# Network clustering via spectral projections

Damien Fay[a], Hamed Haddadi[b,a], Steve Uhlig[c], Liam Kilmartin[d], Andrew W. Moore[b], Jérôme Kunegis[e], Marios Iliofotou[f]

[a]*Computer Laboratory, University of Cambridge, UK.*
[b]*Royal Veterinary College, University of London, UK*
[c]*Deutsche Telekom Laboratories and Technische Universität Berlin, Germany*
[d]*NUI Galway, Ireland*
[e]*University of Koblenz-Landau, Germany*
[f]*University of California at Riverside, USA*

## Abstract

This paper proposes a novel non-parametric technique for clustering networks based on their structure. Many topological measures have been introduced in the literature to characterize topological properties of networks. These measures provide meaningful information about the structural properties of a network, but many networks share similar values of a given measure [1]. Furthermore, strong correlation between these measures occur on real-world graphs [2], so that using them to distinguish arbitrary graphs is difficult in practice [3].

Although a very complicated way to represent the information and the structural properties of a graph, the graph spectrum [4] is believed to be a signature of a graph [5]. A weighted form of the distribution of the graph spectrum, called the weighted spectral distribution (WSD), is proposed here as a feature vector. This feature vector may be related to actual structure in a graph and in addition may be used to form a metric between graphs; thus ideal for clustering purposes.

To distinguish graphs, we propose to rely on two ways to project a *weighted* form of the eigenvalues of a graph into a low-dimensional space. The lower dimensional projection, turns out to nicely distinguish different classes of graphs, e.g. graphs from network topology generators [6, 7, 8], Internet application graphs [9], and dK-random graphs [10]. This technique can be used advantageously to separate graphs that would otherwise require complex sets of topological measures to be distinguished [9].

*Keywords:* Internet topology, Topology generation, Spectral graph theory,

## 1. Introduction

Graphs offer a very versatile means of representing patterns and relationships between entities in many different fields of engineering and science. Significant research has focused on the development of techniques and algorithms to facilitate the identification of patterns or structures within individual graphs [11] and to quantify the characteristics of such graphs [1]. These measures provide meaningful information about the structural properties of a graph, but many graphs share similar values of a given measure [1]. Furthermore, strong correlation between these measures occur on real-world graphs [2], so that using some of them to distinguish arbitrary graphs is difficult in practice [3]. Thus a key problem in clustering of graphs is the selection of an appropriate feature vector. The technique presented here proposes a *universal feature vector methodology* based on a graph metric.

Another way to represent the information and the structural properties of a graph is through the graph spectrum [4]. The spectrum of a graph is often compared to a signature of a graph [5]. Important strutural properties can be captured with the graph spectrum, e.g., its robustness through the algebraic connectivity [12] or the speed at which propagation occurs on it through the spectral radius [13]. However, all of these techniques use particular parts of the spectrum (the first $k$ eigenvalues for example) while ignoring the rest of the information.

In this paper, we aim to distinguish between graphs with different structural properties, without having to make assumptions about which properties actually characterize best the graphs under study. This is a difficult task but as will be shown with appropriate weighting the entire spectrum may be used to represent the structure of a graph. In addition, this *strutural* representation may be used to construct a *metric* and so has many desirable properties which measures do not. Specfically a metric defines consistent distances between graphs and is thus ideally suited to clustering. Given consistent distances between objects allows well known projections onto lower-dimensional spaces. In this paper we demonstrate this with the lower dimensional projections; random projection (RP) and multi-dimensional scaling (MDS). Indeed, in the example applications shown the separation is such that it can be seen clearly in a 2-3 dimensional space. Clustering this data is then an easy task.

The example applications are: graphs produced by network topology generators [6, 7, 8], Internet application graphs [9], and dK-random graphs [10]. Our methodology can be used advantageously to discriminate between graphs that would otherwise require complex sets of topological measures to be clearly distinguished [9].

The rest of this paper is structured as follows. In Section 2 we present the related work. Section 3 explains the theoretical background on the weighted spectral distribution, random projections and multi-dimensional scaling. We provide applications of our technique in Section 4, and conclude in Section 5.

## 2. Related work

Most of the related work for this paper comes from image analysis. In this area, the use of clustering algorithms on multiple graphs has been applied to the problem object identification and the related tasks of image matching or clustering and image indexing in large databases. In [14], the use of continuous time quantum walks, an extension of the classical random walk, applied to an auxiliary graph constructed from two graphs which are to be matched is proposed. A similarity measure is calculated based on a set of probabilities derived from the interference patterns associated with the two graphs and combined with information on edge consistency. The algorithm was evaluated using both synthetic data and by completing a clustering analysis using a graph representation of a database of images of objects viewed from different perspectives. *Multi-Dimensional Scaling* (MDS) is used as a method of visualising the performance of the proposed algorithm with this database of object images.

The use of graph spectral analysis techniques for image clustering is examined in [15]. Various parameters derived from the eigendecomposition of the adjacency matrix are used to form representative feature vectors for individual graphs. Wilson et. al. [16] describes a further enhancement on this approach based on the spectral decomposition of the Laplacian of the graph. A feature vector formed from the coefficients of the elementary symmetric polynomial of the spectral matrix of the Laplacian was proposed due to the fact that it offered a feature vector which was invariant under permutation of the row indices. PCA, MDS and Locality Preserving Project (LPP) techniques were used to illustrate that the resultant feature vectors from graphs representing images of three dimensional objects and image boundaries exhibited well defined clustering behaviour. However, one issue with such spectral approaches is the need to complete a complete eigendecomposition of either the adjacency matrix or some derivation thereof. While this is feasible for graphs with a relatively small number of nodes (as is typical in image analysis problems), it is not a tractable solution for graphs with large numbers of nodes, as encountered in many complex network domains. The WSD presented in this paper also uses the spectrum of the graph Laplacian. However, in contrast to Wilson et. al. [16] we first show how that by weighting the spectrum appropriately a *metric* of graph structure can be formed. This metric forms the basis for graph clustering as compared to the feature vectors used in [16] (which are not metrics but graph measures).

