resubmitted).

How then should the analysis be distributed? Several models for distributed data processing exist. One is to first serialize the data for collection and then farm out the mass of data via a private shadow network to a distributed processing device like a cluster (Han et al., 2002). For this approach to work in real-time, the shadow network and processor would have to be always faster than the central feed. This does not seem like a likely scenario that would scale to future traffic levels. Various methods of distributed fault localization that uses the network in an intelligent way has been discussed in refs. (Steinder & Sethi, 2002, 2003). Other approaches to unsupervised learning in discrete network data include refs. (Barbar et al.; Zano & Savaresi; Stolfo et al., 2001).

The method we choose here is to use the natural topology of the regular communications network to perform part of the processing, thereby combining normal message routing functions with the top level analysis of the data (see ref. (Burgess, resubmitted) for details).

The topology of the network is interesting for several reasons. First, the switching and routing infra-structure naturally sorts traffic into logical and organizational categories. By gradually filtering network arrivals and local data samples through a tree-like decision structure, we naturally distribute the computation of the analysis about the network and defer computation until it is required. We call such a configuration a *network prism*, since it projects one or more data streams into their basic constitutive components, like colours in a spectrum (Burgess, resubmitted).

There is then the question of how to collate an overview of the results that compares the behaviours of the end nodes and finds anomalies that require a distributed view. Having gone to the trouble of distributing the computation of anomalies around the network, one does not wish to then transmit the full data-stream back to some central location for total analysis, for instance. Rather, we propose to use the scaled approach alluded to in previous work (Burgess & Canright, 2003; Burgess, resubmitted) and collect the greatly compressed large-scale data from nodes. In this study, we have collected these samples for off-line

analysis rather than automating the procedure, since it is presently unclear whether there is any utility to the procedure at all.

3. Covariance and correlations

To determine whether there is any statistically significant information available in the comparisons between different locations, we formulate the following hypothesis:

Hypothesis 1. The expected behaviour of hosts and their levels of noise can be used to judge the validity of comparing normality at distributed locations.

The truth or falsity of this hypothesis can then be used to infer the following:

Corollary 1 *If normality at different locations can be compared with statistical significance, it is likely that a centralized detector will see more with its overview than a disjointed set of individual hosts, otherwise it will not.*

We shall use two methods to probe for differences in host properties at distributed locations. Different locations may be modelled with discrete graphs (discrete model) and with geometrical spaces (continuous model). In both cases multivariate correlations provide a measure of associative distance.

In the graph theoretical method of host comparison, we use correlation between expected behaviours as a ‘link’ in a graph of individual nodes. Values are placed at each node (as a vector on the graph), based on the accumulated uncertainties between the node and its neighbours. The self-consistent grooming of the distribution of uncertainties will lead to a score for the relative level of association or dissociation between the hosts’ behaviours. The important feature of this method is that it preserves the individual hosts’ notions of normality.

The geometrical method takes a different approach that tries to compromise the individuality of the node behaviours. It replaces the local measure of normality with one that is common to all hosts. The geometrical search can explore a larger parameter space than of the graphical method using the concept of generalized Pythagorean distance, e.g. ‘half way between host A and host B’.

Both of these methods define the similarity in behaviour (covariance) between pairs of variables, to be measures of likeness. What one then needs is a form of consensus vote about what is *normal* amongst the population. Here the methods differ.

- For the graphical method we shall use the idea of network *centrality*. The uncertainty values and behaviours are rooted in their points of origin, and one uses correlations of noise fluctuations to identify the significance of a logical association that calibrates pairs of nodes in the graph (see below).
- For the geometrical method we use Principal Component Analysis (PCA) in which one looks at the rotationally symmetrical distributions of data in scatter space in order to find

axes of maximal symmetry. The geometrical analysis uses the full parameter space and measures hosts relative to a least squares fit to an expectation value that is common to all the hosts. This method attempts to find a best ‘absolute scale’ that calibrates the whole clique of hosts, binding them into a ‘solid’ scatter region of the parameter space.

Both of these methods require definitions of covariance or correlation. Let us define these for the data samples used by our anomaly detector.

