

курс «Прикладные задачи анализа данных»

# Выделение сообществ (Community Detection)

Александр Дьяконов

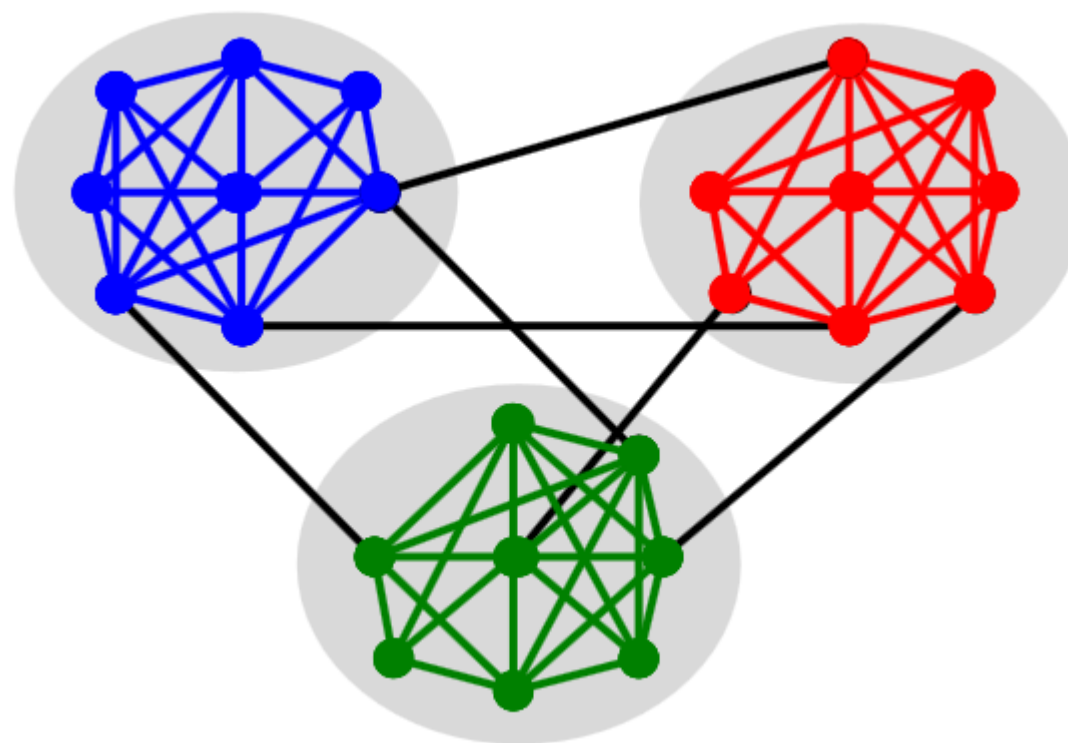
11 декабря 2020 года

## План

**Сообщество в графе**  
**Методы выделения сообществ**  
**Тестирование различных методов**  
**case: выделение кругов в эго-подграфах**

## Сообщество в графе

**нет чёткого определения**  
**рёбер внутри сообщества много,**  
**рёбер соединяющих сообщество с остальными вершинами мало**  
малый радиус сообщества