Interest in the topic of clustering graphs is far more sporadic outside the field of image analysis. Some research in the area exists in the fields of chemo-informatics and bio-informatics, where graphs represent molecules, and the number of sample graphs may be high. Maggiora and Shanmugasumdaram [17] provide a high-level insight into the use of graph clustering as a candidate technique for comparing molecular structures during the process of drug discovery and development. In their work, the clustering of graph-based representations of the structure of molecules is presented as one of several different feature abstraction and machine learning techniques that have been proposed for investigating molecular similarities in the field of chemo-informatics. In a more general setting, another significant contribution is made in Reforgiato et al. [18] which describes an application independent approach to the problem of clustering of graphs. The proposed approach utilises a number of different algorithms, based on identifying important sub-graph structures, to implement a flexible framework in which a clustering analysis can be carried out on any form of graph based dataset. The authors evaluated the performance of their proposed framework by applying it to the task of clustering a chemical [19] and a biological (i.e. RNA) based dataset and by comparing the resultant clustering performance with other non-graph based clustering algorithms.

## 3. Theoretical background

The *weighted spectral distribution* (WSD) is a graph metric based on the normalised Laplacian matrix of a graph. It can be used for comparing the difference in structure between two or more graphs. The metric depends on looking at the distribution of random walk cycles of length $N$ (where $N$ is a parameter of the transform) and how they are distributed across the graph. The technique was first introduced in [20] and further details may be found therein. In this paper we extend this technique in two ways. Firstly, we demonstrate how the WSD may be combined with lower dimensional projection for graph clustering. Secondly, we refine the WSD so that the bins selected for the distribution target the data; in the original paper the technique assumes a uniform bin size.

Specifically, define a graph, $G = (V, E)$, to be a collection of vertices, $V$, and undirected edges, $E$, with number of vertices $|V| = M$. The adjacency matrix of this graph, $A$, is a symmetric matrix with zeros along the diagonal (no self loops) and with:

$$A(G)(u,v) = \begin{cases} 1, & \text{if } u, v \text{ are connected} \\ 0, & \text{if } u, v \text{ are not connected} \end{cases} \tag{1}$$

The Normalised Laplacian $L$ associated with a graph $G = (V, E)$ is constructed from $A$ by normalising the entries of $A$ by the node degrees of $A$ as

$$L(G) = I - D^{-1/2} A D^{-1/2} \tag{2}$$

where $D$ is a diagonal matrix of the degree of $A$, $D = \sum_i A_{i,j}$. Expressing $L$ using the eigenvalue decomposition,

$$L(G) = \sum_i \lambda_i e_i e_i^T \tag{3}$$

where $e_i$ and $\lambda_i$ are the eigenvalues and eigenvectors of $L$ resp [1] the WSD is based on the following theorem from [20]:

**Theorem 3.1.** *The eigenvalues, $\lambda_i$, of the normalised Lapacian matrix for an undirected network are related to the random walk cycle probabilities as:*

$$\sum_i (1 - \lambda_i)^N = \sum_C \frac{1}{d_{u_1} d_{u_2} \dots d_{u_N}} \tag{4}$$

where $N$ is the length of the random walk cycles (Equation (4) is valid for each of $N = 2, 3, \dots.$), $d_{u_i}$ is the degree of node $u_i$ and $u_1 \dots u_N$ denotes a path from node $u_1$ of length $N$ ending at node $u_N$, i.e. an $N$-cycle. For

[1] These are in general different from the eigenpairs of the walk Laplacian.

a proof see [20]. $C$ is a set which contains all the nodes which are part of a random walk cycle in a graph; the set enumerates the walks[2]. Theorem 3.1 states that the probability of taking a random walk of length $N$ that returns to the original node, is directly related to the weighted eigenvalues of $L$. This probability is the 'local structure' of a node, i.e. its local connectivity. Noting that the $\lambda_i$ are unique [3] to a graph it can be seen that the WSD gives a "thumbprint" for the structure of a graph. As shown in [20] this can be used for estimating the parameters of a topology generator that produce graphs which are close (in the WSD sense) to the target graph.

The eigenvalues $\lambda_0, \ldots, \lambda_{n-1}$ represent the strength of projection of the matrix onto the basis elements. This may be viewed from a statistical point of view [21] where each $\lambda_i e_i e_i^T$ may be used to approximate $A(G)$ with approximation error inversely proportional to $1 - \lambda_i$. However, for a graph, those nodes which are best approximated by $\lambda_i e_i e_i^T$ in fact form a cluster of nodes. This is the basis for spectral clustering, a technique which uses the eigenvectors of $L$ to perform clustering of a dataset or graph [22]. The first (smallest) non-zero eigenvalue and associated eigenvector are associated with the main clusters of data. Subsequent eigenvalues and eigenvectors can be associated with cluster splitting and also identification of smaller clusters [23]. Typically, there exists what is called a *spectral gap* in which for some $k$ and $j$, $\lambda_k \ll \lambda_{k+1} \approx 1 \approx \lambda_{j-1} \ll \lambda_j$. That is, eigenvalues $\lambda_{k+1}, \ldots, \lambda_{j-1}$[4] are approximately equal to one and are likely to represent links in a graph which do not belong to any particular cluster. It is then usual to reduce the dimensionality of the data using an approximation based on the spectral decomposition. However, our technique deviates from clustering: the approach proposed here is aimed at representing the global structure of a graph, e.g., the presence or absence of many small clusters (*but not with their specific location*), which is essentially the spread of clustering across the graph. This information is contained in all the eigenvalues of the spectral decomposition.

The number of $N$-cycles is related to many graph properties. The number of 2-cycles is just (twice) the number of edges and the number of 3-cycles

[2] For example, a graph with 3 cycles and with $N = 4$ would result in $C$ containing 3 elements, each containing 4 labels. Note: $C$ is not easy to generate in general and is never actually calculated in practice.

[3] This is not strictly true but the proportion of co-spectral graphs is thought to be insignificant.

[4] i.e., the eigenvalues at the centre of the spectrum.

is (six times) the number of triangles. Hence $\sum_i (1-\lambda_i)^3$ is related to the well known clustering coefficient (as discussed in [20]). An important graph property is the number of 4-cycles. A graph which has the minimum number of 4-cycles, for a graph of its density, is quasi-random, i.e., it shares many of the properties of random graphs, including, typically, high connectivity, low diameter, having edges distributed uniformly through the graph, and so on. This statement is made precise in [24] and [25]. For regular graphs, (4) shows that the sum $\sum_i (1-\lambda_i)^4$ is directly related to the number of 4-cycles. In general, the sum counts the 4-cycles with weights: for the relationship between the sum and the quasi-randomness of the graph in the non-regular case, see the more detailed discussion in [26, Chapter 5]. The right hand side of (4) can also be seen in terms of random walks. A random walk starting at a vertex with degree $d_u$ will choose an edge with probability $1/d_u$ and at the next vertex, say $v$, choose an edge with probability $1/d_v$ and so on. Thus the probability of starting and ending randomly at a vertex after $N$ steps is the sum of the probabilities of all $N$-cycles that start and end at that vertex. In other words exactly the right hand side of (4).