A single time series $\vec{v}_i(t)$, measured at a single location i , has the form of a set T measurements of V component vectors of variables $\vec{v}(h, t)$ at sample time t , measured at host (location) h :

$$\vec{v}(h, t) = \left\{ \left(\begin{array}{c} v_1 \\ v_2 \\ \vdots \\ v_V \end{array} \right)_{t_1}, \dots, \left(\begin{array}{c} v_1 \\ v_2 \\ \vdots \\ v_V \end{array} \right)_T \right\} \quad (1)$$

These components are measured values such as the numbers of processes, the number of TCP packets arriving and so on. Any independent measurable can be used here.

One defines the mean of these data in three independent directions to different ends:

- The mean of each variable $v(h, t)$, $t = t_1, \dots, T$ over time represents the time average behaviour of a single independent variable.
- The mean of each component variable at fixed time over all host locations $h = 1 \dots H$.
- The mean of all variable components, at fixed time and location, represents a scalar characterization of the average state of the observed system. Each component variable and sample is independent in the sense of measurement, but the patterns of behaviour can still be non-independent (i.e. correlated) in a statistical sense by having related variations.

The last of these does not make sense, but the others are both important. Let us define these expectations as follows:

$$\langle \vec{v} \rangle_t(h) = \frac{1}{T} \sum_{t=t_1}^T \vec{v}(h, t) \quad (2)$$

$$\langle \vec{v} \rangle_h(t) = \frac{1}{H} \sum_{h=h_1}^H \vec{v}(h, t), \quad (3)$$

The former is a time-independent vector with V components and the latter are time-dependent vector and scalar respectively. The suffix indicates the type of average being taken.

The evaluation of averages over many time samples is resource consuming business that has been discussed previously in ref. (Burgess, resubmitted). There, an algorithm for

compressing and rationalizing the weighting of sample data, in a periodic time framework, was developed. It eliminates the need for massive data storage. We can apply the same algorithm here to replace the average

$$\langle X \rangle_t \rightarrow \langle\langle X \rangle\rangle_P \quad (4)$$

and then take a regular mean over the remaining weekly variation which we call simply $\langle\langle X \rangle\rangle$. We thus apply the method described in ref. (Burgess, resubmitted) to rank the time importance of the arrival process.

Given a definition of the mean of a data set, one can construct the related fluctuations (deviations from the means):

$$\delta_t \vec{v}(h, t) = v(h, t) - \langle \vec{v} \rangle_t(h) \quad (5)$$

$$\delta_h \vec{v}(h, t) = \vec{v}(h, t) - \langle \vec{v} \rangle_h(t). \quad (6)$$

The complexity of the notation here is a hazard of the number of independent degrees of freedom in the problem.

From the deviations one can construct the *scatter matrices* of fluctuation correlations that describe the mutual uncertainties between pairs of variables, with respect to a particular type of deviation. There are several of these, but we shall focus on the one that will give us a correlation between different host locations.

Scatter matrices are essentially covariances or correlations. The covariance of two variables is usually defined by:

$$\text{Cov}(q_1, q_2) = \langle (q_1 - \langle q_1 \rangle_v)(q_2 - \langle q_2 \rangle_v) \rangle \quad (7)$$

and the correlation function is the normalized version:

$$C(q_1, q_2) = \frac{\text{Cov}(q_1, q_2)}{\sigma_v(q_1)\sigma_v(q_2)}. \quad (8)$$

This lies in the range of values $-1 \leq C \leq +1$. The time averaged matrix for fluctuations of the fixed variable type v_i , at different locations h, h' , is an $H \times H$ matrix:

$$C_h(h, h')|_{v=v_i} = \langle\langle \delta q(h) \rangle\rangle | \langle\langle \delta q(h') \rangle\rangle |_{v=v_i} \quad (9)$$

It is the square matrix of hosts correlated with hosts, for a given variable. In a noisy environment, this correlation function has a disadvantage: it makes quantitative judgements about variations where variations are very uncertain. Thus uncertainty is compounded by this measure. An alternative way of correlating is to use a 'fuzzy' correlator that only measures the qualitative similarity in variations. The rank correlation is defined by:

$$R(q_1, q_2) = \sum_{i=1}^H \frac{(r(q_1) - \frac{H+1}{2})(r(q_2) - \frac{H+1}{2})}{\frac{H(H^2-1)}{12}} \quad (10)$$

where $r(q)$ is the ordinal rank of the data. This replaces the actual value of the data with a natural number that defines a ‘qualitative’ amplitude of the data series, preserving only relative values (ranked order). Instead of comparing actual amplitude fluctuations, it only records whether time-series rise or fall in concert.

