Joint Embedding of Graphs

Shangsi Wang

Johns Hopkins University

February 26, 2017

Motivation

Feature extraction and dimension reduction for networks is critical in a wide variety of domains. There are a few unsupervised feature methods available: Principal Component Analysis, Graph Statistics, Graph Spectral Statistics, Laplacian Engenmap and Adjacency Spectral Embedding. We want a method which

- Take advantage of multiple graphs,
- Utilize low rank property of adjacency matrix,
- Able to apply to sparse large graphs,
- Lead to good inference performance,
- With theoretical support.

Motivate by these properties, we propose a method to jointly embed multiple undirected graphs.

Random Dot Product Graph

Let F be a distribution on a set $\mathcal{X} \in \mathbb{R}^d$, and $\mathbf{X} = [x_1^T, x_2^T, ..., x_n^T] \in \mathcal{X}^n$. The notation is $(\mathbf{X}, \mathbf{A}) \sim RDPG(F)$, if

$$x_1, x_2, ..., x_n \overset{i.i.d.}{\sim} F,$$

$$\mathbf{A}_{st}|\mathbf{X} \sim Bernoulli(x_s^T x_t).$$

Alternatively,

$$P(\mathbf{A}|\mathbf{X}) = \prod_{s<t} (x_s^T x_t)^{\mathbf{A}_{st}} (1 - x_s^T x_t)^{1-\mathbf{A}_{st}}.$$

When the latent positions $\mathbf{X}$ is regarded as parameter, the notation becomes $\mathbf{A} \sim RDPG(\mathbf{X})$.

Adjacency Spectral Embedding

For one graph, d-dimensional Adjacency Spectral Embedding (ASE) approximates $\mathbf{A}$ by the product of rank d matrices.

$$\mathbf{A} \approx \hat{\mathbf{V}}\hat{\mathbf{D}}\hat{\mathbf{V}}^T = \sum_{k=1}^{d} \hat{\mathbf{D}}_{kk}\hat{v}_k\hat{v}_k^T \text{ , or } \min \|\mathbf{A} - \hat{\mathbf{V}}\hat{\mathbf{D}}\hat{\mathbf{V}}\| \qquad (1)$$

Let the embedding $\hat{\mathbf{X}} = \hat{\mathbf{V}}\hat{\mathbf{D}}^{\frac{1}{2}}$ which can be understand as the features of vertices or an estimator for $\mathbf{X}$ under RDPG model. The distance between two embeddings,

$$d(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2) = \min_{\mathbf{Q}, \mathbf{Q}^T\mathbf{Q}=I} \|\hat{\mathbf{X}}_1\mathbf{Q} - \hat{\mathbf{X}}_2\|$$

Can we embed multiple graphs together?

Multiple Random Eigen Graph

Let $\{h_k\}_{k=1}^d$ be a set of norm-1 vectors in $\mathbb{R}^n$, and F be a distribution on a set $\mathcal{X} \in \mathbb{R}^d$. The m pairs $\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^m$ follow a d-dimensional multiple random eigen graphs model, and the notation is $\{(\lambda_i, \mathbf{A}_i)\}_{i=1}^m \sim MREG(F, h_1, ..., h_d)$, if

$$\{\lambda_i\}_{i=1}^m \overset{i.i.d.}{\sim} F,$$

$$\mathbf{A}_i[s,t]|\lambda_i \sim Bernoulli(\sum_{k=1}^d \lambda_i[k] h_k[s] h_k[t]).$$

In cases that $\{\lambda_i\}_{i=1}^m$ are of primary interest,

$$\{\mathbf{A}_i\}_{i=1}^m \sim MREG(\lambda_1, ..., \lambda_m, h_1, ..., h_d).$$

Joint Embedding of Graphs

Given adjacency matrices $\{A_i\}_{i=1}^m$, d-dimensional joint embedding of graphs is defined to be

$$(\hat{\lambda}_1, ..., \hat{\lambda}_m, \hat{h}_1, ..., \hat{h}_d) = \underset{\lambda_i, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \lambda_i[k] h_k h_k^T\|^2. \quad (2)$$

Here, $\hat{h}_k$ is shared across graphs and is considered to be latent positions of vertices; $\hat{\lambda}_i$ is called loadings of graph i and will be treated as features for graphs. Let $\hat{\mathbf{H}} = [\hat{h}_1, ..., \hat{h}_d]$ and $\hat{\mathbf{\Lambda}} = [\hat{\lambda}_1, ..., \hat{\lambda}_m]^T$.