The left hand side of Equation (4) provides an interesting insight into graph structure. The right hand side is the sum of normalised $N$-cycles whereas the left hand side involves the spectral decomposition. We note in particular that the spectral gap is diminished because eigenvalues close to one are given a very low weighting compared to eigenvalues far from one. This is important as the eigenvalues in the spectral gap typically represent links in the network that do not belong to any specific cluster and are not therefore important parts of the larger structure of the network.

We now formally define the *weighted spectrum* as the normalised sum of $N$-cycles as

$$W(G,N) = \sum_i (1-\lambda_i)^N \tag{5}$$

However, calculating the eigenvalues of a large (even sparse) matrix is computationally expensive. In addition, the aim here is to represent the *global* structure of a graph and so precise estimates of *all* the eigenvalue values are not required. Thus, the distribution[5] of eigenvalues is sufficient. In this paper the distribution of eigenvalues $f(\lambda = k)$ is estimated using pivoting and

[5] The eigenvalues of a given graph are deterministic and so *distribution* here is not meant in a statistical sense.

Sylvester's Law of Inertia to compute the number of eigenvalues that fall in a given interval. To estimate the distribution we use $K$ bins[6]. A measure of the graph can then be constructed by considering the distribution of the eigenvalues as

$$\omega(G,N) = \sum_{k \in K} (1-k)^N f(\lambda = k) \tag{6}$$

where the elements of $\omega(G,N)$ form the *weighted spectral distribution*:

$$WSD : G \rightarrow \Re^{|K|}\{k \in K : ((1-k)^N f(\lambda = k))\} \tag{7}$$

In addition, a metric can then be constructed from $\omega(G)$ for comparing two graphs, $G_1$ and $G_2$. This takes the quadratic norm between two WSD's as:

$$\Im(G_1, G_2, N) = \sum_{k \in K} (1-k)^N (f_1(\lambda = k) - f_2(\lambda = k))^2 \tag{8}$$

where $f_1$ and $f_2$ are the eigenvalue distributions of $G_1$ and $G_2$ and the distribution of eigenvalues is estimated in the set $K$ of bins $\in [0,2]$. Equation (8) satisfies all the properties of a metric (see [20]). For a simple worked example we refer to reader to [20] Section IV.

### *3.1. Bin selection based on equalised weightings*

The original WSD proposed using a uniform bin size [20]. However, for a given number of bins, this may not provide the best resolution. It is desirable to have a greater resolution at those points which provide more information at the cost of lower resolution elsewhere. As the weighting in the WSD is polynomial, a uniform bin size does not achieve this. The aim of this section is to assign bins in the WSD given a particular value of $N$ such that *the sum of the weighting in each bin is equal.* The weighting in the WSD may be expressed as:

$$w(x) = (1-x)^N \tag{9}$$

where $w(x)$ is the weight applied to an eigenvalue at $x$. In order to equalise the power within each of the $K$ bins we require that:

$$\int_{k_i}^{k_i+1} w(x)dx = \int_{k_j}^{k_j+1} w(x)dx \ \forall i,j \tag{10}$$

[6] The selection of these bins is considered below.

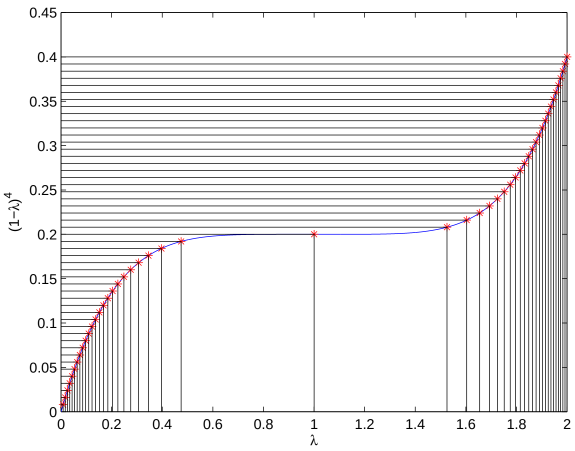

Figure 1: Integral of $w(x) = (1-x)^4$ and bin locations. ($M = 2000$).

i.e. the weight in bin $i \in k_i, k_{i+1}$ should be equal to the weight in bin $j \in k_j, k_{j+1}$. Equation 10 may be solved by a simple integration followed by solving the roots of the equation[7]. The equalised bins for $N = 4$ and $K = 50$ are shown in Figure 1. Note how the weight assigned in each bin is uniform (i.e. on the $y$-axis), given the non-uniform bins on the $x$-axis. Intuitively, the bins should target the most important points in the spectral distribution: those closest to 0 and 2. This is indeed the case as seen in Figure 1. An example of the WSD for a graph using uniform bins and equalised bins is shown in Figure 2. There are 71 bins in each plot. Note how the two WSD's are similar. However, the bins with equalised weightings contains more detail in the region of high amplitude while the uniform bins *waste* effort sampling at points of less importance, i.e. around the spectral gap at 1.

It was found that the clustering resulting from equalised bins gives much improved results, and is therefore used in the remainder of this paper.

### *3.2. Lower Dimensional Projection*

The WSD produces a mapping from $\Re^{M\times M} \longmapsto \Re^{|K|}$ where $|K| = 71$ bins are used in the examples in this paper. However, a 71 dimensional space is still too large to effectively visualise clustering across graphs. In this section,

[7] The roots of a polynomial of order 4 or higher cannot be expressed rationally and so are not presented here.

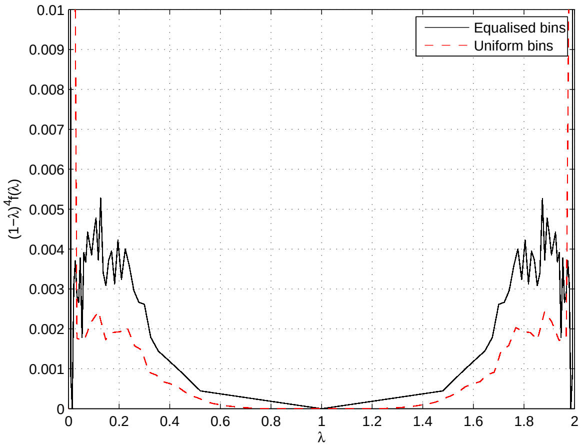

Figure 2: Example of detail in WSD captured by equalised bins compared with uniform bins.

we introduce two commonly used techniques to map the WSD into a lower dimension: *Random Projection* (RP) and *Multi-Dimensional Scaling* (MDS).

