Università della Svizzera italiana

Facoltà di scienze informatiche

Informatics Bachelor Course Numerical Computing **Academic Year 2017/2018**

Instructor: Prof. Olaf Schenk TA: Fabio Verbosio/Aryan Eftekhari

Assignment 2 - Social Networks

Due date: Thursday 19 October 2017, 8:30am

The purpose of this assignment[1] is to learn the importance of sparse linear algebra algorithms to solve fundamental questions in social network analyses. We will use the coauthor graph from the Householder Meeting and the social network of friendships from Zachary's karate club [1]. These two graphs are one of the first examples where matrix methods were used in computational social network analyses.

The Householder XII Meeting

Twenty-one years ago, in June, 1993, the UCLA Lake Arrowhead Conference Center in the San Bernardino Mountains, 90 miles east of Los Angeles, was the site of the Householder XII Symposium on Numerical Linear Algebra, organized by Gene Golub[2] and Tony Chan. Nick Trefethen posted a flip chart with Gene Golub's name in the center. He invited everyone present to add their name to the chart and draw lines connecting their name with the names of all their coauthors. The diagram grew denser throughout the week. At the end it was a graph with 104 vertices (or people) and 211 edges. John Gilbert entered the names and coauthor connections into MATLAB, creating an adjacency matrix A. Let's retrieve the names and matrix as they are stored in the PARC archive (`housegraph.mat`) and available on the course webpage.

```
load housegraph
```

The command `who` shows which variables are stored in the data file.

```
who
```

[1] This document is originally based on a blog from Cleve Moler, who wrote a fantastic blog post about the Lake Arrowhead graph, and John Gilbert, who initially created the coauthor graph from the 1993 Householder Meeting. You can find more information at http://blogs.mathworks.com/cleve/2013/06/10/lake-arrowhead-coauthor-graph/ and ftp://ftp.parc.xerox.com/pub/gilbert/graph.html. Most of this assignment is derived from these two archived work.

[2] http://en.wikipedia.org/wiki/Gene_H._Golub

```
Your variables are:
A            Bunch          Funderlic     Ipsen          Moler          Reichel        VanLoan
ATrefethen   BunseGerstner  George        Jessup         MuntheKaas     Ruhe           Varah
Ammar        Byers          Gilbert       Kagstrom       NHigham        Saied          Varga
Anjos        Chandrasekaran Gill          Kahan          NTrefethen     Sameh          Warner
Arioli       Crevelli       Golub         Kaufman        Nachtigal      Saunders       Widlund
Ashby        Cullum         Gragg         Kenney         Nagy           Schreiber      Wilkinson
Bai          Davis          Greenbaum     Kincaid        Ng             Smith          Wold
Barlow       Demmel         Gu            Kuo            Nichols        Starke         Young
Benzi        Dubrulle       Gutknecht     Laub           OLeary         Stewart        Zha
Berry        Duff           Hammarling    LeBorne        Ong            Strakos        name
Bjorck       Edelman        Hansen        Liu            Overton        Szyld          prcm
Bjorstad     Eisenstat      Harrod        Luk            Paige          TChan          xy
Bojanczyk    Elden          He            Marek          Pan            Tang
Boley        Ernst          Heath         Mathias        Park           Tong
Boman        Fierro         Henrici       Meyer          Plemmons       VanDooren
Borges       Fischer        Hochbruck     Modersitzki    Pothen         VanHuffel
```

and `name` shows the participants as they entered the list:

```
name
```

```
name =
Golub
Wilkinson
TChan
He
Varah
Kenney
Ashby
..
..
```

```
size(A)
```

```
ans =
   104   104
```

The matrix A is 104-by-104 and symmetric. Elements $A_{i,j}$ and $A_{j,i}$ are both equal to one if i and j are the indices of coauthors and equal to zero otherwise. Most of the elements are zero because most pairs of attendees are not coauthors. So the matrix is sparse. In fact, the sparsity, the fraction of nonzero elements, is less that five percent.

```
format short
sparsity = nnz(A)/prod(size(A))
```

```
sparsity =
    0.0486
```

In the left image of Figure 1 we show a picture of the matrix using `spy` plot. It shows the location of the nonzeros, with Gene Golub in the first column and the other authors in the fairly arbitrary order in which they appeared on Trefethen's flip chart.