Other Formulations

The problem can be formulated in different ways

$$\begin{aligned} \underset{\mathbf{D}_i, \|h_k\|=1}{\operatorname{argmin}} \quad & \sum_{i=1}^{m} \|\mathbf{A}_i - \mathbf{H}\mathbf{D}_i\mathbf{H}^T\|^2 \\ \text{subject to} \quad & \mathbf{D}_i \text{ being diagonal.} \end{aligned}$$

Alternatively, if $\{\mathbf{A}_i\}_{i=1}^m$ are stacked in a 3-D array $\mathbb{A} \in \mathbb{R}^{m \times n \times n}$,

$$\underset{\mathbf{\Lambda}, \|h_k\|=1}{\operatorname{argmin}} \|\mathbb{A} - \sum_{k=1}^{d} \mathbf{\Lambda}_{*k} \otimes h_k \otimes h_k\|^2.$$

Relationship to Other Factorization Methods

$$(\hat{\lambda}_1, ..., \hat{\lambda}_m, \hat{h}_1, ..., \hat{h}_d) = \underset{\lambda_i, \|h_k\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \lambda_i[k] h_k h_k^T\|^2.$$

- If $h_k h_k^T$ is replaced by $\mathbf{S}_k$, the problem is equivalent to principal component analysis.
- If $h_k h_k^T$ is replaced by $h_k g_k^T$, the problem is equivalent to tensor rank decomposition or canonical polyadic decomposition.
- If $\mathbf{D}_i$ is not constrained to be diagonal and with more constrains, the problem becomes higher order SVD (Tucker Decomposition).

Greedy Alternating Minimization

We consider an algorithm which solves the problem iteratively. Specifically, at iteration k_0,

$$\underset{\mathbf{\Lambda}_{*k_0}, \|h_{k_0}\|=1}{\operatorname{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{k_0-1} \hat{\mathbf{\Lambda}}_{ik} \hat{h}_k \hat{h}_k^T - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T\|^2.$$

Let $\mathbf{R}_{ik_0} = \mathbf{A}_i - \sum_{k=1}^{k_0-1} \hat{\mathbf{\Lambda}}_{ik} \hat{h}_k \hat{h}_k^T$. The gradients are,

$$\frac{\partial f}{\partial h_{k_0}} = -4 \sum_{i=1}^{m} \mathbf{\Lambda}_{ik_0} (\mathbf{R}_{ik} - \mathbf{\Lambda}_{ik_0} h_{k_0} h_{k_0}^T) h_{k_0}. \tag{3}$$

$$\hat{\mathbf{\Lambda}}_{ik_0} = \langle \mathbf{R}_{ik}, h_{k_0} h_{k_0}^T \rangle, \tag{4}$$

Optimization Algorithm

procedure FIND JOINT EMBEDDING $\hat{\mathbf{\Lambda}}, \hat{\mathbf{H}}$ OF $\{\mathbf{A}_i\}_{i=1}^m$

  Set residuals: $\mathbf{R}_{i1} = \mathbf{A}_i$

  for $k = 1 : d$ **do**

    Initialize h_k and $\mathbf{\Lambda}_{*k}$

    while not convergent **do**

      Fixing $\mathbf{\Lambda}_{*k}$, update h_k by gradient descent (3)

      Project h_k back to the unit sphere

      Fixing h_k, update $\mathbf{\Lambda}_{*k}$ by (4)

