A New Mathematical Space for Social Networks

Anshumali Shrivastava
Department of Computer Science
Computing and Information Science
Cornell University
Ithaca, NY 14853, USA
anshu@cs.cornell.edu

Ping Li
Department of Statistics & Biostatistics
Department of Computer Science
Rutgers University
Piscataway, NJ 08854, USA
pingli@stat.rutgers.edu

Abstract

Finding a new mathematical representation for graphs, which allows direct comparison between different graph structures, is still an open-ended research direction. Having such a representation in a common, well-understood metric space is the first prerequisite for a variety of machine learning algorithms like classification, clustering, etc, over graph datasets. In this study, we propose a symmetric positive semidefinite matrix with the (i, j)-th entry equal to the covariance between normalized vectors $A^i e$ and $A^j e$ as a representation for graph with adjacency matrix A. We argue that the covariance between vectors of the form $A^i e$ and $A^j e$, given some i and j, is an informative feature. We present theoretical results supporting this argument. Our representation, being a covariance matrix in a fixed dimensional metric space, can be directly compared across different graph structures. This naturally provides a measure of similarity on graph objects. In the task of social network classification, our proposal outperforms the state-of-the-art methodologies. In addition, the computation can be performed in operations linear in the number of edges, which makes the proposed approach faster and scalable.

1 Introduction

The study of social networks is becoming increasingly popular. A whole new set of information about an individual is gained by analyzing the data derived from his/her social network. Personal social network of an individual consisting only of neighbors and connections between them, also known as ego network, has recently grabbed significant attention [17, 23]. This new view of the gigantic incomprehensible social network as a collection of small informative overlapping ego networks generate a huge collection of graphs, which leads to a closer and more tractable investigation.

These enormous collections of ego networks, one centered at each user, open doors for many interesting possibilities which were not explored before. For instance, consider the scientific collaboration ego network of an individual. It is known that collaboration follows different patterns across different fields [19]. Some scientific communities are more tightly linked among themselves compared to other fields having less dependencies among the collaborators. For instance, scientists working in experimental high energy physics are very much dependent on specialized labs worldwide (example CERN) and hence it is more likely that scientists in this field have a lot of collaborations among themselves. It is expected that collaboration network in such a scientific domain will exhibit more densely connected structures compared to a field in which people prefer to work independently.

The above peculiarity is also reflected in the ego networks. For an individual belonging to a more tightly connected field, such as high energy physics, it is more likely that there is collaboration among the individual's coauthors. Thus, we can expect the collaboration ego network of an individual contains information about the characteristic of his/her research. By utilizing this information, it should be possible to discriminate (classify) between scientists based on the ego networks. This information can be useful in many applications including user-based recommendations [18, 9], recommending jobs [20], discovering new collaborations [4], citation recommendations [10], etc.

The focus of this paper is on social network classification or equivalently graph classification. The first prerequisite for classifying networks is having the "right" measure of similarity between different graph structures. Finding such a similarity measure is directly related to the problem of comput-

ing meaningful mathematical embedding of social networks. In the present work, we address this fundamental problem of finding an "appropriate" tractable mathematical representation for graphs.

There are many interesting theories which illustrate the peculiarities of social networks [22, 2, 14]. For instance, it is known that the spectrum of adjacency matrix of real-world graph is very specific. In particular, it is been observed that scale-free graphs develop a triangle-like spectral density with a power-law tail, while small-world graphs have a complex spectral density consisting of several sharp peaks [7]. Despite such insight into social graph structures, finding a meaningful mathematical representation for these networks with which these various graph structures can be directly compared or analyzed in a common space is an under-studied area.

Recently it was shown that representing graphs as a normalized frequency vector which counts the number of occurrence of various small k-size subgraphs ($k = 3$ or 4) leads to an informative representation [21, 23]. A sound analysis of such a representation was presented in [23]. It was shown that such a representation naturally models known distinctive social network characteristics like the "*triadic closure*". Computing the similarity between two graphs as the inner product between such a frequency vector representation leads to the state-of-the-art social network classification algorithms.

