A New Space for Comparing Graphs

Anshumali Shrivastava and Ping Li

Cornell University and Rutgers University

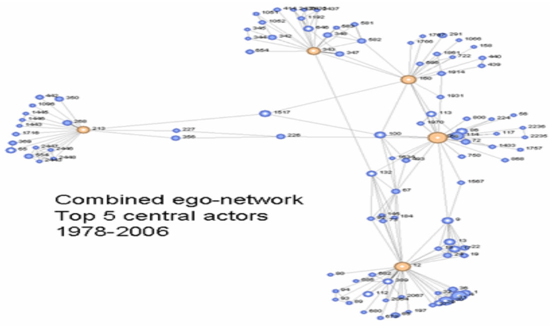
August 18th 2014

Main Contribution

A succinct and informative Covariance Matrix for graph structure.

Gains:

- Can be computed in O(E), scalable.
- Dealing with covariance matrix easier than graphs.
- Can directly compare different graph structures by simply comparing corresponding covariance matrices.
- Directly apply machine learning on matrices. No worry about graph and its combinatorial isomorphic variants.



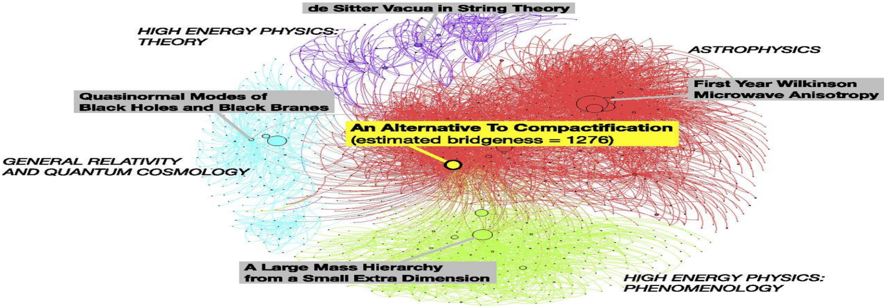
- The connectivity (or the presence of absence of edges) in various networks carries a altogether new set of valuable information.
- The local connectivity structure of an individual (or his/her **ego network**), can be used to infer many peculiarities about him/her.

Analyzing this network connectivity structure can lead to many interesting and useful applications.

Identify users across networks

Get ego networks, try to match it across networks ?

New Classifications Based on Ego Networks



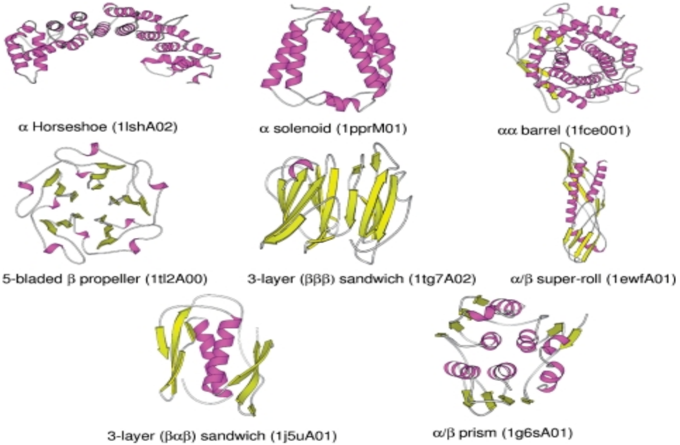
- The, collaboration pattern gets reflected in the ego network of an individual.
- High Energy Physics (HEnP) collaboration network is very dense. Dependence on specialized labs leads to more collaboration.

Can we classify a researcher purely on the basis of his/her collaboration ego network ?

YES !! (This work)

Gains: Personalized Recommendations

Chemical Compound/Activity Classification



Yes !! **(Available in a separate tech report)**
http://arxiv.org/abs/1404.5214

Many More Applications ....

- Synonym extraction using word graphs.
- Structure matching across databases.
- Structured text translation.
- Protein alignment.