## Какие бывают определения сообщества

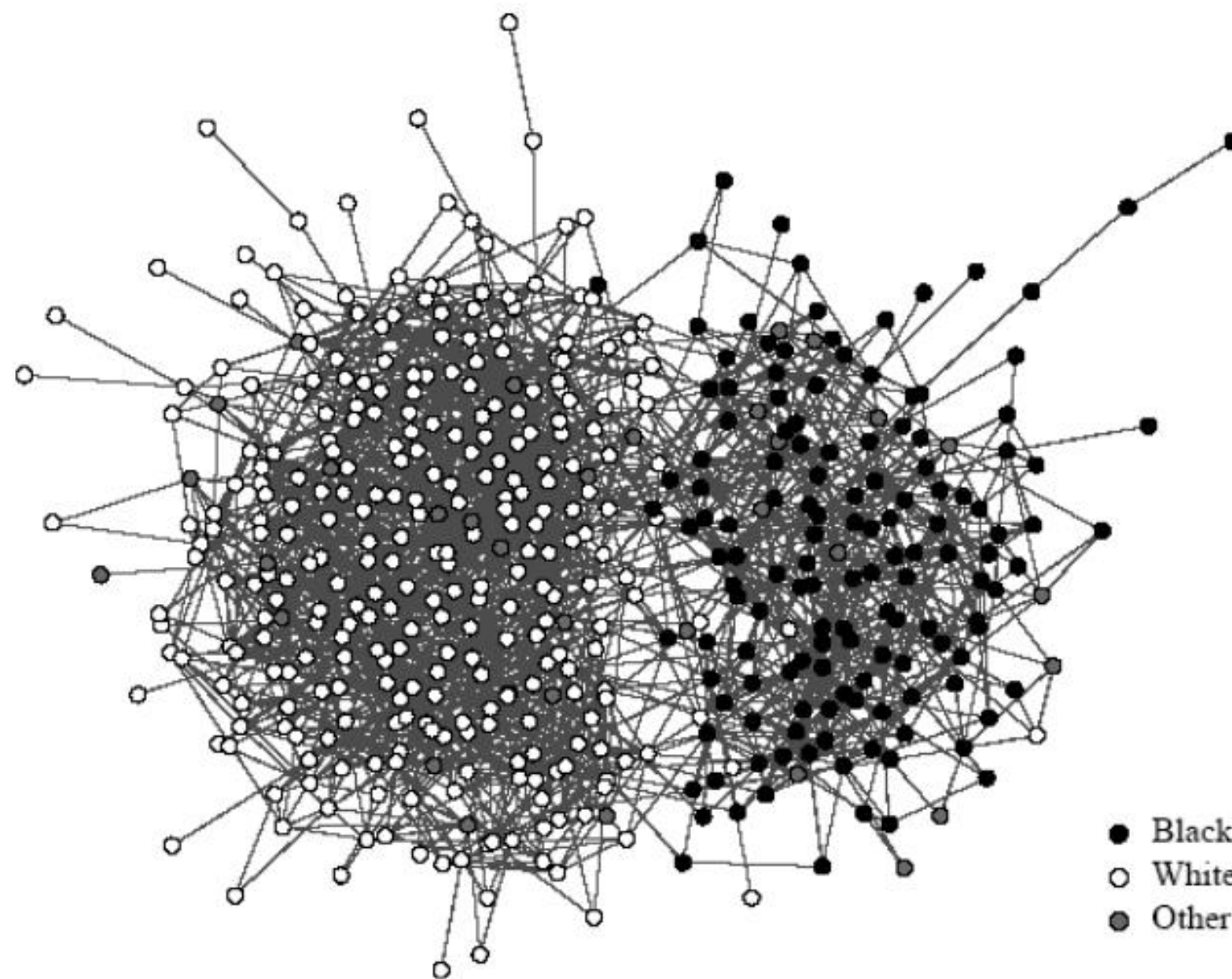
### 1. Чёткие

### 2. Нечёткие (не определения, см. выше)

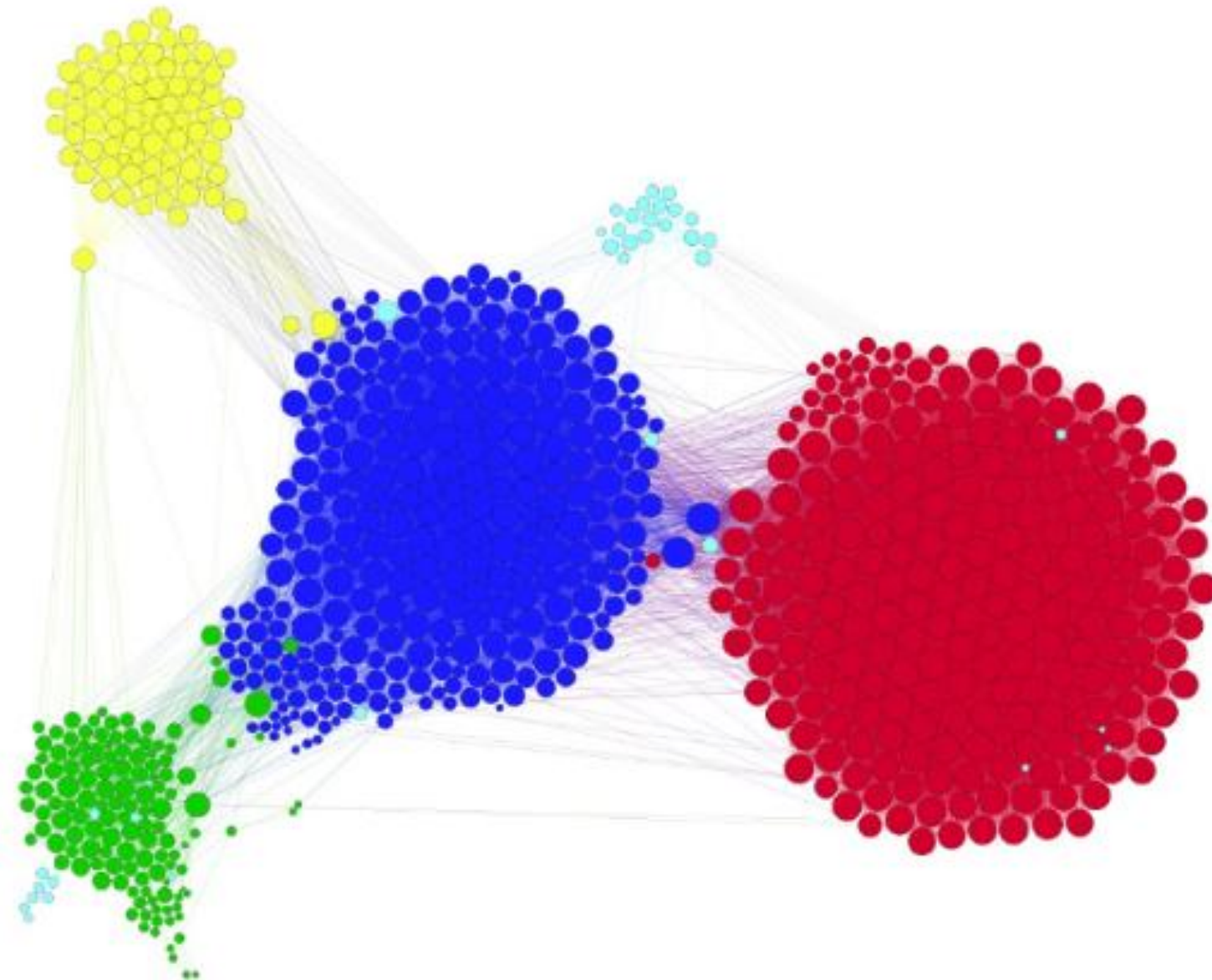
