Wissenschaftliches Rechnen II/Scientific Computing II

Sommersemester 2016
Prof. Dr. Jochen Garcke
Dipl.-Math. Sebastian Mayer

universitätbonn

Exercise sheet 13 To be handed in on **Thursday, 21.07.2016**

Graph Laplacians

1 Group exercises

G 1. A square, nonnegative matrix P is called *right stochastic* if every row is summing to 1. The entry P_{ij} can be interpreted as the probability to jump from vertice i to vertice j in a graph in one time step. Show that all eigenvalues of P are at most 1 in absolute value. Further show that 1 is an eigenvalue of P. Explicitly determine the corresponding right eigenvector.

G 2. Let $G = (V, E, W)$ be an undirected graph with edge weight matrix W, $W_{ij} \geq 0$. A subset of vertices $A \subset V$ is called *connected component* if any two vertices in A can be joined by a path such that all intermediate points also lie in A. Let D be the diagonal matrix with $D_{ii} := d_i := \sum_{j=1}^{n} w_{ij}$. Show that 0 is an eigenvalue of the unnormalized graph Laplacian

$$L = D - W,$$

and that its multiplicity k equals the number of connected components $A_1, \ldots, A_k$ in the graph. Show that the eigenvectors of the eigenvalue 0 are given by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of the connected components.

Vergabe von Terminen für die mündliche Prüfung: Termine für die mündliche Prüfung werden in der Vorlesung am Dienstag, 19.07.2016, vergeben.

Appointments for oral exam: Appointments for the oral exam can be made in the lecture on Tuesday, 19.07.2016.

2 Homework

H 1. Let $G = (V, E, W)$ be a connected, undirected graph with symmetric, nonnegative edge weight matrix W. Let D be the diagonal matrix with $D_{ii} := d_i := \sum_{j=1}^{n} w_{ij}$. What is the difference between the p-dimensional embedding computed by the Laplacian Eigenmap algorithm and the embedding that you obtain by computing the eigenvalue decomposition of $D^{-1}W = V\Lambda V^T$ and using $X = I_{p\times n}\Lambda^{1/2}V^T$ as the p-dimensional embedding.

(5 Punkte)

H 2. Let $G = (V, E, W)$ be a undirected graph with $n = |V|$ vertices and symmetric, nonnegative edge weight matrix W. We want to partition G into k clusters such that cluster sizes are balanced, edges between different clusters have low weight and edges within a cluster have high weight. One approach to achieve this is *Ncut*. The goal of Ncut is to solve the optimization problem

$$\min_{A_1,\ldots,A_k\subset V, \bigcup A_k = V} \mathrm{Ncut}(A_1, \ldots, A_k), \tag{1}$$

where

$$\mathrm{Ncut}(A_1, \ldots, A_k) := \frac{1}{2}\sum_{i=1}^{k} \frac{W(A_i, \bar{A}_i)}{\mathrm{vol}(A_i)}$$

with $\bar{A}_i = V \setminus A_i$, $\mathrm{vol}(A_i) = \sum_{j\in A_i} d_j$, and $W(A_i, \bar{A}_i) = \sum_{j\in A_i, l\in \bar{A}_i} w_{jl}$.

a) Argue why $\mathrm{Ncut}(A_1, \ldots, A_k)$ is a reasonable objective function for the partitioning problem. You do not have to prove anything, just give plausible arguments.

b) For $k = 2$, show that (1) is equivalent to

$$\min_{A\subset V} f_A^T L f_A \quad \text{subject to} \quad \mathbb{1}_V^T D f_A = 0,\ f_A^T D f = \mathrm{vol}(V),$$

where $L = D - W$ is the graph Laplacian and the vector $f_A \in \mathbb{R}^n$ is given by

$$(f_A)_i = \begin{cases} (\mathrm{vol}(\bar{A})/\mathrm{vol}(A))^{1/2} & \text{if } i \in A, \\ -(\mathrm{vol}(A)/\mathrm{vol}(\bar{A}))^{1/2} & \text{if } i \in \bar{A}. \end{cases}$$

c) Generalize the result in b) for $k > 2$. That is, show that (1) is equivalent to

$$\min_{A_1,\ldots,A_k} \mathrm{tr}(H^T L H) \quad \text{subject to} \quad H^T D H = I_n, \tag{2}$$

where $H = H(A_1, \ldots, A_k)$ is a $n \times k$ matrix such that the ith column of H is the properly normalized indicator vector of A_i.

d) Unfortunately, the discrete optimization problem (1) is NP-hard. Therefore, one relaxes the discreteness condition in (2) and solves

$$\min_{H\in\mathbb{R}^{n\times k}} \mathrm{tr}(H^T L H) \quad \text{subject to} \quad H^T D H = I_n. \tag{3}$$

Assuming that the optimal solution U of (3) is a good approximation of the optimal solution of (2), how can you use k-means to obtain the desired graph partition from U?

(15 Punkte)