Анализ социальных сетей

Методы выделения сообществ

Славнов Константин

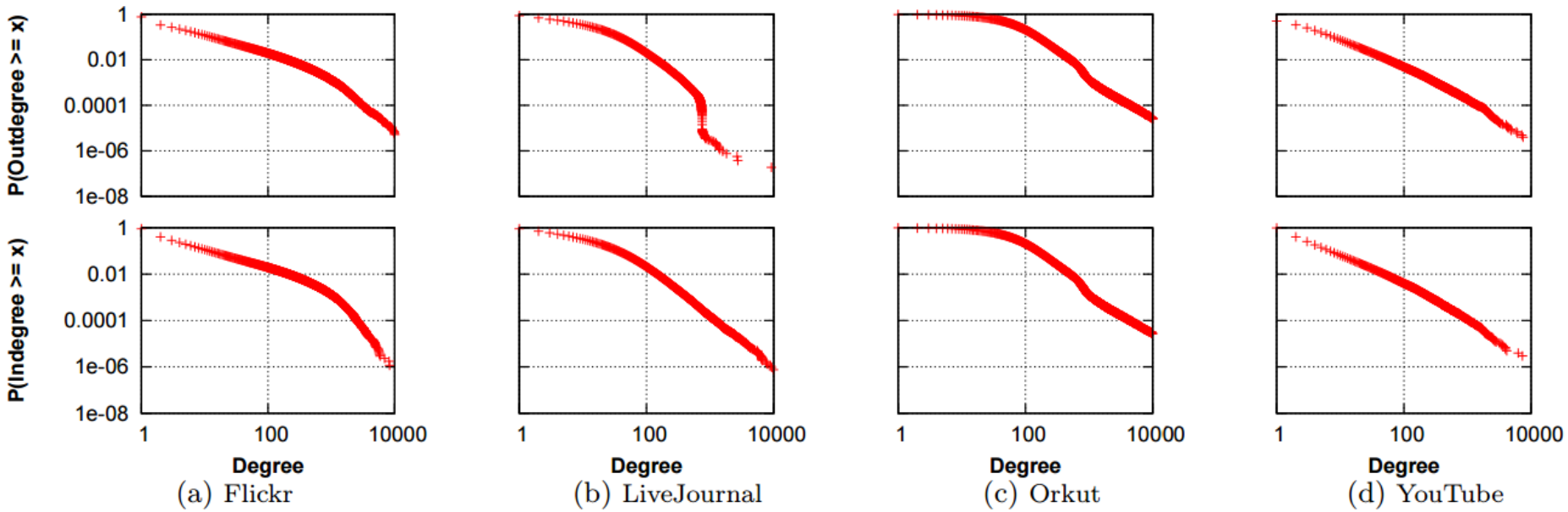
11.11.2015

НИУ ВШЭ, ИППИ РАН (Москва, Россия)

Свойства социальных сетей

- **Одна большая общая компонента связности**
  - **Facebook – 99.91%**

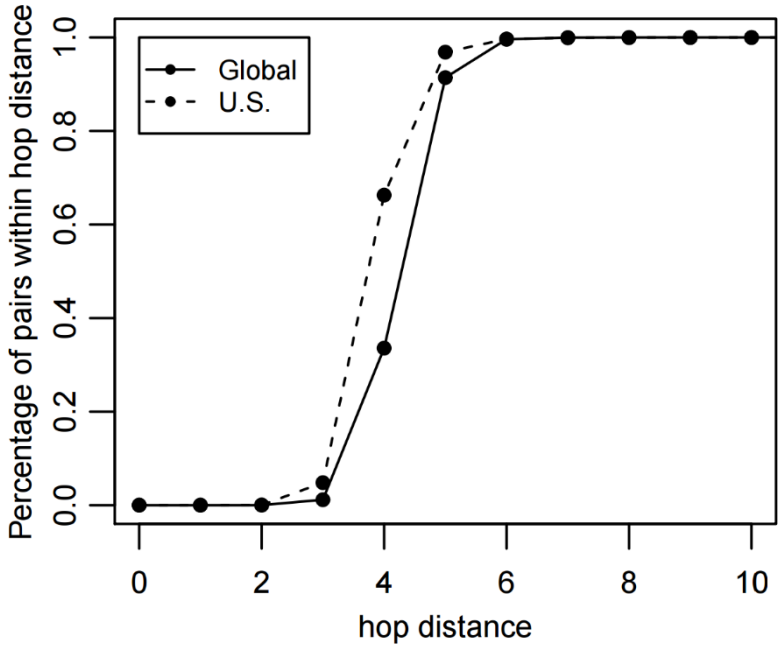
- **Распределение на степенях вершин**
  - **Scale-free**



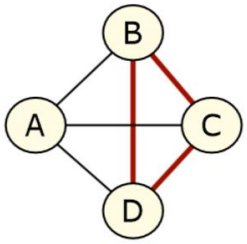
(a) Flickr (b) LiveJournal (c) Orkut (d) YouTube

Свойства социальных сетей

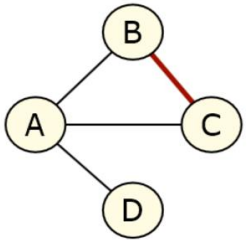
- **Среднее расстояние**



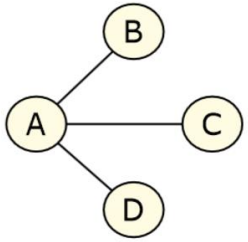
- **Коэффициент кластеризации**



$$CC(A) = \frac{3}{3}$$



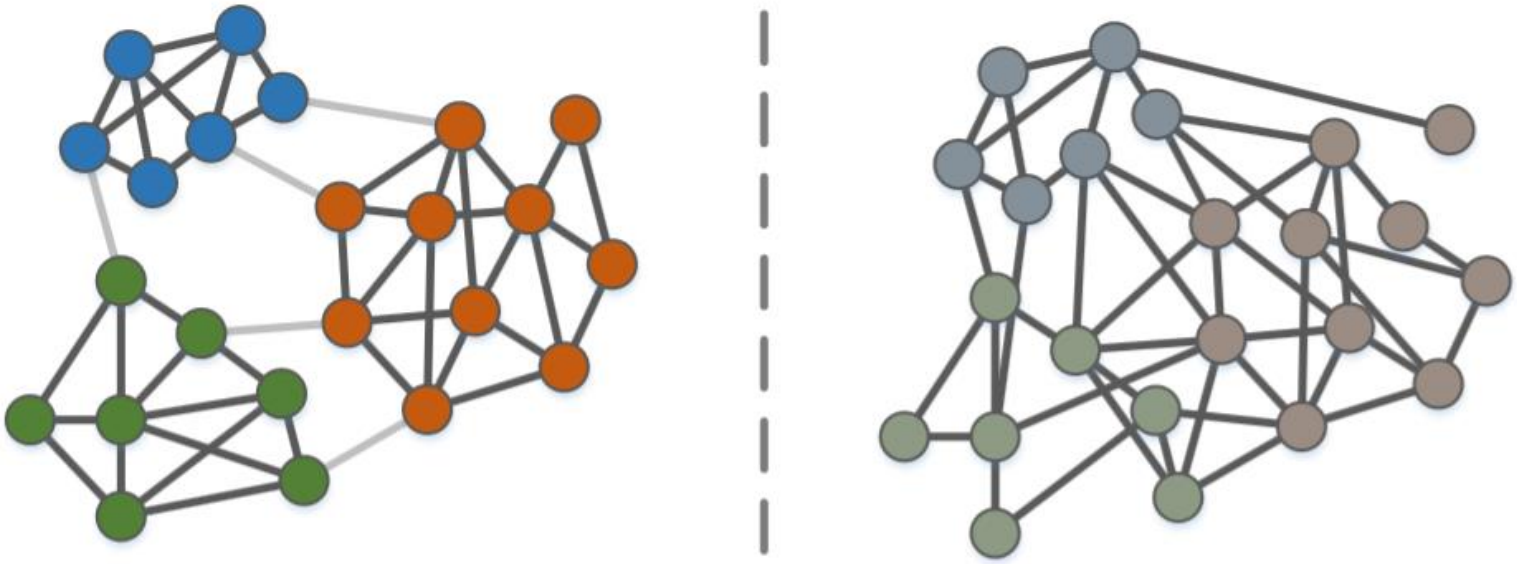
$$CC(A) = \frac{1}{3}$$



$$CC(A) = \frac{0}{3}$$

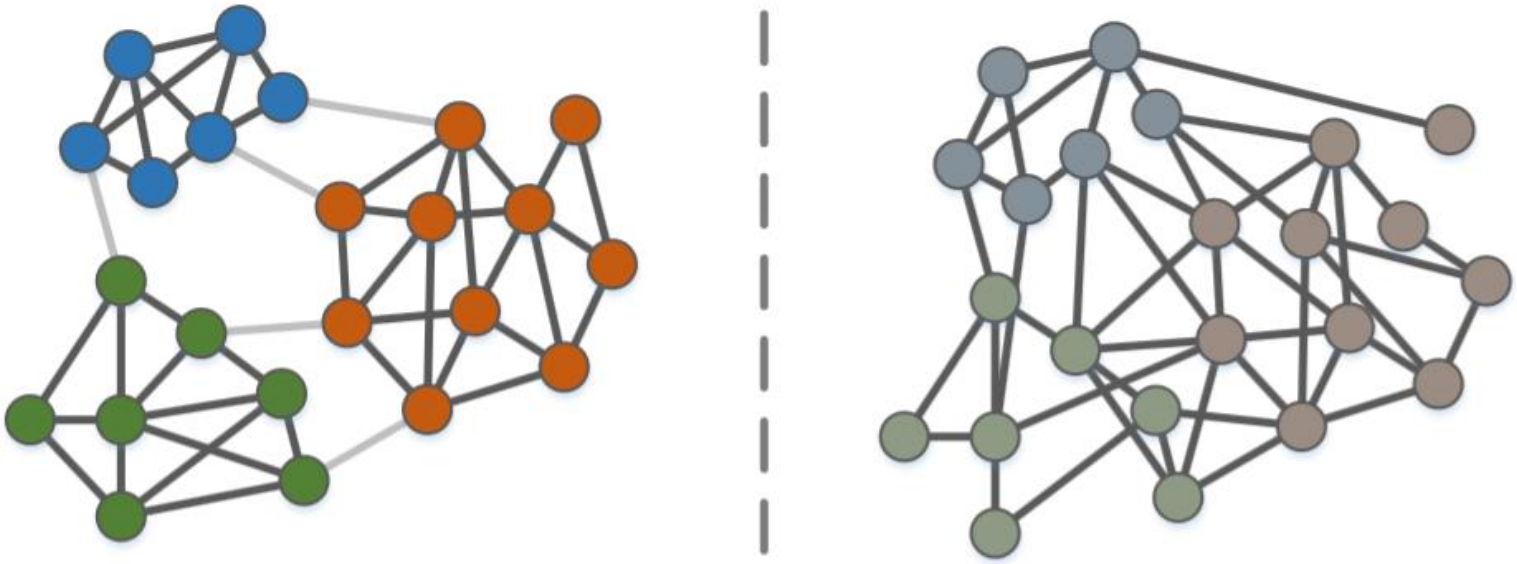
Свойства социальных сетей

- **Структура сообществ**



Сообщество

Связи внутри группы гораздо плотнее межгрупповых



Как формализовать задачу выделения сообществ?