### 3. Алгоритмические

(то что получается в результате действия алгоритма)

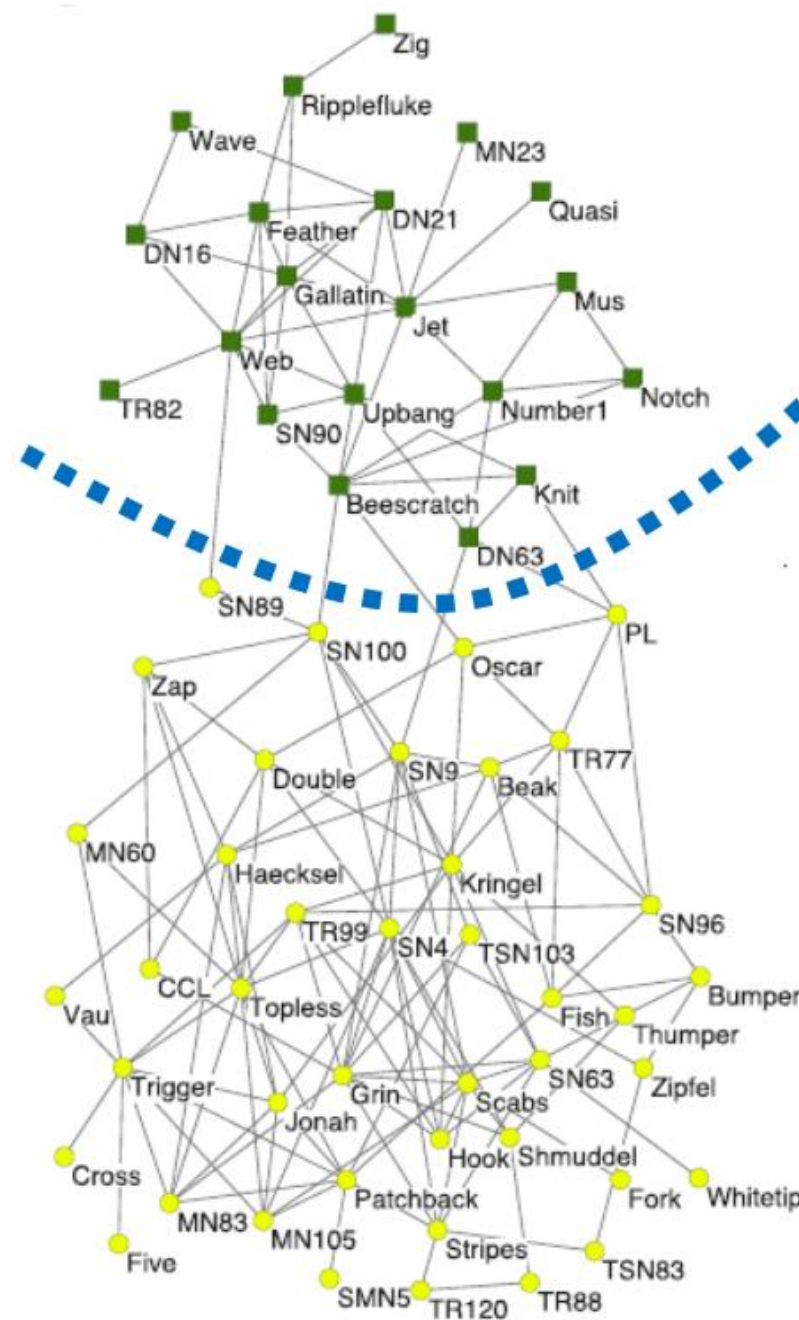
## Примеры сообществ: сеть социальных отношений в high-school