The Underlying Fundamental Problem

What is the right common space (with a well defined metric) for graph structure.

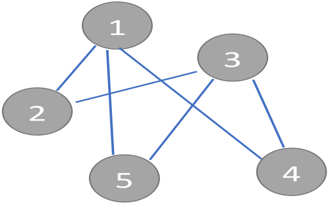
Challenges:

- Varying sizes.
- Node correspondence is usually not available.
- Same graph object exhibits many isomorphic forms.

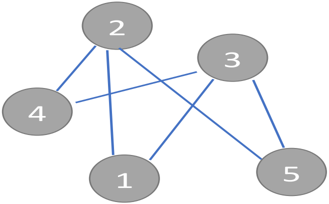
Succinct summarization of graphs is a wide open research direction.

Graph Representation : Permutation Invariance

A property that does not change with node renumbering.



$$\begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{vmatrix}$$



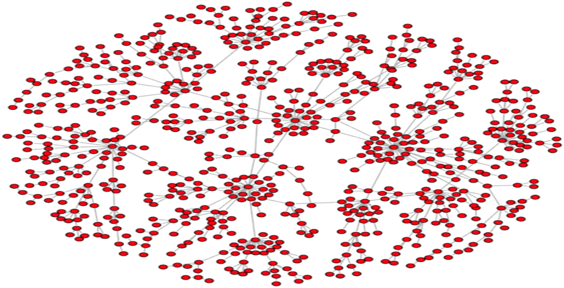
$$\begin{vmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{vmatrix}$$

Adjacency Matrices are not permutation invariant, they are not comparable.

Hardness

- Graph Isomorphism is a special case of graph comparison (checking equality).
- Graph Isomorphism is a hard problem. (Its belongingness in P or NP is still open)
- Its hopeless to have an efficient exact embedding for all possible graph.

Good News: Real World Graphs are Special



- Very specific spectrum.
- Has triadic closures and local clustering.
- The degree distribution follows power law.
- Lot of hubs.

We can hope to capture all of these in a succinct representation.

The Right Object for Studying Graphs

- Should be **Permutation invariant**.
- Should be sensitive to variations in the spectral properties.
- Should be sensitive to distributions of different substructures (or subgraphs).
- The last two are related, so its not clear what is the right balance.

Should be efficient to compute !!

Existing Approach 1

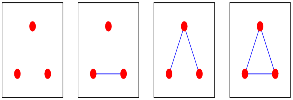
A normalized feature vector representation of various known graph invariants.

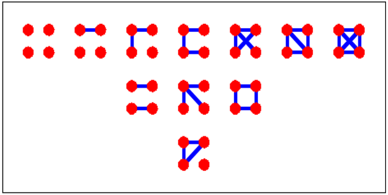
- Top eigenvalues of adjacency matrix.
- Clustering coefficient.
- Mean and Variance of Degrees, edges, etc.

Problems

- Graphs of different sizes have different number of eigenvalues.
- Not really clear if their values are directly comparable.
- How many graph invariants are enough ?
- What relative importance to give to different invariants ?(some characteristic might dominate others)

Existing Approach 2





- A histogram based on frequency of small graphs (graphlets) contained in the given graph.
- Usually frequency of small graphs of size 3-4 is used.
- The histogram can be efficiently and accurately estimated by sampling.

Problems

- Small graphs do not always capture enough structural information. We need frequency of larger graphs for richer information.
- Counting graphs of size ≥ 5 is very costly.
  - For every sampled subgraphs, we need to match it with one of the many isomorphic variants.
  - Every sampling step encode the graph isomorphism problem. (costly for large substructures)
  - Computation time increases exponentially with size of graph, even after sampling.

Alternative View: Power Iteration of Adjacency Matrix

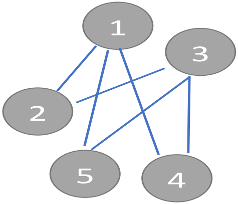
Power iteration is a cheap and effective way of summarizing any matrix.