It is not clear that a histogram based only on counting small sub-graphs can sufficiently capture all essential properties of the graphical structure. It is expected that only counting small k-subgraphs ($k = 3$ or 4) will loose information. It is also not clear what is the right value of k which provides the right tradeoff between computation and expressiveness. For instance, we observe that (Section 6) $k = 5$ leads to significant improvement over $k = 4$ at the cost a significant increase of computations. Although, it is known that histograms based on counting subgraphs of size k can be reasonably approximated by simply sampling few induced subgraphs of size k, counting subgraphs with $k \geq 5$ is still computationally expensive as it requires testing the given sampled subgraphs with the representative set of graphs for isomorphism (see Section 6). Thus, finding a rich graph representation, which aptly captures its behavior and is computational inexpensive, is an important research problem.

One challenge in meaningfully representing a graph in a common space is the basic requirement that isomorphic graphs should map to the same object. Features based on counting substructures, for example the frequency of subgraphs, satisfy this requirement by default, but ensuring this property is not trivial if we take a non-counting based approaches.

Our Contributions: We take an alternate route and characterize graph based on the truncated power iteration of the corresponding adjacency matrix A, starting with the vector of all ones denoted by e. Such a power iteration generates vector $A^i e$ in the i^{th} iteration. We argue that the covariance between vectors of the form $A^i e$ and $A^j e$, given some i and j, is a very informative feature for a given graph. We present theoretical results supporting this argument.

Instead of using a histogram-based feature vector representation, we represent graph as a symmetric positive semidefinite covariance matrix C^A, whose $(i, j)^{th}$ entry is the covariance between vectors $A^i e$ and $A^j e$. To the best of our knowledge this is the first representation of its kind. We further compute the similarity between two given graph as the standard Bhattacharya similarity between the corresponding covariance matrix representations. Our proposal follows a simple procedure involving only matrix vector multiplications and summations. The entire procedure can be computed in time linear in the number of edges which makes our approach scalable in practice. Similarity based on this new representation outperforms exiting methods on a real social network classification task.

In addition, this paper also shows some interesting insight in the domain of the collaboration networks. We show that it is possible to distinguish researchers working in different experimental physics sub-domains just based on the ego network of the a researcher's scientific collaboration. To the best of our knowledge this is the first work that explores the information contained in the ego network in scientific collaboration. The results presented could be of independent interest in itself.

Notations: Graph $G = \{V, E\}$, with $|V| = n$ and $|E| = m$, is represented by a binary symmetric adjacency matrix $A \in \mathbb{R}^{n \times n}$, where $A_{i,j} = 1$ if and only if $(i, j) \in E$. We use $A_{(i),(:)} \in \mathbb{R}^{1 \times n}$ to denote the i^{th} row of matrix A, and $A_{(:),(j)} \in \mathbb{R}^{n \times 1}$ to denote its j^{th} column. We use e to denote the vector of all 1s. Dimension of vector e will be implicit depending on the operation. Vectors are by default column vectors ($\mathbb{R}^{n \times 1}$).The transpose of a matrix A is denoted by A^T, defined as $A^T_{i,j} = A_{j,i}$. The covariance between two vectors $x \in \mathbb{R}^{n \times 1}$ and $y \in \mathbb{R}^{n \times 1}$ is defined as $Cov(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$, where $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$.

2 Graphs as a Symmetric Positive Semidefinite Matrix

A graph is fully described by its adjacency matrix. A good description of a matrix operator is a set of vectors generated from its *power iteration*. *Power iteration* of a matrix $A \in \mathbb{R}^{n\times n}$ on a given starting vector $v \in \mathbb{R}^{n\times 1}$ computes normalized $A^i v \in \mathbb{R}^{n\times 1}$ in the i^{th} iteration.

In a seminal result [13], it was shown that the characteristic polynomial of a matrix can be computed by using the set of vectors generated from its truncated power iterations, $\{v, Av, A^2v, ..., A^kv\}$, commonly known as the *"k-order Krylov subspace"* of matrix A. *"Krylov subspace"* lead to some of the fastest linear algebraic algorithms for sparse matrices. In the web domain, power iterations were used in algorithms including "Page-rank" and "HITS" [12]. A truncated power iteration of the data similarity matrix also lead to an informative representation for clustering [16].

To meaningfully represent graphs in a common mathematical space, a basic requirement is that isomorphic graphs should be mapped to the same object. Although the k-order *"Krylov subspace"* sufficiently characterizes a matrix, it cannot be directly used as a common representation for the associated graph because it is sensitive to the reordering of nodes. In other words the mapping $M : A \rightarrow \{v, Av, A^2v, ..., A^kv\}$ is not a *"graph invariant"* mapping.