## Примеры сообществ: эго-сеть фейсбука



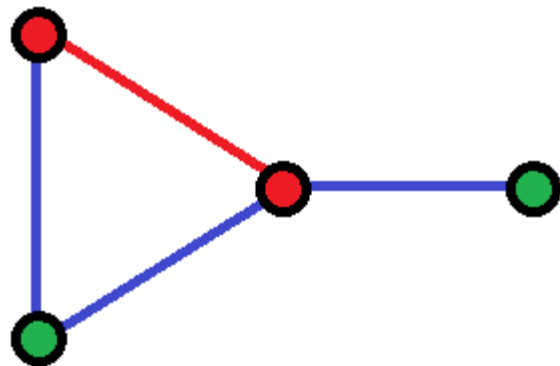
## Примеры сообществ



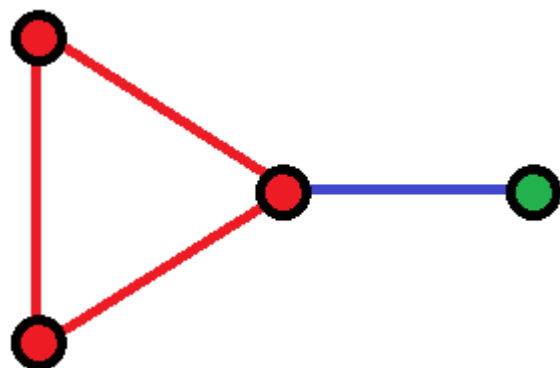


## Сообщество в графе

Идеальный кандидат – клика