Задача выделения сообществ

$G = (V, E)$ – граф

A_{ij} – матрица смежности,

d_i – степень i-ой вершины,

C_i – сообщество i-ой вершины

$m = |E|$ – количество ребер

1. Поиск разбиения, максимизирующего функционал:

$$\mathbb{V} = \{V_k\} \text{ – разбиение на } k \text{ сообществ}$$

$$V_k = \{i \in V, j \in V \mid C_i = C_j\}$$

2. Задача кластеризации вершин:

$$D_G(i,j) \text{ – расстояние на вершинах}$$

Задача выделения сообществ

$G = (V, E)$ – **граф,**

A_{ij} – **матрица смежности,**

d_i – **степень i-ой вершины**

C_i – **сообщество i-ой вершины,**

$m = |E|$ – **количество ребер**

1. **Поиск разбиения, максимизирующего функционал:**

$$\mathbb{V} = \{V_k\} \text{ – разбиение на } k \text{ сообществ}$$

$$V_k = \{i \in V, j \in V \mid C_i = C_j\}$$

2. **Модулярность:**

$$Q = \frac{1}{2m} \sum_{(i,j) \in E} (A_{ij} - \frac{d_i d_j}{2m}) \delta(C_i, C_j)$$

Модулярность

$\boldsymbol{G = (V, E)}$ **– граф,**

A_{ij} **– матрица смежности,**

d_i **– степень** $\boldsymbol{i}$**-ой вершины**

$\boldsymbol{C_i}$ **– сообщество** $\boldsymbol{i}$**-ой вершины,**

$m = |E|$ **– количество ребер,**

$\delta(C_i, C_j)$ **– дельта-функция**

Модулярность:

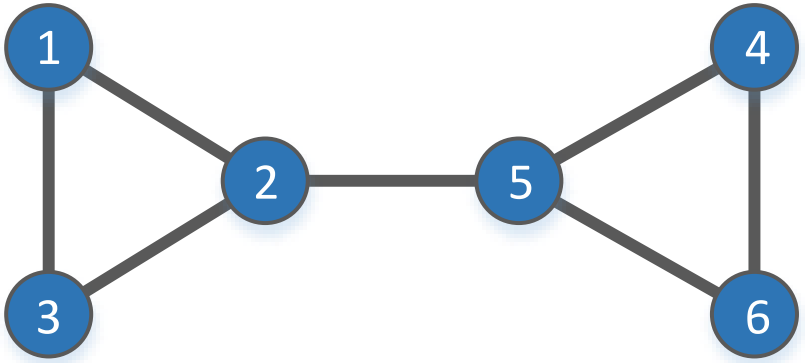
$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j), \quad Q \in [-1,1]$$

Интерпретация:

Разность между долей ребер внутри сообщества и ожидаемой доли связей в случайном графе.

Дискретная задача оптимизации!

Модулярность



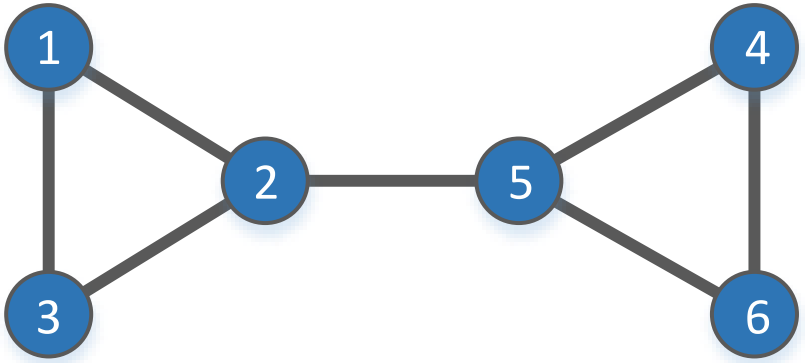
$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j)$$

$$\begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & \\ & & & 0 & 1 & 1 \\ & 1 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} - \frac{1}{2 \cdot 6} \cdot \begin{pmatrix} 4 & 6 & 9 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \end{pmatrix} = \frac{1}{12} \cdot \begin{pmatrix} -4 & 6 & 8 & -4 & -6 & -4 \\ 6 & -9 & 6 & -6 & 3 & -6 \\ 8 & 6 & -4 & -4 & -6 & -4 \\ -4 & -6 & -4 & -4 & 6 & 8 \\ -6 & 3 & -6 & 6 & -9 & 6 \\ -4 & -6 & -4 & 8 & 6 & -4 \end{pmatrix}$$

Матрица Модулярности

Модулярность

Каждая вершина в своем сообществе:



$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j)$$

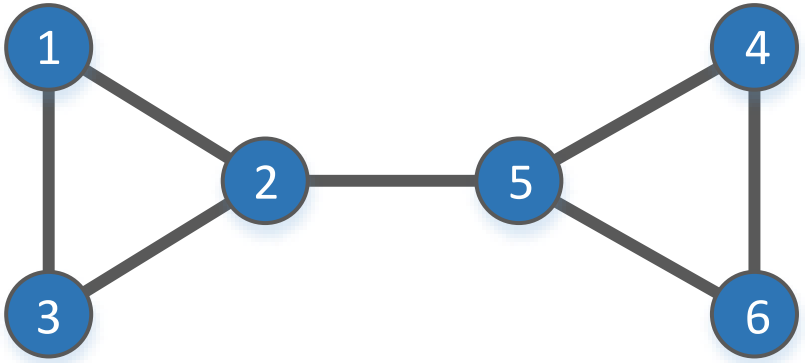
$$\begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & \\ & & & 0 & 1 & 1 \\ & 1 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} - \frac{1}{2 \cdot 6} \cdot \begin{pmatrix} 4 & 6 & 9 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \end{pmatrix} = \frac{1}{12} \cdot \begin{pmatrix} \mathbf{-4} & 6 & 8 & -4 & -6 & -4 \\ 6 & \mathbf{-9} & 6 & -6 & 3 & -6 \\ 8 & 6 & \mathbf{-4} & -4 & -6 & -4 \\ -4 & -6 & -4 & \mathbf{-4} & 6 & 8 \\ -6 & 3 & -6 & 6 & \mathbf{-9} & 6 \\ -4 & -6 & -4 & 8 & 6 & \mathbf{-4} \end{pmatrix}$$

Матрица Модулярности

$$Q = -\frac{34}{144}$$

Модулярность

Все вершины в одном сообществе:



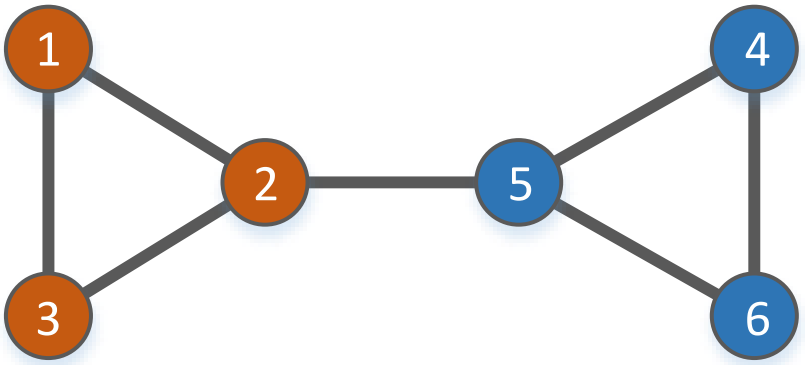
$$Q = \frac{1}{2m}\sum_{(i,j)\in E}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta(C_i, C_j)$$

$$\begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & \\ & & & 0 & 1 & 1 \\ & 1 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} - \frac{1}{2\cdot 6}\cdot \begin{pmatrix} 4 & 6 & 9 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \end{pmatrix} = \frac{1}{12}\cdot \begin{pmatrix} \mathbf{-4} & \mathbf{6} & \mathbf{8} & \mathbf{-4} & \mathbf{-6} & \mathbf{-4} \\ \mathbf{6} & \mathbf{-9} & \mathbf{6} & \mathbf{-6} & \mathbf{3} & \mathbf{-6} \\ \mathbf{8} & \mathbf{6} & \mathbf{-4} & \mathbf{-4} & \mathbf{-6} & \mathbf{-4} \\ \mathbf{-4} & \mathbf{-6} & \mathbf{-4} & \mathbf{-4} & \mathbf{6} & \mathbf{8} \\ \mathbf{-6} & \mathbf{3} & \mathbf{-6} & \mathbf{6} & \mathbf{-9} & \mathbf{6} \\ \mathbf{-4} & \mathbf{-6} & \mathbf{-4} & \mathbf{8} & \mathbf{6} & \mathbf{-4} \end{pmatrix}$$

$$Q = -\frac{28}{144}$$

Модулярность

Два сообщества:



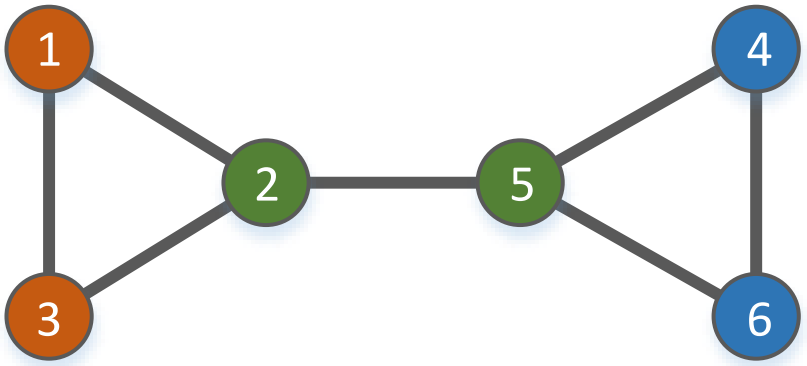
$$Q = \frac{1}{2m}\sum_{(i,j)\in E}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta(C_i, C_j)$$

$$\begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & \\ & & & 0 & 1 & 1 \\ & 1 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} - \frac{1}{2\cdot 6}\cdot \begin{pmatrix} 4 & 6 & 9 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \end{pmatrix} = \frac{1}{12}\cdot \begin{pmatrix} \mathbf{-4} & \mathbf{6} & \mathbf{8} & -4 & -6 & -4 \\ \mathbf{6} & \mathbf{-9} & \mathbf{6} & -6 & 3 & -6 \\ \mathbf{8} & \mathbf{6} & \mathbf{-4} & -4 & -6 & -4 \\ -4 & -6 & -4 & \mathbf{-4} & \mathbf{6} & \mathbf{8} \\ -6 & 3 & -6 & \mathbf{6} & \mathbf{-9} & \mathbf{6} \\ -4 & -6 & -4 & \mathbf{8} & \mathbf{6} & \mathbf{-4} \end{pmatrix}$$

$$Q = \frac{23}{144}$$

Модулярность

Три сообщества:



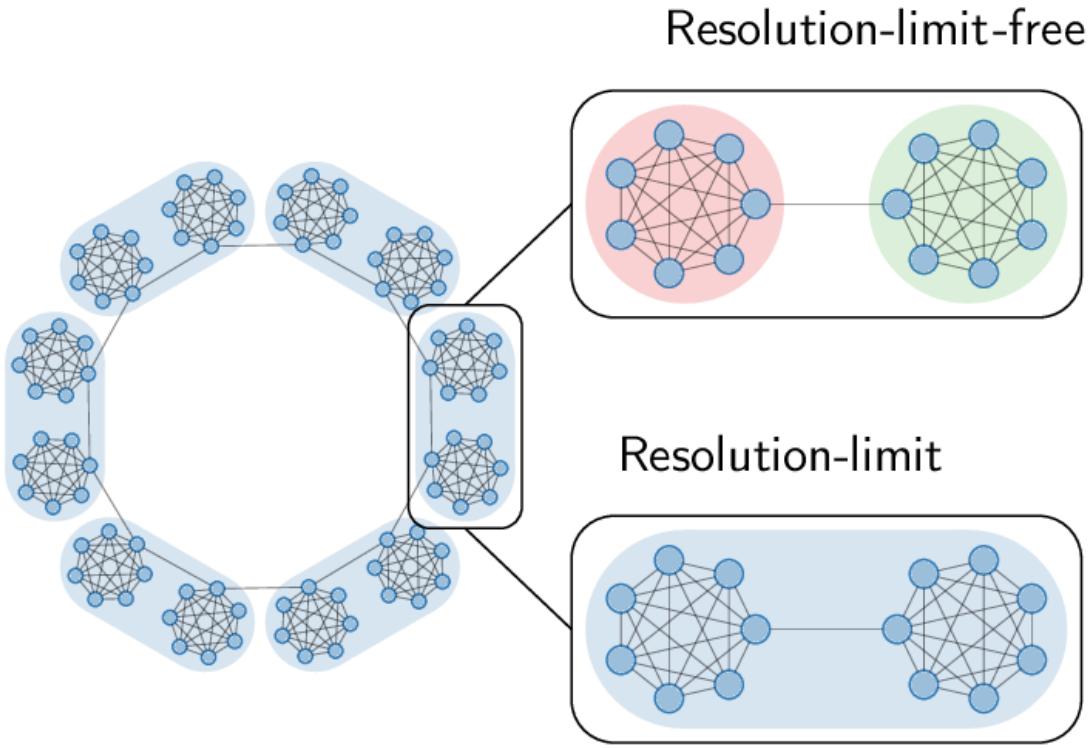
$$Q = \frac{1}{2m} \sum_{(i,j) \in E} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(C_i, C_j)$$

$$\begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & \\ & & & 0 & 1 & 1 \\ & 1 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} - \frac{1}{2 \cdot 6} \cdot \begin{pmatrix} 4 & 6 & 9 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 4 & 6 & 4 & 4 & 6 & 4 \\ 6 & 9 & 6 & 6 & 9 & 6 \\ 4 & 6 & 4 & 4 & 6 & 4 \end{pmatrix} = \frac{1}{12} \cdot \begin{pmatrix} \mathbf{-4} & 6 & \mathbf{8} & -4 & -6 & -4 \\ 6 & \mathbf{-9} & 6 & -6 & \mathbf{3} & -6 \\ \mathbf{8} & 6 & \mathbf{-4} & -4 & -6 & -4 \\ -4 & -6 & -4 & \mathbf{-4} & 6 & \mathbf{8} \\ -6 & \mathbf{3} & -6 & 6 & \mathbf{-9} & 6 \\ -4 & -6 & -4 & \mathbf{8} & 6 & \mathbf{-4} \end{pmatrix}$$

$$Q = \frac{4}{144}$$

Проблема Модурярности

Маленькая разрешающая способность (resolution limit)



Fortunato, Santo, and Marc Barthélemy.
"Resolution limit in community detection."
Proceedings of the National Academy of Sciences 104.1 (2007): 36-41.

Проблема Модурярности

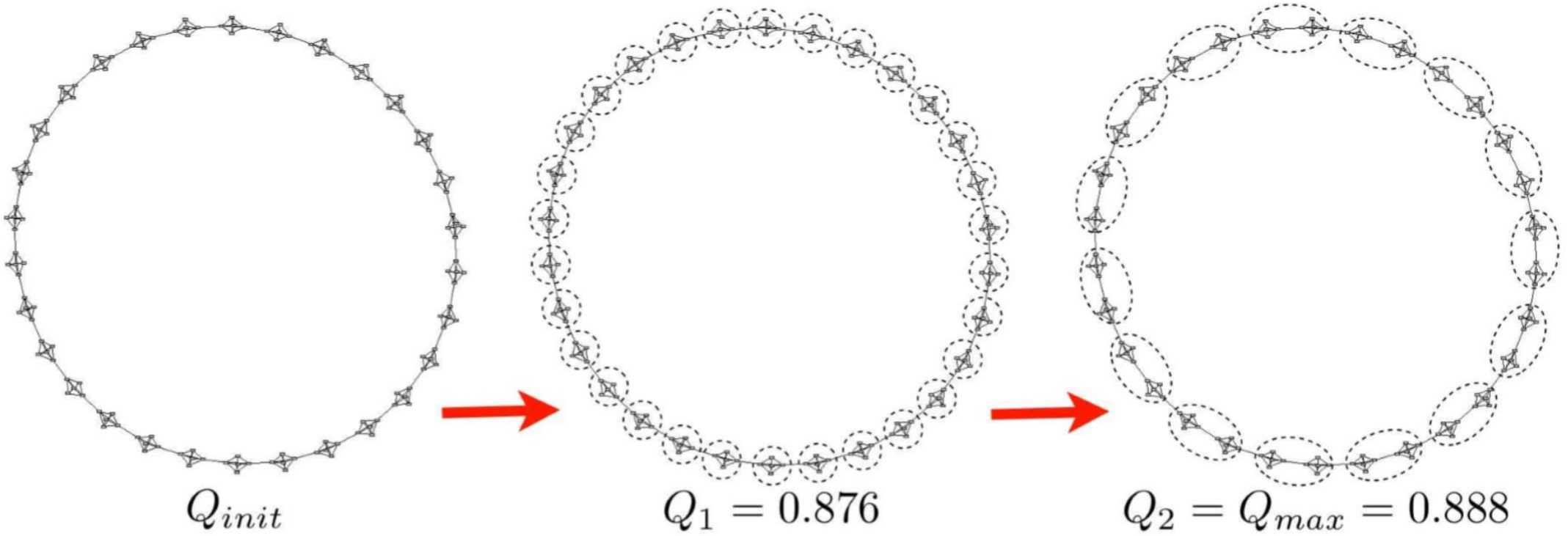
Почему так происходит?

Неверные предположения Модулярности:

1. **Любая вершина имеет равную вероятность соединиться с любой**
2. **Количество связей между сообществами уменьшается при увеличении размера графа.**

Проблема Модурярности

Пример



Методы выделения сообществ

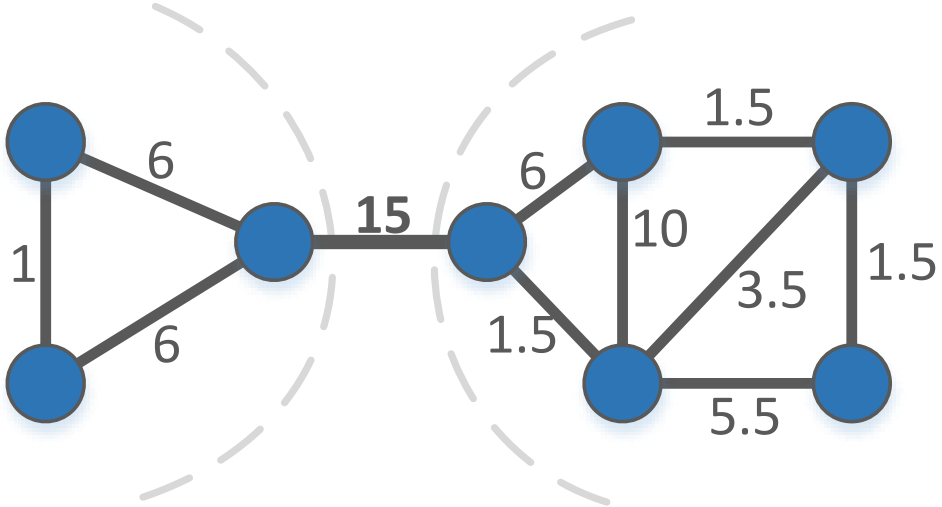
3. **Betweenness** — алгоритм на основе “центральности по посредничеству”
4. **Fastgreedy** — жадная оптимизация модулярности
5. **Multilevel** — многоуровневая оптимизация модулярности
6. **LabelPropogation** — какие соседи такой и я
7. **Walktrap** — короткие блуждания не приводят к выходу из сообщества
8. **Infomap** — минимизация кода для случайного пути
9. **Eigenvector** — собственные вектора матрицы модулярности

Методы выделения сообществ

Betweenness

Betweenness – число кратчайших путей, проходящих через данное ребро

1. **Подсчет коэффициентов “центральности по посредничеству”**
2. **Поочередное удаление ребер с наибольшим коэффициентом**
3. **Сообщества – компоненты связности**
4. **Выбор разбиения по максимуму модулярности**



Сложность $O(m^2 n)$ **=(**

Методы выделения сообществ

Fastgreedy

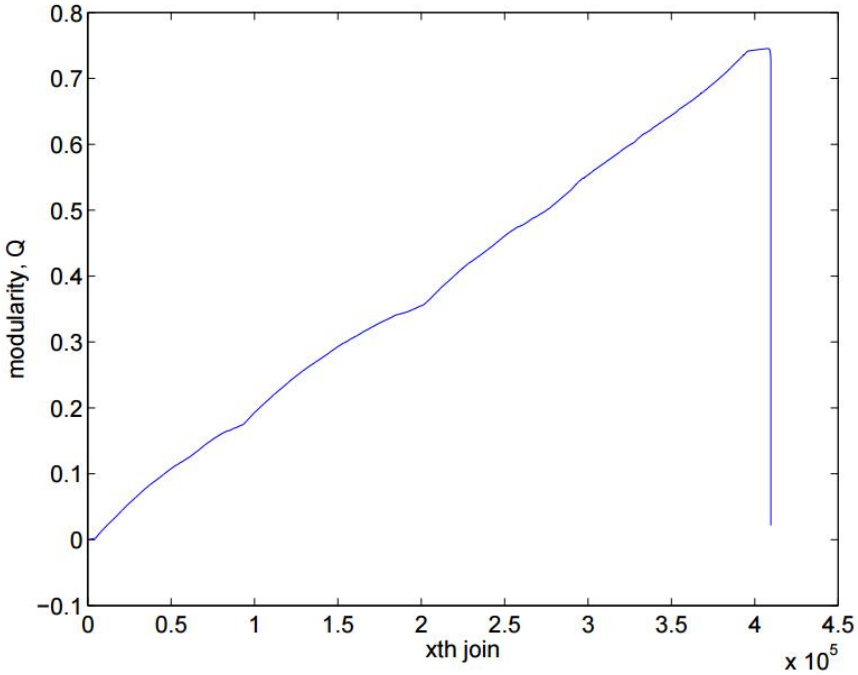
Жадная оптимизация модулярности.

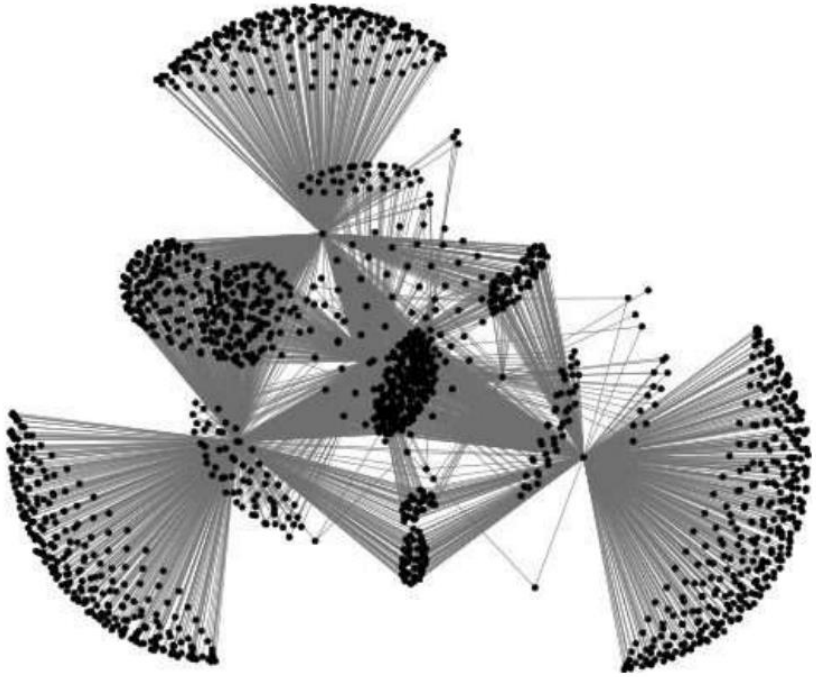
1. **Инициализация сообществ в каждой вершине.**
2. **Объединение сообществ, максимизирующее модулярность.**
3. **Выход, если нельзя увеличить модулярность.**

Сложность $O(mn)$**.**

Методы выделения сообществ

Fastgreedy





Тест: Рекомендательная система Amazon.com

409 687 вершин, 2 464 630 ребер.

Методы выделения сообществ

Fastgreedy

Rank	Size	Description
1	114538	General interest: politics; art/literature; general fiction; human nature; technical books; how things, people, computers, societies work, etc.
2	92276	The arts: videos, books, DVDs about the creative and performing arts
3	78661	Hobbies and interests I: self-help; self-education; popular science fiction, popular fantasy; leisure; etc.
4	54582	Hobbies and interests II: adventure books; video games/comics; some sports; some humor; some classic fiction; some western religious material; etc.
5	9872	classical music and related items
6	1904	children's videos, movies, music and books
7	1493	church/religious music; African-descent cultural books; homoerotic imagery
8	1101	pop horror; mystery/adventure fiction
9	1083	jazz; orchestral music; easy listening
10	947	engineering; practical fashion

10 самых больших сообществ в сети рекомендаций Amazon. 87% вершин

Методы выделения сообществ

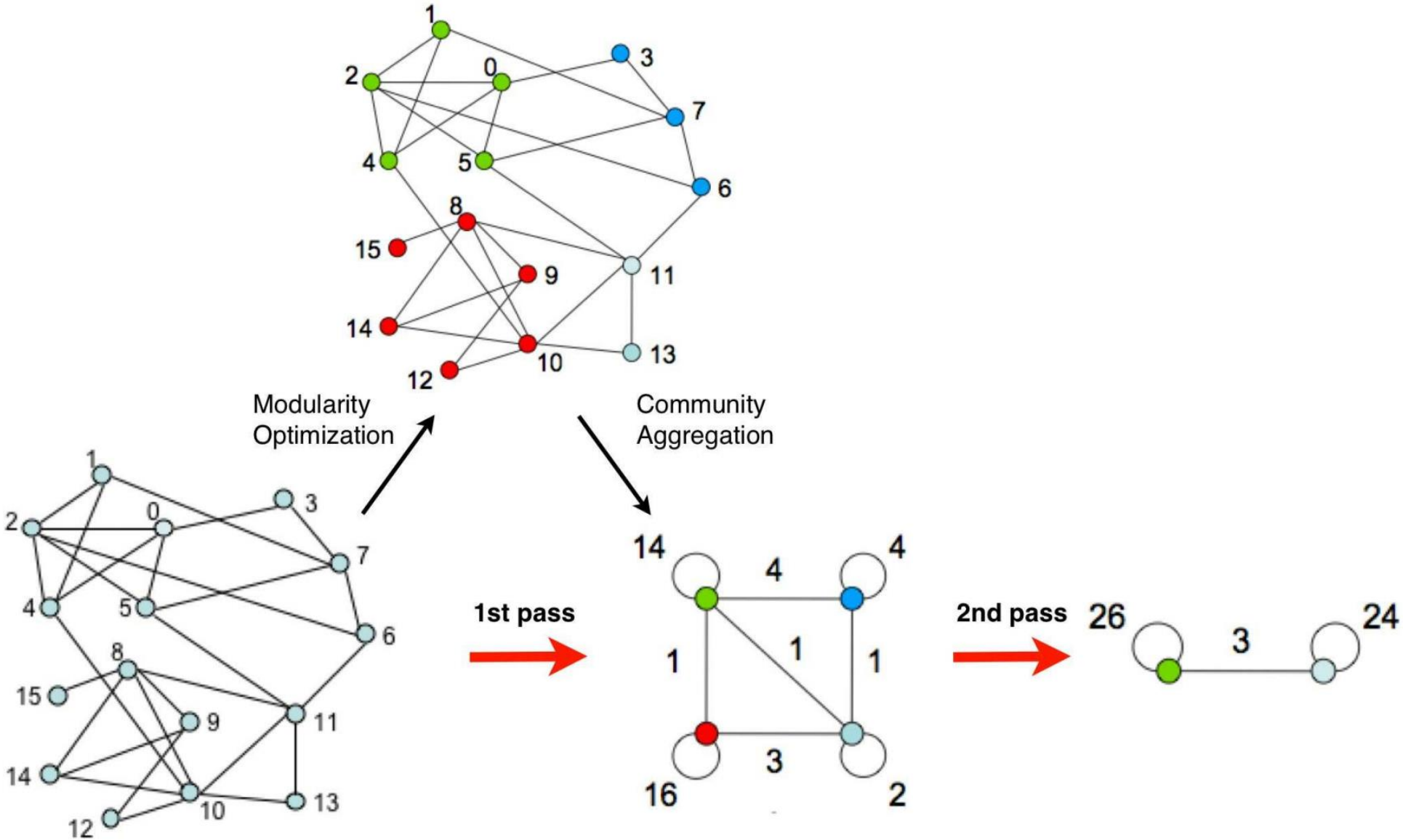
Multilevel

Многоуровневая оптимизация модулярности.

1. **Инициализация сообществ в каждой вершине**
2. **Первый Этап**
   a. **Жадно максимизируем модулярность путем перемещения вершины в сообщество вершины-соседа**
   b. **Нет перемещений – Выход.**
3. **Второй этап**
   a. **Создать метаграф на вершинах-сообществах**
   b. **Перезапустить алгоритм на метаграфе**

Методы выделения сообществ

Multilevel



Методы выделения сообществ

LabelPropogation

Вершина относится к тому сообществу, что и большинство ее соседей.

