nb-2021-04-15

April 29, 2021

1 Exercises for 2021-04-29

Note: You will need to do `import Pkg; Pkg.add("GraphRecipes")` if you do not already have it installed.

```
[ ]: # import Pkg; Pkg.add("GraphRecipes")
```

```
[ ]: using LinearAlgebra
     using Plots
     using GraphRecipes
     using SparseArrays
     using MatrixNetworks
```

1.1 Data setup

The Zachary karate club is a standard example in network analysis; for this and some other examples, see Mark Newman's web site, the Network Repository, and the KONECT repository. Other example repositories include SNAP and UCI.

```
[ ]: m = 78
     n = 34
     adj   = [1 2
              1 3
              2 3
              1 4
              2 4
              3 4
              1 5
              1 6
              1 7
              5 7
              6 7
              1 8
              2 8
              3 8
              4 8
              1 9
              3 9
```

```
3 10
1 11
5 11
6 11
1 12
1 13
4 13
1 14
2 14
3 14
4 14
6 17
7 17
1 18
2 18
1 20
2 20
1 22
2 22
24 26
25 26
3 28
24 28
25 28
3 29
24 30
27 30
2 31
9 31
1 32
25 32
26 32
29 32
3 33
9 33
15 33
16 33
19 33
21 33
23 33
24 33
30 33
31 33
32 33
9 34
10 34
14 34
```

```
        15 34
        16 34
        19 34
        20 34
        21 34
        23 34
        24 34
        27 34
        28 34
        29 34
        30 34
        31 34
        32 34
        33 34]
labels = [0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1]
A = sparse(adj[:,1], adj[:,2], ones(m), n, n)
A = A+A'
spy(A)
```

[ ]:
```
graphplot(A, markersize=0.4, curves=false, names=labels)
```

1.2 Semi-supervised labeling

Suppose we label node 1 as 0 and node 34 as 1.

(4 points): Compute the vector of "soft" labels using the Laplacian-based approach from the 4/22 lecture. Plot the soft labels vs the "ground truth" labels. Does this approach work well?

[ ]:
```

```

1.3 Spectral partitioning

This is a small enough matrix that we can just use the Julia `eigen` function. For larger matrices, we would want to use `eigs` (which is now in the `Arpack.jl` package).

We start by plotting the labeling vs the first eigenvector of the combinatorial Laplacian. What we notice is that this ordering does a pretty good job of separating the two "ground truth" clusters in the graph.

[ ]:
```
d = [sum(A[i,:]) for i = 1:n]
D = spdiagm(0 => d)
L = D-A
Ldense = Matrix(L)
 s, V = eigen(Ldense)
spectral1d = V[:,2]
scatter(spectral1d, labels')
```

(3 points): In your own words, explain what was meant by "does a pretty good job of separating the two 'ground truth' clusters in the graph."

The spectral approximation to maximizing the modularity (defined in terms of $B = A - \frac{dd^T}{2m}$ involves similarly looking at the eigenvector associated with the largest eigenvalue of B.

(3 points): Plot the "ground truth" labels against the components of this eigenvector – you should see it again does a pretty good job of separating out the two pieces.

```
[ ]:
```

1.4 Students choice

Suppose you had to ask a question to gently probe students knowledge of the graph learning methods discussed in the past two weeks.

(2 points): What question do you think you would ask?

(2 points): What would your answer to that question be?