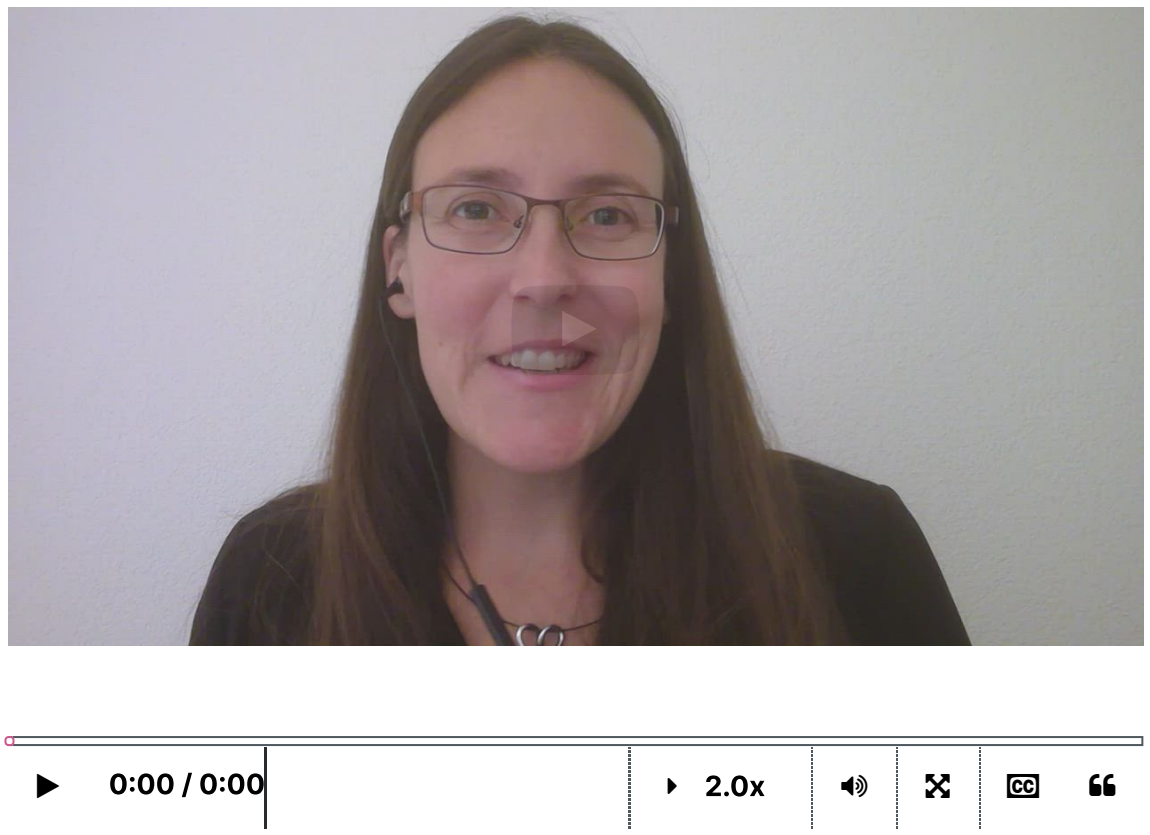
3. Steiner trees

Bookmark this page

Exercises due Oct 20, 2021 17:29 IST Completed

Cooffending networks, Steiner trees

Start of transcript. Skip to the end.



Prof Uhler: OK, welcome back to this last lecture

of this networks module, where we're

discussing other ways of analyzing the criminal network

that you're actually getting to analyze for your problem set.

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Introduction to Steiner trees

When examining a network, we may have identified a few nodes of particular interest. We may then want to ask the question: what is the smallest sub-network that connects all of these interest nodes? If we define “smallest" to mean the sum of all edge weights in the sub-network, then the this problem is known as the Steiner tree problem in graphs.

For example, we may have a weighted graph where each node is a city, and the weighted edges represent the costs to build an electrical distribution line between the connected cities. If the goal is to connect all cities on the graph while minimizing costs, then we should find the minimum spanning tree of the graph — which can be done in polynomial time.

