STATISTICAL LEARNING SYSTEMS

LECTURE 13: ON LOCAL CLUSTERING FOR MASSIVE GRAPHS

Jacek Koronacki

Institute of Computer Science, Polish Academy of Sciences

Ph. D. Program 2013/2014

The project is co-financed by the European Union within the framework of European Social Fund

On local clustering with Nibble

This section will be based on the paper by Daniel A. Spielman and Shang-Hua Teng, *A Local Clustering Algorithm for Massive Graphs and its Application to Nearly-Linear Time Graph Partitioning* (2008), and on a few more in which modifications of the algorithm were proposed.

During the lecture only some basic information will be provided, namely that on the relationship between clustering and conductance, random walk with a truncated version of the distribution; also the algorithm Nibble and its modifications will be sketched.

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

On local clustering with Nibble

Let $G = (V, E)$ be an undirected graph with $V = \{1, \dots, n\}$. Let $d(i)$ denote the degree of vertex i and, for $S \subset V$, let its volume $\mu(S) = \sum_{i \in S} d(i)$. So, $\mu(V) = 2|E|$ (we shall be assuming that $|E| = m$). Let $E(S, V - S)$ be the set of edges connecting a vertex in S with a vertex in $V - S$. Define the conductance of set S by

$$\Phi(S) = \frac{|E(S, V - S)|}{\min(\mu(S), \mu(V - S))}$$

Let $\mathbf{A}$ be the adjacency matrix:

$$\mathbf{A}(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \text{ and } u \neq v \\ k & \text{if u=v and this vertex has } k \text{ self-loops} \\ 0 & \text{otherwise} \end{cases}$$

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

On local clustering with Nibble

Define the following two vectors on a set of vertices S:

$$\chi_S(u) = \begin{cases} 1 & \text{for } u \in S \\ 0 & \text{otherwise} \end{cases}$$

and

$$\psi_S(u) = \begin{cases} d(u)/\mu(S) & \text{for } u \in S \\ 0 & \text{otherwise} \end{cases}$$

Consider the random walk that at each step stays at the current vertex with probability $1/2$ and otherwise moves to the endpoint of a random edge attached to the current vertex. The matrix realizing this walk equals $\mathbf{M} = (\mathbf{AD}^{-1} + I)/2$, where $\mathbf{D} = \mathrm{diag}(d(1), \ldots, d(n))$.

For a random walk which starts at a node v, its distribution at time t is $p_t = \mathbf{M}^t \chi_v$. ψ_V is the steady-state distribution and ψ_S is the restriction of that walk to the set S.

On local clustering with Nibble

Define the following truncation operation:

$$[p]_\epsilon(u) = \begin{cases} p(u) & \text{if } p(u) \geqslant d(u)\epsilon \\ 0 & \text{otherwise} \end{cases}$$

Nibble generates the following sequence of vectors (starting at χ_v):

$$q_t = \begin{cases} \chi_v & \text{if } t = 0 \\ Mr_{t-1} & \text{otherwise} \end{cases}$$

where

$$r_t = [q_t]_\epsilon.$$

For a vector q_t, let $S_j(q_t)$ be the set of j vertices u maximizing $q_t(u)/d(u)$. Letting t increase, the algorithm finds a j (if it exists) such that, for an a priori chosen ϕ, $\Phi(S_j(q_t)) \leqslant \phi$, the set $S_j(q_t)$ does contain neither too much nor too little volume and many elements of $S_j(q_t)$ have large probability mass. The $S_j(q_t)$ found is the cluster sought. So described algorithm is a basis for an efficient and well balanced clustering of a massive graph.

Digression on the Google PageRank Algorithm

Reid Andersen, Fan Chung and Kevin Lang, *Using PageRank to Locally Partition a Graph* (2007) proposed several improvements over the Nibble, their generic version being known as PageRank-Nibble Algorithm.

The original PageRank algorithm, as proposed by Page and Brin, considers a webpage to be important if many other webpages point to it. (In our exposition we closely follow [HTF].)

The linking webpages that point to a given page are not treated equally: the algorithm also takes into account both the importance (PageRank) of the linking pages and the number of outgoing links that they have.

Linking pages with higher PageRank are given more weight, while pages with more outgoing links are given less weight. These ideas lead to a **recursive** definition for PageRank:

PhD STUDIES KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
The project is co-financed by the European Union within the framework of European Social Fund

Digression on the Google PageRank Algorithm

Let $A_{ij} = 1$ if page j points to page i, and zero otherwise. Let $d_j = \sum_{i=1}^{N} A_{ij}$ equal the number of pages pointed to by page j (number of outlinks).
Then the Google PageRanks p_i are defined by the recursive relationship

$$p_i = (1-\beta)/N + \beta \sum_{j=1}^{N} \left(\frac{A_{ij}}{d_j} \right) p_j$$

where β is a positive constant (according to [HTF] apparently set to 0.85).

The idea is that the importance of page i is the sum of the importances of pages that point to that page. The sums are weighted by $1/d_j$, that is, each page distributes a total vote of 1 to other pages.

The project is co-financed by the European Union within the framework of European Social Fund

Digression on the Google PageRank Algorithm

In matrix notation

$$\mathbf{p} = (1-\beta)\mathbf{e}/N + \beta\mathbf{A}\mathbf{D}^{-1}\mathbf{p}.$$

Originally, the authors considered PageRank as a model of user behavior, where a random web surfer clicks on links at random, without regard to content. The surfer does a random walk on the web, choosing among available outgoing links at random. The factor $1-\beta$ is the probability that he or she does not click on a link, but jumps instead to a random webpage.

The page rank solution is the stationary distribution of an irreducible, aperiodic Markov chain over the N webpages (to see this note that $\mathbf{e}'\mathbf{p} = 1$).

The project is co-financed by the European Union within the framework of European Social Fund

Digression on the Google PageRank and the PageRank-Nibble

Remark 1: We may write

$$\mathbf{p} = (1-\beta)\mathbf{e}/N + \beta\mathbf{A}\mathbf{D}^{-1}\mathbf{p}$$

as

$$\mathbf{p}(\mathbf{s}) = (1-\beta)\mathbf{s} + \beta\mathbf{A}\mathbf{D}^{-1}\mathbf{p}(\mathbf{s}),$$

where

$$\mathbf{s} = \mathbf{e}/N.$$

Observe that nothing prevents us from switching for this uniform probability distribution **s** to any fixed probability distribution **s**, opening the way to personalized PageRank algorithm.

Remark 2: In the original PageRank, as wee see, the standard random walk matrix $\mathbf{A}\mathbf{D}^{-1}$ is used. Andersen et al. proved that PageRank with the lazy random walk matrix $M = (AD^{-1} + I)/2$ is equivalent to the original one up to a change in β.

Just a word on the PageRank-Nibble Algorithm

Most briefly put, PageRank-Nibble algorithms are versions of Nibble with the random walk borrowed from PageRank.

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY