Анализ сложных сетей.
Домашнее задание 2.

Анвар Курмуков

7 Ноября

1 Прежде чем начать 2

1. Всего баллов за дз: 105+.
2. Дата выдачи дз: 07.11.2019.
3. Мягкий делайн: 21.11.2019.
4. Жесткий дедлайн: 28.12.2019
   (-5 баллов из 110 за каждый день просрочки).
5. Все даты указаны по UTC+3.

1.1 Куда и что присылать

1. Сюда → kurmukovai@gmail.com.
2. **Тема письма** → Имя-Фамилия-complex-networks-HW2.
3. Присылать сформированный из .ipynb файла **.pdf документ**, рядом прикладывать сам .ipynb.

2 Network community detection

Given graph G with n nodes, find non-overlapping node "communities": k groups of nodes that are densely intra connected and have low number of inter connections.

2.1 Spectral clustering

2.1.1 Algorithm.

1. Compute square diagonal matrix of node degrees D.

$$D_{ii} = \sum_i A_{ij}, D_{ij} = 0, i \neq j$$

2. Construct graph Laplacian

$$L_{unnormed} = D - A$$

3. Find $0 = \lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ smallest eigenvalues of L and construct matrix X by stacking m corresponding eigenvectors $(v_1, \ldots v_m)$ as columns of X. Matrix X has size $n \times m$, its rows are "spectral representaion"of graph nodes.

4. Run k-means algorithm on matrix X and assign nodes with labels obtained by k-means.

2.1.2 Task. Total 35

1. Implement this algorithm using different graph Laplacians:
   (a) Unnormalized Laplacian: $L = D - A$, (4 points)
   (b) Symmetric normalization: $L_{sym} = I - D^{-\frac{1}{2}} \cdot A \cdot D^{-\frac{1}{2}}$, (4 points)
   (c) Random Walk normalization: $L_{rw} = I - D^{-1} \cdot A$, (4 points)

2. Theoretical questions:
   (a) From network point of view, what does Symmetric normalization do? (4 points)
   (b) Under what conditions Symmetric and Random walk normalizations yield the same result? (4 points)

3. Compare results of 1(a), 1(b) and 1(c) on Karate club network and artificial network (network.npy):

(a) Reorder rows and columns of these networks according to obtained clustering structure in order to obtain diagonal block structure of an adjacency matrix. (5 points)

(b) Using v_1 and v_2 as node coordinates draw them on 2D plot. (5 points)

(c) Using Adjusted Rand Index (from sklearn package). (5 points)

2.1.3 Materials.

1. Andrew Ng paper on spectral clustering `https://ai.stanford.edu/~ang/papers/nips01-spectral.pdf`

2. Tutorial on spectral clustering with multiple theoretical views on the problem `http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Luxburg07_tutorial_4488%5b0%5d.pdf`

3. Amazing explanation from James R. Lee `https://www.youtube.com/watch?v=8XJes6XFjxM&index=3&list=LLsQoc9tDb2toYF28XWGYRKQ&t=0s`

2.2 Label Propagation

Label propagation is neither the most accurate nor the most robust method. It is, however, without doubt one of the simplest and fastest clustering methods.[1] The idea of label propagation is very simple and kinda related to K-nearest neighbours approach. Given initial node coloring, algorithm "propagates"these colors to node's neighbours.

2.2.1 Algorithm.

Given graph G with n nodes:

1. Set $t = 0$

2. Initialize vector of labels $C(t)$ ($C_i(t)$ is the color of node i on t'th step) with numbers from 1 to n (each node has its own color/label)

3. Iterate over all nodes of a graph in a fixed (but random) order. For each node change it color $C_i(t)$) according to the most frequent color of it neighbours (including i itself). All ties are broken uniformly at random.

[1] `https://arxiv.org/pdf/1709.05634.pdf`

4. If every node has a label that the maximum number of their neighbors have, then stop the algorithm. Else increase t by 1 and repeat step 2.

This algorithm is not deterministic, the result hardly depends on the chosen traversal order and ties randomness. However it is amazingly (almost linear on number of nodes) fast and easily extents on weighted networks.

2.2.2 Task. Total 30

1. Implement label propagation algorithm (10 points)
2. Run label propagation on artificial network (network.npy) multiple times. Compare results of different runs (using Adjusted Rand Index and reordering). Compare results with the results of spectral clustering. (10 points)
3. Propose a reasonable approach to deal with the uncertainty (10 points)

2.2.3 Materials.

1. Label propagation review `https://arxiv.org/pdf/1709.05634.pdf`
2. Raghavan, Albert, Kumara original paper `https://arxiv.org/pdf/0709.2938.pdf`

2.3 Unsupervised image segmentation

You are asked to implement simple image segmentation algorithm. Segmentation is a task of partitioning image into different segments such that pixels in each segment are somehow similar/related.

2.3.1 Algorithm.

Given $n \times m$ RGB image:

1. Construct full graph with nodes corresponds to image pixels and edges are euclidean distance between pixel's intensities (image of $n \times m$ size yield graph with nm nodes so be carefull)
2. Cluster graph nodes using any community detection algorithm
3. Use nodes labels as image pixels labels. Reshape vector of labels (of size nm) into rectangular matrix of size $n \times m$.

2.3.2 Task. Total 40+

You task is to implement this segmentation algorithm and run it on different simple images.

1. Implement different (simple) variants of an algorithm (40+ points):
   (a) Compute edge weights using 3 dimensions (R and G and B) (10 points)
   (b) Construct 3 graphs for each color channel separately and then combine the resulting segmentations. (10 points)
   (c) Use gray scale image to compute graph weights. (10 points)
2. This algorithm is computationally expensive, for typical image of size 256 x 256 pixels you have graph with 65536 nodes and 2 147 450 880 edges. Propose an improvements to make your task computationally tractable (5 points each, no more than 15)
3. Run this algorithm on given images (apple.jpg, pineapple.jpg) and analyze the results. Use your ideas from the previous task. (10 points)

To combine different segmentation (1(b)) you may use the following approach. Given two different labelings C_1 and C_2 construct new labeling $C_{1,2}$ in a following way: $C_{1,2}(i) = \{C_1(i), C_2(i)\}$ (concatenate labels of C_1 and C_2). Such (aggregated) segmentation will have no more segments then C_1 and C_2 combined.