Specifically, given $C$ different graphs the aim is to seek a mapping from their WSD's into an $l$ dimensional space: $\Re^{C\times|K|} \longmapsto \Re^{C\times l}$ where $l << |K|$. Typically $l = 2$ or 3 makes visual inspection possible. Note that the methods used are parameter-free and so a *natural* clustering of the data is sought, as opposed to a supervised method which applies a mapping learned from training data.

### *3.2.1. Random Projection*

Random projection [27] is a technique often used in compressed sensing, in which a high dimensional matrix is reduced to a low dimensional matrix by multiplying the data by a random matrix as:

$$Z = XT \tag{11}$$

where $Z \in R^{C\times l}$ is the projected data matrix, $X \in R^{C\times|K|}$ are the WSD's of the $C$ graphs, $T \in R^{|K|\times l}$ is the random projection matrix where each of the elements of $T$ are drawn from a Gaussian distribution $T \sim N(0,1)$. As the rows of $T$ are normally distributed independent variables, their correlation is zero in expectation and so they form (in expectation) orthogonal vectors. In addition the norm of the vector is 1 and so $T$ forms a reduced basis in the original data.

### *3.2.2. Multi-Dimensional Scaling*

MDS [28] is a well-known technique which maps *distances* between objects onto a reduced dimensional space. An intuitive example involves taking the distance matrix commonly shown in the bottom corner of many road maps and using it to reconstruct the map itself. Unlike random projection, the technique uses the *distance* between the graphs here defined in terms of the metric introduced in Equation 8, $\Im(G_1, G_2, N)$. First, a dissimilarity matrix, $R$, is constructed as:

$$R_{(i,j)} = \begin{cases} \Im(G_i, G_j, N) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{12}$$

The goal of MDS is to find a set of vectors $Z_1, Z_2, ...Z_{|K|}$ that incrementally approximate the distance in the dissimilarity matrix. Specifically, we wish to minimise the distance between the projected vectors and the orignal data as:

$$C = \min_{Z_1,Z_2,...Z_{|K|}} \sum_{i<j} (\|Z_i - Z_j\| - R_{(i,j)})^2 \tag{13}$$

where $C$ is the cost function to be minimised. The minimisation in this paper is performed using numerical optimisation based on the eigenvector decompostion of $R$ [29]. Typically, the first and second vectors, $Z_1$ and $Z_2$ are sufficient to allow visualisation of clustering within the data. In the sequel, we denote by WSD+RP and WSD+MDS the random projection and multi-dimensional scaling techniques, respectively.

## 4. Applications

This Section provides some real-world examples of the use of the proposed technique. We examine three scenarios using the WSD as a feature vector and then project this feature vector into 2/3-D showing the clear separation of the different classes of objects. Specifically, the three examples, chosen from the areas of computer networking, are:

- *Network topology generators:* Existing topology generators rely on very different rules to build graphs. We show that the generated graphs can be clustered in a low dimensional space. This makes it possible to distinguish the different graph structures that are sampled by these generators.

- *Network application identification:* Graphs constructed from the interaction of nodes using the same application can be distinguished with our techniques. Previous work required multiple metrics, identified through manual inspection in order to classify applications [9].

- *Orbis based topologies and the dK-series:* dK-series were introduced in [10] to capture degree correlations in real-world graphs. The resulting topology generator, called Orbis, creates subsets of graphs embedded according to the dK-series paradigm. We show that the graphs generated by Orbis are much more similar than previously thought and this is caused by a strong implicit prior on the graph structures generated.

### *4.1. Topology generator projections*

The aim of this section is to demonstrate how the WSD+RP may be used to distinguish between topology generators. A topology generator is a set of rules which are used to build up a synthetic graph. For example, the Waxman topology generator first generates $M$ nodes distributed uniformly on a square and then connects points according to probability:

$$p(u, v) = \alpha e^{-\beta h_{u,v}} \tag{14}$$

where $p(u, v)$ is the probability of connecting nodes $u$ and $v$, $\alpha$ and $\beta$ are parameters of the generator and $h_{u,v}$ is the Euclidean distance between $u$ and $v$ on the square. The AB and GLP topology generators are based on preferential attachment while the INET model is based on a complex model for how connections are formed in the Internet (see [30] for more details).

For each type of topology generator a *family* of WSD's may be generated by varying the parameters of the generator. The aim at this point is to show that these WSD families map onto different curves for different topology generators. The WSD's generated by an AB model should not correspond to any of those of the GLP model or the Waxman model, etc.. We begin by sampling from the family of WSD's for each topology generator. Specifically we generate 100 topologies of each using random parameters. 71 bins are used in this experiment, resulting in a data matrix of 400 WSD's (4 topology generators) of size 400 × 71. Figure 3 shows these families of WSD's side by side. Note, it is not immediately clear from Figure 3 that these WSD's do in fact map to different points (clusters) in the 71 dimensional space.

The next stage is to reduce the dimension of this data to 400 × 2, so as to be able to visualize the clustering in the data. As we only need *any*

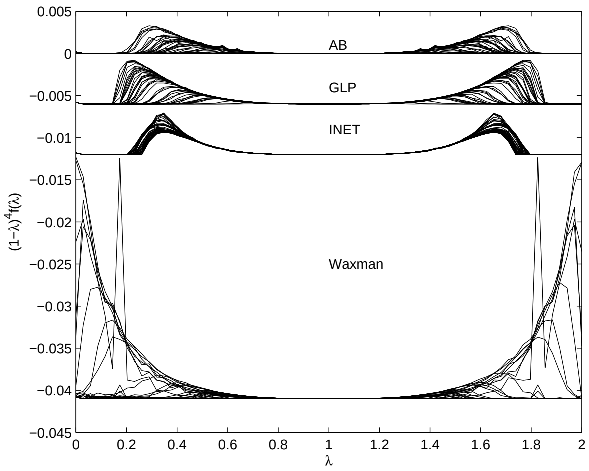

Figure 3: The WSD's for 4 types of topology generator are shown above. The figures are separated for clarity, i.e. the WSD's for the GLP model are artificially moved downward by $-0.006$ ($M = 2000$).

projection that separates the data classes (generators), *not* specifically an optimal projection, the RP technique is used.