- View adjacency matrix A as a dynamical operator (function) operating on vectors.
- If two operators A and B are "similar" then vectors $\{Ax, A^2x, ...A^kx\}$ should be "similar" to $\{Bx, B^2x, ...B^kx\}$
- The subspace $\{Ax, A^2x, ...A^kx\}$ is a well studied object known as **k-th order "Krylov" subspace.**
- "Krylov" subspace based methods are to some of the fastest known linear algebraic algorithms for sparse matrices.

Problem: "Krylov" subspace are **not permutation invariant** in general.

Summarization with Power Iteration on Unit Vector

Start with vector as e, the vector of all $1's$.
Given adjacency matrix A, the generated subspace will be
$\{Ae, A^2e, A^3e, ...\}$

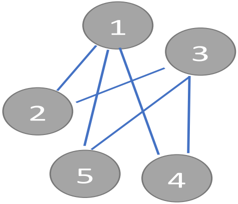


$$M_A = \begin{vmatrix} (Ae)_1 & (A^2e)_1 & (A^3e)_1 & (A^4e)_1 & (A^5e)_1 \\ (Ae)_2 & (A^2e)_2 & (A^3e)_2 & (A^4e)_2 & (A^5e)_2 \\ (Ae)_3 & (A^2e)_3 & (A^3e)_3 & (A^4e)_3 & (A^5e)_3 \\ (Ae)_4 & (A^2e)_4 & (A^3e)_4 & (A^4e)_4 & (A^5e)_4 \\ (Ae)_5 & (A^2e)_5 & (A^3e)_5 & (A^4e)_5 & (A^5e)_5 \end{vmatrix}$$

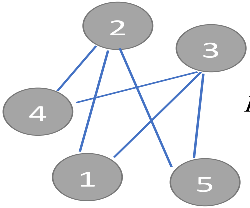
Here $(Ae)_i$ is the i^{th} component of vector Ae.
Truncated power iteration are very informative.
Power iteration over unit vector is key ingredient in many web algorithms including the famous HITS.

Observation: Power Iteration on Unit Vector is Special

If A and B are the adjacency matrices of same graph under reordering, then rows of M_B is simply row shuffled version of M_A.

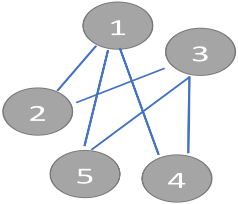


$$M_A = \begin{vmatrix} (Ae)_1 & (A^2e)_1 & (A^3e)_1 & (A^4e)_1 & (A^5e)_1 \\ (Ae)_2 & (A^2e)_2 & (A^3e)_2 & (A^4e)_2 & (A^5e)_2 \\ (Ae)_3 & (A^2e)_3 & (A^3e)_3 & (A^4e)_3 & (A^5e)_3 \\ (Ae)_4 & (A^2e)_4 & (A^3e)_4 & (A^4e)_4 & (A^5e)_4 \\ (Ae)_5 & (A^2e)_5 & (A^3e)_5 & (A^4e)_5 & (A^5e)_5 \end{vmatrix}$$



$$M_B = \begin{vmatrix} (Be)_1 & (B^2e)_1 & (B^3e)_1 & (B^4e)_1 & (B^5e)_1 \\ (\mathrm{Be})_2 & (B^2e)_2 & (B^3e)_2 & (B^4e)_2 & (B^5e)_2 \\ (Be)_3 & (B^2e)_3 & (B^3e)_3 & (B^4e)_3 & (B^5e)_3 \\ (Be)_4 & (B^2e)_4 & (B^3e)_4 & (B^4e)_4 & (B^5e)_4 \\ (Be)_5 & (B^2e)_5 & (B^3e)_5 & (B^4e)_5 & (B^5e)_5 \end{vmatrix}$$