However, suppose the goal is instead to connect just a few nodes of this graph together at minimum cost, perhaps due to a mandate or government contract. Then, the problem is the Steiner tree problem, and the computational complexity for finding the solution is NP-hard.

The cities that must be connected are called the terminals of the graph. The difficulty of the problem comes from the need to decide which additional cities should be used to create the subgraph that connects these terminals. If the mandate specified a few cities, and required that no other cities should be connected, then we could just discard the remaining nodes from the graph and the problem reduces to the minimum spanning tree.

Steiner graph example

1/1 point (graded)
Consider the following adjacency matrix:

$$A = \begin{pmatrix} 0 & 3 & 0 & 5 & 2 & 0 \\ 3 & 0 & 5 & 0 & 2 & 0 \\ 0 & 5 & 0 & 3 & 0 & 2 \\ 5 & 0 & 3 & 0 & 0 & 2 \\ 2 & 2 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

$$\begin{pmatrix} 0 & 0 & 2 & 2 & 2 & 0 \end{pmatrix}$$

Draw this graph for yourself, including the edge weights (which are represented by value of the corresponding element of the adjacency matrix).

Let the node labels be $i = 1, 2, 3, 4, 5, 6$, where $i = 1$ is the node in the left-most column, and $i = 6$ is the node in the right-most column. Now, the terminals of this graph will be defined as $\{1, 2, 3, 4\}$.

Find the Steiner tree by hand. Remember that the Steiner tree is a subgraph of A, that must include the terminals (nodes $1, 2, 3$, and 4), that minimizes the sum of all edge weights in the graph such that all terminals are part of a connected component. Additional nodes (nodes 5 and 6) may be added to the subgraph if they allow for a more minimal tree that spans the terminals.

Which nodes are part of the Steiner tree for this graph?

- [x] Node 1
- [x] Node 2
- [x] Node 3
- [x] Node 4
- [x] Node 5
- [x] Node 6

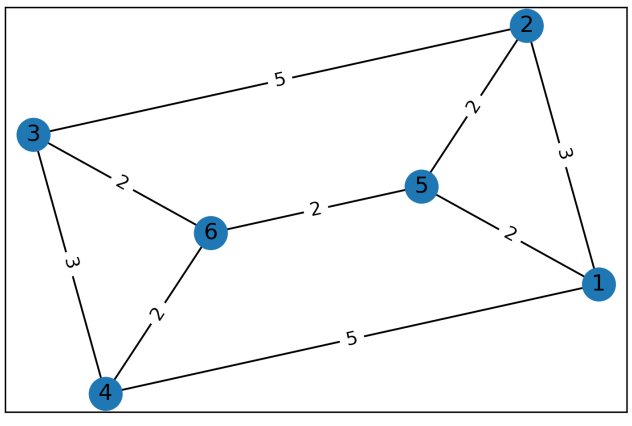
✔

Solution:

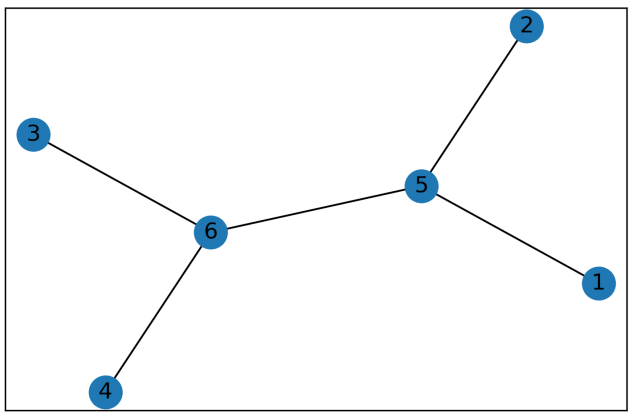
We can visualize the graph using the following Python:

```
A = [[0, 3, 0, 5, 2, 0],
    [3, 0, 5, 0, 2, 0],
    [0, 5, 0, 3, 0, 2],
    [5, 0, 3, 0, 0, 2],
    [2, 2, 0, 0, 0, 2],
    [0, 0, 2, 2, 2, 0]]
graph = networkx.from_numpy_matrix(np.array(A), create_using=networkx.Graph)
pos=networkx.kamada_kawai_layout(graph)
networkx.draw_networkx(graph,pos, labels={ i: str(i+1) for i in range(len(A)) })
networkx.draw_networkx_edge_labels(graph,pos,edge_labels=networkx.get_edge_attributes(graph,'weight'))
```