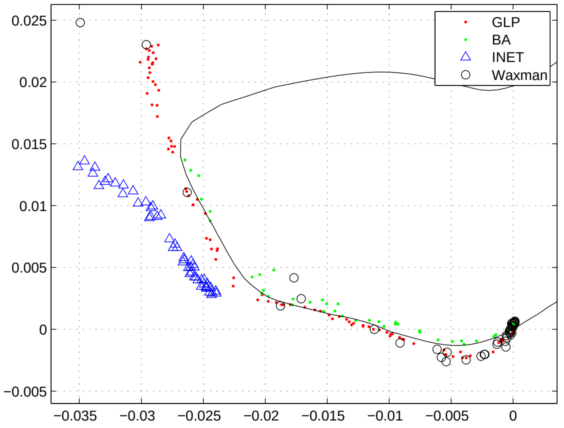

Figure 4: 2-D projection of topology generator graphs. Note the $x$ and $y$ axis are not relevant, only the separation of points. ($M = 2000$).

Figure 4 shows the projection of the sampled families onto 2 dimensions using random projections. This Figure may be thought of as a 2-D representation of 'structural space' in which a greater separation of points represents a greater separation of the corresponding graph sturctures.

The first thing to note about Figure 4 is that most of the Waxman WSD's lie well outside the range of the figure. This is to be expected as Waxman topologies differ significantly from the others. Second, the actual units of the graph are irrelevant: only the separation of the points is meaningful. At the right of the graph (around $(0, 0)$), there is a clustering in which the GLP, AB and Waxman models all overlap. This occurs at low parameter values when the graphs contain few links and are therefore difficult to distinguish. The GLP and AB graphs are very close for a large section of the families. This occurs as GLP is similar in structure to AB but not equal. In order to demonstrate this, a Support Vector Machine (SVM) [31] was used to determine the boundary between the the AB class and the GLP class. The decision boundary is shown in Figure 4 as a solid black line. Note the boundary value is irrelevant outside of the training range. As can be seen, the boundary separates the two classes efficiently, with an 11% false classification rate. The Inet models generate a different cluster of projections which is shown in Figure 4. We conclude that our technique is able to empirically distinguish different generated topological structures.

### *4.2. Network application identification*

**Traffic monitoring setup.** The traffic traces used in this work are collected from an OC48 optical link of the Metromedia Fiber Network (MFN) backbone in San Jose, CA. The data was provided to CAIDA [32] by the WAND Research Group (University of Waikato, New Zealand) using an OC48 DAG interface card. The data used in our experiments are from the 08/14/2002; between 09:00 - 10:00AM and 11:00-12:00PM. Over this time period, the captured traffic contains on average close to 500K flows (TCP and UDP) in every five minute interval. The overall volume of data approximates 1 TByte of raw IP packets. Several traces from MFN are publicly available by CAIDA [32].

**Representative sample.** Our experience with traffic graphs [33, 9, 34, 35] showed that the MFN traces are a representative sample of a large Tier-1 backbone link. Other locations we studied in the past [33, 9] include data from the Palo Alto Internet eXchange (PAIX) and Internet2 (Abilene) backbones collected over different times of the day, different days of the week, and over several years. Using publicly available traces [32] allows other researchers to extend and verify our findings and contributions. All traces are IP-anonymized and contain traffic from both directions of the link.

**Flow processing details.** Throughout this paper we group packets into flows using the standard method based on the five tuple {`SrcIP`, `SrcPort`, `DstIP`, `DstPort`, `Protocol`}. For a TCP flow, we generate a directed edge starting from the node that sent the `SYN` packet. For the UDP flows, we create a directed edge starting from the sender of the first packet. To establish the ground truth for flows (e.g., eDonkey, Web, etc.), we use a combination of signature- and port-based traffic classifiers [9, 34, 35, 36]. The monitor used for collecting the MFN trace captured 44 bytes for each packet, which includes IP and TCP/UDP headers and an initial 4 bytes of payload for some packets. Approximately 40% of the flows are classified using standard payload-based signature matching techniques as used in [37, 38, 34, 35] and for the remaining flows we used the port-based classifier from CoralReef [36], which performs very well for the MFN data as observed in [9].

On the upper left plot of Figure 5, we observe a rather clean separation between the projections of the graphs belonging to different applications. Some overlap exists between some applications, e.g. Gnutella and eDonkey or SSH and MP2P. Overlap between WINMX and MP2P or eDonkey and Gnutella are expected, as the communication patterns generated by these P2P applications are similar. However, as can be seen in Figure 5, the separation increases as we include a third dimension.

To investigate further, the data was split evenly into two randomly selected groups; a training set and a test set. The training set was then used to train a classifier using standard discriminant analysis [39]. It was found that overall there was a 14% misclassification error. More specifically the confusion matrix is shown is Table 1. In this matrix, entry $i, j$ is the proportion of class $i$ objects that are misclassified as class $j$. Ideally, this matrix should have 1 along the diagonal. Note also that the sum of any row is equal to 1 (i.e. 100%). Also note that the entry for $i, j$ is not necessarily the same as $j, i$. The diagonal and interesting values are shown in bold font.

From Table 1 it can be seen that 22% of the Gnutella graphs have problems being distinguished from the E-Donkey graphs (the reverse is not true). This is expected given the similarity of their communication patterns. In addition, 22% of FTP graphs are misclassified as HTTPS, and 13% as MP2P. We expect FTP graphs to be similar to HTTPS since both protocols have similar graph sizes and both protocols are based on the client-server architecture, with low degree nodes (clients) being connected with high degree