Graphs as Bag-of-Vectors



$$M_A = \begin{vmatrix} (Ae)_1 & (A^2e)_1 & (A^3e)_1 & (A^4e)_1 & (A^5e)_1 \\ (Ae)_2 & (A^2e)_2 & (A^3e)_2 & (A^4e)_2 & (A^5e)_2 \\ (Ae)_3 & (A^2e)_3 & (A^3e)_3 & (A^4e)_3 & (A^5e)_3 \\ (Ae)_4 & (A^2e)_4 & (A^3e)_4 & (A^4e)_4 & (A^5e)_4 \\ (Ae)_5 & (A^2e)_5 & (A^3e)_5 & (A^4e)_5 & (A^5e)_5 \end{vmatrix}$$

- We can associate a set of n vectors, corresponding to rows of M_A, with a graph having n nodes.
- Reordering of nodes does not change this set. (It simply permutes them)
- The dimension k of these vectors (no of columns) is the no of power iteration performed.

We are looking for an object that richly **describes a set of vectors.**

What Describes a Set (Bag) of Vectors ?

Set of vectors easier to deal with than graphs.

The cardinality of set n can vary.

Two objects

- Subspace spanned by n vectors. (bad choice as $n \gg k$)
- Most likely probability distribution generating these vectors. (Fit the most likely Gaussians)

Key component of M.L.E multivariate Gaussian over n samples, **"The Covariance Matrix".**

Our Proposal: The Covariance Matrix Representation

We propose $C^A \in \mathbb{R}^{k\times k}$, **the covariance matrix of M_A** as a representation for graph with adjacency matrix A.

Input: Adjacency matrix $A \in \mathbb{R}^{n\times n}$, k, no 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 $C^A \in \mathbb{R}^{k\times k}$

Whats Nice ?: For a fixed k, all graphs (irrespective of size) represented in a common space of $\mathbb{R}^{k\times k}$ p.s.d Covariance Matrix.

Property 1

Theorem

C^A is a graph invariant.

Proof Idea: The covariance matrix is independent of the ordering of rows.

Implications: C^A **can be used as a representation for graph.** Covariance matrix is a well studied object which is easier to handle than graphs.

Property 2

Theorem

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

where λ_t is the t^{th} eigenvalue and s_t is the component wise sum of the t^{th} eigenvector.

Proof Idea: The mean of vector $A^i e$ can be written as $\frac{[e^T A^i e]}{n}$. Some algebra $C^A_{i,j} = \left(n \frac{[e^T A^{i+j} e]}{[e^T A^i e][e^T A^j e]} \right) - 1$, use representation of e in the eigenbasis of A to complete the proof.

Implications: C^A **encodes the spectrum.** Components of matrix C^A (weighted and exponentiated) combination of all $\lambda_t's$ and $s_t's$.

Property 3

Theorem

Given the adjacency matrix A of an undirected graph with n nodes and m edges, we have

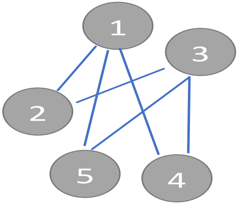
$$C^A_{1,2} = \frac{n}{2m}\left(\frac{3\Delta + P_3 + n(Var(deg)) + m\left(\frac{4m}{n} - 1\right)}{(P_2 + m)}\right) - 1$$

where Δ denotes the total number of triangles, P_3 is the total number of distinct simple paths of length 3, P_2 is the total number of distinct simple 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.

Proof Idea: We get $C^A_{i,j} = \left(n\frac{[e^TA^3e]}{[e^TA^1e][e^TA^2e]}\right) - 1$. Terms of the form $[e^TA^ie]$ is the sum of total number of paths of length i although with lot of repetition. Careful counting leads to the above expression.

Implications:Components of C^A is sensitive to counts of substructures.