We now use these measures of similarity in two eigenvalue methods to calibrate distributed normality in the network of associations.

4. Principle component analysis

The identification of principal axes and values within a region of a multi-dimensional space is a method that is used in many branches of science and engineering. In statistics, this method is referred to Principal Component Analysis (PCA). The idea is to determine the intrinsic symmetry properties of a region and to find axes that best express them in terms of the original parameterization. In whimsical terms, the determination of the principal component is like finding an axis that skewers the region most like a kebab. The region of space does not have to be convex, though it is often assumed to be.

In PCA, one looks for an ellipsoidal solid of rotation formed by rotating Gaussian profiles about a number of special directions in the parameter space. These directions are the intrinsic vectors of the region and are found by a method of stationary variation. The resulting vectors also have the interpretation of a multi-dimensional least-squares fit. Thus the discovered trend is a fair weighted compromise between all independent samples.

Stationary vectors of constant length L in a vector space may be derived from the Lagrangian

$$\mathcal{L} = \vec{e}^T C \vec{e} - \lambda(\vec{e}^T \vec{e} - L) \quad (11)$$

with Lagrange multiplier λ enforcing the constant length $\vec{e}^T \vec{e} = L$ constraint. Stationary variations satisfy:

$$\frac{\partial \mathcal{L}}{\partial \vec{e}^T} = C \vec{e} - \lambda \vec{e} = 0, \quad (12)$$

giving the eigenvalue equation

$$C \vec{e} = \lambda \vec{e}. \quad (13)$$

See ref. (Duda, Hart, & Stork, 2001) for a more complete discussion of this method. The eigenvalues correspond to the intrinsic dimensions of the ellipsoidal region and thus provide a calibrating scale for the uncertainty.

The axis of largest scatter corresponds to the maximum eigenvalue. This direction is that of the most important trend in the data, or conversely the direction that maps out the greatest range of values. Positive correlation means that points determine the trend whereas negative correlation means that they oppose it. A negative correlation is also a pattern however—a

large negative correlation -1 represents a see-saw relationship between two variables (when one goes up, the other goes down).

5. Graphical importance ranking

A graph $\Gamma(v, \eta)$ is a set of nodes v joined by edges or links η , characterized by an adjacency matrix (Balakrishnan, 1997), which is a square matrix of nodes versus nodes. If a matrix row-column entry is zero, there is no edge or link joining the pair or nodes corresponding to the row and column. If there is a unit value 1, then a link joins the nodes. A symmetric matrix represents an undirected graph (a graph with no arrow heads).

The covariance matrix is like a graph in which a positive value represents a connection and a negative number represents a repulsion between the nodes. The normalized covariance matrix yields a value between ± 1 , i.e. the correlation matrix. The covariance, or correlation matrix that we have defined above represents a non-directed graph, between host nodes, in which the vertices or links represent probable correlations. We can simplify the probabilistic viewpoint by using a threshold for deciding when a correlation is strong enough to represent a link. This threshold policy is analogous to a fuzzy ranking of values (see below).

In a non-directed graph, the number of links connecting node i to all other nodes is called the degree k_i of the node. We are interested in this connectivity because it signifies similarity of hosts in our picture (a correlation link makes a node similar to its neighbour). Large scale similarity, in turn, represents normality, by population consensus. All else being equal, abnormality (anomaly) means low connectivity in this graph. Importance ranking is usually used to find the most connected graph nodes (Bonacich, 1987; Canright, Engø-Monsen, & Weltzien, 2003; Page et al., 1998; Kleinberg, 1999), but here we are interested in those that are the least well connected (the complement graph).

To find the complement graph it is easier to discuss the best connected nodes in a graph. These hosts define the calibrated standard of normality.

A simple starting definition of well-connected could be ‘of high degree’: that is, count the neighbours. We want however to embellish this simple definition in a way that looks beyond just nearest neighbours. To do this, we borrow an old idea from both common folklore and social network theory (Bonacich, 1987): an important person is not just well endowed with connections, but is well endowed with connections to important persons.

The motivation for this definition is clear from the example in figure 2. It is clear from this figure that a definition of ‘well-connected’ that is relevant to the diffusion of information (harmful or otherwise) must look beyond first neighbours. In fact, we believe that the circular definition given above (important nodes have many important neighbours) is the best starting point for research on influence in networks.