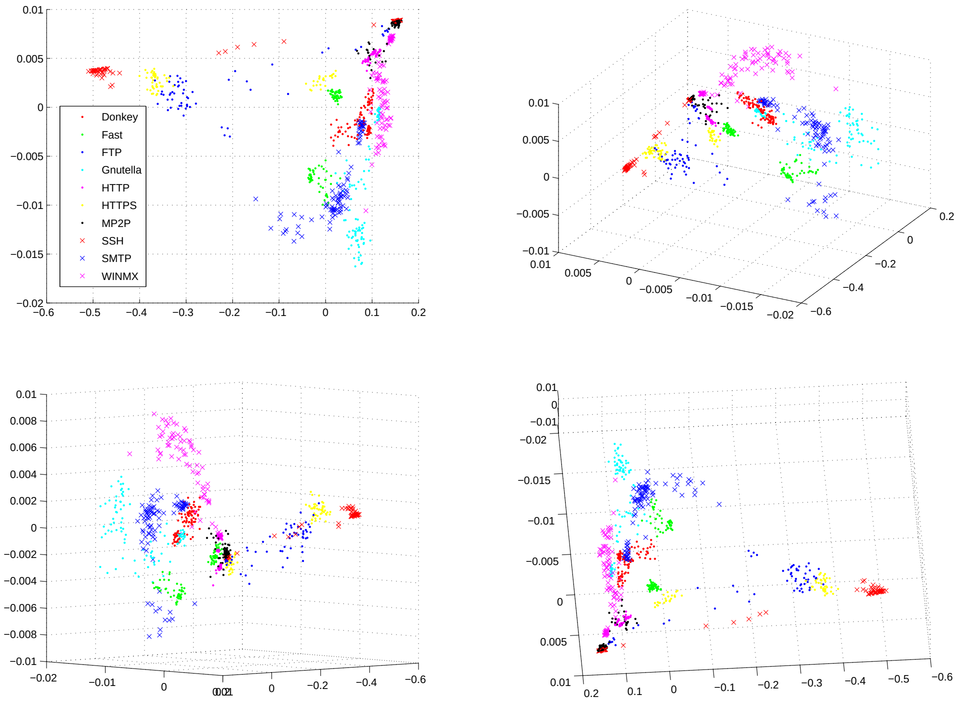

Figure 5: A 3-D plot of the application graphs mapped using WSD+RP. The upper left plot shows the $x$-$y$ plane (i.e. first two dimensions); subsequent plots show differing angles of view.

Table 1: Confusion matrix for Internet applications.

| | Donkey | Fast | FTP | Gnutella | HTTP | HTTPS | MP2P | SSH | SMTP | WINMX |
|---|---|---|---|---|---|---|---|---|---|---|
| Donkey | **0.96** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | 0.00 |
| Fast | 0.00 | **1.00** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| FTP | 0.00 | 0.00 | **0.67** | 0.00 | 0.00 | **0.20** | 0.13 | 0.00 | 0.00 | 0.00 |
| Gnutella | **0.22** | 0.02 | 0.00 | **0.74** | 0.00 | 0.00 | 0.00 | 0.00 | 0.02 | 0.00 |
| HTTP | 0.00 | 0.00 | 0.06 | 0.00 | **0.91** | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 |
| HTTPS | 0.00 | 0.03 | 0.03 | 0.00 | 0.00 | **0.94** | 0.00 | 0.00 | 0.00 | 0.00 |
| MP2P | 0.00 | 0.00 | 0.04 | 0.00 | 0.07 | 0.00 | **0.89** | 0.00 | 0.00 | 0.00 |
| SSH | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.06 | **0.31** | **0.60** | 0.00 | 0.00 |
| SMTP | 0.17 | 0.04 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.78** | 0.00 |
| WINMX | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | **0.22** | 0.00 | 0.04 | **0.74** |

hosts (servers). 31% of the SSH graphs are misclassified as MP2P. 17% of the SMTP graphs are misclassified as eDonkey. Also, SMTP and eDonkey were also confused in [34] and [9], because of the similarity in the architecture of the two applications. While we leave it for further work to study the exact reasons for these misclassifications, we expect that the limited dimensionality of the projections, as well as the similarity of the static graph structures are the two most important reasons for the misclassifications.

### *4.3. Orbis and dK-series.*

The Orbis topology generator [10] is based on the *configuration model* developed by Bollobas in [40]. The configuration model constructs a topology with a given degree distribution. First a list of edges is constructed, with both ends of the edge unlabeled. The edges are then assigned node labels, at random, to satisfy the required degree distribution. For example, a given degree distribution may require one node with degree 2 and one node with degree 3, etc. The first node is assigned to two of the, as of yet, unlabeled edges and then the second node to 3 of the edges and so on. At the end, all the edges are labeled at both ends and these are connected to form the final graph. For the Orbis topology generator, this process may be taken to the next level by considering the joint degree distribution: the probability that a node of degree $k$, say, is connected to a node of degree $j$. This is achieved by an adjustment to the configuration model in which the labeling of edges requires the satisfaction of the joint degree distribution rather than being simply random (see [10] for more details).

Note that the degree distribution is implicitly given as a marginal of the joint degree distribution. Likewise the average degree is implicitly specified in the degree distribution. In the Orbis terminology, these form the *dK series* in which we may express graphs as subsets of each other: 0K is the set of graphs with average degree, $\bar{k}$; 1K is the set of graph with degree distribution, $p(k)$; 2K are the set of graphs with joint degree distribution $p(k, j)$; 3K and higher elements of the series represent higher order cumulants. In [10], it is proposed that $dK \subset (d-1)K \cdots \subset 2K \subset 1K \subset 0K$ as shown diagrammatically in Figure 6.

The aim of this section is to generate 100 [8] from 0K, 1K and 2K topologies each, and then map these into two dimensions using the WSD+MDS;

[8] This number was found to be sufficient as can be seen by the clustering in Figure 9.

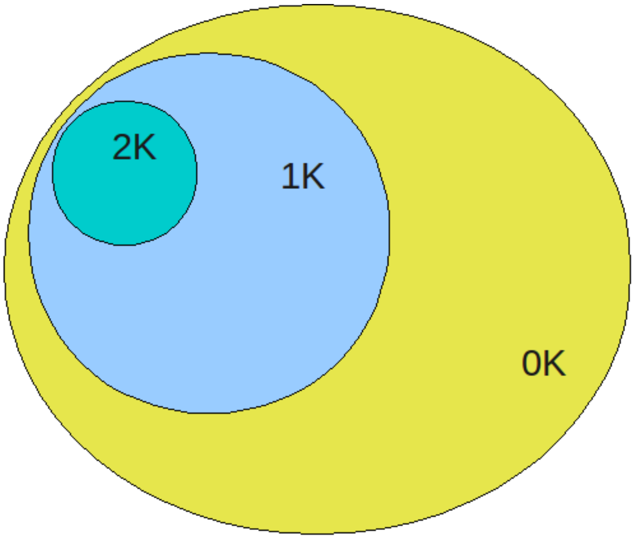

Figure 6: WSD of dK-series graphs. (Note: this figure is a recreation of Figure 2 from [41]).

essentially sampling the areas of Figure 6. The first stage involves generating a (any) model with a specific joint degree distribution. For this we used an AB model with 3000 nodes and parameters [0.3,0.1]. This creates a relatively dense AB graph with a power law degree distribution and connections based on preferential attachment (see [42] for more details). The joint degree distribution, degree distribution and average degree of this topology are then used to generate the 300 graphs using Orbis. Figure 7 shows the resulting 300 WSD's of these graphs. As can be seen in Figure 7, three distinct WSD patterns are produced.