It turns out that if we use $v = e$, the vector of all ones, then the covariances among the different vectors in the power iteration are *"graph invariant"* (see Theorem 2), i.e., their values do not change with the spurious reordering of the nodes. Power iteration starting on vector e is the key ingredient in HITS and Page-rank algorithms and are known to be quite informative. We start by defining our covariance matrix representation for the given graph, and the algorithm to compute it. In later sections we will argue why such a representation is suitable for discriminating graph structures.

Given a graph with adjacency matrix $A \in \mathbb{R}^{n\times n}$ and a number k, we compute the first k terms of power iteration, to generate normalized vectors of the form $A^i e$, $i \in \{1, 2, ..., k\}$. Since we start with e, we choose, without lose of generality, to normalize the sum equal to n for the ease of analysis. We then compute matrix $C^A \in \mathbb{R}^{k\times k}$ with $C^A_{i,j} = Cov(\frac{nA^i e}{||A^i e||_1}, \frac{nA^j e}{||A^j e||_1})$. See Algorithm 1.

Algorithm 1 *Covariance Representation(A,k)*

Input: Adjacency matrix $A \in \mathbb{R}^{n\times n}$, k, the number of power iterations.
Initialize $x^0 = e \in \mathbb{R}^{n\times 1}$.
for $t = 1$ **to** k **do**
$M_{(:),(t)} = n \times \frac{Ax^{t-1}}{||Ax^{t-1}||_1}$
$x^t = M_{(:),(t)}$
end for
$\mu = e \in \mathbb{R}^{k\times 1}$
$C^A = \frac{1}{n}\sum_{i=1}^{n}(M_{(i),(:)} - \mu)(M_{(i),(:)} - \mu)^T$
return $C^A \in \mathbb{R}^{k\times k}$

Theorem 1 *The matrix C^A is symmetric positive semidefinite.* □

Theorem 2 *For any permutation matrix π we have $C^\pi = C^{\pi A \pi^T}$ i.e., C^A is a graph invariant.* □

Note that the converse of Theorem 2 is not true. We can not even hope for it because then we would have solved the intractable *Graph Isomorphism Problem* by using this tractable matrix representation. For example, consider adjacency matrix of a regular graph. It has e as one of its eigenvectors with eigenvalue equal to d, the constant degree of the regular graph. So, we have $A^i e = d^i e$ and $Cov(d^i e, d^j e) = 0$. Thus, all regular graphs are mapped to the same zero matrix. Perfectly regular graphs never occur in practice and there is always some variation in the degree distribution of real-world graphs. For non regular graphs, i.e., when e is not a eigenvector of the adjacency matrix, we will show in the next section that the proposed C^A representation is quite informative.

3 More Properties of Matrix C^A

In this Section, we argue that the representation C^A encodes various key features of the given graph, making it an informative representation. In particular, we show that C^A contains information about the spectral properties of A as well as the counts of small substructures present in the graph.

For adjacency matrix A, let $\lambda_1 \geq \lambda_2 \geq ...\lambda_n$ be the eigenvalues and $v_1, v_2, ...v_n$ be the corresponding eigenvectors. We denote the component wise sum of the eigenvectors by $s_1, s_2, s_3, ..., s_n$, i.e., s_i denotes the component-wise sum of the eigenvector v_i.

Theorem 3 $C^A_{i,j} = \left(\frac{n\left(\sum_{t=1}^n \lambda_t^{i+j} s_t^2\right)}{\left(\sum_{t=1}^n \lambda_t^i s_t^2\right)\left(\sum_{t=1}^n \lambda_t^j s_t^2\right)} \right) - 1.$ □

Theorem 3 illustrates that the representation C^A is tightly linked with the spectrum of adjacency matrix A, which is an important characteristic of the given graph. It is further known that the counts of various small local substructures contained in the graph such as the number of triangles, number of small paths, etc., are also important features [23] for a given graph. We next show that the matrix C^A is also sensitive to these counts of various local sub-structures.

Theorem 4 *Given the adjacency matrix A of an undirected graph with n nodes and m edges:*

$$C^A_{1,2} = \frac{n}{2m} \left(\frac{3\Delta + P_3 + n(Var(deg)) + m\left(\frac{4m}{n} - 1\right)}{(P_2 + m)} \right) - 1$$

where Δ denotes the total number of triangles, P_3 is the total number of distinct paths of length 3, P_2 is the total number of distinct paths of length 2 and $Var(deg) = \frac{1}{n}\sum_{i=1}^n deg(i)^2 - \left(\frac{1}{n}\sum_{i=1}^n deg(i)\right)^2$ is the variance of degree.

The above two Theorems tell that our proposed representation C^A encodes crucial information for discriminating network structures. Theorem 2 says that this object is a graph invariant and a covariance matrix in a fixed dimensional space. Thus, C^A is directly comparable between different graph structures, as will be supported by our experiments.

4 Computing Similarity between Graphs

We have argued that the matrix C^A captures critical information of the underlying graph. Given a fixed k, we have a representation for graphs in a common mathematical space, the space of symmetric positive semidefinite matrices $\mathbb{S}_{k\times k}$, whose mathematical properties are well understood. In particular, there are standard notions of similarity between such matrices.

The fact that matrix C^A is actually a covariance matrix motivates us to define the similarity between two graphs, with adjacency matrices $A \in \mathbb{R}^{n_1 \times n_1}$ and $B \in \mathbb{R}^{n_2 \times n_2}$ respectively, as the Bhattacharya similarity between two Gaussian distributions with zero means and covariance matrices C^A and C^B respectively. This similarity can be computed as follows

$$Sim(C^A, C^B) = e^{-\frac{1}{2}\log\left(\frac{det(\Sigma)}{\sqrt{(det(C^A)det(C^B))}}\right)} \tag{1}$$

Here, $det()$ is the determinant and $\Sigma = \frac{C^A + C^B}{2}$. Note that $C^A \in \mathbb{R}^{k\times k}$ and $C^B \in \mathbb{R}^{k\times k}$ are computed using the same value of k.

Theorem 5 *The similarity $Sim(C^A, C^B)$, defined in (1), between graphs with adjacency matrix A and B is positive semidefinite and a valid kernel.* □

Thus, the similarity function defined in this section is a valid kernel [11] and hence can be directly used in existing machine learning algorithms operating over kernels such as SVMs. We will see performance of this kernel on social network classification task later in Section 5.

4.1 Computation Complexity

For a fixed k, computing the set of vectors $\{Ae, A^2e, A^3e, ..., A^ke\}$ recursively as in Algorithm 1 has computation complexity of $O(mk)$. Note that the number of non-zeros in matrix A is $2m$ and each operation inside the for-loop is a sparse matrix vector multiplication, which has complexity $O(m)$. Computing the covariance matrix C^A requires summation of n outer products of vectors of dimension k, which has complexity $O(nk^2)$. Computing Eq. (1) involves the determinants of $k \times k$ matrices, which requires $O(k^3)$. Thus, the complexity for computing similarity is $O(mk+nk^2+k^3)$.

The value of k should be small like 4 or 5, because power iteration converges quickly and large values of k will make the C^A representation singular. Thus, the total time complexity of computing the similarity between two graphs reduces to $O(m + n) = O(m)$ (as usually $m \geq n$).

5 Evaluations on Real Social Network Classification

5.1 Task and Data

The task is to classify the research area of an individual taking into account the information contained in his/her ego collaboration network. We used three public collaboration network datasets [15] (`http://snap.stanford.edu/data/`): 1) High energy physics collaboration network (ca-HepPh.html), 2) Condensed matter physics collaboration network (ca-CondMat.html), and 3) Astro physics collaboration network (ca-AstroPh.html). These networks are generated from e-print arXiv and cover scientific collaborations between authors papers submitted to the respective categories. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by p authors, this generates a completely connected (sub)graph on p nodes.

To generate meaningful ego-networks from each of these huge collaboration networks, we select different users who have collaborated with more than 50 researchers and extract their ego networks. The ego network is just the subgraph containing the selected node and all its neighbors. We randomly choose 1000 such users from each of the high energy physics collaboration network and the astro physics collaboration network. In case of condensed matter physics, the collaboration network only had 415 individuals with more than 50 neighbors and so for this domain we take all of them.

Table 1: Graph statistics of ego-networks used in the paper.

STATS	High Energy	Condensed Matter	Astro Physics
No of Graphs	1000	415	1000
Mean No of Nodes	131.95	73.87	87.40
Mean No of Edges	8644.53	410.20	1305.00
Mean Clustering Coefficient	0.95	0.86	0.85

We have 2415 undirected ego network structures in total. The basic statistics of these ego networks is summarized in Table 1. We label each of the graph according to which of the three collaboration network it belongs to. Thus, our classification task is to take a researcher and his/her ego collaboration network and determine whether he/she belongs to high energy physics group, condensed matter physics group, or Astro physics group. This is a specific version of a general problem that arises in social media: "how audiences differ with respect to their social graph structure ?" [1].

For a better insight, we break the problem into 4 classification tasks: 1) high energy physics v.s. condensed matter physics (HEP Vs CM), 2) high energy physics v.s. astrophysics (HEP Vs ASTRO), 3) astrophysics v.s. condensed matter physics (ASTRO Vs CM), and 4) all the three domains (Full).

5.2 Competing Methodologies

We run the standard Kernel C-SVMs [3] on the data all pairwise similarity for classification. We evaluate the following five measures for computing similarity between two given graphs.

The Proposed Similarity (PROP): Given two graphs, we compute the similarity between them using Eq. (1). Instead of tuning k individually for each of the tasks, for easy replication of results we used 3 fixed values of $k = \{4, 5, 6\}$ for all of them, denoted by PROP-4, PROP-5 and PROP-6.

4-Subgraph Frequency(FREQ-4): Following [23], for each of the graphs we first generate a feature vector of normalized frequency of subgraphs of size four. It is well-known that the subgraph frequencies of arbitrarily large graphs can be accurately approximated by sampling a small number of induced sub-graphs. In line with the recent work, we computed such a histogram by sampling 1000 random subgraphs over 4 nodes. This process generates a normalized histograms of dimension 11 for each graph since there are 11 non-isomorphic different graphs with 4 nodes [23]. The similarity between two graphs is the inner product between the corresponding feature vectors.

5-Subgraph Frequency (FREQ-5): Recent success of counting induced subgraphs of size 4 in the domain of social networks leads to a natural curiosity "whether counting all subgraphs of size 5 improves the accuracy values over only subgraphs of size 4 ?". To answer this question, we also consider the histogram of normalized frequency of subgraphs of size 5. Similar to the case of FREQ-4, we sample 1000 random induced subgraphs of size 5 to generate a histogram representation. There are 34 non-isomorphic different graphs on 5 nodes and hence we obtain a vector of dimension 34.

3-Subgraph Frequency (FREQ-3): To understand the importance of size 4 subgraphs, we also compare with the histogram representation based on frequencies of subgraphs of size 3. There are 4

non-isomorphic graphs with 3 nodes and hence here we generate a histogram of dimension 4. Since this is a cheaper task, we use all the size 3 subgraphs instead of sampling only few of them. Again the similarity value is the inner product between the corresponding vectors.

Random Walk Similarity (RW): Random walk similarity is a widely used similarity measure over graphs. This similarity is based on a simple idea: given a pair of graphs, perform random walks on both, and count the number of similar walks (see [24]). There is a rich set of literature regarding connections of this similarity with well-known similarity measures in different domains such as Binet-Cauchy Kernels for ARMA models [25], rational kernels [5], r-convolution kernels [8]. The random walk similarity [25] between two graphs with adjacency matrix A and B is defined as $RWSim(A,B) = \frac{1}{n_1 n_2} e^T M e$, where M is the solution of Sylvester equation $M = (A^T M B) exp^{-\lambda} + ee^T$. This can be computed in closed forms efficiently in $O(n^3)$ time. We use standard recommendations for the value of λ.