Now we make this circular definition precise. Let v_i denote a vector for the importance ranking, or connectedness, of each node i . Then, the importance of node i is proportional to the sum of the importances of all of i ’s nearest neighbours:

$$v_i \propto \sum_{j=\text{neighbours of } i} v_j. \quad (14)$$

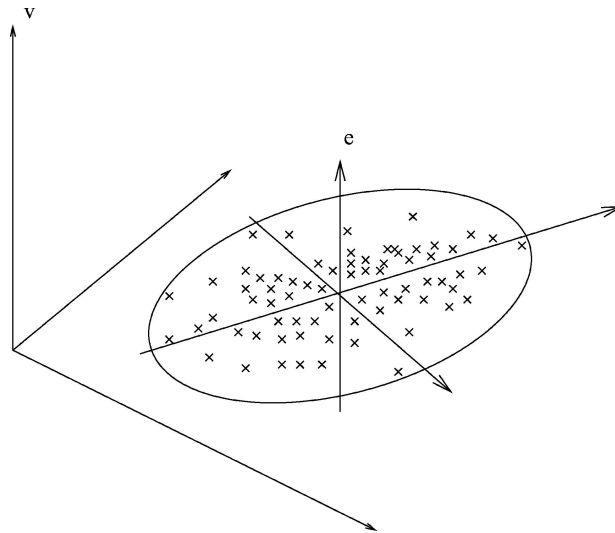


Figure 1. The principal components of a solid characterize the natural measurements of the solid, given that its natural directions \vec{e} might be rotated in relation to the actual parameter axes \vec{v} .

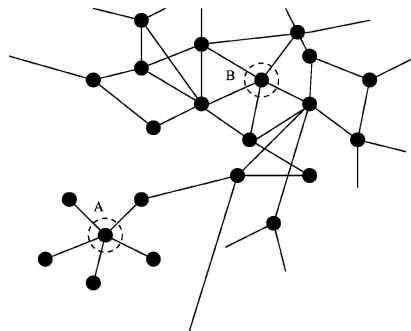


Figure 2. Links represent ‘likeness’ of neighbours. Nodes A and B are both connected by five links to the rest of the graph, and therefore seem equally normal if one merely counts local links. But node B is clearly more ‘normal’ because its neighbours are also well connected—i.e. very alike.

This may be written as

$$v_i \propto \sum_j A_{ij} v_j, \quad (15)$$

where A is the *adjacency matrix*, whose entries A_{ij} are 1 if i is a neighbour of j , and 0 otherwise. Notice that this self-consistent equation is scale invariant; we can multiply \vec{v} by any constant and the equation remains the same. We can thus rewrite Eq. (15) as

$$A \vec{v} = \lambda \vec{v}, \quad (16)$$

and, if non-negative solutions exist, they solve the self-consistent sum; i.e. the importance vector is hence an eigenvector of the adjacency matrix A . If A is an $N \times N$ matrix, it has N eigenvectors (one for each node in the network), and correspondingly many eigenvalues. The eigenvector of interest is the principal eigenvector, i.e. that with highest eigenvalue, since this is the only one that results from summing all of the possible pathways with a positive sign. The components of the principal eigenvector rank how ‘central’ a node is in the graph. Note that only ratios v_i/v_j of the components are meaningfully determined. This is because the lengths $v^i v_i$ of the eigenvectors are not determined by the eigenvector equation.

This form of well-connectedness is termed ‘eigenvector centrality’ (Bonacich, 1987) in the field of social network analysis, where several other definitions of centrality exist. The method is closely related to methods of importance ranking used in Internet search engines.

6. Results

We have tested the two foregoing distributed anomaly methods in a cluster of 37 hosts that are well known to us in order to see if we can learn their behavioural relationships from the machine-learned data. Once again, we emphasize that the data are collected using the lazy approach of ref. (Burgess, resubmitted) and that we treat each variable type as an independent analysis. Correlations between different types of variable have not been studied.

For each variable independently in the set, we obtain the ranking values from the principal eigenvectors, scaled so that the maximum value is 1. These have then been inspected manually at length. The full data are too numerous to quote here, nor are they sufficiently interesting to cite in full. An excerpt of the data is shown in Table 1.