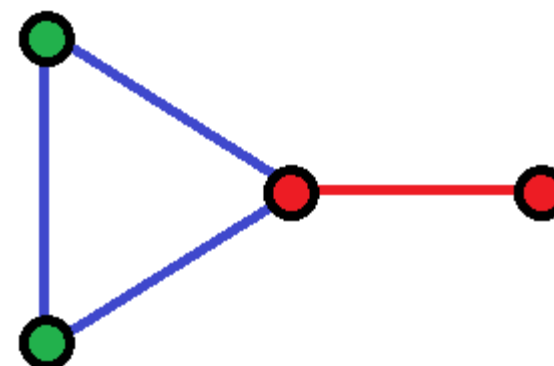


Клика



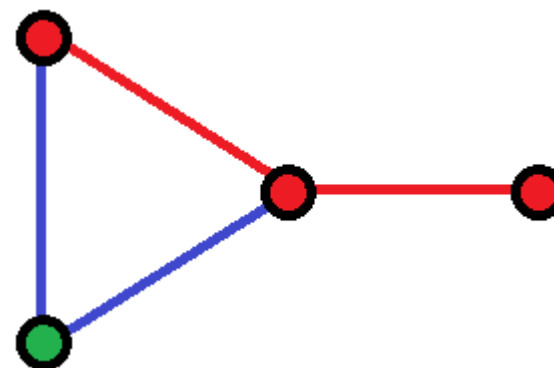
Наибольшая клика

Клика наибольшего размера



Максимальная клика

