New Advances for the Analysis & Tracking of Networks
A White Paper

Joseph E. Johnson, PhD
Professor of Physics, University of South Carolina
jjohnson@sc.edu
August 16, 2006

The Discovery & Product:

We have discovered new mathematical methods for analyzing any type of network and specifically identifying anomalies. We have rendered this algorithm into a software product and seek applications. This white paper is an informal overview of our product: ExaSphere – A Network Analysis Engine.

Introduction

Our research team has made a set of very important advances in the description of networks. These advances are both at the fundamental mathematical level as well as a practical level where we have developed new software tools for network analysis and tracking. Most of the other articles and material represented on this site are either highly technical or highly mathematical and are relatively formal. This paper is intentionally very informal and at the lay level mathematically and technically. Yet at the same time this paper addresses what we believe to be the critical philosophical and foundational issues of this new technological domain that is absolutely critical in importance in the new 'information age'. In spite of the informal nature and non-mathematical nature, this presentation will nevertheless center on the issues that will face our society as we move forward in this 21st century. We will see that these issues are of incredible complexity and difficulty and at the same time represent some of the greatest opportunities for technological and commercial advances in addition to truly fundamental advances in pure mathematics and computer science. The primary effort here is to present a complete foundation on the descriptive level from a new perspective

What are networks?

A network is constructed from a set of points, called nodes. The nodes can have any distinctive names but for convenience we use the numbers 1, 2, 3, … N to distinguish the nodes. The network is defined as a set of connections between these nodes where these connections are labeled by numerical values that are not negative. Thus we can represent any network by an array or matrix of numbers like $C_{1,8} = 46$ where 46 is the 'weight' or degree of connection between nodes '1' and node '8'. Thus a network is exactly specified by a square array of non-negative numbers but with the diagonal undefined. Typically one takes the diagonal arbitrarily as a '0' or a '1' or other value but this arbitrariness rests on the fact that the strength of connection of a thing to itself is not defined. Also one might note that the array might be symmetric (ie $C_{1,8} = C_{8,1}$ called an undirected graph) or not symmetric ($C_{1,8} \neq C_{8,1}$ called a directed graph).

An example of a network might be the number of airline flights that are flown between two given airports in one day. Another example is the number of emails that are sent from one computer to another computer in a week. Still another example is the amount of money in units of $1,000 that was transferred in a given month between two given bank accounts. Two final examples might be the power transferred between two electrical substations in units of Mega Watts, or the number of cars that travel between two interstate interchanges in a one hour period.

Thus one quickly sees that networks are totally pervasive in our society and can be represented in some of the following groups:

1. Communications Networks
    a. Internet transfers (email, FTP files, images..)
    b. Telephone calls
    c. Mail, Fed Ex, UPS
2. Transportation Networks
    a. Air traffic between airports
    b. Highway traffic at different levels of highway networks
    c. Waterway traffic
    d. Railroad networks and shipments
    e. Pipelines transfers among junction points
3. Financial Networks
    a. Banking transfers among bank accounts
    b. Accounting flows in a corporate or government agency
    c. Ownership and investment linkages among stocks and entities
    d. Input-Output economic flows of a nations or states economy
4. Utility and Energy
    a. Electrical grids with the transfer of power between substation, plants and consumers.
    b. Natural gas pipelines
    c. Water and sewer services among sources and disposal points
    d. Solid trash and waste flows
5. Social Networks
    a. Criminal and terrorist networks
    b. Organizational charts and relationships
    c. Social organizations
    d. Financial-social alliances, cooperative links
6. Manufacturing Processes
    a. Tracking of just-in-time flows for manufacturing and work flows
    b. Assembly line flows
7. Electrical & Mechanical Networks
    a. All electrical devices with electrical flows
    b. Specifically all computers and collections of computers
    c. Mechanical systems with energy flows among components
8. Biological, Health, and Disease Networks
    a. Disease networks and contagious flows among living entities
    b. Blood, lymph, digestive, nutrient flows in the body
    c. Neural networks

Why are networks different from other problems in the sciences?

If we consider a physical system such as a baseball, we can monitor its state (where it is, how fast it is going, its acceleration, etc) using a vector that points to the ball.

A vector is described by a set of numbers. If it is a baseball moving in the air then we need three numbers (x, y, z) to specify its position (ie the vector). These might give latitude, longitude, and altitude of the ball. Or we might use the position in feet from a given coordinate system. Likewise the velocity of the ball can be given by its velocity in each of the three directions: (V_x V_y, V_z) again a vector in three dimensions. If we are going to be more complete, then we would also monitor the angular position and spin (angular velocity) of the ball. But again these are vectors in three dimensions thus giving us two vectors in three dimensions. Furthermore if we have a thousand such baseballs that we wish to monitor then it just becomes a vector in three thousand dimensions – complicated but understandable in principle and executable on a computer. The same situation even holds in the theories of relatively and quantum mechanics: the state of a system is given by a vector which is a list of numbers. The change of these vectors in time is given by a matrix or two dimensional array of numbers that transform one from the old values to the new values. Thus in summary one can see that all systems in the physical world are basically described by vectors (a one dimensional list of numbers) and the change in time is given by a matrix (two dimensional array of numbers). Furthermore the changes over time are usually understood in principle and obey know laws of classical and quantum physics. But ….

Networks are far more complex objects:

But we have seen from the examples of networks above that a network is specified by a matrix or two dimensional array of numbers. This is a whole order of complexity more difficult that classical, relativistic, and quantum mechanics. This fundamental difference is perhaps better seen when we realize that the vector that describes a particle or even a set of particles, is a single arrow or point in a space whereas a network is represented by a large number of vectors all at the same time. Thus a network of a million computers is represented by a matrix of a million squared or a trillion numbers. Furthermore, these trillion numbers are rapidly changing and do not obey known laws and almost certainly are not linear. Thus we are faced with not just a million coordinates for a single point but rather a million numbers for each of a million points simultaneously – and without known laws of change with time!

Do we not have similar situations in the natural world such as with gases?

In physics and chemistry, we can have a very large number of particles as we do with a gas or a fluid consisting of trillions of trillions of particles. We certainly are not interested in knowing where each particle is and how it is moving. Rather, we seek summary values to know what state or condition the system is in and how the system is changing. These 'metrics', the 'summary values' of the trillions of positions and velocities of the individual particles, are the familiar concepts of temperature, pressure, volume, heat, internal energy, and entropy. With these few variables, we can assess the state of the system, how it is changing, and whether or not it is in equilibrium.

Can we define and develop 'summary metrics' for networks?

This is difficult for the following reasons: First, networks do not have an innate sense of 'distance' from one node to another. On the internet for example, one is as close to a user in China as in the same county here in the US. While it is true that some networks such as airlines and highways have longitude-latitude values for each airport or intersection, these values are actually of little meaning to the network per se which is defined by the existence of the nodes and the internodes' connections, independent of and abstracted from the distance. Without a concept of distance, we cannot form the notions of 'volume' or of 'pressure as force/area'.

Next we realize that on networks there is no conserved energy as there is in a gas that is isolated. Thus we cannot define the random energy of 'heat' or the average random energy which represents 'temperature'. In particular this is emphasized because there is no well defined "equilibrium" but only average behavior. Finally, one can consider entropy but entropy is always defined as a measure of order or disorder and is defined on probability distributions. The network does not have intrinsic probability distributions but only the set of connections given by the connection matrix C_{ij}.

A nice thing about networks.

One advantage to dealing with networks is that one can expand or collapse the network based upon the nodes. Specifically one can treat a single computer as a single node, or treat a whole department of computers as a single node, or even all computers at a single company as a single node. Thus one can effectively collapse a network to ignore the internal traffic system.

A deeper look at the problem.

Thus it is not obvious how to summarize all of the values of the connection matrix down into a few representative variables as one does with thermodynamics. Like the positions and velocities of particles in a gas, the connections in the C matrix are all of equal importance and value. Thus it is not clear how to summarize the data in a useful way. One of the core problems is that the connection matrix is not even unique. If for example another person were to look at the same network, they would probably not number the nodes in the same way. Thus when the second person writes the C matrix, since the rows and columns are in different orders, the matrix will look entirely different. Whereas a given C matrix gives a definite network topology, the converse is not true and the same network can be represented by N! (i.e. N*(N-1)*(N-2)*……1) different matrices corresponding to the different ways one can number the nodes. Here one gets to one of the fundamental problems in network theory that the nodes have no natural order and thus any numbering of them is equivalent. To simply know if two networks are equivalent in their topology (connectedness), one must perform all possible renumbering of nodes and then compare the resulting C matrices. It is easily verified that no existing computer now or in the foreseeable future will be able to even compare medium size networks to see if they are the same. So one would ask what is done now if there are no metrics. First let us review the requirements for an ideal system and then review how things are currently done.