Figure 8 shows the 2-D projection of these 300 WSD's using MDS. The first thing to note is that the 0K, 1K and 2K models form distinct clusters in the 2-D plane; this is not the expected result [9]. The reason for this unexpected clustering can be understood through the examination of the degree distributions of the 0k models (see Figure 9). The average, $\mu_{p(k)}$, and the standard deviation, $\sigma_{p(k)}$ of the 100 0K and the 100 1K degree distributions is shown in Figure 9. The degree distribution of the 0K models is highly

[9] The *expected* result is to have the 0K projections (i.e. dots) spread across a large area with the 1K projections somewhere within that area and the 2K projections inside the 1K area.

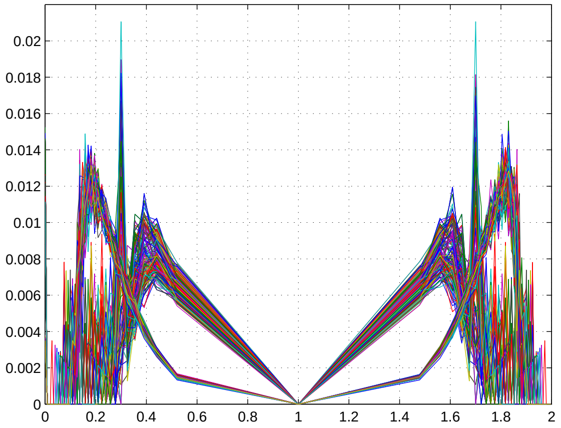

Figure 7: WSD of dK-series graphs.

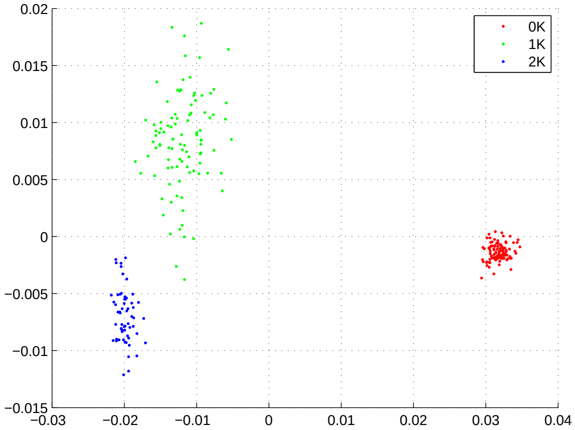

Figure 8: 2-D MDS Projection of dK-series graphs.

concentrated around the mean degree, $\bar{k}$, [10] while that of the 1K models is power-law (this is not obvious in a linear plot). In addition, the distributions themselves are highly concentrated around their means, i.e. $\sigma_{p(k)}$ is relatively

[10] This is because the 0K graphs are Erdos-Renyi graphs which are well known to have a concentrated degree distribution.

small. In summary, while it is possible that a 0K model would produce a topology with power-law distribution, the probability is vanishingly small.

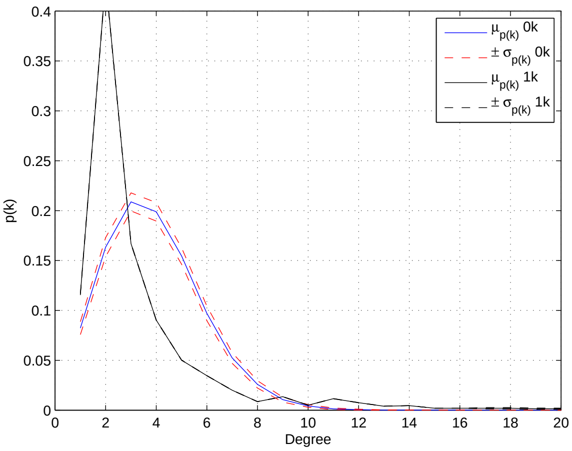

Figure 9: Degree distribution of 0K and 1K graphs with confidence intervals.

A similar situation arises with the 1K and 2K graphs: the chance of a 1K model generating a 2K model with the same joint degree distribution is again extremely small. This can be seen from the lack of overlap in their projections in Figure 8. The key problem with this approach is that the mechanism used to generate the 0K model does not result in degree distributions which are picked uniformly from the set of all degree distributions.

Likewise, the 1K graphs do not have joint degree distributions uniformly sampled from the set of possible joint degree distributions. In summary, while the three sets in Figure 6 do exist, the current mechanism only generates a very small region of those sets in practice. The Orbis generator therefore does not sample uniformly graphs as would be suggested from Figure 6 (Note: there is a discussion in the Conclusion of the consequences).

The key problem identified is that while Figure 6 is strictly true the topology's are not generated uniformly in the sample space. In fact the 0K model effectively places a very tight prior on the distribution of degree distributions in the 0K graphs. That is, the 0K graphs have a concentrated degree distribution; they should be completely random (no prior). One way to circumvent this would be to sample a degree distribution randomly from a distribution of degree distributions such that the average degree satisfies the specified one. This could then be used to generate a 1K model. The problem here of course is that this approach requires using a $dK+1$ model to generate a $dK$ topology. As the highest known practical topology generator is a $d2$ generator this restricts the topologies to $d1$.

## 5. Conclusions

Graphs offer a very versatile means of representing patterns and relationships between entities in many different fields of engineering and science. In this paper, we have proposed a technique to distinguish between graphs with different structural properties, without having to make assumptions about which properties actually characterize best the graphs under study. Our technique consists projections of a weighted graph spectrum onto lower-dimensional spaces, through random projections (RP) and multi-dimensional scaling (MDS).

We showed that these two projections (RP and MDS) turn out to be able to distinguish different types of graphs: from synthetic ones produced by topology generators to real ones resulting from the interactions between nodes participating in specific applications. Throughout these applications, we demonstrate that our technique can be used advantageously to discriminate between graphs that would otherwise require complex sets of topological measures to be clearly distinguished, e.g., [9].

The WSD+MDS technique presented may have many future applications in the growing area of real-world graph analysis from dynamic visualisation of graph structural changes to evolution of graph based systems and identification of undesirable structural regions for graphs (for example in network security).

## References

[1] L. da Costa, F. Rodrigues, G. Travieso, P. V. Boas, Characterization of complex networks: A survey of measurements, Advances of Physics 56 (2007) 167–242.

[2] A. Jamakovic, S. Uhlig, I. Theisler, On the relationships between topological metrics in real-world networks, in: European Conference on Complex Systems, Dresden, Germany, 2007.

