7. Spectral Clustering

Bookmark this page

Exercises due Oct 20, 2021 17:29 IST Completed

Relaxation of Ideal Clustering

The optimization problem for a given n_1, n_2

$$C = \min_{\mathbf{s}\in\{-1,1\}^n} \mathbf{s}^T L \mathbf{s}, \quad \text{such that } \sum_k s_k = n_1 - n_2,$$

is computationally hard to solve for any real-life graph of decent size. We can therefore relax the integer constraints on the $\mathbf{s}$ vector and consider the relaxed problem

$$\hat{C} = \min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}^T L \mathbf{x}, \quad \|\mathbf{x}\| = 1.$$

where we have also ignored the constraint that the components of $\mathbf{x}$ have to sum up to $n_1 - n_2$. The normalization of $\mathbf{x}$ is necessary or otherwise there is only one trivial solution to the problem – the all-zeros vector.

Spectral Clustering: Eigenvector Corresponding to Second Smallest Eigenvalue

The relaxed problem leads to an approximate but intuitive solution to the ideal clustering problem.

Self-exercise: First, we can show that

$$\mathbf{x}^T L \mathbf{x} = \frac{1}{2} \sum_{i,j} A_{ij} (x_i - x_j)^2.$$

Derivation

$$\begin{aligned} x^T L x &= \sum_{i,j=1}^{n} L_{ij} x_i x_j \\ &= \sum_{i,j=1}^{n} \left(D_{ij} - A_{ij}\right) x_i x_j \\ &= \sum_{i=1}^{n} D_{ii} x_i^2 - \sum_{i,j=1}^{n} A_{ij} x_i x_j \\ &= \frac{1}{2} \sum_{i=1}^{n} D_{ii} x_i^2 + \frac{1}{2} \sum_{j=1}^{n} D_{jj} x_j^2 - \sum_{i,j=1}^{n} A_{ij} x_i x_j \\ &= \frac{1}{2} \sum_{i,j=1}^{n} A_{ij} \left(x_i^2 + x_j^2 - 2 x_i x_j\right) \\ &= \frac{1}{2} \sum_{i,j=1}^{n} A_{ij} (x_i - x_j)^2, \end{aligned}$$

where we use the fact that $D_{ii} = \sum_{j=1}^{n} A_{ij}$, and $D_{jj} = \sum_{i=1}^{n} A_{ij}$

Hide

Minimizing the above would lead to a solution $\mathbf{x}$ that can be interpreted as follows: In the original ideal clustering setup, the multiplier to A_{ij} is equal to 0 when two nodes i, j belong to the same cluster. The same way, we can treat any two nodes i, j whose x_i, x_j values are close as being in the same cluster. And, if the x_i, x_j values are far apart we can classify them into different clusters.

Beyond this intuitive understanding of the relaxed problem, the relaxed problem also has important properties. First the graph Laplacian L is a symmetric matrix. Then, using the above expansion we can clearly see that if $A_{ij} \geq 0,\ \forall i, j$ (which is the likely scenario in most applications) L is a **positive semi-definite** matrix. For a positive semidefinite matrix, the eigenvalues are nonnegative.

In particular, for the Laplacian the smallest eigenvalue is equal to 0. This can be seen from the fact that

$$L\mathbf{1} = 0,$$

where $\mathbf{1}$ is a vector of ones. **Self-exercise:** It can also be shown that the multiplicity of the zero eigenvalue is the number of connected components of the simple, undirected graph.