How many Iterations ?



$$M_A = \begin{vmatrix} (Ae)_1 & (A^2e)_1 & (A^3e)_1 & (A^4e)_1 & (A^5e)_1 \\ (Ae)_2 & (A^2e)_2 & (A^3e)_2 & (A^4e)_2 & (A^5e)_2 \\ (Ae)_3 & (A^2e)_3 & (A^3e)_3 & (A^4e)_3 & (A^5e)_3 \\ (Ae)_4 & (A^2e)_4 & (A^3e)_4 & (A^4e)_4 & (A^5e)_4 \\ (Ae)_5 & (A^2e)_5 & (A^3e)_5 & (A^4e)_5 & (A^5e)_5 \end{vmatrix}$$

k is the number of power iteration, or the number of columns in M_A.

- Power iteration converges to the dominant eigenvector geometrically.
- Near convergence the new columns are uninformative.
- We only need **very few iterations, like 4 or 5.**

Similarity between Graphs

We compare the corresponding covariance matrices. We use standard **Bhattacharrya similarity** between $C_A \in \mathbb{R}^{k \times k}$ and $C_B \in \mathbb{R}^{k \times k}$.

$$Sim(C^A, C^B) = exp^{-Dist(C^A, C^B)}$$

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

$$\Sigma = \frac{C^A + C^B}{2}$$

Theorem

$Sim(C^A, C^B)$ is a positive semi-definite (hence a valid kernel).

Note: Covariance matrix has special properties (e.g. symmetric), so the similarity measure should respect that structure.

Computation Time

- Given a choice of k, computing the set of vectors $\{Ae, A^2e, A^3e, ..., A^ke\}$ recursively is $O(E * k)$. (**A is sparse !!**)
- Computing the covariance matrix C^A is $O(nk^2)$.
- Computing similarity is $O(k^3)$.

Usually, we need very small k like 4 or 5. Hence, the overall complexity is $O(E)$.

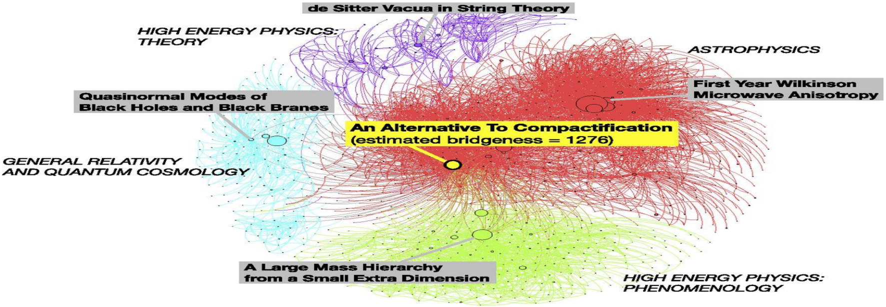
Evaluation Tasks: How Good is this Representation ?

We test the effectiveness on two **graph classification task**.

- **Classifying researcher's subfield based on his/her ego network structure**
- **Discriminating random Erdos-Reyni graphs from real graphs**

A good representation (or similarity measure) should have better discriminative power.

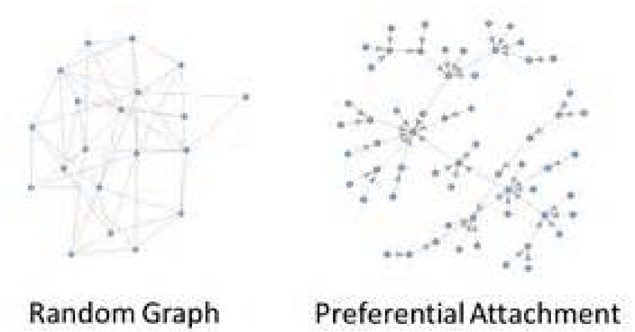
Task 1



Classify Researcher's Subfield :

- We take three publicly available collaboration network dataset:
  1. High energy physics (HEnP)
  2. Condensed matter physics (CM)
  3. Astro physics (ASTRO)
- Sample ego networks from them to generate a dataset of graphs.
- Given a researcher's ego collaboration network, determine whether he/she belongs to HenP, CM, or ASTRO.

