

Атрошенко 11.06.2017 : 11.23

Плотность графа

$$p = \frac{m}{n(n-1)/2}$$

Плотность ребер в графе (m – кол-во ребер, n – кол-во вершин)

Плотность сообщества

$$p_{in}(c) = \frac{m_c}{n_c(n_c-1)/2}$$

Аналогичные рассуждения для сообщества

$$\max(p_{in} - p_{out})$$

Идеальное разбиение на сообщества достигается при
максимуме разности внутренней и внешней плотностей

Центральность по посредничеству

$$C(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

$\sigma_{st}(e)$ - количество кратчайших путей между вершинами s и t проходящих через ребро e

σ_{st} - количество кратчайших путей между вершинами s и t

Модулярность

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

$$\delta(c_i, c_j) = \begin{cases} 1 : c_i = c_j \\ 0 : c_i \neq c_j \end{cases}$$

K_i – степень i -ой вершины, m – количество(сумма) ребер, A_{ij} – матрица смежности, C_i – номер сообщества i -ой вершины. Модулярность – метрика качества разбиения графа на сообщества, чем она выше, тем разбиение лучше.

Костищина 11.06.2017: 12.37 - когда и как вы меняете номер сообщества каждой вершины в алгоритме?

На 3ем шаге алгоритма (напомню)

3) То ребро, центральность которого максимальна, удаляется из графа. В случае, когда таких рёбер в графе несколько, одно из них выбирается случайным образом. Если рёбер в графе нет, то алгоритм завершает свою работу.

я удаляю ребро в графе. После удаления проверяю: если 2 смежные с этим ребром вершины оказались в разных компонентах связности, то одну из этих компонент называю новым сообществом. (Одну, т.к. другая будет иметь номер сообщества этой компоненты связности до удаления ребра)

Хетеева 11.06.2017 15:46 - В 4 шаге Вашего алгоритма говорится, что он завершает свою работу , когда значение модулярности уменьшилось. Есть ли какие-то альтернативные условия завершения работы алгоритма?

Да, существуют разные варианты.

1. Заканчивать работу, когда будут удалены все ребра, тогда разбиение, при котором достигается максимум модулярности, будет наилучшим.
2. Заканчивать работу, когда достигнут локальный максимум модулярности, а не после 1ого её падения, как написано у меня.

Исаева 11.06.17 15:52 В первом шаге алгоритма Вы изначально разбиваете граф на сообщества, где каждая компонента связности - сообщество, если исходный граф не связный. Не могли бы Вы подробнее пояснить этот шаг, а лучше написать его алгоритм?

На случай несвязного графа в начале моей программы есть цикл, который пробегает по всем вершинам и, если она не принадлежит никакому сообществу, присваивает ей номер очередного сообщества и запускает для нее функцию, которая обходом в глубину присваивает этот номер всем вершинам, в которые можно попасть из данной.