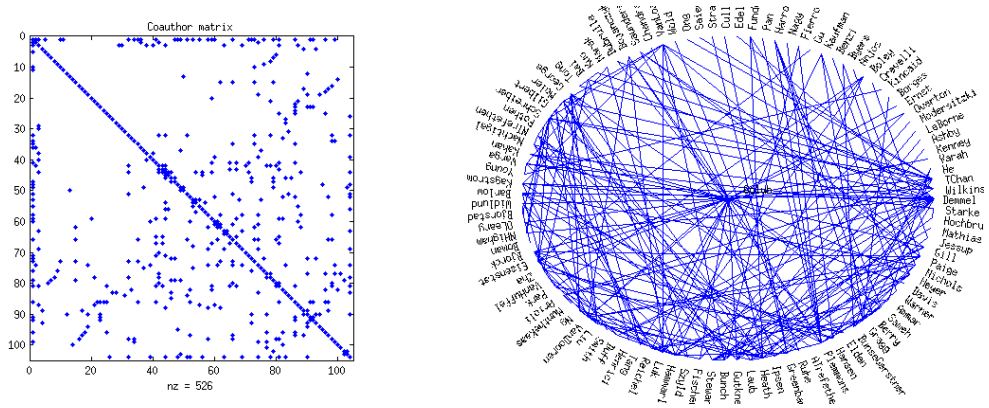

Figure 1. [Left] The coauthor matrix from the meeting (created with `spy(A)`, [Right] The coauthor circle from the meeting.

Most Prolific Author

As the adjacency matrix is loaded from the archive, the most prolific coauthor is already in the first column. It was not a surprise to the participants at the conference to find that the most prolific coauthor is Gene Golub.

```
m = find(sum(A) == max(sum(A)))
name(m,:)
```

```
m =
    1
ans =
Golub
```

Circular of all the Authors

In the right image of Figure 1 we created a circular plot with Golub in the center, the other authors around the circumference, and edges connecting the coauthors. If we place the authors around the circumference in the order we retrieve them from the chart, the edges would cross the circle pretty much randomly as shown in Figure 1.

```
drawit;
snapnow
```

Reorder the Authors

We want to rearrange the authors so that the coauthor connections are as close as possible to the circumference of the circle. This corresponds to a symmetric permutation of the matrix A that minimizes, or at least reduces, its bandwidth. In 1969, Elizabeth Cuthill and John McKee described a heuristic for reordering the rows and columns of a matrix to reduce its bandwidth and it is now known as the *Reverse Cuthill McKee* ordering.

```
r = symrcm(A(2:end,2:end));
prcm = [1  r+1];
spy(A(prcm,prcm))
```

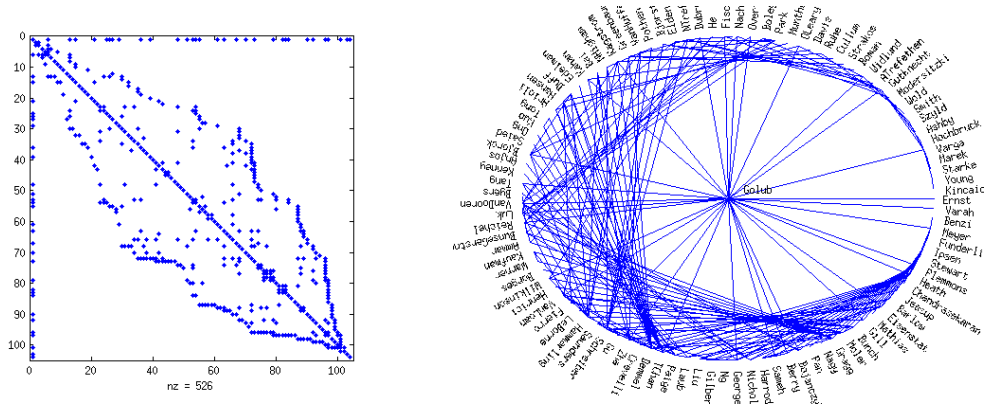

Figure 2. [Left] The reordered coauthor matrix PAP^T from the meeting, [Right] The reordered coauthor circle.

```
title('Coauthor matrix with reduced bandwidth');
drawitrcm;
snapnow
```

Figure 2 demonstrates the impact of the reordering P to both the matrix PAP^T and the circle.

1. Zachary's Karate Club Graph

Spectral Graph Partitioning and the Laplacian with Matlab

In this segment, we will plant an artificial partition in a graph, and then use the second smallest eigenvector λ_2 to find it. As always, the first step is to generate our dataset. In this example, we will be a little more ambitious and use a larger number of vertices.

```
n=1000;
```

To plant an artificial partition in our test dataset, we need to determine the vertices in each of the two groups. To do this, we randomly generate a permutation of all the vertices and select one set as group 1 and the rest as group 2. The variable `gs` controls how big the size of the first group is.

```
x = randperm(n);
gs = 450;
group1 = x(1:gs);
group2 = x(gs+1:end);
```

Now, we need to decide on the probabilities of edges within each group and between the two groups. Because we are planting a partition, the probabilities of edges between the groups should be much lower than the probability of edges within each group. Suppose that group 1 is a little more tightly connected than group 2. (Please insert your own amusing names for an actual identification of group 1 and group 2, e.g. politicians and mathematicians.)

```
p_group1 = 0.5;
p_group2 = 0.4;
p_between = 0.1;
```

With these probabilities in hand, we can construct our graph. The last few operations symmetrize the matrix.