      Compute objective $\sum_{i=1}^{m} \|\mathbf{R}_{ik} - \mathbf{\Lambda}_{ik} h_k h_k^T\|^2$

    end while

    Update residuals: $\mathbf{R}_{i(k+1)} = \mathbf{R}_{ik} - \mathbf{\Lambda}_{ik} h_k h_k^T$

  end for

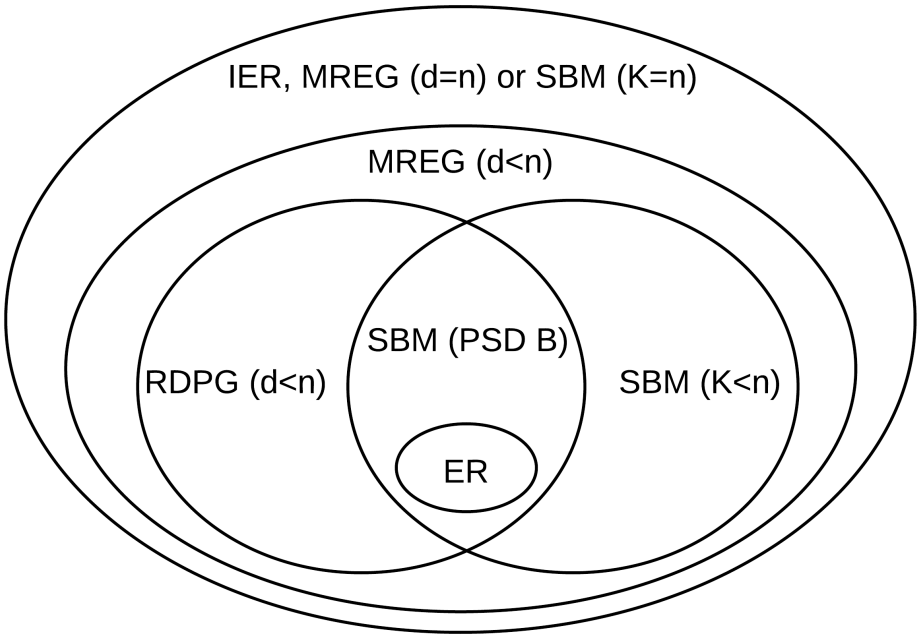
  Output $\hat{\mathbf{\Lambda}} = [\mathbf{\Lambda}_{*1}, ..., \mathbf{\Lambda}_{*d}]$ and $\hat{\mathbf{H}} = [h_1, ..., h_d]$

end procedure

Theory

Theorem

Given any distribution $\mathcal{F}$ on binary graphs and a random adjacency matrix $\mathbf{A} \sim \mathcal{F}$, there exists a dimension d, a distribution F on $\mathbb{R}^d$, and a set of vectors $\{h_k\}_{k=1}^d$, such that $\mathbf{A} \sim MREG(F, h_1, ..., h_d)$.



Theory

Let $\rho(\mathbf{A}_i, h) = \|\mathbf{A}_i - \langle \mathbf{A}_i, hh^T \rangle hh^T\|^2$ and $D(h, h_1) = E(\rho(\mathbf{A}_i, h))$. Under 1-dimensional MREG model, we can show

Theorem

If $D(h, h_1)$ has a unique global minimum at h', then $\hat{h}_1^m$ converges almost surely to h' as m goes to infinity. That is,

$$\hat{h}_1^m \xrightarrow{p} h'.$$

Theorem

If h' is a minimizer of $D(h, h_1)$, then

$$\|h' - h_1\| \leq \frac{2E(\lambda_i)}{E(\lambda_i^2)(h_1^T h')^2}.$$

Experiment 1: Setup

We generate 200 graphs from 2 dimensional MREG $(\lambda_i, \mathbf{A}_i)\}_{i=1}^{200} \sim MREG(F, h_1, h_2)$. The generating scheme is equivalent to the following stochastic black model, let Y_i indicator for the class membership.