In general, we define the Principle Component Analysis (PCA) score to be the component of host h of the principle eigenvector of the PCA matrix. This ranking tends to identify with high score the hosts that define the trends for the others to follow (the direction of closest

Table 1. Example values for the principal eigenvector rankings of two measured variables (incoming NFS and incoming E-mail).

Host	NFS client			SMTP server		
	PCA	Centrality	Ranked	PCA	Centrality	Ranked
1	0.05	0.00	0.19	0.52	0.00	0.00
2	0.42	0.00	0.00	0.10	0.00	0.00
3	0.08	0.00	0.31	0.63	0.00	0.88
4	1.00	1.00	0.99	1.00	1.00	0.00
5	-0.05	0.00	0.00	0.90	1.00	0.00
6	0.03	0.00	0.99	0.33	0.00	0.00
7	-0.04	0.00	0.99	0.49	0.00	0.48
8	-0.13	0.00	0.99	0.41	0.00	1.00

perpendicular fit). The lowest scores can be negative and represent the most anomalous behaviours (the behaviours about which the average scatter is greatest) or even contradicts the observed trends.

The problem with PCA is that it is amplitude sensitive, and thus detects noisy participants. PCA components are large if the average variation in activity is large ($\sigma^2(q)$ is large), or if the fluctuations about the expected value are large. Thus a big PCA can mean an abnormal value, but it need not—it can simply mean a very uncertain value. One might try to compensate for this uncertainty by dividing the component or the principal eigenvector by its corresponding eigenvalue, which is the natural measuring scale for the parameter. However, when this is done, the results tend to become only more noisy.

The centrality score is the component of the principal eigenvector of the threshold graph. To find this, we choose an arbitrary threshold value between 0 and 1 for the correlations. Any value above the threshold is normalized to 1 and any number below (including a negative number) is set to 0. The resulting score tells us which host behaves most like all of its neighbours. Centrality means “there are many others like me”, so it represents normality. A low value represents an abnormal host amongst the population. (The case of zero adjacency matrix is singular and requires special attention.)

The interpretation of the rankings presents us with a challenge. Since statistical data blur cause-effect relationships, the reasons for high or low scores using the two methods are not always clear. Most of the rankings represent noise. Machines that have identical working situations generate significantly different values that display order of magnitude changes comparable to 1. In other cases, we see special features of the data reflected in the numbers. For the NFS client, hosts 2 and 5 show low ranked centralities, meaning that these are the principal generators of NFS requests. By contrast, the other measures show no sign of this result.

Confusingly, the NFS data show one pattern of interpretation for the ranking mechanisms, while the SMTP data show a different one.

The PCA score for NFS clients gives a high score to an apparently random host in our network. The high score is presumably due to its relatively high level of usage by users. The lowest scores are in fact assigned to the NFS servers in the network (hosts 2 and 5). This might appear significant, however, a comparison with a few other random hosts shows that similar scores are obtained for them. Unranked centrality gives no useful insight into the data: it picks out the same maximum as PCA—presumably for the same reason (amplitude weighting). Ranked centrality gives a minimum to hosts 2 and 5. While all the other ranked centrality values are close to 1, the regular centrality values are close to zero. The low scores for ranked centrality pick out the NFS servers and the network host that performs backups.

In the SMTP data, we see the two highest PCA scores coincide with the two highest scores for centrality (hosts 4 and 5). This unusual contradiction of anomalous behaviour picks out the two E-mail servers on the network. Centrality gives a high score to these machines alone. This result is strange but can be understood if one realizes that only two hosts on the network actually have this variable non-zero. Thus they vary in the same way *and* they are weighted by high activity compared to the others. The zero terms have essentially zero weight (zero eigenvalue) since there is no activity. PCA on the other hand fails to notice that most of the hosts have no activity since it tries to compromise on a value for all hosts.

It therefore gives artificial weight to some hosts. The most unusual hosts are those that have zero ranked centrality. It gives a lowest score to the two mail servers, but assigns the same score to several others. This result is merely confusing.

Several examples of these types of feature exist in the data, swathed in noise. There is no need to recite the full litany. What is disappointing is that the methods reveal quite different features—sometimes one method is better than another, but there is no firm conclusion. The least useful measure is the plain centrality.

