HEART: Statistics and Data Science With Networks

Joshua Agterberg

Johns Hopkins University

Fall 2021

Outline

Outline

Overall Idea

- Starting with a graph, obtain its adjacency matrix **A** (or in general a similarity matrix)

Overall Idea

- Starting with a graph, obtain its adjacency matrix **A** (or in general a similarity matrix)
- Plot the eigenvalues of **A** in decreasing order and try to find an elbow at level d (computationally or eyeball)

Overall Idea

- Starting with a graph, obtain its adjacency matrix **A** (or in general a similarity matrix)
- Plot the eigenvalues of **A** in decreasing order and try to find an elbow at level d (computationally or eyeball)
- Compute an $n \times d$ matrix called the *graph embedding* of **A** using
  1. `eigen` <- `irlba`$(\mathbf{A}, d)$
  2. $\hat{X}$ <- `eigen`$\$u$ %*% `diag`$\left($`sqrt`$($`eigen`$\$d)\right)$

Overall Idea

- Starting with a graph, obtain its adjacency matrix **A** (or in general a similarity matrix)
- Plot the eigenvalues of **A** in decreasing order and try to find an elbow at level d (computationally or eyeball)
- Compute an $n \times d$ matrix called the *graph embedding* of **A** using
  1. `eigen <- irlba(`**A**$, d)$
  2. $\hat{X}$ `<- eigen$`u `%*% diag(sqrt(eigen$`d`))`
- Treat the n rows of $\hat{X}$ as your data set, performing clustering or hypothesis testing with $\hat{X}$

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)
- `irlba` can be replaced with other (ideally faster) computational method to compute singular values and singular vectors

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)
- `irlba` can be replaced with other (ideally faster) computational method to compute singular values and singular vectors
- **A** may have negative eigenvalues, but if you use an SVD-based approach (like `irlba`), this will be ignored

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)
- `irlba` can be replaced with other (ideally faster) computational method to compute singular values and singular vectors
- **A** may have negative eigenvalues, but if you use an SVD-based approach (like `irlba`), this will be ignored
- Extremely large networks will need different methods to obtain the embedding, since `irlba` works primarily on medium-sized graphs ($n = 2000$ for my computer)

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)
- `irlba` can be replaced with other (ideally faster) computational method to compute singular values and singular vectors
- **A** may have negative eigenvalues, but if you use an SVD-based approach (like `irlba`), this will be ignored
- Extremely large networks will need different methods to obtain the embedding, since `irlba` works primarily on medium-sized graphs ($n = 2000$ for my computer)
- Finding the elbow is not a science and more of an art, so you may have to redo a few different times with different choices of d

Practical Considerations

- Can use any similarity matrix **S** (need not be the adjacency matrix)
- `irlba` can be replaced with other (ideally faster) computational method to compute singular values and singular vectors
- **A** may have negative eigenvalues, but if you use an SVD-based approach (like `irlba`), this will be ignored
- Extremely large networks will need different methods to obtain the embedding, since `irlba` works primarily on medium-sized graphs ($n = 2000$ for my computer)
- Finding the elbow is not a science and more of an art, so you may have to redo a few different times with different choices of d
- Idea works for DCSBMs, MMSBMs, RDPGs, GRDPGs, and any low-rank model

Practical Considerations

- Sometimes there may be a “sign difference” meaning that the dimensions of the plots might switch signs

Practical Considerations

- Sometimes there may be a “sign difference” meaning that the dimensions of the plots might switch signs
- This is because eigenvectors are still eigenvectors when multiplied by -1

Practical Considerations

- Sometimes there may be a “sign difference” meaning that the dimensions of the plots might switch signs
- This is because eigenvectors are still eigenvectors when multiplied by -1
- For one graph, this is essentially no problem, but for more graphs, this is something to keep in mind

Practical Considerations

- Sometimes there may be a "sign difference" meaning that the dimensions of the plots might switch signs
- This is because eigenvectors are still eigenvectors when multiplied by -1
- For one graph, this is essentially no problem, but for more graphs, this is something to keep in mind
- Suppose you have two graphs and two graph embeddings $\hat{X}$ and $\hat{Y}$ (both in dimension $n \times d$