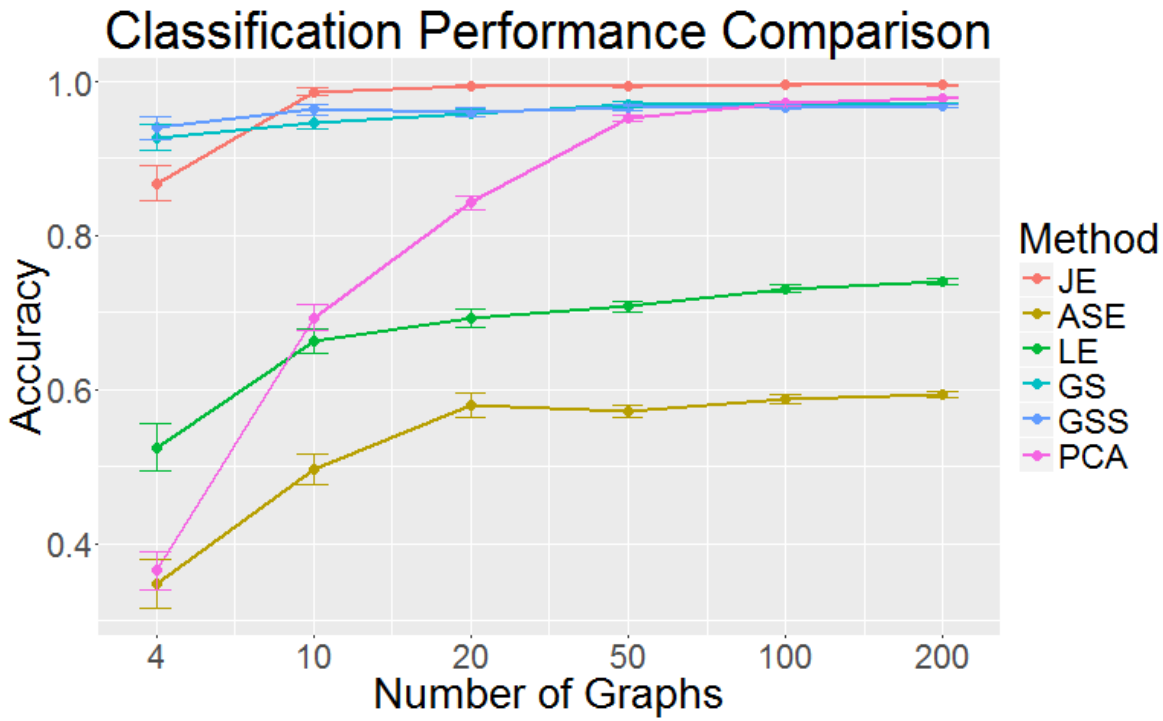
$$A_i|Y_i = 0 \sim SBM((1, ..., 1, 2, ..., 2), \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix})$$

$$A_i|Y_i = 1 \sim SBM((1, ..., 1, 2, ..., 2), \begin{bmatrix} 0.25 & 0.2 \\ 0.2 & 0.25 \end{bmatrix})$$

These graphs are jointly embedded and classified by applying 1-Nearest Neighbor on $\hat{\lambda}_i$.

Experiment 1: Result

We compare joint embedding to extract features to five other feature extraction approaches: Adjacency Spectral Embedding, Laplacian Eigenmap, Graph Statistics, Graph Spectral Statistics, and PCA.



Experiment 2: Predict CCI

Functional magnetic resonance images of 113 subjects are collected and converted to graphs. A covariate named creativity index is also measured for each subject. We try to predict creativity index based on the neural connectivity. We first jointly embed 113 graphs, then regress creativity index on $\hat{\lambda}_i$,

$$CCI_i \sim \beta_0 + \hat{\lambda}_i^T \beta + \epsilon_i.$$

The p-value compared to null model is around 0.0018.