There is clearly information in these data, but it is swamped by noise and the correlation measures are sensitive to the noise in qualitatively different ways. Knowing the data in a central location would not help us. The problem is not the location or difficulty of combining the results, but in the variability. It is possible that a greater population size would help to create a more stable set of results. However, as the number of machines grows, the cost-effectiveness of any cross host analysis diminishes at least as fast as the square of the number of hosts. A centralized analysis would scale even worse.

7. Conclusions

The validity of the distributed hypothesis seems far fetched at best. From the data we have collected it would seem that it is quite false and hence by corollary that there is little benefit to centralized anomaly detection, of the type we discuss here. Such evidence is not proof, of course, but there are deeper reasons for supposing that the situation would not improve with different data or different calibration measures: even apparently equivalent hosts have different interactions with the larger environment of users and clients. This makes their environments naturally noisy. Only long term, programmed relationships are likely to show up in correlations, as we have seen here.

One possible application of the noisy correlations would be to extract a zoology of machine types by classifying data in relation to known policy, and to feed such data to brand new hosts. Rather than starting with a blank slate, one could provide the new hosts with a trained immune response, based on experience elsewhere (Somayaji, Hofmeyr, & Forrest, 1997; Burgess, 1998). This has potential advantages, if it can be made to work, since it takes cfengine several weeks to learn a reliable profile and the profile will adapt to any local differences in time.

The principal component results and the general centrality results use actual value amplitudes in their correlations. This makes them very sensitive to noise. Another problem here is that the covariance assumes that the distribution of points about the mean is symmetrical (typically Gaussian). Few of the distributions are in fact symmetrical—the resource constraints of computer systems lead to a natural skew (Burgess et al., 2001). This leads to spurious values that are difficult to interpret. PCA gives no consistent discernable meaning to the data. We speculate that this is because it gives up the individuality of the hosts in seeking a best fit amongst them all, whereas the centrality methods preserve this important perspective.

The ranked correlations are far less sensitive to noise, since the signal ranking assigns a pro forma amplitude to the variation. Ranked correlation is not directly implementable in real time however, though an approximation might be possible. A centralized server would

not be able to perform this analysis and extract data, since it needs a long term perspective that is impractical to store and analyze for the whole network at once.

We conclude then that, while there is a certain amount of information about functional relationships contained in the correlation data, the little that is to be extracted could be found more easily by reference to the network policy in many sites. In a pervasive computing environment, one might not have access to a global policy for the network and the learning of these relationships might be the only way to determine them. However, the counterpoint is that such relationships take several weeks to emerge in statistical data, whereas ad hoc collaborations in pervasive environments might simply lead to random noise.

The problem with the pan-host analysis is not the methods used; these reveal features even though the data are poor. The main problem here is the variability in the levels. Hosts have no clear correlations except in cases where they are already obvious for logistic reasons. The benefit of pan-machine analysis is therefore dubious. We can express this logically as follows: Let the observable ‘effect’ of policy $P(h)$ at host h be $\mathcal{E}(P(h))$, and let the effect of environmental influences $e(h)$ on host h be $\mathcal{E}(e(h))$. Since $P(h)$ represents what we already know, a system is normal and controlled if:

$$\mathcal{E}(P(h)) \gg \mathcal{E}(e(h)). \quad (17)$$

An anomaly analysis is worth while, only if two abstract criteria are met:

$$\begin{aligned} \mathcal{E}(e(h)) &> \mathcal{E}(P(h)) \\ \mathcal{S}(\mathcal{E}(e(h))) &\sim \mathcal{S}(\mathcal{E}(P(h))), \end{aligned} \quad (18)$$

i.e. a causal signal \mathcal{S} in the environment exists and is of the same order of magnitude as the causal signal in the policy, so that the two compete for the system’s attention. This will say something about the entropy of the environment’s projection into the system.

Although we have not explored a precise meaning for these measures, it is clear that none of the data here can be seen as fulfilling requirements of this kind. It will be of interest in the future to consider measures for \mathcal{E} and \mathcal{S} , perhaps in information theoretical terms, related to the policy maintenance theorem (Burgess, 2004). The language is also suggestive of fact that a signal (anomaly) that becomes noticeable in the environment depends upon the way it is projected into the measurement device and therefore measurements centrally and in a distributed setting are unlikely to be equivalent.

There does not seem to be a compelling argument for central analysis of the patterns of behaviour that emerge at the network level. We have not exhausted the possibilities for distributed analysis in the present paper however. We shall return to consider a number of qualitatively different techniques in later work.