Practical Considerations

- Sometimes there may be a "sign difference" meaning that the dimensions of the plots might switch signs
- This is because eigenvectors are still eigenvectors when multiplied by -1
- For one graph, this is essentially no problem, but for more graphs, this is something to keep in mind
- Suppose you have two graphs and two graph embeddings $\hat{X}$ and $\hat{Y}$ (both in dimension $n \times d$
- $\hat{X}$ and $\hat{Y}$ might not be aligned, so you need to run an alignment procedure – called a `procrustes` algorithm, designed to align point clouds on top of each other

Practical Considerations

- Sometimes there may be a "sign difference" meaning that the dimensions of the plots might switch signs
- This is because eigenvectors are still eigenvectors when multiplied by -1
- For one graph, this is essentially no problem, but for more graphs, this is something to keep in mind
- Suppose you have two graphs and two graph embeddings $\hat{X}$ and $\hat{Y}$ (both in dimension $n \times d$
- $\hat{X}$ and $\hat{Y}$ might not be aligned, so you need to run an alignment procedure – called a `procrustes` algorithm, designed to align point clouds on top of each other
- If the graphs have different numbers of vertices, this alignment procedure doesn't work, but luckily I wrote a paper about this

Outline

Community Detection with Graph Embeddings

- Starting with a graph embedding in $\hat{X}$, cluster the rows using your favorite clustering method:

Community Detection with Graph Embeddings

- Starting with a graph embedding in $\hat{X}$, cluster the rows using your favorite clustering method:
  - K-means
  - Gaussian Mixture Modelling (library `mclust`)
  - K-Medoids (`pam` algorithm)

Community Detection with Graph Embeddings

- Starting with a graph embedding in $\hat{X}$, cluster the rows using your favorite clustering method:
  - K-means
  - Gaussian Mixture Modelling (library `mclust`)
  - K-Medoids (`pam` algorithm)
- Practical issue: for real data, you need to interpret the clusters yourself (always a problem with unsupervised learning)

Community Detection with Graph Embeddings

- Starting with a graph embedding in $\hat{X}$, cluster the rows using your favorite clustering method:
  - K-means
  - Gaussian Mixture Modelling (library `mclust`)
  - K-Medoids (`pam` algorithm)
- Practical issue: for real data, you need to interpret the clusters yourself (always a problem with unsupervised learning)
- For simulated data or partially labelled data, you can compare output using the Adjusted Rand Index (ARI)

Community Detection with Graph Embeddings

- Starting with a graph embedding in $\hat{X}$, cluster the rows using your favorite clustering method:
  - K-means
  - Gaussian Mixture Modelling (library `mclust`)
  - K-Medoids (`pam` algorithm)
- Practical issue: for real data, you need to interpret the clusters yourself (always a problem with unsupervised learning)
- For simulated data or partially labelled data, you can compare output using the Adjusted Rand Index (ARI)
  - ARI accounts for the problem that any relabeling of the clusters is "the same"

Outline

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution
  - Test whether two SBMs have the same B matrix

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution
  - Test whether two SBMs have the same B matrix
  - Test whether the rank is in fact d

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution
  - Test whether two SBMs have the same B matrix
  - Test whether the rank is in fact d
  - ...more!

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution
  - Test whether two SBMs have the same B matrix
  - Test whether the rank is in fact d
  - ...more!
- E.g. test whether a stochastic blockmodel has a particular B matrix by clustering and estimating the probabilities of membership with the clustering aggregation

Hypothesis Testing with Graph Embeddings

- First, determine what you would like to test. Some tests are:
  - Test whether two vertices have the same community
  - Test whether two graphs have the same general distribution
  - Test whether two SBMs have the same B matrix
  - Test whether the rank is in fact d
  - ...more!
- E.g. test whether a stochastic blockmodel has a particular B matrix by clustering and estimating the probabilities of membership with the clustering aggregation

Practical Considerations

- This requires knowing some more statistics to do these tests

Practical Considerations

- This requires knowing some more statistics to do these tests
- If you can translate the network-level test to a test on graph embeddings, you are done

Practical Considerations

- This requires knowing some more statistics to do these tests
- If you can translate the network-level test to a test on graph embeddings, you are done
- Sometimes it's not clear how to do that, and sometimes things get kind of complicated

Practical Considerations

- This requires knowing some more statistics to do these tests
- If you can translate the network-level test to a test on graph embeddings, you are done
- Sometimes it's not clear how to do that, and sometimes things get kind of complicated
- Math can be difficult to understand for these problems, but the idea is the same for many of these papers