With gives the following visualization:



The Steiner tree for this graph with terminals $\{1, 2, 3, 4\}$ is shown below:



Submit You have used 2 of 4 attempts

Answers are displayed within the problem

Steiner graph sum

1/1 point (graded)
For the Steiner tree that you found in the previous problem, what is the sum of all edge weights of the tree?

10 ✔ **Answer:** 10

Solution:

From the answer to the previous problem, all edges in the Steiner tree have weight 2 and there are 5 edges, so the total is 10.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Approximation to the Steiner tree problem

6/6 points (graded)
Although finding the Steiner tree for the previous problem was relatively easy, the NP-hard aspect of the problem makes an exact solution an unrealistic goal for most real world problems. Rather than attempting to the find the true minimal tree, we can instead relax the problem to finding an approximately minimal tree. The following algorithm describes one such approach.

First, we need to compute the graph distance matrix, $\boldsymbol{D}$, for just the terminal nodes. Each element of the distance matrix is the smallest path length between two nodes; that is, $D_{i,j} = d_{i,j}$ in the usual notation.

Compute this matrix for the adjacency matrix, $\boldsymbol{A}$, used in the previous two problems, and fill in the entires below (the matrix is symmetric, so you need only fill in the upper triangular part).

$\boldsymbol{D} =$

0	3 ✔ **Answer:** 3	6 ✔ **Answer:** 6	5 ✔ **Answer:** 5
0		5 ✔ **Answer:** 5	6 ✔ **Answer:** 6

0

3

✔ **Answer:** 3

0

Solution:

The full distance matrix is

$$\begin{pmatrix} 0 & 3 & 6 & 5 & 2 & 4 \\ 3 & 0 & 5 & 6 & 2 & 4 \\ 6 & 5 & 0 & 3 & 4 & 2 \\ 5 & 6 & 3 & 0 & 4 & 2 \\ 2 & 2 & 4 & 4 & 0 & 2 \\ 4 & 4 & 2 & 2 & 2 & 0 \end{pmatrix}.$$

We are only interested in the terminal nodes, which make up the upper left 4×4 block:

$$\boldsymbol{D} = \begin{pmatrix} 0 & 3 & 6 & 5 \\ 3 & 0 & 5 & 6 \\ 6 & 5 & 0 & 3 \\ 5 & 6 & 3 & 0 \end{pmatrix}.$$

Submit You have used 2 of 4 attempts

Answers are displayed within the problem

Approximate Steiner tree 2

1/1 point (graded)

We can now use this distance matrix, $\boldsymbol{D}$, as an adjacency matrix for a new complete graph.

From this matrix, we can create a minimal tree than spans all the terminal nodes by finding a minimal spanning tree.

Find the minimal spanning tree for $\boldsymbol{D}$, either by hand, or use the **networkx** function `networkx.algorithms.tree.mst.minimum_spanning_tree`.

What is the sum of all the edge weights in the minimal spanning tree?

11

✔ **Answer:** 11

Solution:

Python:

```
D = [[0, 3, 6, 5],
    [3, 0, 5, 6],
    [6, 5, 0, 3],
    [5, 6, 3, 0]]
graphD = networkx.from_numpy_matrix(np.array(D), create_using=networkx.Graph)
posD=networkx.kamada_kawai_layout(graphD)
mst = networkx.algorithms.tree.mst.minimum_spanning_tree(graphD)
print(mst.size(weight='weight'))
```

Which prints a sum of edge weights of 11.

Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Approximate Steiner tree 3

1/1 point (graded)
Finally, we need to map this minimal spanning tree back to the original graph, A.

If an edge, D_{ij}, is part of the minimum spanning tree, then we find the shortest part between node i and node j in A. This shortest path is a list of edges in A, so we add all of these edges to our Steiner graph approximation S.

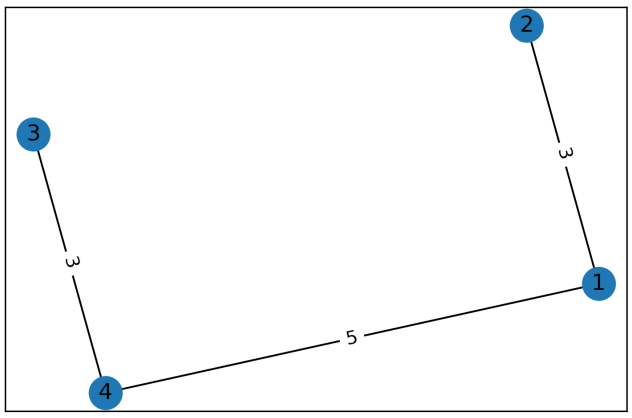
We then repeat this for all edges in the minimum spanning tree, until we have connected all terminals in A to form a tree in S.

Do this for your solution to the minimal spanning tree in the previous question. What can you say about the edges in the approximation Steiner tree and the exact Steiner tree you found earlier?

- All but one of the edges in the exact Steiner tree are in the approximation.
- Most of the edges in the exact Steiner tree are in the approximation.
- None of the edges in the exact Steiner tree are in the approximation. ✔

Solution:

Here we have the approximate solution:



We can see that this subgraph, and the subgraph for the exact solution are completely disjoint in the edges! Despite this, they only differ by 10% (11 vs 10) in the sum of all edge weights.

In fact, it requires a very artificial graph to get this kind of disjoint approximate solution. Changing any of the edge weights even a little causes the exact and approximate solutions to come into closer alignment. For most real world problems, the approximation is good.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Discussion

Hide Discussion

Topic: Module 3: Network Analysis:Spectral Clustering / 3. Steiner trees

Add a Post

All Posts

More than one way to solve approximate Steiner tree 2?

question posted 2 months ago by **DWKOC**

Is there more than one correct solution to approximate Steiner tree 2? For example, my tree looks like:

1→2 (weight=3) 2→3 (weight=5) 3→4 (weight=3)

I think this is correct, and gives me the correct answer for part 2, but now the given solution for part 3 is no longer valid

This post is visible to everyone.

Add a Response

1 response

vilgalys (Staff)
2 months ago

There should just be one approximation to the steiner tree, if you're using this method - maybe check how you're constructing the minimal spanning tree?

Also, in the future, try to avoid sharing your solution, even if you think there's a problem with it.

Could you please clarify that in 18:48 prof. mentioned that it's possible to have multiple spanning trees. Does it contradict to your above comment? Or did you mean there's only one approx. steiner tree for this particular example?

On the other hand, I think this prolem is symmetric so I found both approximation give the same total weight. Is there any consideration that restrict us from not choosing the other same weight spanning tree? Thank you

posted 2 months ago by **trungaero**

Okay, good clarification and sorry for being misleading - the minimum spanning tree is not guaranteed to be unique, although generally the python package should deliver a consistent result. Since the weight is the same, you've found another acceptable minimum spanning tree and solution to the relaxed problem.

The problem is not in the spanning tree you found, but in the comparison between this tree and the optimal Steiner tree. Happy to discuss more after the deadline if you're not able to resolve this issue.

posted 2 months ago by **vilgalys** (Staff)

@vilgalys: Thank you for your clarification!

posted 2 months ago by **trungaero**

Add a comment

Showing all responses

Add a response:

Preview

Submit

‹ Previous Next ›