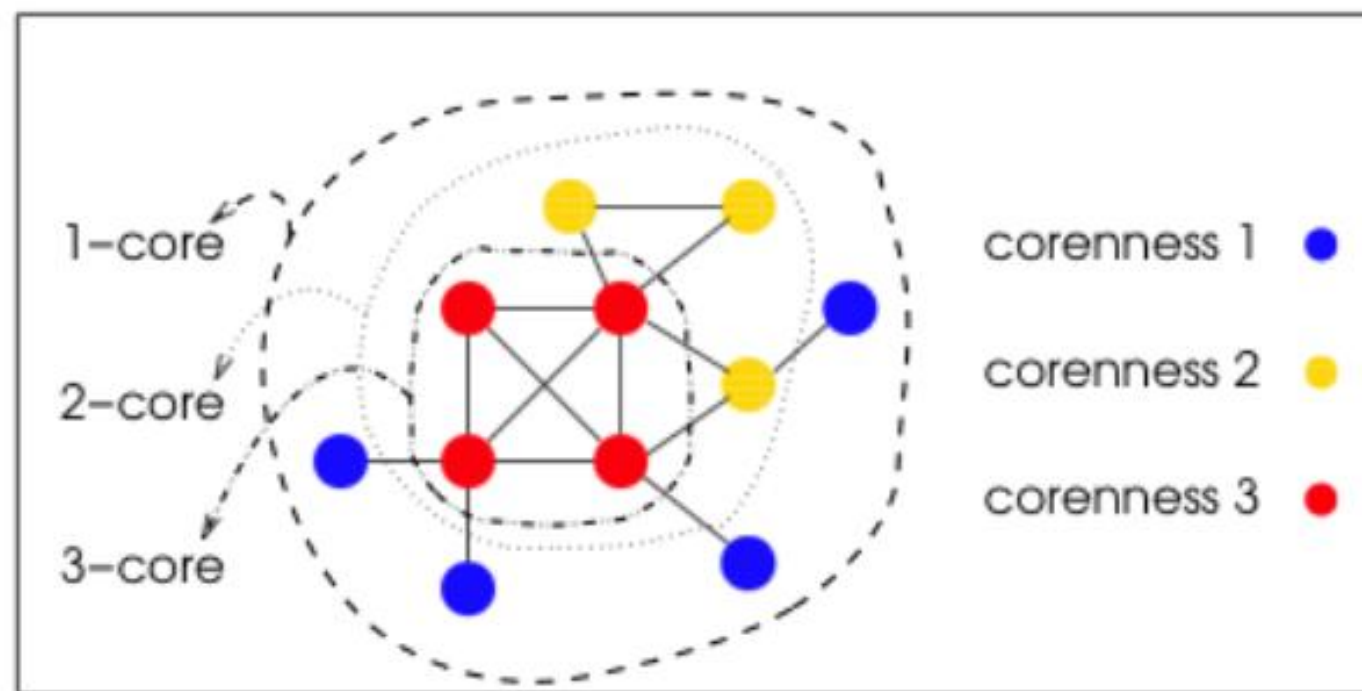
Не может быть расширена



Не клика

**Но вычислительные сложности...**

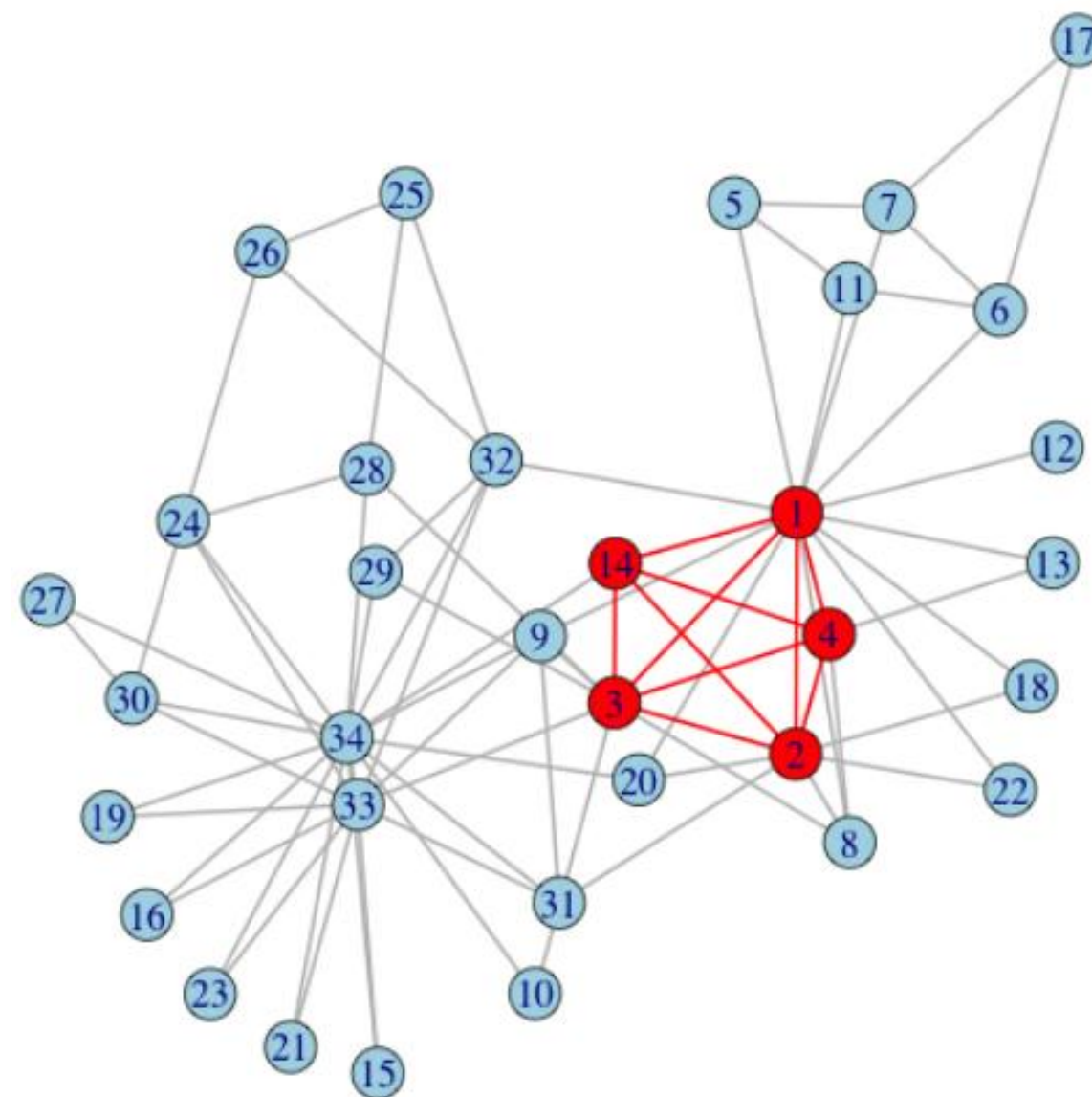
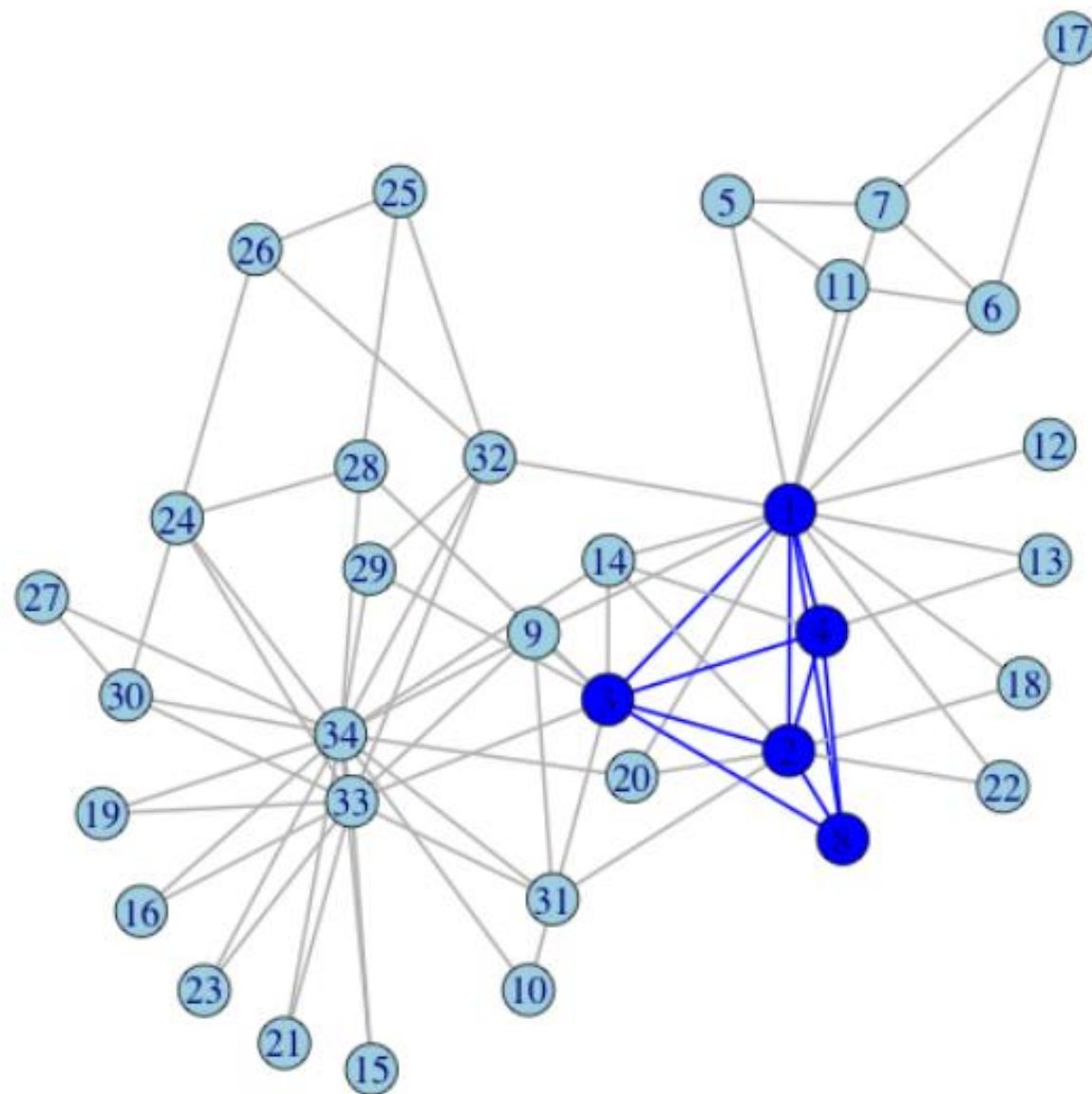