1. **Инициализация сообществ в каждой вершине**
2. **По всем вершинам в случайном порядке:**
   a. **Переопределить метку как метки большинства соседей**
3. **Если нет изменений – выход.**

Почти линейная сложность.

Одно большое сообщество. =(

Методы выделения сообществ

Walktrap

Короткие блуждания не приводят к выходу из сообщества.

Вероятность перехода из i в j

$$P_{ij} = \frac{A_{ij}}{d_i} \qquad P = D^{-1}A, \quad D = diag(d_1, \dots, d_n) \qquad P_{ij}^t = (P^t)_{ij}$$

Метрика на вершинах

$$r_{ij} = \left\| D^{-\frac{1}{2}} P_{i\square}^t - D^{-\frac{1}{2}} P_{j\square}^t \right\| = \sqrt{\sum_{k=1}^{n} \frac{\left(P_{ik}^t - P_{jk}^t\right)^2}{d_k}}$$

Обобщение на сообщества

$$P_{C,k}^t = \frac{1}{|C|} \sum_{i \in C} P_{ij}^t$$

$$r_{C_1 C_2} = \left\| D^{-\frac{1}{2}} P_{C_1\square}^t - D^{-\frac{1}{2}} P_{C_2\square}^t \right\| = \sqrt{\sum_{k=1}^{n} \frac{\left(P_{C_1,k}^t - P_{C_2,k}^t\right)^2}{d_k}}$$

Методы выделения сообществ

Walktrap

1. Инициализация сообществ в каждой вершине.
2. Вычисление расстояния между всеми смежными вершинами.
3. На каждом шаге:
   a. Выбрать два сообщества C_1 и C_2
   b. Объединить эти сообщества $C_3 = C_1 \cup C_2$
   c. Обновить расстояния между сообществами

Минимизируем среднее квадратов расстояний от вершины до ее сообщества

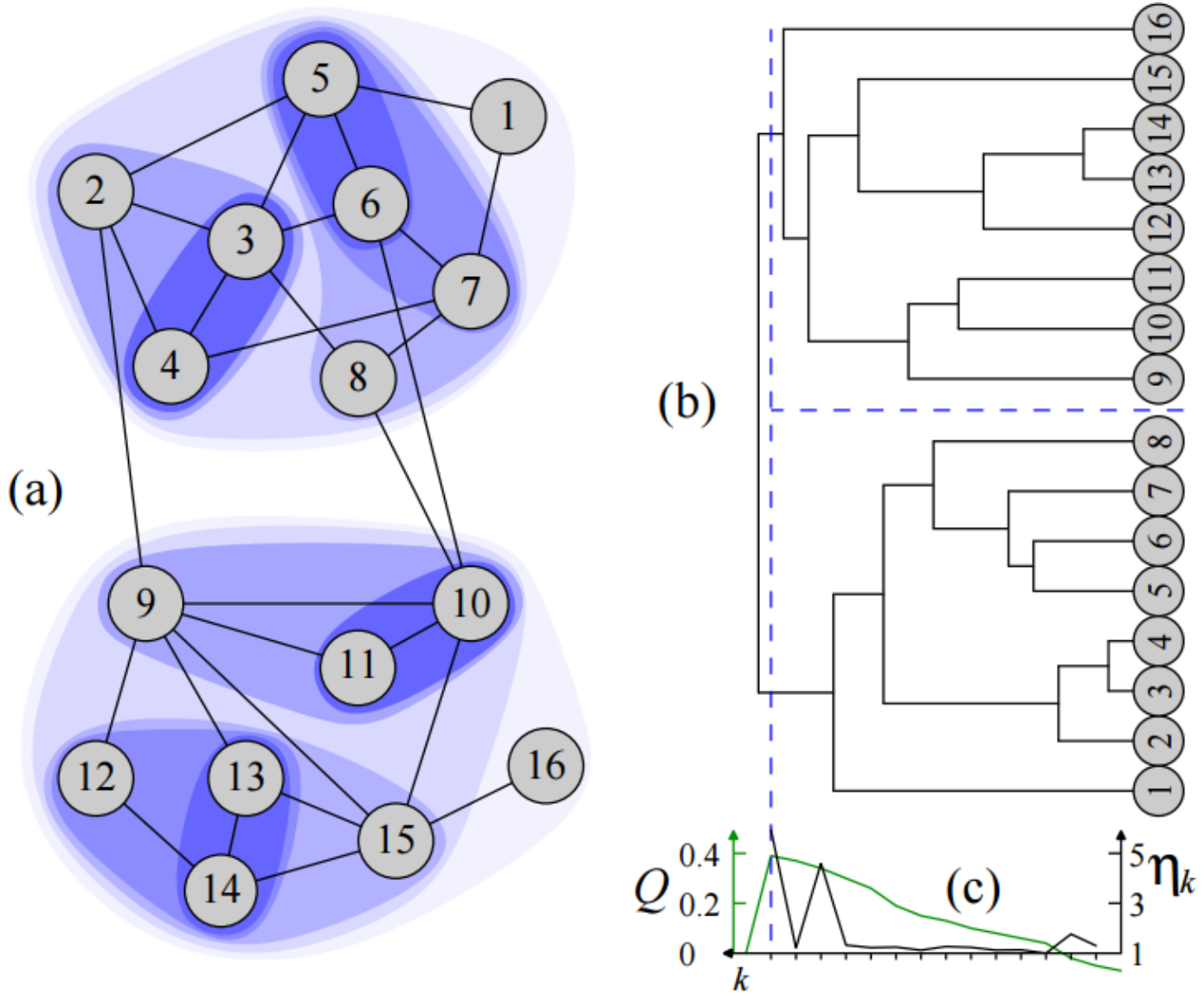
$$\sigma_k = \frac{1}{n} \sum_{C \in \mathbb{P}_k} \sum_{i \in C} r_{iC}^2 \to \min$$

$$\Delta\sigma(C_1, C_2) = \frac{1}{n} \left(\sum_{i \in C_3} r_{iC_3}^2 - \sum_{i \in C_1} r_{iC_1}^2 - \sum_{i \in C_2} r_{iC_2}^2 \right) \to \min$$

Хорошо работает на практике.

Методы выделения сообществ

Walktrap



Методы выделения сообществ

Infomap

Минимизация кода для случайного пути

2 уровня кодирования: уровень сообществ и вершин.

Коды Хафмана.

$$L(M) = q_{out}H(C) + \sum_{i=1}^{m} p_{in}^{i} H(C_i) \to \min$$

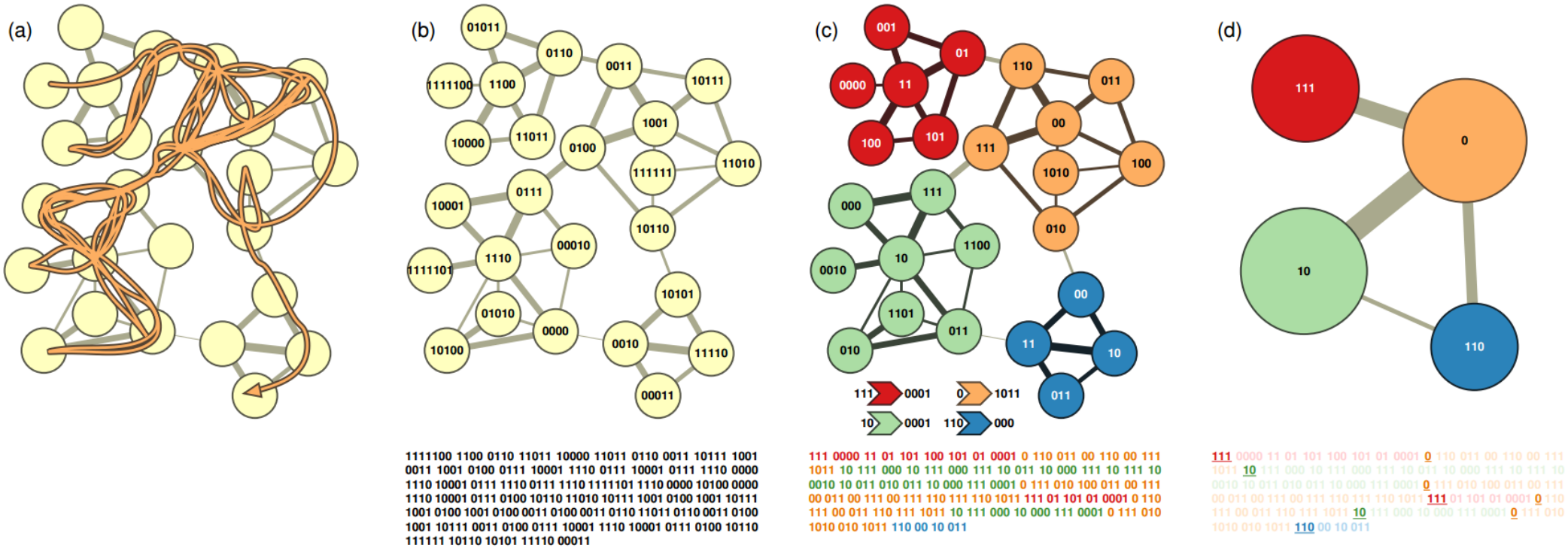
q_{out} **– вероятность покинуть сообщество**

$H(C)$ **– энтропия кода**

p_{in}^{i} **– доля перемещений внутри сообщества** C_i **(учитывая выход)**

Методы выделения сообществ

Infomap



Одно большое сообщество =(

На практике работает чуть хуже остальных

Методы выделения сообществ

Eigenvector

Собственные вектора матрицы модулярности

A_{ij} **– матрица смежности,**

d_i **– степень** i**-ой вершины**

$d = (d_1, \dots d_n)$ **– вектор из степеней вершин**

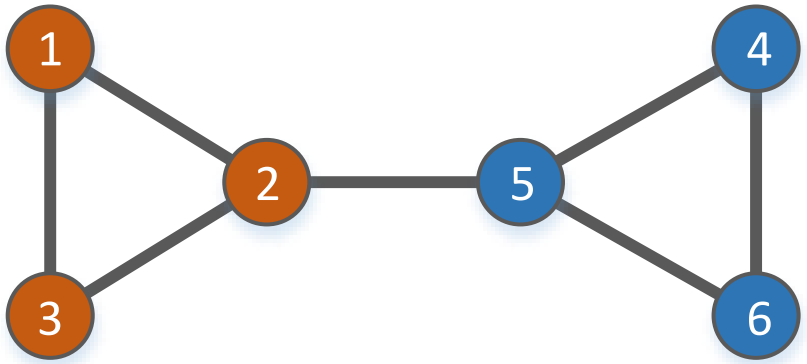
$m = |E|$ **– количество ребер**

$$M = A - \frac{dd^T}{2m}$$

Первый собственный вектор – отрицательная компонента одно сообщество, положительное – другое.