Requirements for the network metrics:

What are the characteristic requirements that we would hope that our metrics would satisfy? (a) Capture topological essence: We would expect our metrics to be indicative of dominant structure of the network. In particular, if the network topology is about the same then we expect the metrics would have about the same values. (b)

Hierarchical in Detail: We would actually like a ‘sequence of metrics’ or multiple metrics that begin with the most dominant, course, and holistic aspects of the network and sequentially go to finer and finer levels of detail. (c) Intuitive: We would like for the metrics to have an intuitive meaning in order to guide our understanding of the values and ranges and general behavior of the metrics. (d) Computationally Fast: We would hope that the metrics can be rapidly computed from simple sums and powers of the elements as opposed to complex mathematical functions (such as eigenvectors and eigenvalues which take far too long to compute) that can only be computed on very small networks. (e) Nodal Independence: We would like to have metrics that are unchanged by any renumbering of the nodes and thus reflective of the topology and not the numbering. (eigenvalues are an example that are independent). (f) Well Defined Mathematically: The metrics should rest on a solid mathematical foundation and not be arbitrary and thus be totally unambiguous. (recall that the C matrix specifying the topology is not defined on the diagonal!). (g) Complete: We would like to have metrics that in the hierarchical numeration can potentially be an exhaustive description of the topology (not that we wish to recreate all of the trillion values but rather that we would like to know that the hierarchical expansions are in some sense “complete”. In summary, we would like to have something like the expansion of sound waves in harmonics (Fourier analysis of orthogonal functions).

What do people do without well defined metrics?

Actually they can make a lot of progress but it is at the lowest of levels. One can count the activity of a matrix such as C(t) at a given node (computer server or bank account). The average transmission out or transmission in will have certain a certain mean value and range and one can test this continuously for each node. This is traditionally done on computers that function as servers in that the administrator simply looks at the number of incoming transmissions on each port and looks at the outgoing transmissions from each port. By knowing what is normal, the administrator can judge if these transmissions are appropriate or if something seems wrong. The same is done on electrical grids and bank accounts. Naturally one can develop detailed statistics and displays of the behaviors. Work in the last few years has indicated that most networks are not random but are what are called ‘scale free’ meaning that a few nodes become very highly connected to a very large number of other nodes (and act as hubs similar to large airport hubs) while most nodes just connect to a few nodes. The pattern for this is linear in logarithms of the variables and is still not understood or expected. But in spite of vast research, there is no general set of metrics that can provide a “space” for the representation of networks in general and which satisfies the criteria above.

Our Initial Discovery (sorry - one technical paragraph)

The intent of this white paper is to avoid the highly technical aspects already presented in the attached documents. So we will summarize by making just one technical point: We have been able to show that every possible network is in exact one to one (isomorphic) correspondence to an infinitesimal generator (Markov Lie algebra element with deterministic setting of the diagonal of C) of a Markov Monoid transformation (whose columns are automatically probability distributions). Now that we have a set of probability distributions for each node, we can compute the entropy associated with the transmission out from and into each node.

This reduces the N^2 values down to 2N values – a considerable reduction. These 2N values of entropy (N rows plus N columns) give us a measure of the order or disorder associated with the topology of connectives at that node. But it is easy for a lay reader to understand this entropy which we now describe. In lay terms, we make a specific assignment of the diagonal elements of the C matrix and get a new matrix that reflects the complexity of all the topology of the system in a set of probability distributions. These probabilities will be shown to be associated with the infinitesimal flows of a conserved entity that flows over this network at rates proportional to the values for the topology. We will return to this later.

What does entropy mean?

Most of us have heard of the term 'entropy' and know that it is a measure of the disorder in a system. We also know that entropy always increases and a perfect example is that stuff just gets into disorder all by itself. The reason is that 'order' or 'information' is not the natural state of things. It is just not highly probable that all the grass will grow at exactly the same rate and stay uniform, nor that all the pens and pencils are going to naturally gravitate back to the same desk drawer any more than dust is all going to fall and collect in the trash can all by itself. These are the fundamentals that we now want to make more tangible for our use.

Information is usually defined as the negative of entropy and represents order. We will use the terms 'information' and 'entropy' interchangeably as each is the negative of the other and they measure the same concept. Information was first defined by Shannon in 1948 as a method of monitoring information loss and redundancy of transmission and communications signals. It was Shannon that defined information as the logarithm of the probability (based upon the fact that probabilities of independent systems multiplied thus information must be the log of the probability in order to add for independent systems).

