Out of sample extension of graph ASE

General idea:

- Embed some graph into Euclidean space with ASE
- now, find a new vertex that wasn't in the original graph
- where does that vertex go in the embedded space?

Questions

- Why do we have latent positions be represented as row rather than column vectors?
- How does the LLS equation work?
- What is an elbow?
- Why does `out_sample_A @ eig_vectors @ np.diag(1/np.sqrt(eig_values))` give us our embedding?
- what is the relevance of `self._profile_likelihood_maximization`?

Introduction

- Graph embedding is the problem of embedding some vertex $v \in V$ to some lower-dimensional representation $x_v \in S$ in a way that preserves the topology of its graph G in some way.
- ASE and LSE are the two primary methods for doing this.

Background and Notation

- formally, you have some data D and some embedding of it $X \in R^{nxn}$
  - such that the embedding of $z_i \in D$ is given by the i_{th} row of X
- now you have some new observation z, and you want to embed it under the same scheme used to produce X
- how do you do this without redoing the whole embedding?
- there is some previous work here, you can solve least squares problems, some people used a neural net
- not too much work on the ASE oos extension besides Tang et al., 2013a

ways to do it

- least squares

Notation

- for $B \in R^{n_1xn_2}$, $\sigma_i(B)$ denotes the i_{th} singular value of B, where $\sigma_1(B) \geq \sigma_2(B) \geq \ldots \geq \sigma_k(B) \geq 0$
- there are k singular values, where $k = min\{n_1, n_2\}$
- n indexes the number of vertices in a hollow graph G
- E_n is some event that occurs with high probability

- bunch of stuff I don't understand that well, something about the Borel-Cantelli Lemma
- for some vector x \in R^dx ∈ Rd, ||x||||x|| means the Euclidean norm, and ||x||_\infty||x||∞ means the largest norm of xx across all possible dd.

OOS algorithm discussion

- ASE decomposes adjacency matrix into their eigenmatrix and then makes it smaller
- RDPGs make graphs from dot products
- ASE recovers vertices in RDPGs (and SBMs) really well
- this is only up to a rotation though
- ok so given that, say we compute some embedding matrix \hat{X}X^ where its rows are the latent positions of the i_{th}ith vertex
- Now add a new vertex vv to GG

The Juice: Linear Least Squares (LLS)

- embed vertex v as the least-squares solution to \hat{X}w = \vec{a}X^w = a⃗
  - why can you treat X as a linear transformation, and why will transforming w produce \vec{a}a⃗?
- solves this equation:
  - \min{w\in R^d}\sum_{i=1}^n (a_i-\hat{X}_i^Tw)^2min w ∈ Rd ∑i=1n(ai − X^iT w)2

important results

- ASE recovers latent positions with error of order n^{-1/2}log(n)n−1/2log(n) uniformly over the nn vertices.

09/11 office hours

- other use cases: high or low degree nodes tend to distort spectral embedding, so you can omit them from original embedding and then add back in with oos (pretty cool)
- also comes up a bit with omni in a way that I currently don't understand
- linear least squares version
- oos is bizarre: simple implementation once you get it, essentially all in the line pedigo linked
- we could never figure out an API that really made sense though

API

- start with a subgraph, then embed
- need edges from oos vertices, and how they connect to in-sample vertices
- API hard
- hayden wrote a lot of random stuff that might be broken now on top of the relatively simple oos algorithm, so unclear whether playing with his code is super worth it at this point