Table 2: Prediction accuracy in percentage for proposed and the state-of-the-art similarity measures on different ego network classification tasks. The reported results are averaged over 10 repetitions of 10-fold cross-validation. Standard errors are indicated using parentheses. Best results in bold

Methodology	COLLAB (HEnP Vs CM)	COLLAB (HEnP Vs ASTRO)	COLLAB (ASTRO Vs CM)	COLLAB (Full)
PROP-4	98.06(0.05)	**87.70(0.13)**	89.29(0.18)	82.94(0.16)
PROP-5	**98.22(0.06)**	87.47(0.04)	89.26(0.17)	**83.56(0.12)**
PROP-6	97.51(0.04)	82.07(0.06)	**89.65(0.09)**	82.87(0.11)
FREQ-5	96.97 (0.04)	85.61(0.1)	88.04(0.14)	81.50(0.08)
FREQ-4	97.16 (0.05)	82.78(0.06)	86.93(0.12)	78.55(0.08)
FREQ-3	96.38 (0.03)	80.35(0.06)	82.98(0.12)	73.42(0.13)
RW	96.12 (0.07)	80.43(0.14)	85.68(0.03)	75.64(0.09)

5.3 Evaluations and Results

The evaluations consists of running kernel SVM on all the tasks using five different kernels (similarity measures) as described. The evaluation procedure is the standard cross validation estimation of classification accuracy. First, we split each dataset into 10 folds of identical size. We then combine 9 of these folds and again split it into 10 parts, then use the first 9 parts to train the kernel C-SVM [3] and use the 10th part as validation set to find the best performing value of C from $\{10^{-7}, 10^{-6}, ..., 10^{7}\}$. With this fixed choice of C, we train the C-SVM on all the 9 folds and predict on the 10th fold acting as an independent evaluation set. The procedure is repeated 10 times with each fold acting as an independent test set once. For each dataset the procedure is repeated 10 times randomizing over partitions. The mean classification accuracy and the standard errors are shown in Table 2. Note that we did not tune any parameter other than the "C" for kernel SVM.

In all of the tasks, similarity measure based on the proposed representation outperforms all the other competing state-of-the-art measures, most of the time with a significant margin. This clearly demonstrates that the covariance matrix representation captures sufficient information about the ego networks and is capable of discriminating between them. The accuracy for three different values of k are not very much different form each other, except in some cases with $k = 6$, which shows that slight variations in k do not have any significant change in the performance. Ideally, k can be tuned based on the dataset, but for easy replication of results we used 3 fixed choices of k.

As expected, counting subgraphs of size 4 (FREQ 4) always improve significantly over just counting subgraphs of size 3 (SUBGREQ-3). This is in line with the recent work [23] which shows the effectiveness of counting subgraphs of size 4. Interestingly, counting subgraphs of size 5 (FREQ-5) improves significantly over FREQ-4 on all tasks, except for HEnP Vs CM classification. This clearly shows the sub-optimality of histogram obtained by counting very small graphs ($k \leq 4$). Even with sampling, FREQ-5 is an order of magnitude expensive than other methodologies. Unfortunately, as we will see in the next section, with increasing k, we loose the computational tractability of counting induced k-subgraphs in the given graph (even with sampling).

Our covariance methodology consistently performs better than (FREQ 5). This clearly demonstrates the superiority of the C^A representation. As argued in Section 3, the matrix C^A even for $k = 4$ or 5, does incorporate information regarding the counts of bigger complex sub-structures in the graph.

This along with the information of the full spectrum of the adjacency matrix leads to a very sound representation which outperforms state-of-the-art similarity measures over graphs.

6 Run-Time Comparisons

To obtain an estimate of the computational costs of various similarity methods, we compare the times required to compute the similarity values between two given graphs using different methodologies as presented in Section 5.2. We record the cpu time taken for computing similarity between all possible pairs of graphs. As summarized in Table 3, All experiments were performed in MATLAB on an Intel(R) Xenon 3.2 Ghz CPU machine having 72 GB of RAM.

It can be seen that other than the FREQ-5 and RW all other methods are quite competitive. It is not surprising that RW kernels are slow since they have cubic runtime complexity. Although computing histogram based on counting all the subgraphs of size 4 is much more computationally expensive than counting subgraphs of size 3, approximating the histogram by sampling is fairly efficient.

Table 3: Time (in sec) required for computing all pairwise similarity

Prop-4	Prop-5	Prop-6	Freq-3(All)	Freq-4(1000 samp)	Freq5 (1000 samp)	RW
260.56	276.56	286.87	369.83	265.77	7433.41	25195.54

Even with sampling, FREQ-5 is an order of magnitude slower. To understand this phenomenon, we review the process of computing the histogram by counting subgraphs. There are 34 graph structures over 5 nodes unique up to isomorphism. Each of these 34 structures has $5! = 120$ many isomorphic variants (one for every permutation). To compute a histogram over these 34 structures, we first sample an induced 5-subgraph from the given graph. The next step is to match this subgraph to one of the 34 structures. This requires determining which of the 34 graphs is isomorphic with the given sampled subgraph. The process is repeated 1000 times for every sample. Thus every sampling step requires solving graph isomorphism problem. Even FREQ-4 has the same problem but there are only 11 possible subgraphs and the number of isomorphic structures for each graph is only $4! = 24$. The problem becomes intractable as we move beyond 5 as the graph isomorphism problem is combinatorially hard.

The proposed similarity based on C^A is significantly less expensive than FREQ-5 and at the same time performs better. Counting-based approaches do capture information but quickly loose tractability once we start counting bigger substructures. Power iterations of the adjacency matrix is a nice way of capturing information about the underlying graph and at the same time is computationally efficient. The biggest problem with these power iterations is that they are not directly comparable. The representation C^A removes this concern and makes power iterations of different adjacency matrix comparable in a common space.

7 Conclusion

We embed graphs into a new mathematical space of positive semidefinite matrices $\mathbb{S}_{k\times k}$. We take an altogether different approach for characterizing graphs based on the covariance matrix of vectors obtained from the power iteration of the adjacency matrix. Our analysis indicates that the proposed matrix representation C^A contains most of the important characteristic information about the networks structure. Since the C^A representation is a covariance matrix in a fixed dimensional space, it naturally provides a measure of similarity between different graphs. The procedure is simple and can be computed in time linear in number of edges, making our approach scalable in practice.

Our experimental evaluations demonstrate the superiority of the proposed C^A representation, over other state-of-the-art methodologies, in ego network classification tasks. The run-time comparisons indicate that the proposed approach provides the right balance between the expressiveness of representation and the computational tractability. Since finding tractable and meaningful representations of graph is a fundamental problem, we believe that our results will provide good motivation for using the new representation C^A in analyzing real networks.

Acknowledgement

The work is supported by NSF-SES-1131848, NSF-Bigdata-1250914, ONR-N00014-13-1-0764, and AFOSR-FA9550-13-1-0137.

References

[1] L. A. Adamic, J. Zhang, E. Bakshy, and M. S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *WWW*, pages 665–674. 2008.

[2] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer networks*, 33(1):309–320, 2000.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[4] H.-H. Chen, L. Gou, X. Zhang, and C. L. Giles. Collabseer: a search engine for collaboration discovery. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 231–240. 2011.

[5] C. Cortes, P. Haffner, and M. Mohri. Rational kernels. In *NIPS*, pages 601–608, 2002.

[6] A. Das Sarma, D. Nanongkai, G. Pandurangan, and P. Tetali. Distributed random walks. *Journal of the ACM*.

[7] I. J. Farkas, I. Derényi, A.-L. Barabási, and T. Vicsek. Spectra of real-world graphs: Beyond the semicircle law. *Physical Review E*, 64(2):026704, 2001.

[8] D. Haussler. Convolution kernels on discrete structures. Technical report, 1999.

[9] J. He and W. W. Chu. *A social network-based recommender system (SNRS)*. Springer, 2010.

[10] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *WWW*, pages 421–430. 2010.

[11] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, pages 1171–1220, 2008.

[12] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[13] A. N. Krylov. On the numerical solution of equations whose solution determine the frequency of small vibrations of material systems (in Russian (1931)).

[14] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Link Mining: Models, Algorithms, and Applications*, pages 337–357. Springer, 2010.

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[16] F. Lin and W. W. Cohen. Power iteration clustering. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 655–662, 2010.

[17] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, pages 548–556, 2012.

[18] G. D. F. Morales, A. Gionis, and C. Lucchese. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *WSDM*, pages 153–162, 2012.

[19] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[20] I. Paparrizos, B. B. Cambazoglu, and A. Gionis. Machine learned job recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 325–328, New York, 2011.

[21] N. Shervashidze, T. Petri, K. Mehlhorn, K. M. Borgwardt, and S. Viswanathan. Efficient graphlet kernels for large graph comparison. In *AISTATS*, pages 488–495, 2009.

[22] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[23] J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In *WWW*, pages 1307–1318, 2013.

[24] S. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 99:1201–1242, 2010.

[25] S. Vishwanathan and A. Smola. Binet-cauchy kernels. 2004.