Experiment 2: Result

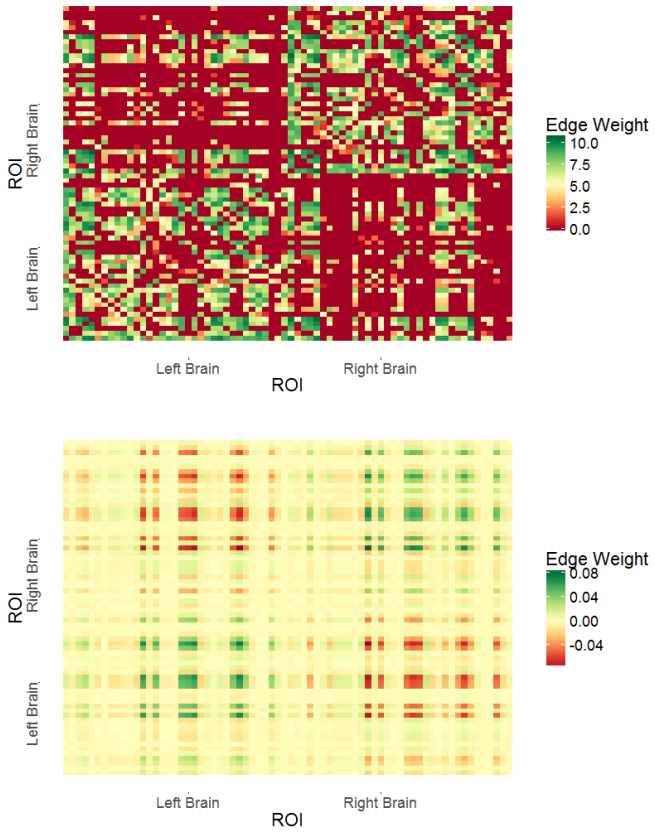


Figure: A typical brain graph and $\hat{h}_6 \hat{h}_6^T$.

Experiment 2: Result

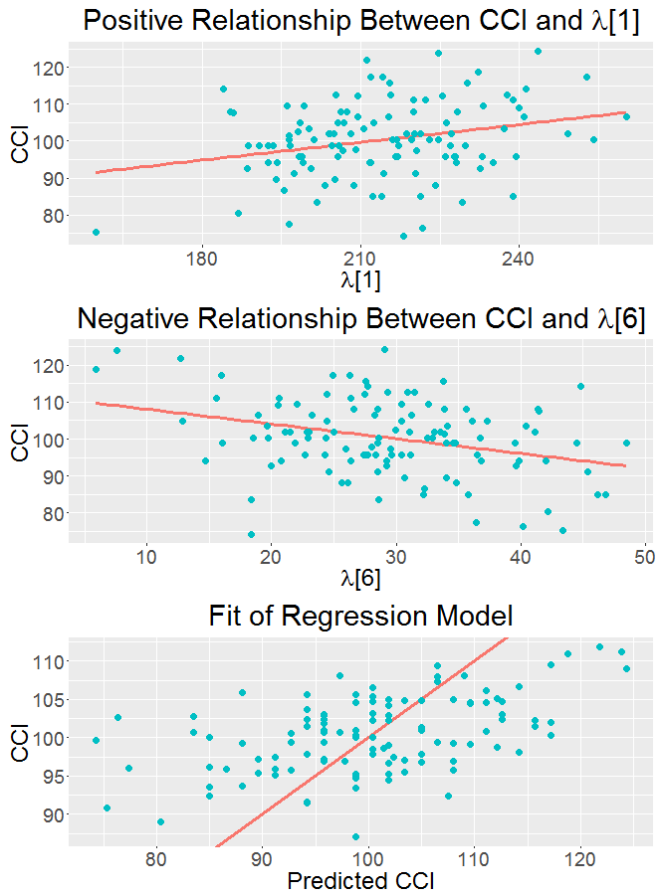


Figure: The relationship between CCI and $\hat{\lambda}$.

Experiment 3: Clustering Wikipedia Pages

We apply the spectral clustering approach to Wikipedia graphs. The vertices of these graphs represent Wikipedia article pages. The two vertices are connected by an edge if either of the associated pages hyperlinks to the other. Two graphs are constructed based on English webpages and French webpages.We consider a subset vertices from 3 categories: People, Things, Math Terms.

Method	ASE+EN	ASE+FR	JE+EN	JE+FR
ARI	0.147	0.115	**0.158**	0.156
Purity	0.549	0.520	**0.551**	0.549

Table: Clustering Performance on Wikipedia Graphs.