Методы выделения сообществ

Eigenvector



$$M = \frac{1}{12} \cdot \begin{pmatrix} -4 & 6 & 8 & -4 & -6 & -4 \\ 6 & -9 & 6 & -6 & 3 & -6 \\ 8 & 6 & -4 & -4 & -6 & -4 \\ -4 & -6 & -4 & -4 & 6 & 8 \\ -6 & 3 & -6 & 6 & -9 & 6 \\ -4 & -6 & -4 & 8 & 6 & -4 \end{pmatrix}, \quad v_1 = \begin{pmatrix} 0.444 \\ 0.325 \\ 0.444 \\ -0.444 \\ -0.325 \\ -0.444 \end{pmatrix}$$

Методы агрегирования результатов

1. Объединение кластеризацией

2. Объединение базовыми методами

Методы агрегирования результатов

Объединение кластеризацией

1. Номер сообщества из каждого разбиения – признак
2. Запустить любой метод кластеризации с категориальными признаками

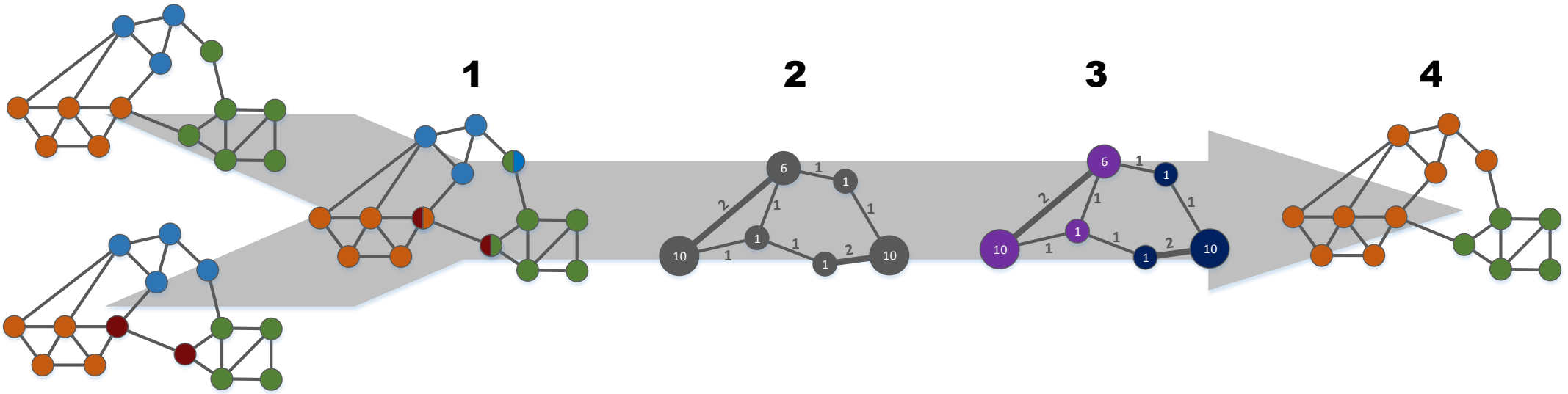
1. Чувствителен к методу
2. Чувствителен к метрике
3. Не гарантирует локальность оптимума
4. Не гарантирует улучшение результата

5. Но работает

Методы агрегирования результатов

Объединение базовыми методами

1. Построить новое разбиение путем измельчения исходных
2. Построить взвешенный граф на сообществах
3. Запустить любой базовый метод на новом графе
4. Вернуться к исходному графу



Тесты

1. Модельные данные
   a. l-partition benchmark. Тест Гирвана-Ньюмана
      4 сообщества по 32 вершины, $p_{in} = 0.5$

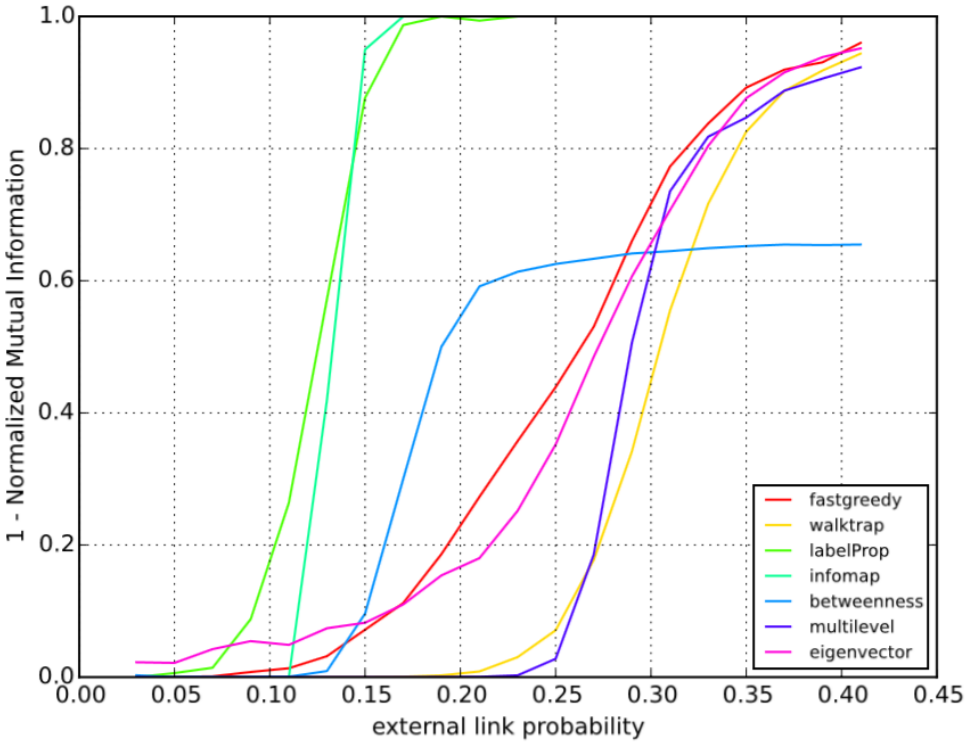
   b. Тест Lancichinetti
      1000 вершин, средняя степень вершины – 40,
      распределения на степенях вершин и размерах сообществ $\sim n^{-2}$,
      размеры сообществ от 30 до 100.

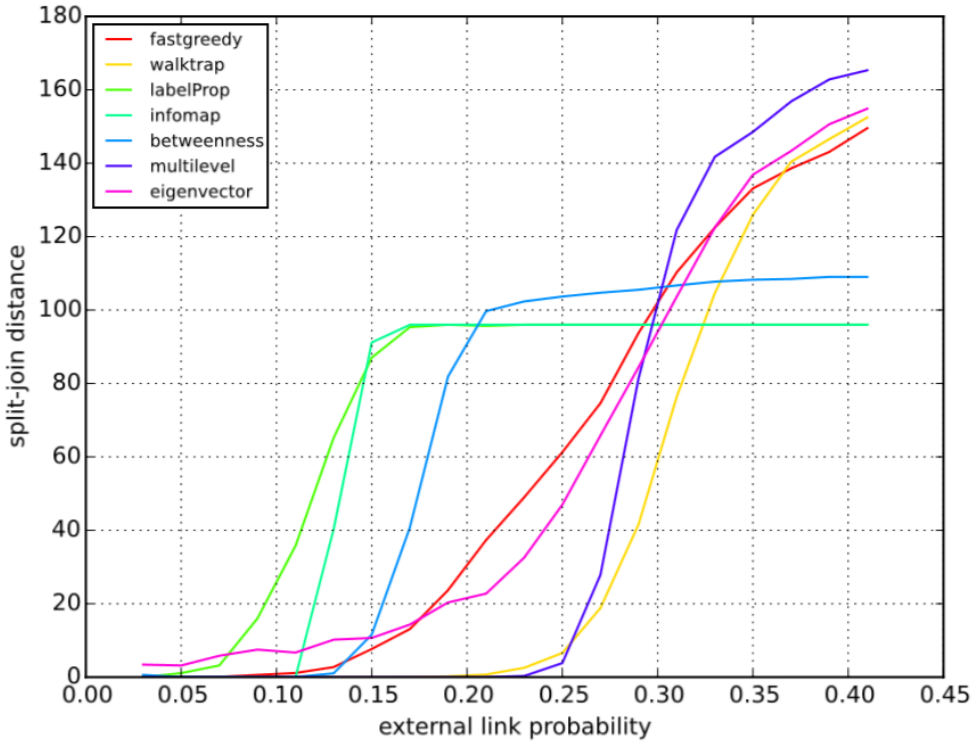
2. Тест на реальных данных
   a. Эго-графы из конкурса Learning Social Circles in Networks
   b. Тест на реальных данных, эго-графы моих друзей “В Контакте”

Тест Гирвана-Ньюмана

Все базовые методы

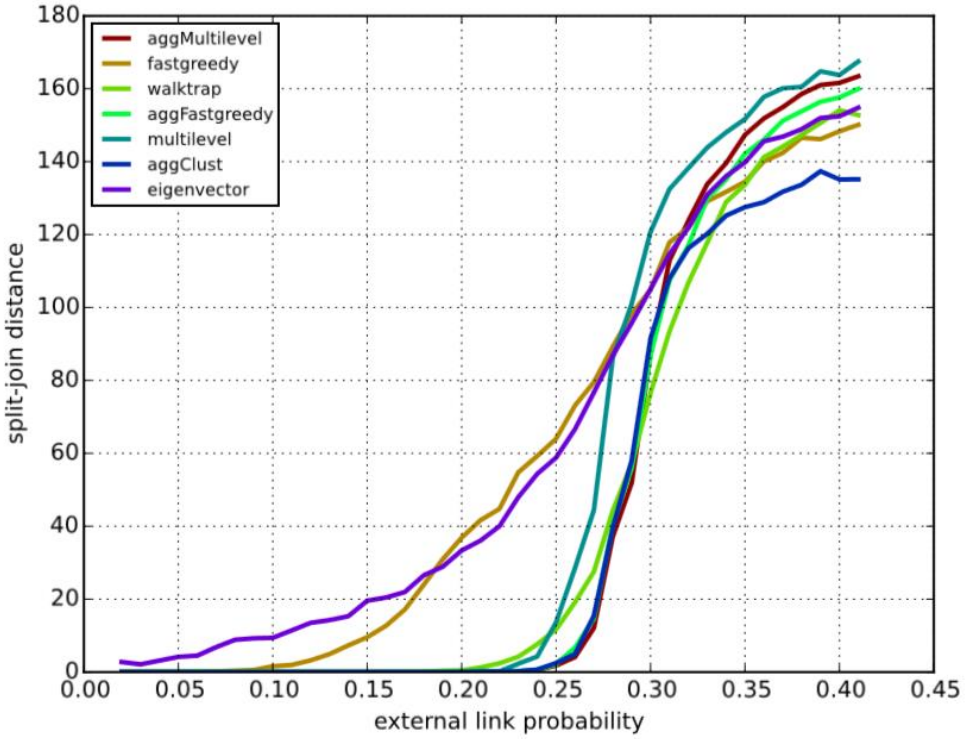
Сравниваем с известной разметкой по $1 - nmi$ и sjd

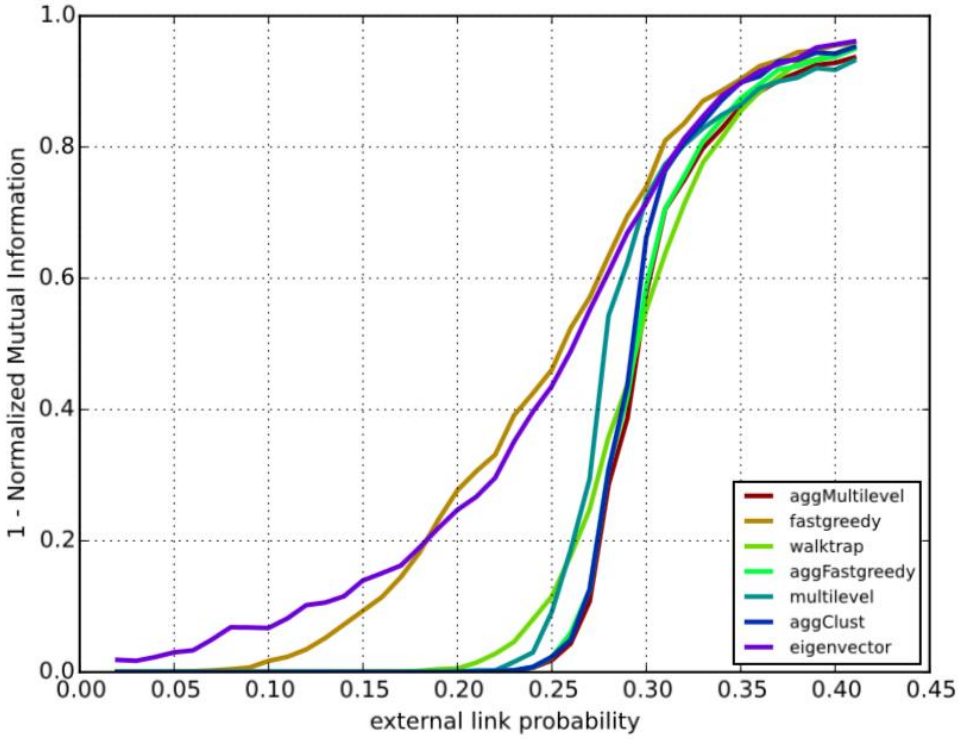




Тест Гирвана-Ньюмана

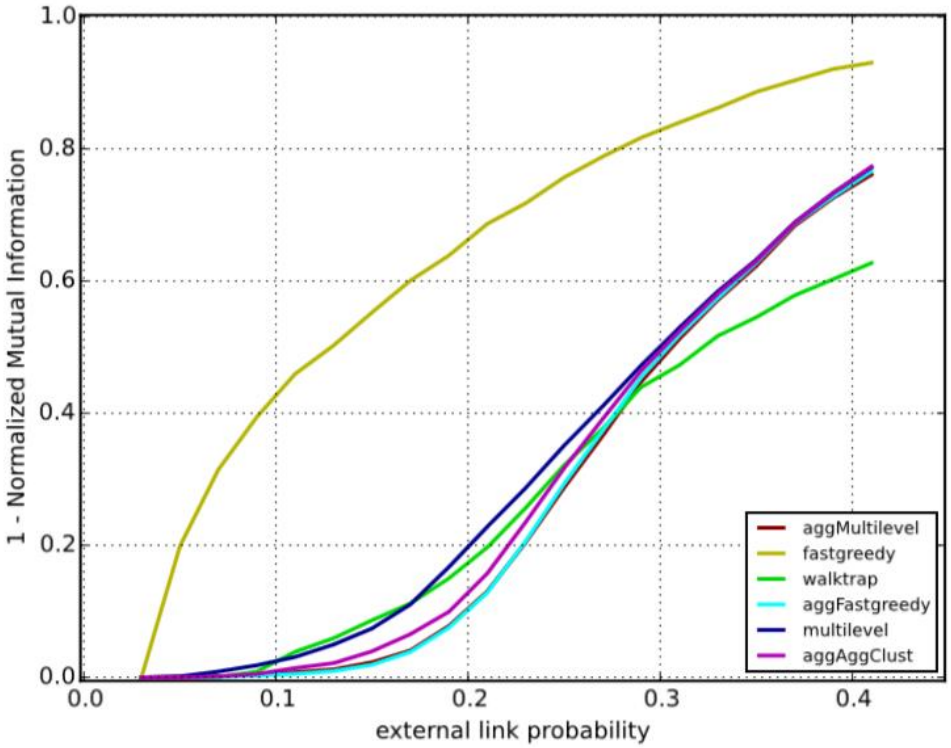
Базовые методы VS объединение

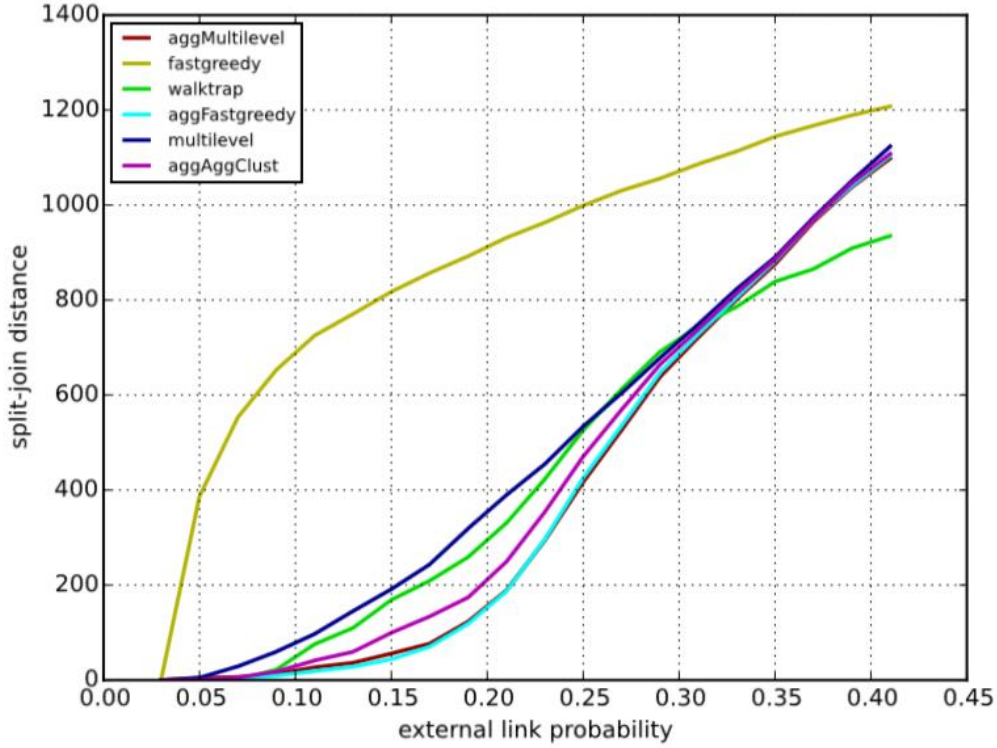




Тест Lancichinetti

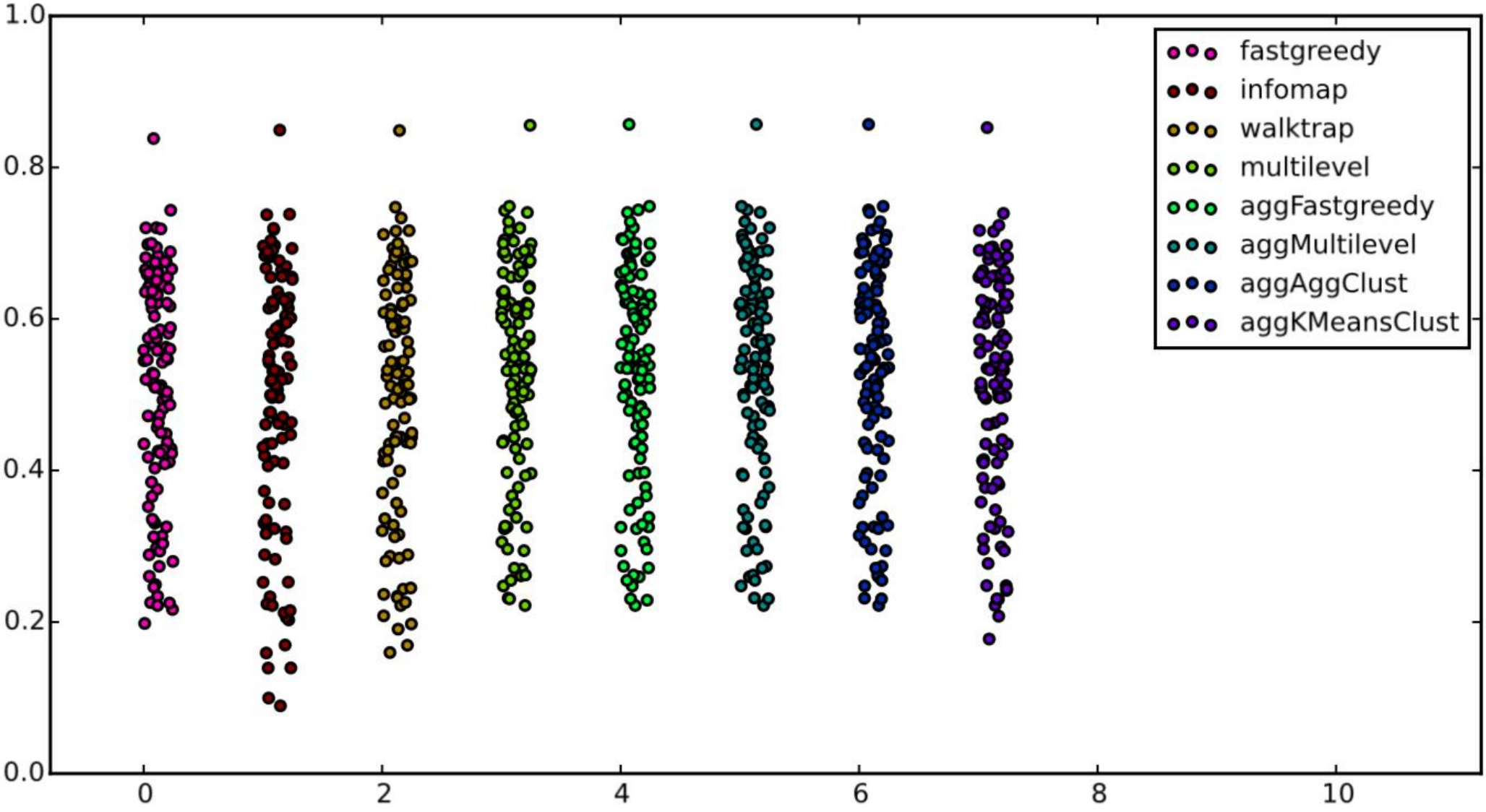
Базовые методы VS объединение





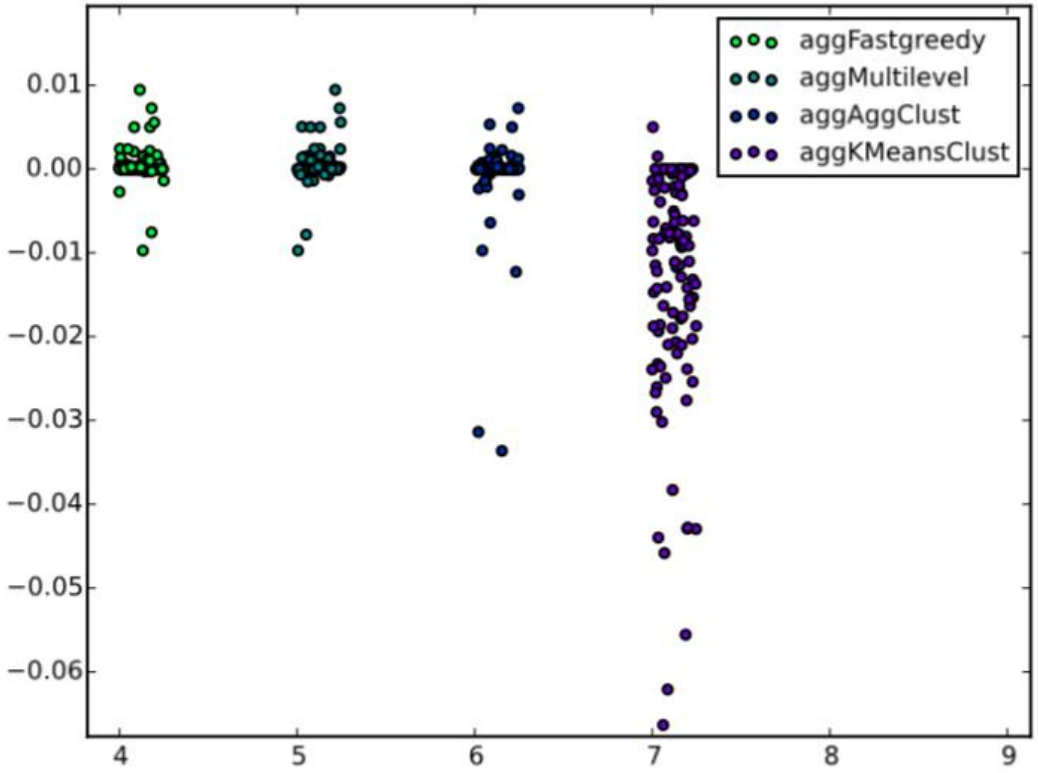
Тест на реальных данных

Learning Social Circles in Networks



Тест на реальных данных

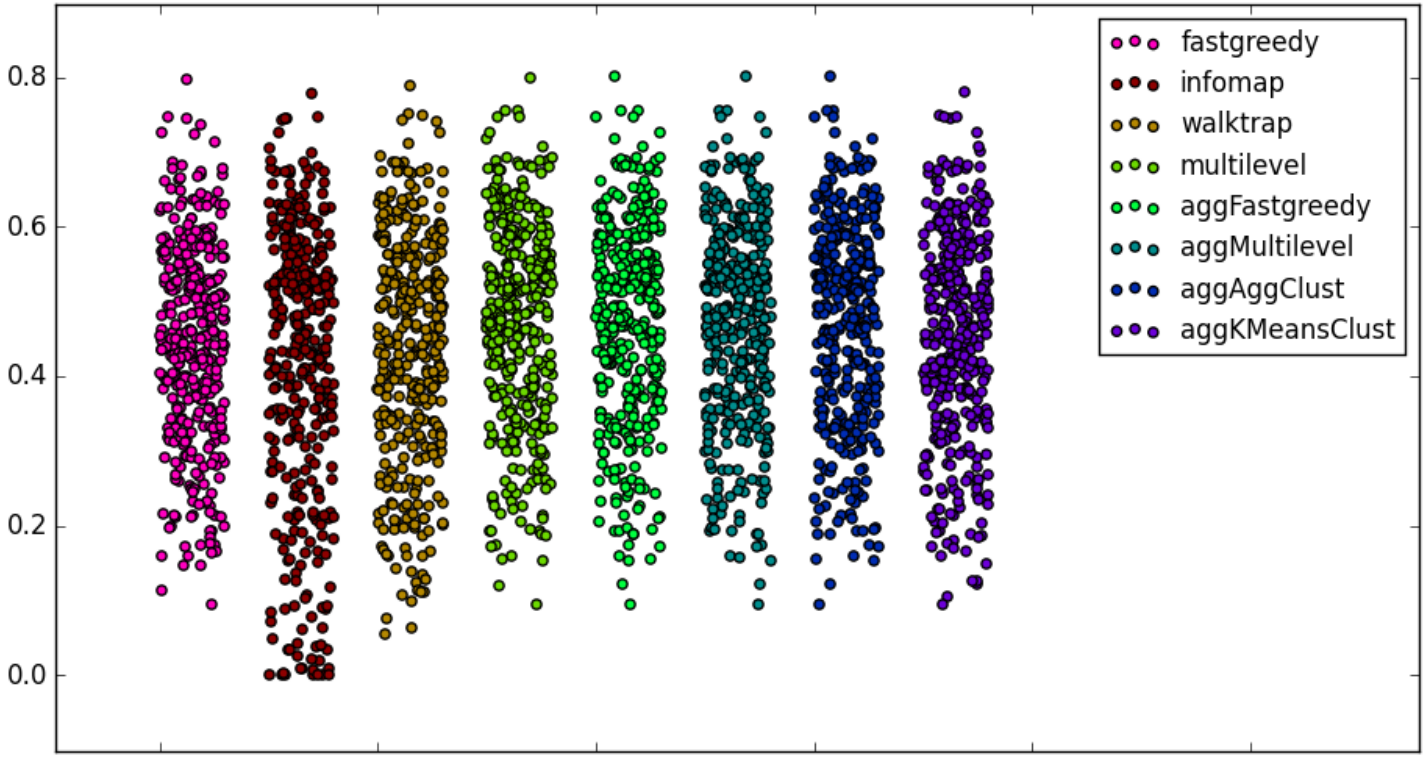
Learning Social Circles in Networks



Метод	средний прирост	среднее для максимума из 0 и прироста
aggFastgreedy	0.000336	0.00053
aggMultilevel	**0.000364**	**0.00057**
aggAggClust	-0.000629	0.00037
aggKMeansClust	-0.014295	0.00006

Тест на реальных данных

Эго-графы друзей “В Контакте”



Метод	средний прирост	среднее для максимума из 0 и прироста
aggFastgreedy	0.000699	0.000809
aggMultilevel	0.000719	0.000801
aggAggClust	-0.000733	0.000734
aggKMeansClust	-0.019532	0.00008

igraph & python

Реализация всех описанных базовых методов.

Все что угодно для графов.

http://igraph.org/python/

```
import igraph as ig

methodsList = {
    'infomap': lambda G: G.community_infomap(),
    'fastgreedy': lambda G: G.community_fastgreedy().as_clustering(),
    'eigenvector': lambda G: G.community_leading_eigenvector(),
    'labelProp': lambda G: G.community_label_propagation(),
    'multilevel': lambda G: G.community_multilevel(),
    'optModularity': lambda G: G.community_optimal_modularity(),
    'betweenness': lambda G: G.community_edge_betweenness().as_clustering(),
    'walktrap': lambda G: G.community_walktrap().as_clustering(),
    'spinglass': lambda G: G.community_spinglass(),
}
```

igraph & python

Реализация всех описанных базовых методов

http://igraph.org/python/doc/tutorial/tutorial.html

```
G = ig.Graph.Read_GraphML("myGraph.GraphML")
print G.edge_betweenness()
VC = methodsList['walktrap'](G)
print VC.membership
print VC.modularity


print ig.compare_communities(VC1, VC2, 'nmi')
```