References

- Balakrishnan, V. (1997). *Graph theory*. New York: Schaum’s Outline Series, McGraw-Hill.
 Barbará, D., Li, Y., Couto, J., Lin, J.-L., & Jajodia, S. (2003). Bootstrapping a data mining intrusion detection system. In *Proceedings of the 2003 ACM Symposium on Applied Computing*. New York, NY: ACM Press.

- Bonacich, P. (1987). Power and centrality: A family of measures. *American Journal of Sociology*, 92, 1170–1182.
- Burgess, M. (1993). Cfengine WWW site. <http://www.iu.hio.no/cfengine>.
- Burgess, M. (1995). A site configuration engine. Cambridge MA: Computing Systems, MIT Press, vol. 8, p. 309.
- Burgess, M. (1998). Computer immunology. In *Proceedings of the Twelfth Systems Administration Conference (LISA XII)*. USENIX Association: Berkeley, CA, p. 283.
- Burgess, M. (2002). Two dimensional time-series for anomaly detection and regulation in adaptive systems. *IFIP/IEEE 13th International Workshop on Distributed Systems: Operations and Management (DSOM 2002)* (p. 169).
- Burgess, M. (2004). *Analytical network and system administration—Managing human-computer systems*. Chichester: J. Wiley & Sons.
- Burgess, M. (resubmitted). Probabilistic anomaly detection in distributed computer networks. *Science of Computer Programming*.
- Burgess, M. & Canright, G. (2003). Scalability of peer configuration management in partially reliable and ad hoc networks. In *Proceedings of the VIII IFIP/IEEE IM Conference on Network Management* (p. 293).
- Burgess, M., Haugerud, H., Reitan, T., & Straumsnes, S. (2001). Measuring host normality. *ACM Transactions on Computing Systems*, 20, 125–160.
- Canright, G., Engø-Monsen, K., & Weltzien, Å. (2003). Multiplex structure of the communications network in a small working group. *Social Networks—An International Journal of Structural Analysis* (submitted for publication).
- Duda, R., Hart, P., & Stork, D. (2001). *Pattern Classification*. New York: Wiley Interscience.
- Grimmett, G. & Stirzaker, D. (2001). *Probability and random processes*, 3rd edition. Oxford: Oxford Scientific Publications.
- Han, S.-H., Kim, M.-S., Ju, H.-T., & Hong, J.-K. (2002). The architecture of NG-MON: A Passive Network Monitoring System for High-Speed IP Networks. In *IFIP/IEEE 13th International Workshop on Distributed Systems: Operations and Management (DSOM 2002)* (p. 16).
- Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 604.
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The PageRank citation Ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project.
- Ranum, M. J., Landfield, K., Stolarchuk, M., Sienkiewicz, M., Lambeth, A., & Wall, E. (1997). Implementing a generalized tool for network monitoring. In *Proceedings of the Eleventh Systems Administration Conference (LISA XI)* (p. 1). Berkeley, CA: USENIX Association.
- Snort, Intrusion detection system. <http://www.snort.org>.
- Somayaji, A., & Forrest, S. (2000). Automated reponse using system-call delays. In *Proceedings of the 9th USENIX Security Symposium* (p. 185).
- Somayaji, A., Hofmeyr, S., & Forrest, S. (1997). Principles of a computer immune system. *New Security Paradigms Workshop, ACM* (pp. 75–82).
- Steinder, M. & Sethi, A. (2002). Distributed fault localization in hierarchically routed networks. In *IFIP/IEEE 13th International Workshop on Distributed Systems: Operations and Management (DSOM 2002)* (p. 195).
- Steinder, M. & Sethi, A. (2003). A survey of fault localization techniques in computer networks. *Science of Computer Programming* (to appear).
- Stolfo, S. J., Lee, W., Chan, P. K., Fan, W., & Eskin, E. (2001). Data mining-based intrusion detectors: An overview of the columbia IDS project. *ACM SIGMOD*, 30:4.
- Zanero, S., & Savaresi, S. M. Unsupervised learning techniques for an intrusion detection system. In *Proceedings of the 2004 ACM Symposium on Applied Computing*.

Received March 31, 2004

Revised October 6, 2004

Accepted October 6, 2004

Final manuscript October 6, 2004