Tracking the Adjacency Spectral Embedding for Streaming Graphs

Federico Larroca, Paola Bermolen, Marcelo Fiori, Bernardo Marenco and Gonzalo Mateos

Asilomar Conference on Signals, Systems, and Computers
November 2022
Random dot product graphs

Consider a latent space $\mathcal{X}_d \subset \mathbb{R}^d$ such that for all

$$x, y \in \mathcal{X}_d \implies x^\top y \in [0, 1]$$

⇒ Inner-product distribution $F : \mathcal{X}_d \mapsto [0, 1]$

Random dot product graphs (RDPGs) are defined as follows:

$$x_1, \ldots, x_{N_v} \overset{i.i.d.}{\sim} F, \quad A_{ij} \mid x_i, x_j \sim \text{Bernoulli}(x_i^\top x_j)$$

for $1 \leq i, j \leq N_v$, where $A_{ij} = A_{ji}$ and $A_{ii} \equiv 0$

A particularly tractable latent position random graph model

⇒ Vertex positions $X = [x_1, \ldots, x_{N_v}]^\top \in \mathbb{R}^{N_v \times d}$

Estimation of latent positions

- **Q:** Given $G$ from an RDPG, find the ‘best’ $X = [x_1, \ldots, x_{N_v}]^\top$?

- MLE is well motivated but it is intractable for large $N_v$

  $$\hat{X}_{ML} = \arg\max_X \prod_{i<j} (x_i^\top x_j)^{A_{ij}} (1 - x_i^\top x_j)^{1-A_{ij}}$$

- Instead, let $P_{ij} = P((i, j) \in \mathcal{E})$ and define $P = [P_{ij}] \in [0, 1]^{N_v \times N_v}$
  
  $\Rightarrow$ The RDPG model specifies that $P = XX^\top$

  $\Rightarrow$ **Key:** Observed $A$ is a noisy realization of $P$ ($\mathbb{E}[A] = P$)

- Suggests a **LS regression** approach to find $X$ s.t. $XX^\top \approx A$

  $$\hat{X}_{LS} = \arg\min_X \|XX^\top - A\|_F^2$$

Adjacency spectral embedding

- Since $A$ is real and symmetric, can decompose it as $A = U\Lambda U^\top$
  - $U = [u_1, \ldots, u_{N_v}]$ is the orthogonal matrix of eigenvectors
  - $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_{N_v})$, with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_{N_v}$

- Define $\hat{\Lambda} = \text{diag}(\lambda_1^+, \ldots, \lambda_d^+)$ and $\hat{U} = [u_1, \ldots, u_d]$ ($\lambda^+ := \max(0, \lambda)$)

- Best rank-$d$, positive semi-definite (PSD) approximation of $A$ is $\hat{U}\hat{\Lambda}\hat{U}^\top$

  $\Rightarrow$ Adjacency spectral embedding (ASE) is $\hat{X}_{LS} = \hat{U}\hat{\Lambda}^{1/2}$ since

  $$A \approx \hat{U}\hat{\Lambda}\hat{U}^\top = \hat{U}\hat{\Lambda}^{1/2}\hat{\Lambda}^{1/2}\hat{U}^\top = \hat{X}_{LS}\hat{X}_{LS}^\top$$

- Q: Is the solution unique? Nope, inner-products are rotation invariant

  $$P = XW(XW)^\top = XX^\top, \quad WW^\top = I_d$$

  $\Rightarrow$ RDPG embedding problem is identifiable modulo rotations
Embedding an SBM graph

- **Ex:** SBM with $N_v = 1500$, $Q = 3$ and mixing parameters

$$\alpha = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \quad \Pi = \begin{bmatrix} 0.5 & 0.1 & 0.05 \\ 0.1 & 0.3 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

- Sample adjacency $A$ (left), $\hat{X}_{LS} \hat{X}_{LS}^T$ (center), rows of $\hat{X}_{LS}$ (right)

- Use embeddings to bring to bear geometric methods of analysis
Interpretability of the embeddings

- Ex: Zachary’s karate club graph with $N_v = 34$, $N_e = 78$ (left)

- Node embeddings (rows of $\hat{X}_{LS}$) for $d = 2$ (right)
  - Club’s administrator ($i = 0$) and instructor ($j = 33$) are orthogonal

- Interpretability of embeddings a valuable asset for RDPGs
  - Vector magnitudes indicate how well connected nodes are
  - Vector angles indicate nodes’ affinity
Streaming Graphs

- **Goal**: track the underlying model of a stream of graphs \( G_t \)
  
  Ex 1: monitoring a wireless network
  Ex 2: evolving social network

- **Naive approach**: estimate \( \hat{X}_t \) by finding the ASE for each \( A_t \) separately
  
