Прикладные задачи анализа данных

Семинар 10

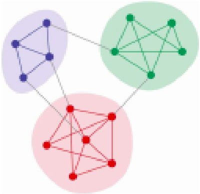
Спектральная кластеризация

Национальныи Исследовательский Университет
Высшая Школа Экономики

26 апреля 2018

Optimization criterion: modularity

- Let n_c - number of classes, c_i - class label per node



- Modularity:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

$$\delta(c_i, c_j) = \begin{cases} 1: & \text{if} \quad c_i = c_j \\ 0: & \text{if} \quad c_i \neq c_j \end{cases} \text{ - kronecker delta}$$

[Maximization!]

- Random network
  - consider an edge e attached to node i, degree k_i
  - probability that it is attached to node j, degree k_j is $k_j/2m$
  - expected number of edges (average) between i and j is $k_i k_j/2m$
- expected number of edges within the same class, c_i- class, $\delta(c_i, c_j)$- kronecker delta

$$\langle m_c \rangle = \frac{1}{2} \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j)$$

- Modularity:

$$Q = \frac{m_c - \langle m_c \rangle}{m} = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

- Modularity matrix:

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

- Single class, $\delta(c_i, c_j) = 1$, $Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) = 0$

- Direct modularity maximization (bi-partitioning), [Newman, 2006]
- For two classes C_1, C_2 indicator variable $s = \pm 1$

$$\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1) = \begin{cases} 1: & i,j \in C_1 \text{ or } i,j \in C_2 \\ 0: & i \in C_1, j \in C_2 \text{ or } i \in C_2, j \in C_1 \end{cases}$$

- Modularity

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{i,j} B_{ij} s_i s_j$$

where

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

- Qudratic form:

$$Q(\mathbf{s}) = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}$$

- Integer optimization - NP, relaxation $s \to x$, $x \in R$
- Keep norm $||x||^2 = \sum_i x_i^2 = \mathbf{x}^T\mathbf{x} = n$
- Quadratic optimization

$$Q'(\mathbf{x}) = \frac{1}{4m}\mathbf{x}^T\mathbf{B}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n)$$

- Eigenvector problem

$$\mathbf{B}\mathbf{x}_i = \lambda_i'\mathbf{x}_i$$

- Approximate modularity

$$Q'(\mathbf{x_i}) = \frac{n}{4m}\lambda_i$$

- Maximization - maximal λ

- Can't choose $\mathbf{s} = \mathbf{x}k$, can select optimal $\mathbf{s}$
- Decompose in the basis: $\mathbf{s} = \sum_j a_j \mathbf{x_j}$, where $a_j = \mathbf{x}_j^T \mathbf{s}$
- Modularity

$$Q(\mathbf{s}) = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s} = \frac{1}{4m}\sum_i (\mathbf{x}_i^T\mathbf{s})^2\lambda_i$$

- max $Q(\mathbf{s})$ reached when $\lambda_1 = \lambda_{max}$ and max $\mathbf{x}_1^T\mathbf{s} = \sum_j x_{1j}s_j$
- Choose $\mathbf{s} \parallel \mathbf{x}_1$, $\mathbf{s} = sign(\mathbf{x}_1)$

Algorithm: Spectral moduldarity maximization: two-way parition

Input: adjacency matrix $\mathbf{A}$

Output: class indicator vector $\mathbf{s}$

compute $\mathbf{k} = deg(\mathbf{A})$;

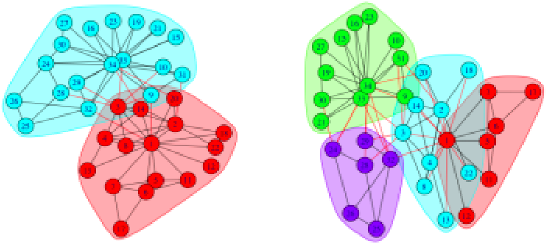
compute $\mathbf{B} = \mathbf{A} - \frac{1}{2m}\mathbf{k}\mathbf{k}^T$;

solve for maximal eigenvector $\mathbf{B}\mathbf{x} = \lambda\mathbf{x}$;

set $\mathbf{s} = sign(\mathbf{x}_1)$

Recursive bisection

- Compute modularity matrix $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$
- Solve for maximal eigenvalue $\mathbf{Bx} = \lambda \mathbf{x}$
- set $\mathbf{s} = sign(\mathbf{x}_1)$
- recurse on each partition



Скачиваем тут

При подготовке семинара использовались

- материалы лекции Леонида Жукова