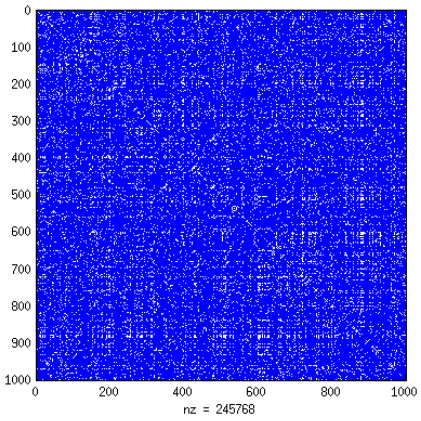

Figure 3. An almost dense artificial dataset.

```
A(group1, group1) = rand(gs,gs) < p_group1;
A(group2, group2) = rand(n-gs,n-gs) < p_group2;
A(group1, group2) = rand(gs, n-gs) < p_between;
```

Next, let's see if we can see the partition.

```
spy(A);
```

Figure 3 shows the result of the spy command. While some might claim to see a partition in this data, I think it is not particularly obvious. Now, let's investigate what the second smallest eigenvector λ_2 tells us about this graph. We will use the eig command in Matlab to compute all eigenvectors (stored in V) and eigenvalues (stored in D)

```
A = triu(A,1);
A = A + A';
deg = sum(A);
L = diag(deg)-A;
[V D] = eig(L);
D(2,2)
ans =
  199.9732
```

The second smallest eigenvalue states that we shouldn't expect to find any very good cuts in our dataset. To see what we found, let's plot the second smallest eigenvector V(:,2).

```
plot(V(:,2), '.-');
```

The result is shown in the left graphic of Figure 4. That isn't all that helpful. Maybe sorting the vector. The result is shown in the right graphic of Figure 4

```
plot(sort(V(:,2)), '.-');
```

Now, that picture is much more informative! We see a large gap in the middle of the values. Interestingly enough, the number of points on the right gap is the same as gs, the size of our planted group. Let's see what happens when we permute the vertices of the graph to this ordering.

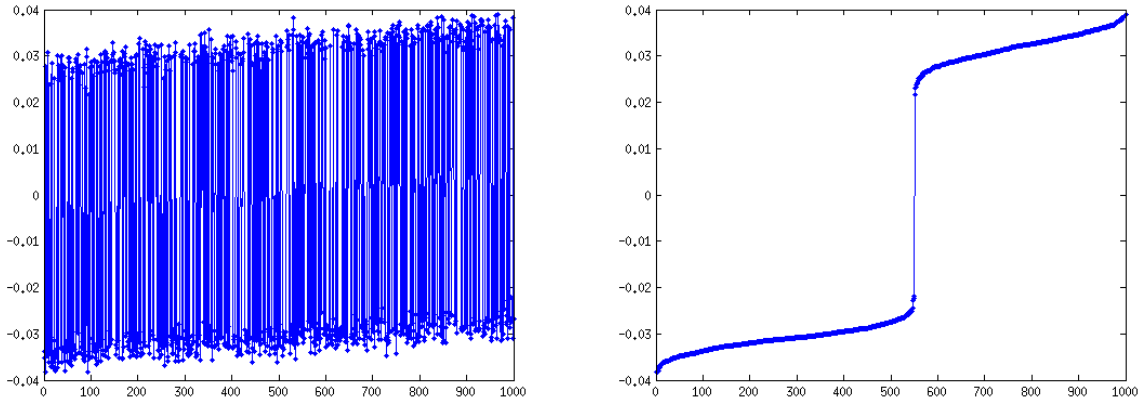

Figure 4. [left] The second smallest eigenvector, [right] The second smallest sorted eigenvector.

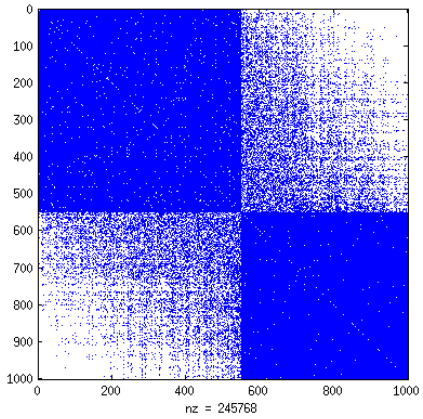

Figure 5. The results after the partitioning of our artificial dataset.

```
[ignore p] = sort(V(:,2));
spy(A(p,p));
```

Looks like we found our partitions in Figure 5. In spectral analysis of a graph, the second eigenvalue λ_2 is **very helpful** in finding a partitioning that splits the graph into **two subgraphs** while at the same time **minimizing the edges** between these two subgraphs.

2. Solve the following Social Networks problems:

1. **The Reverse Cuthill McKee Ordering [20 points]:** Read section 5.7 *(Permutations and ordering strategies, pp. 122-127)* from the book *A First Course in Numerical Methods* and explain in two paragraphs the effects of the three lines in the Matlab code below.

```
r = symrcm(A(2:end,2:end));
prcm = [1 r+1];
spy(A(prcm,prcm))
```

   Explain the impact of this ordering to the circle of the author graph.

2. **Degree Centrality [10 points]:** In graph theory and network analysis, centrality refers to indicators which identify the most important vertices within a graph. Applications include identifying the most influential person(s) in a social network, key infrastructure nodes in the Internet or urban networks, and super spreaders of disease. Here we are interested in the **Degree centrality**, which is conceptually simplest. It is defined as the number of links incident upon a node (i.e., the number of vertices that a node has). The degree centrality of a vertex v, for a given graph $G := (V, E)$ with $|V|$ vertices and $|E|$ edges, is defined as the numbers of edges of vertex v, e.g., the numbers of nonzeros in column v or in row v.

   Compute the degree centrality for all authors and order the authors of the Householder meeting in descending order (showing both the authors, the coauthors and the degree centrality.

3. **The Connectivity of the Coauthors [10 points]:** How many coauthors have authors in common? Think about a general expression how to compute the list of common coauthors of two authors in matrix notation. Use this matrix notation to compute common coauthors of the pairs (Golub, Moler), (Golub, Saunders), and (TChan, Demmel)? Who are those common coauthors?

4. **PageRank of the Coauthor Graph [10 points]:** Compute the PageRank value (e.g. using a modified version from `pagerank.m` from the mini-project 1) for all authors and order the authors in descending order according to the page-rank. Show a list similar to

```
page-rank  in  out author
```

5. **Zachary's karate club: social network of friendships between 34 members [50 points]:** Figure 6 shows the social network of a karate club at a US university in the 1970s. At the end of the study, a conflict between club members arose. As a consequence,the club formally split into two separate organizations (white circles vs. grey squares). Please use the adjacency matrix `karate.adj` and answer the following numerical questions.

   - Write a Matlab code that ranks the five nodes with the largest degree centrality? What are their degrees?
   - Rank the five nodes with the largest eigenvector (PageRank) centrality. What are their (properly normalized) eigenvector centralities?
   - Are the rankings in (a) and (b) identical? Give a brief verbal explanation of the similarities and differences. The top five nodes are the same, regardless if we use the degree or eigenvector pagerank

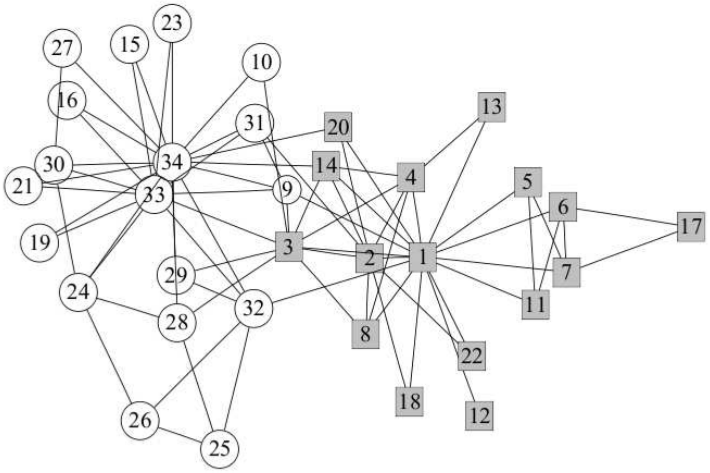

Figure 6. The social network of a karate club at a US university. Image from M. E. J. Newman and M. Girvan, Phys. Rev. E 69,026113(2004)

centrality. Since both are measures for the importance of nodes, it is not completely surprising that we receive similar answers.

- Use spectral graph partitioning to find a near-optimal split of the network into two groups of 16 and 18 nodes, respectively. List the nodes in the two groups. How does spectral bisection compare to the real split observed by Zachary?

In-class assistance

If you experience difficulties in solving the problems above, please come to class either on Tuesday 10:30-12:15, SI-006 or on Thursday 8:30-10:15, SI-006

Additional technical notes

You only need Matlab for this assignment.

References

[1] The social network of a karate club at a US university, M. E. J. Newman and M. Girvan, Phys. Rev. E 69,026113 (2004) pp. 219-229.