Experiment 3: Result

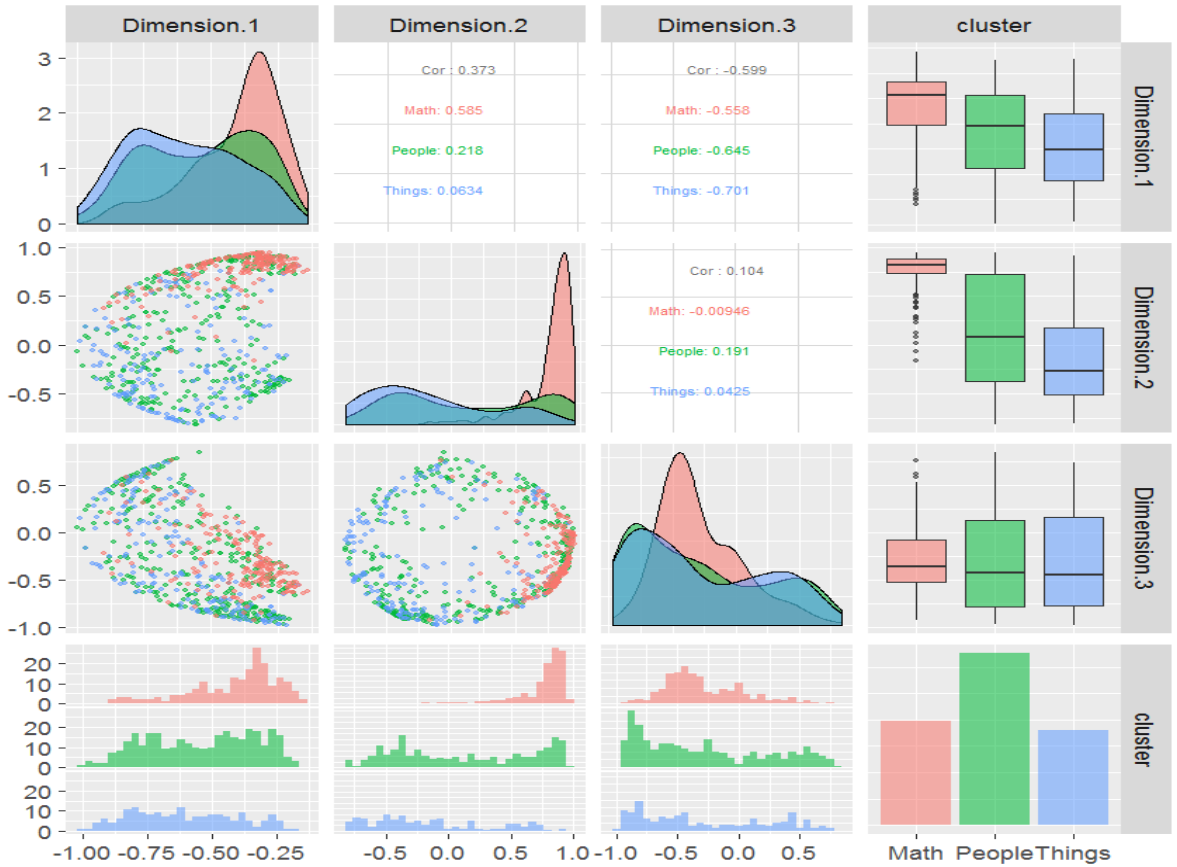


Figure: The latent positions of English Graph estimated by the joint embedding are shown.

Variations

$$(\hat{\lambda}_1, ..., \hat{\lambda}_m, \hat{h}_1, ..., \hat{h}_d) = \underset{\lambda_i, \|h_k\|=1}{\text{argmin}} \sum_{i=1}^{m} \|\mathbf{A}_i - \sum_{k=1}^{d} \lambda_i[k] h_k h_k^T\|^2.$$

- Constrain λ_i to be the same within the group.
- Constrain λ_i to be non-negative.
- Add penalty on λ_i.
- Use logistic loss.
- Include matrix such as $h_k h_{k'}^T$.

This is joint work with Joshua T. Volgelstein and Carey E. Priebe.

Thank you! Questions?