Getting back to the optimization problem

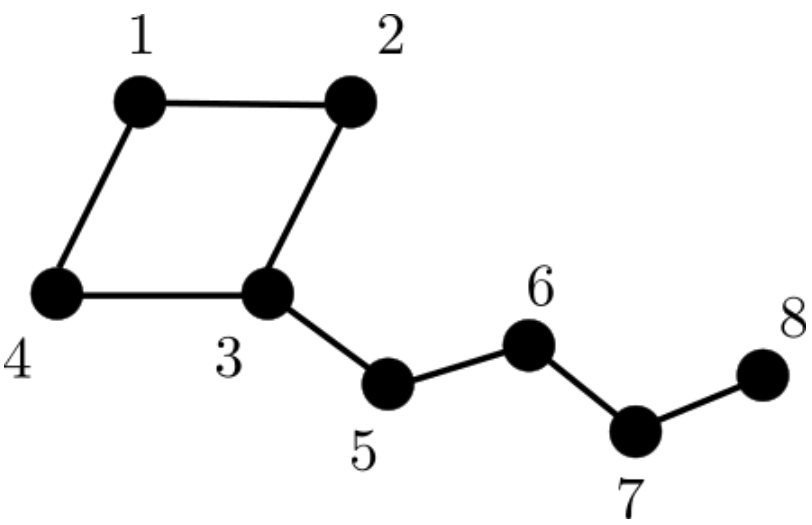
$$\hat{C} = \min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T L \mathbf{x}, \ \ \|\mathbf{x}\| = 1,$$

we now know that the optimal value of this problem is equal to 0 since there is an eigenvector (which can be normalized) with 0 eigenvalue. This solution is not satisfying for our clustering since the eigenvector(s) corresponding to the zero eigenvalue(s) pick out essentially the connected components of the graph. In particular, if there is only one component in the graph then $\mathbf{1}$ is the optimal vector that obtains the optimal cost of $\hat{C} = 0$ and this vector provides no information as to how to pick out different clusters in the graph.

Let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the n eigenvalues of L and let $\hat{\lambda}$ be the second smallest eigenvalue or the smallest non-zero eigenvalue of L. If there is only one connected component in the graph, this value is also equal to λ_2. The eigenvector corresponding to this eigenvalue, denoted $\hat{\mathbf{x}}$, turns out to be a good heuristic vector to cluster the graph according to the following rule: nodes whose corresponding x_i values are close to each other can be assigned a cluster. We can use this procedure to obtain any number of clusters that we wish to in the graph. With $\hat{\mathbf{x}}$ the cost of the relaxed clustering problem becomes $\hat{\mathbf{x}}^T L \hat{\mathbf{x}}$.

Spectral Clustering - I

8/8 points (graded)
Consider the kite graph shown in the figure.



7: A Kite Graph.

Compute the eigenvector corresponding to the second smallest eigenvalue. You may use any computational tool at your disposal. We recommend `networkx.linalg.algebraicconnectivity.fiedler_vector`. Ensure your eigenvector is normalized to $\sqrt{\sum_i v_i^2} = 1$. Round your values to **three significant figures**.

Node 1: -0.37147146 ✔ **Answer:** -0.3715

Node 2: -0.33561673 ✔ **Answer:** -0.3356

Node 3: -0.23497399 ✔ **Answer:** -0.235

Node 4: -0.33561673 ✔ **Answer:** -0.3356

Node 5: 0.01167124 ✔ **Answer:** 0.0117

Node 6: 0.25606344 ✔ **Answer:** 0.256

Node 7: 0.45102474 ✔ **Answer:** 0.451

Node 8: 0.55891948 ✔ **Answer:** 0.5589

Solution:

Python:

```
A = [[0, 1, 0, 1, 0, 0, 0, 0],
    [1, 0, 1, 0, 0, 0, 0, 0],
    [0, 1, 0, 1, 1, 0, 0, 0],
    [1, 0, 1, 0, 0, 0, 0, 0],
    [0, 0, 1, 0, 0, 1, 0, 0],
    [0, 0, 0, 0, 1, 0, 1, 0],
    [0, 0, 0, 0, 0, 1, 0, 1],
    [0, 0, 0, 0, 0, 0, 1, 0]]
graph = networkx.from_numpy_matrix(np.array(A), create_using=networkx.Graph)
networkx.linalg.algebraicconnectivity.fiedler_vector(graph)
```