Task 2



Classify Random Vs Social:

- Discriminate random Erdos-Reyni graphs from Twitter ego networks.
- For every twitter ego network Erdos-Reyni graph is generated with same number of nodes and edges.

A good similarity measure should be able to discriminate between graphs following different distributions.

Data Statistics

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

STATS	High Energy	Condensed Matter	Astro Physics	Twitter	Random
Number of Graphs	1000	415	1000	973	973
Mean Number of Nodes	131.95	73.87	87.40	137.57	137.57
Mean Number of Edges	8644.53	410.20	1305.00	1709.20	1709.20
Mean Clustering Coefficient	0.95	0.86	0.85	0.55	0.18

Competing Methodologies

- Proposed similarity based on covariance matrix. We report results for ($k = 4,5,6$). No tuning.
- Subgraph frequency histogram with graphs of size 3,4, and 5. Going beyond 5 is way too costly.
- Random Walk kernels.
- Feature vector of eigenvalues. (Use Top-5 and Top-10 eigenvectors)

Experimental Details

- We run **kernel SVMs,** on the similarity values computed from competing representations.
- Generate 10 partition, use 9 for train and cross-validate for svm parameter C, 10th part for testing.
- Each experiment repeated 10 times randomizing over partitions.

We report classification accuracy and time required to compute similarity.

Classification Accuracy

Methods	COLLAB (HEnP Vs CM)	COLLAB (HEnP Vs ASTRO)	COLLAB (ASTRO Vs CM)	COLLAB (Full)	SOCIAL (Twit- ter Vs Random)
Our(k =4)	98.06(0.05)	**87.70(0.13)**	89.29(0.18)	82.94(0.16)	99.18(0.03)
Our(k =5)	**98.22(0.06)**	87.47(0.04)	89.26(0.17)	**83.56(0.12)**	99.43(0.02)
Our(k =6)	97.51(0.04)	82.07(0.06)	**89.65(0.09)**	82.87(0.11)	**99.48(0.03)**
FREQ-5	96.97(0.04)	85.61(0.1)	88.04(0.14)	81.50(0.08)	99.42(0.03)
FREQ-4	97.16(0.05)	82.78(0.06)	86.93(0.12)	78.55(0.08)	98.30(0.08)
FREQ-3	96.38(0.03)	80.35(0.06)	82.98(0.12)	73.42(0.13)	89.70(0.04)
RW	96.12(0.07)	80.43(0.14)	85.68(0.03)	75.64(0.09)	90.23(0.06)
EIGS-5	94.85(0.18)	77.69(0.24)	83.16(0.47)	72.02(0.25)	90.74(0.22)
EIGS-10	96.92(0.21)	78.15(0.17)	84.60(0.27)	72.93(0.19)	92.71(0.15)

Running Time Comparisons

Table: Time (in sec) required for computing all pairwise similarities of the two datasets.

	SOCIAL	COLLAB (Full)
Total Number of Graphs	1946	2415
Our (k =4)	177.20	260.56
Our (k =5)	200.28	276.77
Our (k =6)	207.20	286.87
FREQ-5 (1000 Samp)	5678.67	7433.41
FREQ-4 (1000 Samp)	193.39	265.77
FREQ-3 (All)	115.58	369.83
RW	19669.24	25195.54
EIGS-5	36.84	26.03
EIGS-10	41.15	29.46

Lessons

- Simply computing many graph invariant does not give the right representation.
  1. Issue of relative importance.
  2. Never know how many are enough.
  3. One graph usually has more invariants than others.
- Histogram of subgraphs is a good representation but very costly when computing for subgraphs of size ≥ 5.
- Power iteration is a very cheap way of summarizing graphs which caputres information of various substructures.
- Finding right representation is the key in machine learning with graphs.

Thanks!!