  ❌ Computationally expensive
  ❌ Challenging to align separate embeddings

- ASE does not account for the all-zero diagonal of \( A \). Truly we want to solve

\[
\hat{X} \in \arg\min_{X} \| M \circ (A - XX^\top) \|_F^2
\]

\[
\Rightarrow M := 11^\top - I \text{ is a mask matrix, with zero-diagonal and ones everywhere else}
\]
Gradient descent

- Let $f : \mathbb{R}^{N_v \times d} \mapsto \mathbb{R}$ be the objective function $f(X) = \|M \circ (A - XX^\top)\|_F^2$
  $\Rightarrow$ Non-convex w.r.t. $X$, convex w.r.t. $P = XX^\top$

- Gradient descent (GD) method (a.k.a. factorized GD or Procrustes flow)
  $$X_{t+1} = X_t - \alpha \nabla f(X_t), \quad t = 0, 1, 2, \ldots$$
  $\Rightarrow$ Step size $\alpha > 0$ and $\nabla f(X) = 4 [M \circ (XX^\top - A)] X$, for symmetric $A$ and $M$

- Convergence: if $X_0$ is close to the solution, iterations converge with linear rate to $\hat{X}$

Proposition. There exist $\delta > 0$ and $0 < \kappa < 1$ such that, if $\|X_0 - \hat{X}\|_F \leq \delta$, then

$$d(X_t, \hat{X}) \leq \kappa^t d(X_0, \hat{X}), \quad \text{for all } t > 0,$$

where $d(X, \hat{X}) := \min_{W \in O_{d \times d}} \|XW - \hat{X}\|_F^2$ accounts for the rotational ambiguity.

Y. Chi et al., “Nonconvex optimization meets low-rank matrix factorization: An overview,” TSP, 2019
Tracking via warm restarts

**Idea:** Update $\hat{X}_t$ through GD initialized with the previous estimate $\hat{X}_{t-1}$

**Example:**
- $N_v = 6$ Wi-Fi APs in a Uruguayan school
- Hourly measurements over 4 weeks (655 graphs)
- AP 4 was moved at $t \approx 310$

Our approach in context

Q Isn’t this the classic problem of recursively updating eigenvalues/vectors?
A Yes, but
  × Computationally expensive except for specific types of changes (e.g. rank-1)
  × Available methods accumulate error and/or still produce unaligned estimates

Example: an SBM with two communities, at each $t$ a random node changes affiliation

Varying number of nodes

- Dynamic graphs typically include deletions/additions of nodes
  - Deletions are easy to handle, but additions?
- Assume a single node $i = N_v + 1$ is added
  - The new $A_{t+1} \in \{0, 1\}^{N_v + 1 \times N_v + 1}$ has an extra row (column) $a_{N_v + 1} \in \{0, 1\}^{N_v}$
  - What about $\hat{x}_{N_v + 1}$?
- Reasonable approximation: project $a_{N_v + 1}$ to the column space of $\hat{X}_t$

$$\hat{x}^{\text{proj}}_{N_v + 1} = a_{N_v + 1} \hat{X}_t^{\text{norm}}$$

with $\hat{X}_t^{\text{norm}}$ the column-wise normalized version of $\hat{X}_t$

- ✔ Simple and consistent as $N_v \to \infty$
- ✔ Preserves alignment
- ✗ Assumes embeddings do not change over time
- ✗ Error accumulates as new nodes are added in the finite $N_v$ regime

K. Levin et al., “Out-of-sample extension of graph adjacency spectral embedding,” PMLR, 2018
Varying number of nodes

**Idea:** Update $\hat{X}_{t+1}$ using GD where old nodes are initialized at $\hat{x}_t$ and new ones at $\hat{x}^{\text{proj}}$

- Still simple and consistent as $N_v \to \infty$
- Preserves alignment
- Embeddings may change over time
- Constant error as new nodes are added in the finite $N_v$ regime

Simple example:
- $G_0 = G_{100,0.1}$. We add new nodes that are also from an ER with $p = 0.1$
Real-world data

- $G_t$ consisting of:
  - Nodes: national football teams
  - (Weighted) edges: number of matches between years $t - 3$ and $t$
- Start at $t = 1930$ ($N_v = 41$) and finish at $t = 2015$ ($N_v = 222$). We use a fixed $d = 7$.
  - Example: Australia left the OFC and joined the AFC in 2005
  - Plot: Asia and Oceania’s embeddings best 2-d approximation

Y. Li et al., “Networks of international football: Community structure, evolution and globalization of the game,” Applied Network Science, 2022
Concluding remarks

- ASE to estimate latent nodal positions in RDPGs ⇒ Non-convex matrix factorization
- Convergent, first-order gradient descent algorithm for refined formulation
  ⇒ Scalable and fast computation of nodal representations
  ⇒ Track dynamic network representations even when $N_v$ changes

Future work

⇒ Directed case implies constraints on the optimization problem
⇒ Consistency, asymptotic normality, stability ($N_v \to \infty$)

🔗 https://github.com/git-artes/