So rather than talk about entropy lets talk about information and order in a system. When we want to get things organized, we put all the trash in the trash can, the paper in the paper drawer etc. In other words we move the probability distribution so that it peeks at one place thus we know where to find things. This gives us order. How do we measure it? One good answer and the one we predominantly use in our work is that we take the square of the probability distribution at each place, add it up, multiply by N and take the logarithm of the result. (i.e. we define information for column 'j' as $E_j = \log_2 (N\Sigma_i M_{ij}^2)$ ie the log of N times the sum of the squares of the Markov probabilities). How does this work? Well when most of the dirt is in the trash can, then the probability at that point of finding dirt is very high so if we square the value, it becomes much higher still. Adding up all the squares of the values of the probabilities will give us a representation of how well all the trash is 'collected together'. It is true that we could use another higher power such as cube or the fourth power but that is not necessary to consider now as the second power will suffice.

When we use the connection matrix to make this Markov matrix, we get a probability distribution at each node that gives the probability that such a transformation would transfer something to that node based upon the strength of the associated value in the connection matrix. It takes quite a bit of explanation but it turns out that the connection matrix generates an infinitesimal flow of a ‘probability’ from one node to another. This is not a real flow but it does reflect the topology of the connection matrix in the form of the Markov matrix of probabilities. As a consequence, when we compute the entropy (or information) of each column in the Markov matrix then we get a measure of the degree of organization associated with transfers to that node from other nodes. If all other nodes are about equally transferring to that node (most values are about the same) then the entropy is high and the information function will be low. But if only a few nodes are transferring to that node in this time frame, then the information will be high exactly like having all the trash in one trash can. All this is to say that the value E_j is a measure of how organized the ‘connections’ are flowing to the node j. .

Restating this we can say that the Markov theory gives us some mathematical method for getting an incoming probability distribution of the Markov matrix say M_{11} M_{21} … M_{N1} representing a probability for each of the N nodes. We know that the sum of these is unity since the probability of doing something is one thus $M_{11} + M_{21} +\ldots + M_{N1} =1$. This is true of all probability distributions.

But one can have an even distribution where $M_{11} = M_{21} =\ldots = M_{N1}$ … and this distribution represents the maximum disorder or entropy and thus the least information. We do not know where something is located. It is like having all the dirt and trash in a room evenly spread out all over with the same probability. But when we clean up we make all the probabilities for dirt and paper equal to zero except for one place – the trash can. Thus the maximum information or order is where say $M_{21} = 1$ and thus the rest are zero $M_{11,}$ $M_{31\ldots} = 0$.

<u>Now we have measures of the ‘organization’ of the network:</u>

We now have this information metric for each column E^c_i of the Markov matrix that is generated from the connection matrix that represents all the connections in the network. We have gone from N^2 values to N. We can do the same thing for the rows that we did for the columns and get the N information metrics E^r_i for each transmission out from the nodes. So what good is this?

The next thing to realize is that it does not matter in this given time frame which node is doing what but rather <u>what is the pattern of the information values</u>. In other words we can sort the values E^c_i in order and form a curve that is always decreasing (or sometimes flat) and this curve give the profile or ‘spectra’ of the organization of the transfers into the nodes. Likewise we can use the same sort order to obtain a curve for the transfers out (i.e. the row values).

Now here is the main thrust of the idea: If the network transfers in and out from nodes is about the same as it was in the previous period of time, then the curve will not change. Thus we can study this curve for a given network and determine when it changes by overlaying it on the average curve for that network. If the curve rises up or falls below the normal curve, then the nodes at that point are behaving in an anomalous fashion. We can use the lookup table (from prior to the sort) to see which nodes are doing this and then take further action in analysis.

So how does this software work?

Well the first thing is to realize that the software is general purpose and will work on electrical power grid networks as well as on financial networks. The first issue is to define the 4 network variables: Time, Node i, Node j, Weight. This is the data that must flow electronically out of your database and into this program. Time will normally include the date and might be the number of seconds since a fixed date such as Jan 1, 1900. The node identification (node i and node j) can be in any form that is unique such as a bank account number, air port three letter designation, or a substation for a power grid. Our software will automatically number the nodes in a sequential method, and set up a lookup table so you can later see what node 522 really references. Also this allows other analysts to study the data without any confidential information such as IP addresses for computers or individual bank accounts being seen (as long as they do not have the look up table). Finally the weight is the measure of the strength of connection. There are reasons that you might want to use the logarithm of a fund transfer rather than the dollar value itself as it is probably not true that a $5,000 transfer is 5 times as ‘strong’ as a $1,000 transfer.