Mathematica:

```
A = {{0, 1, 0, 1, 0, 0, 0, 0},
    {1, 0, 1, 0, 0, 0, 0, 0},
    {0, 1, 0, 1, 1, 0, 0, 0},
    {1, 0, 1, 0, 0, 0, 0, 0},
    {0, 0, 1, 0, 0, 1, 0, 0},
    {0, 0, 0, 0, 1, 0, 1, 0},
    {0, 0, 0, 0, 0, 1, 0, 1},
    {0, 0, 0, 0, 0, 0, 1, 0}}
{evals, evecs} = Eigensystem[DiagonalMatrix[Total[A]] - A]
evecs[[ -2 ]]/Sqrt[ Total[evecs[[ -2 ]]^2 ]]
```

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Spectral Clustering - II

2/2 points (graded)
Answer the following questions based on the eigenvector computed above:

1. If we clustered the kite graph into two clusters each of 4 nodes, which of the following would be in the cluster with node 1?

- [x] 2
- [x] 3
- [x] 4

- [ ] 5
- [ ] 6
- [ ] 7
- [ ] 8

✔

2. If we instead had only 3 nodes in the cluster that contained node 1, which of the following nodes will make it into that cluster?

- [x] 2
- [ ] 3
- [x] 4
- [ ] 5
- [ ] 6
- [ ] 7
- [ ] 8

✔

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Modularity Maximization

We can approach clustering via several objectives. An objective that is similar to the ideal clustering problem that we saw earlier is **modularity maximization** (we recommend the learner to review the definition of modularity of a graph). Formally, modularity maximization in two groups can be formulated by the following problem:

$$C = \max_{\mathbf{s}\in\{-1,1\}^n} \sum_{i,j} B_{i,j}\left(1 + s_i s_j\right), \quad \text{where } B_{ij} = A_{ij} - P_{ij}, \quad \text{such that } \sum_k s_k = n_1 - n_2.$$

Here, P_{ij} is the expected number of edges between nodes i, j in a random graph model. We will study random graph models in the next lecture, but we provide the expressions for a few models here:

- Erdos-Renyi: $P_{ij} = \frac{2m}{n(n-1)}$.
- Configuration model: $P_{ij} = \frac{k_i k_j}{2m-1}$.

Once again, the integer-valued optimization problem is computationally hard and we need to resort to approximate relaxations of the problem (see lecture slides for Louvain method by Blondel et al., 2008).

Discussion

Hide Discussion

Topic: Module 3: Network Analysis:Spectral Clustering / 7. Spectral Clustering

Topic: Module 3: Network Analysis:Spectral Clustering / 7. Spectral Clustering

Add a Post

All Posts

Spectral Clustering - I

question posted 2 months ago by **Thekindlyone**

I've used two different approaches to calculate the eigenvector (using the proposed method and through the Laplacian) and got exactly the same result with exception of the sign.

- Q1: Does the sign bear any additional meaning beyond clustering into two groups?
- Q1: I got all the values correct except for the Node 5. What could I be doing wrong? (I've rounded all numbers to 3 significant digits)

Thanks!

This post is visible to everyone.

MaksM (Community TA)
2 months ago - marked as answer 2 months ago by **lam_trinh** (Community TA)

Q1: It's the usual deal - an eigenvector of length 1 is only defined up to sign (in general, any multiple of an eigenvector is an eigenvector, requiring unit length removes most of this ambiguity, but not all). There is no principled way to choose one of the two such eigenvectors.

Q2 (presumably): As a guess, you rounded to a fixed number of digits, not significant digits. Since node 5 has smaller value, and hence more leading zeros (which are digits, but not significant ones), it ended up less precisely reported than the rest (relative to its magnitude, which is what significant digits are meant to capture). Maybe try reporting more digits on it (despite what the bold text, it seems to actually mean "at least 3 significant digits"; I reported 7 significant digits in each value and the grader accepted this).