[3] A. Banerjee, J. Jost, Spectral plot properties: Towards a qualitative classification of networks, in: European Conference on Complex Systems, 2007.

[4] D. Cvetković, M. Doob, H. Sachs, Spectra of Graphs, Theory and Applications, Johann Ambrosius Barth Verlag, 1995.

[5] E. R. van Dam, W. Haemers, Which graphs are determined by their spectrum?, Linear Algebra and its Applications 373 (2003) 241–272.

[6] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, W. Willinger, Network topology generators: degree-based vs. structural, in: Proceedings of ACM SIGCOMM 2002, Pittsburgh, PA, 2002, pp. 147–159.

[7] T. Bu, D. Towsley, On distinguishing between Internet power law topology generators, in: Proceedings of IEEE Infocom 2002, 2002.

[8] H. Haddadi, D. Fay, S. Uhlig, A. Moore, R. Mortier, A. Jamakovic, M. Rio, Tuning topology generators using spectral distributions, in: Lecture Notes in Computer Science, Volume 5119, SPEC International Performance Evaluation Workshop, Springer, Darmstadt, Germany, 2008.

[9] M. Iliofotou, M. Faloutsos, M. Mitzenmacher, Exploiting Dynamicity in Graph-based Traffic Analysis: Techniques and Applications, in: ACM CoNEXT, 2009.

[10] P. Mahadevan, C. Hubble, D. Krioukov, B. Huffaker, A. Vahdat, Orbis: rescaling degree correlations to generate annotated Internet topologies, SIGCOMM Computer Communications Review 37 (4) (2007) 325–336. doi:http://doi.acm.org/10.1145/1282427.1282417.

[11] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network Motifs: Simple Building Blocks of Complex Networks, Science 298 (5594) (2002) 824–827.

[12] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Mathematical Journal 23 (1973) 298–305.

[13] B. Mohar, Y. Alavi, G. Chartrand, O. Oellermann, A. Schwenk, The laplacian spectrum of graphs, Graph Theory, Combinatorics and Applications 2 (1991) 871–898.

[14] D. Emms, R. C. Wilson, E. R. Hancock, Graph edit distance without correspondence from continuous-time quantum walks, in: Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, 2008.

[15] B. Luo, R. C. Wilson, E. R. Hancock, Spectral clustering of graphs, in: CAIP, 2003, pp. 540–548.

[16] R. C. Wilson, E. R. Hancock, B. Luo, Pattern vectors from algebraic graph theory, IEEE Trans. Pattern Anal. Mach. Intell. 27 (7) (2005) 1112–1124.

[17] G. M. Maggiora, V. Shanmugasundaram, Molecular similarity measures 275.

[18] D. Reforgiato, R. Gutierrez, D. Shasha, Graphclust: A method for clustering database of graphs, Journal of Information & Knowledge Management (JIKM) 7 (04) (2008) 231–241.

[19] J. W. Raymond, C. J. Blankley, P. Willett, Comparison of chemical clustering methods using graph- and fingerprint-based similarity measures, J Mol Graph Model (2003) 421–433.

[20] D. Fay, H. Haddadi, A. Thomason, A. Moore, R. Mortier, A. Jamakovic, S. Uhlig, M. Rio, Weighted Spectral Distribution for Internet Topology Analysis: Theory and Applications, IEEE/ACM Transactions on Networking (TON) 18 (1) (2010) 164–176.

[21] A. Seary, W. Richards, Spectral methods for analyzing and visualizing networks: an introduction., in: Dynamic Social Network Modeling and Analysis, National Academic Press, 2003, pp. 209–228.

[22] B. Nadler, S. Lafon, R. Coifman, I. Kevrekidis, Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators, in: Neural Information Processing Systems (NIPS), (2005).

[23] A. Ng, M. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, (2002).

[24] A. G. Thomason, Pseudo-random graphs, Random Graphs ’85, North-Holland Mathematical Study 144 (1987) 307–331.

[25] F. R. K. Chung, R. L. Graham, R. M. Wilson, Quasi-random graphs, Combinatorica 9 (4) (1989) 345–362.

[26] F. R. K. Chung, Spectral Graph Theory (CBMS Regional Conference Series in Mathematics), American Mathematical Society, (1997).

[27] X. Fern, C. Brodley, Random projection for high dimensional data clustering: A cluster ensemble approach, in: Proceedings of the International Conference on Machine Learning (ICML), 2003, pp. 186–193.

[28] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, 1994.

[29] G. A. F. Seber, Multivariate Observations, John Wiley & Sons, 1984.

[30] H. Haddadi, G. Iannaccone, A. Moore, R. Mortier, M. Rio, Network topologies: Inference, modelling and generation, in: IEEE Communications Surveys and Tutorials, Vol. 10, 2008.

[31] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A practical guide to support vector classification, department of Computer Science, National Taiwan University, Taipei 106, Taiwan. `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

[32] CAIDA Trace Project, http://www.caida.org.

[33] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, G. Varghese, Network monitoring using traffic dispersion graphs (tdgs), in: ACM IMC, 2007.

[34] M. Iliofotou, H. Kim, P. Pappu, M. Faloutsos, M. Mitzenmacher, G. Varghese, Graph-based P2P Traffic Classification at the Internet Backbone, in: IEEE Global Internet, 2009.

[35] B. Gallagher, M. Iliofotou, T. Eliassi-Rad, M. Faloutsos, Homophily in Application Layer and its Usage in Traffic Classification, in: IEEE INFOCOM, 2010.

[36] D. Moore, K. Keys, R. Koga, E. Lagache, K. C. Claffy, The coralreef software suite as a tool for system and network administrators, in: USENIX LISA, 2001.

[37] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: Multi-level Traffic Classification in the Dark, in: ACM SIGCOMM, 2005.

[38] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, K. Lee, Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices, in: ACM CoNEXT, 2008.

[39] R. Johnson, D. Wichern, Applied Multivariate Statistical Analysis, 5th Edition, Prentice Hall, 2002.

[40] B. Bollobas, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, Eur. J Combinatorics 1 (1980) 311 – 316.

[41] P. Mahadevan, D. Krioukov, K. Fall, A. Vahdat, Systematic topology analysis and generation using degree correlations, in: Proceedings of ACM SIGCOMM 2006, Pisa, Italy, 2006, pp. 135–146.

[42] R. Albert, A.-L. Barabasi, Topology of evolving networks: local events and universality, Physical Review Letters 85.