The next decision is how long in time to make the window for the construction of each C matrix. You will want to make it long enough to get representative data on the structure of the network but not so long that the changes in time are lost. For example, the window of time might be one whole day for financial transactions but might only be a few seconds for an electrical grid. The data records will be added up into a new C matrix for the selected window of time with weights being added to any values already in the cells. For example a record like 1354.5, 34, 758, 12.2 for t, i, j, would add the value 12.2 into the element $C_{34,758}$ and so forth for all of the period of time in that window say from t=1300 to t=1400.

After the C matrix is constructed for this window of time, we must sum the elements in each of the rows and put the sum in that corresponding diagonal element with a minus sign. (This is to create the infinitesimal generator for the Markov transformation that will give us the probability distributions in the columns). We next normalize the matrix by dividing all matrix elements by the negative of the total of all diagonal elements. This somewhat normalizes the matrix to a unit trace and makes all matrices of the same intensity. We keep that previous value of the total diagonal, A(t), and plot it as the amplitude later on separately from the entropy spectra.

Next we form the M matrix using the series expansion $M=e^{\lambda C} =1+ \lambda C+\lambda C^2/2! +_{.}$ Here we must choose the number of terms, k, that will be used in the expansion as well as the expansion parameter λ. One must be somewhat familiar with series expansions to realize that a choice of a very, very small λ will mean that higher powers are extremely small and thus the number of powers used, k, should not be high. Likewise a larger value of λ will lead to a need to use more powers of C in the expansion.

Next, the program computes the entropy of each column and sorts the results into a spectra (distribution of entropies) of values in order. It is this spectral curve that changes each time the window changes. It will normally be near its average value but wherever it deviates significantly, we can check those nodes to try to see why there is an abnormal deviation. How do we know about the deviations? One must first establish what the normal distribution is and then compare the normal with the spectral curve in that window of time. This can be done by summing the squares of the differences between the two curves and arriving at one value say $E^c(t)$. The software will redo this entire procedure also for the rows and arrive at a value $E^r(t)$. The software then monitors both $E^c(t)$ and $E^r(t)$ along with the amplitude value of the trace A(t) that we referred to earlier. In conclusion, the software plots these three functions over time as well as the superposition of the instantaneous entropy spectral curves with normal spectra for both the rows and columns. It is these two curves that let one identify where in the network the anomalies are occurring.

But what is 'normal' behavior for the entropy spectra?

The normal behavior might in fact be different for 9 AM on Monday morning than it is on a Sunday afternoon due to the differences in work schedules. So we must use a 'normal' spectra that will reflect the pattern of behavior for that time of day, day of week, time of month, and in consideration of holidays. Additionally, those who are familiar with network behavior are aware that other factors can alter the values such as weather and its influence on electricity consumption in a power grid. Our software can 'lock in' the spectra under different circumstances and use this as normal for future comparisons. More sophisticated computations are possible.

What will this software tell me about my network?

The ExaSphere Network Analysis Engine is a general purpose algorithm that determines "if things are about normal and if not then where are they abnormal". More specifically, the software compares the entropy profile at a given time to the normal profile. It is important to realize that the comparison is made not with the idea that each node continuously has the same behavior but rather that some node will behave the way that some other node behaved in the normal profile in analogy with thermodynamics. Other more detail analysis can be done with these tools but this gives the user a general overview.

What is the current objective in ExaSphere?

Our applications of the ExaSphere Network Engine software has thus far been exclusively for monitoring internet traffic on servers to look for attacks, malicious processes, and system malfunctions. We are now are ready to deploy our software in other environments such as power grids and gas distribution; financial network structures and ownership networks; telephony networks; social, biological, and disease networks; and specifically the diverse types of transportation networks (highway traffic, international shipping, airline networks, and trucking networks).

We seek to deploy our software in well established companies, organizations, and military environments that will commit to testing the software on a no-cost basis for a limited time in these and related specific application areas. The specific objective is to learn the degree of functionality and applicability in such diverse network areas.

Please contact:
Joseph E. Johnson, PhD
Email: jjohnson @ sc . edu or Phone: 803-777-6431