Thank you for the explanation MaksM, this makes sense. And your comment about the grader and 3 significant digits was bang on. Thank you.

posted 2 months ago by **Thekindlyone**

Add a comment

Add a Response

2 other responses

RogerZeng
2 months ago

Looks like the eigenvetor dimension 3 and 4 got flipped around in the answer.

I think the answer is correct. If you use networkx: fiedler vector is presented in the node order while node order may be different from 1, 2, 3....8. You can check this with eg. G.nodes(). (If you provided an edge list, node order appears to follow the individual new nodes added in subsequent pairs.)

posted 2 months ago by **horvathsz_hu**

Add a comment

bhbenam
2 months ago

I created and plotted the adjacency matrix of the kite graph and it looks the same, only the node starts with zero

I created and plotted the adjacency matrix of the kite graph and it looks the same, only the node starts with zero instead of 1. I used the function fiedler_verctor from networkx and got the fiedler vector with 8 eigenvalues. But graders refuse my all answers. shall we sort them? the first four are negative and the last four are positive. I also tried with 6 significant digits. Any tip?

Put the parameter normalized in **False** not **True**

```
nx.linalg.algebraicconnectivity.fiedler_vector(G, normalized=False)
```

posted 2 months ago by **destinyZ**

Thanks @destinyZ, it works. why? actually we had to calculate normalized values. They are normalized with normalized = False, strange?

posted 2 months ago by **bhbenam**

Wondering if there is any hint to check the answers to spectral clustering 1. I used the eigenvectors of L and the eigenvector corresponding to my second smallest eigenvalue seems to make sense numerically, but the grader is marking all of them wrong. It may help to have more attempts than 3 for this problem also for the future. thanks.

posted 2 months ago by **simon-templar**

Hello TAs: I want to draw your attention to a comment made by one of you earlier: Q1: It's the usual deal - an eigenvector of length 1 is only defined up to sign (in general, any multiple of an eigenvector is an eigenvector, requiring unit length removes most of this ambiguity, but not all). There is no principled way to choose one of the two such eigenvectors. This could be critical for spectral clustering 1 as the fiedler vector provides an eigenvector that is opposite in sign to the eigen vector computed from first principles from L = D-A. The grader is only accepting the fiedler version. This is probably ok but I think the reversed sign is correct as well? Not sure. Can some one respond with technical detail as to this? Thank you for your help.

posted 2 months ago by **simon-templar**

Node order. Just in case anyone else gets initially caught as I was. In my browser (Firefox), the kite graph renders as a very small clump of dots. so small I did not notice the nodes are actually labelled. Therefore, I guessed at the labels and got all right except 3 and 4 wrong way round, so my answers to these were flipped from the grader's. Seeing an earlier post (RogerZeng – thanks!) about flipped nodes tipped me off to make correction. Then it occurred to me to use a magnifier tool to look closely again at graph and then I saw the labels!

posted 2 months ago by **Kg33wh**

Hi @simon-templar, according to my understanding of eigenvector, then yes, either sign should be okay. If x is an eigenvector, then so is -x.

$$A(-x) = -Ax = -\lambda x = \lambda(-x)$$

My hypothesis: I think the reason why the grader didn't accept your Fiedler values computed from L = D - A has to do with the values that were programmed into the answer boxes. The answer boxes expect certain exact values that were programmed in there.

I'll let the staff chime in if they see this post as they'd probably know for sure.

posted 2 months ago by **lam_trinh** (Community TA)

@destinyZ Thanks!!!

posted 2 months ago by **MPaulaWilde**

Add a comment

Showing all responses

Add a response:

Preview

Submit

‹ Previous Next ›

edX

About
Affiliates
edX for Business
Open edX
Careers
News

Legal

Terms of Service & Honor Code
Privacy Policy
Accessibility Policy
Trademark Policy
Sitemap

Connect

Blog
Contact Us
Help Center
Media Kit
Donate

GET IT ON Google play
Download on the App Store