**к-ядра**

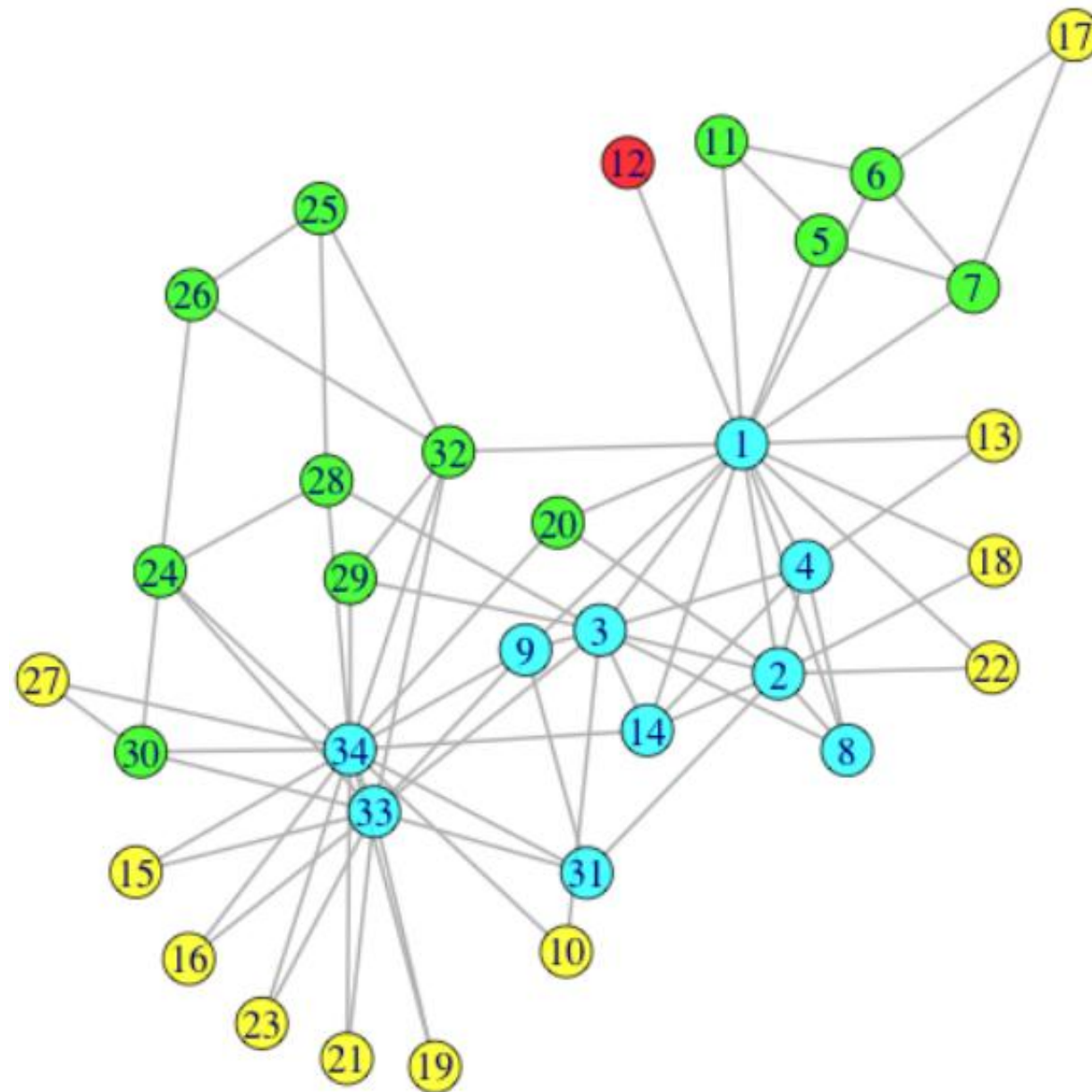
Alvarez-Hamelin et.al., 2005

**к-ядро = степень каждой вершины  $\geq k$**

## Наибольшие клики (Карате клуб)

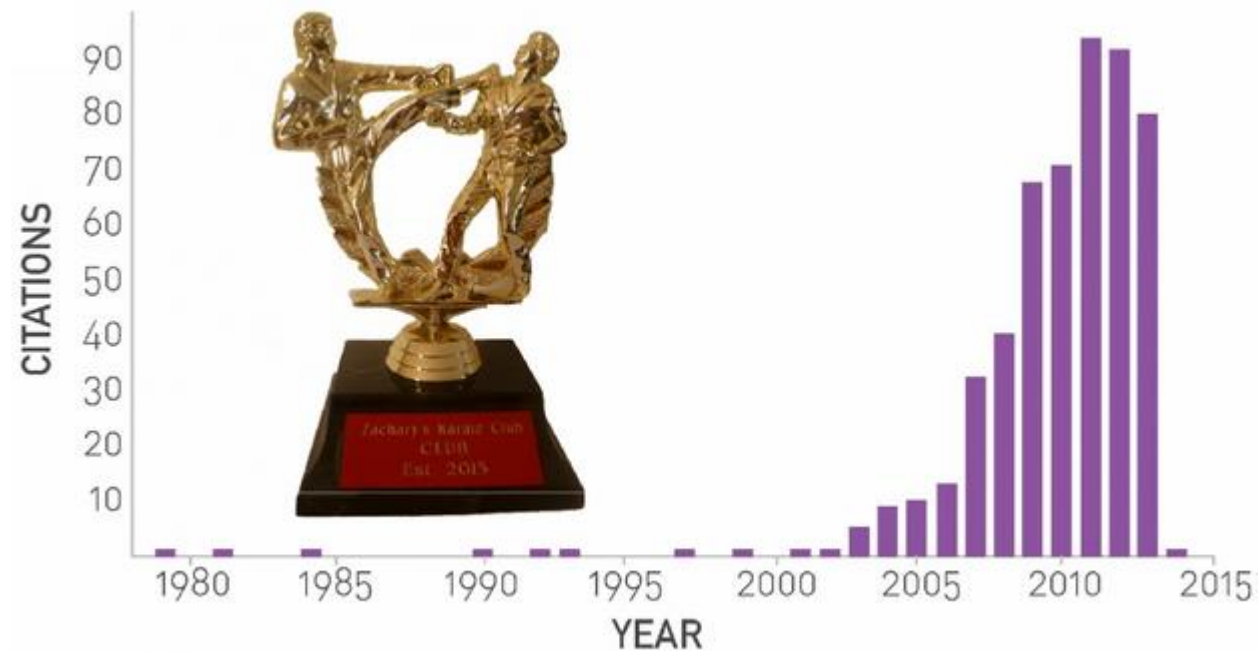


## Ядра (Карате клуб)



## Датасет Карате-клуб

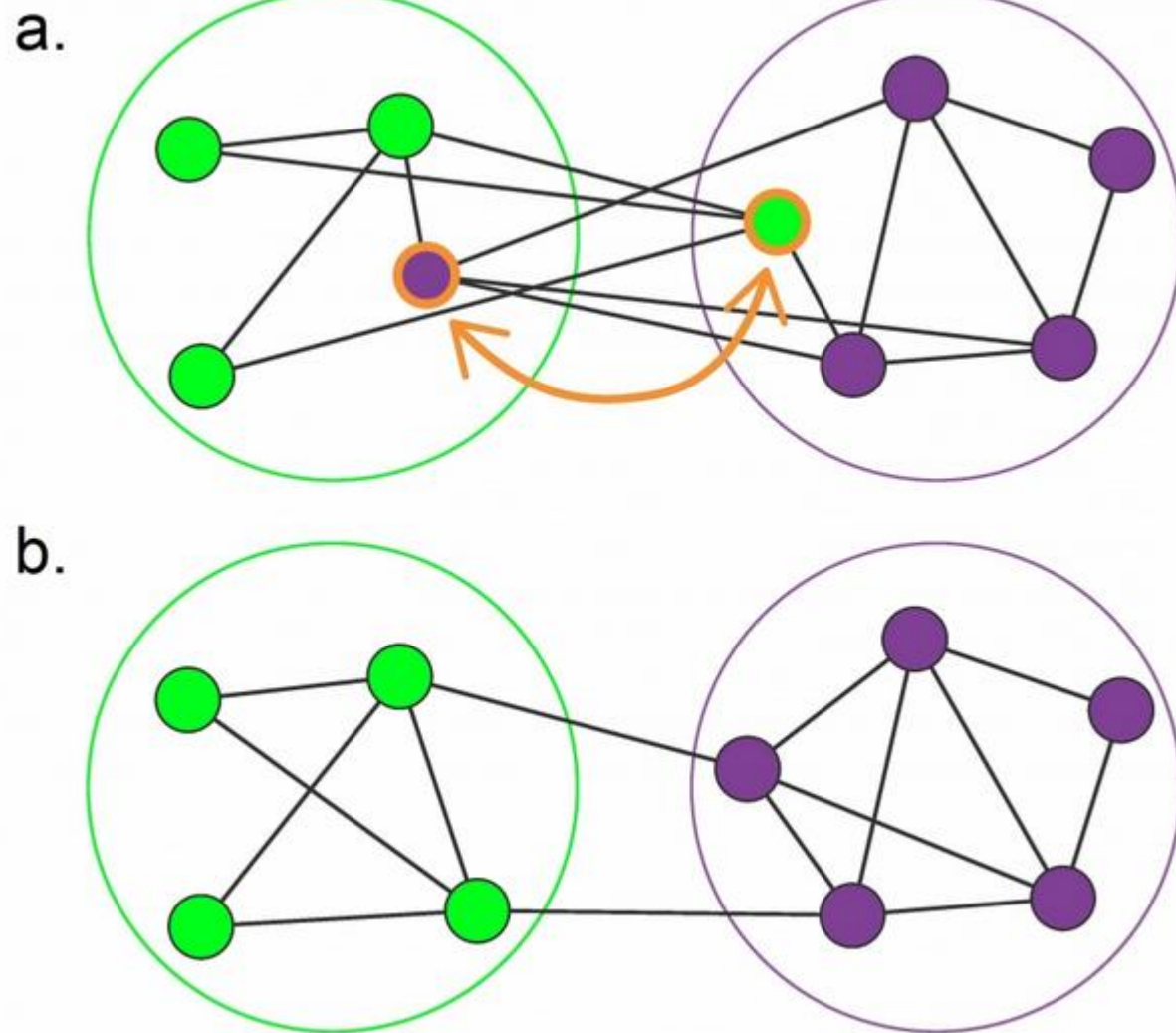
**Конфликт привёл к разделению членов клуба на 2 сообщества  
связи – общение вне клуба**



**Приз за первое упоминание на конференциях по SNA**

## Выделение сообществ: 0й способ

### Разбиение графа: Kernighan-Lin Algorithm



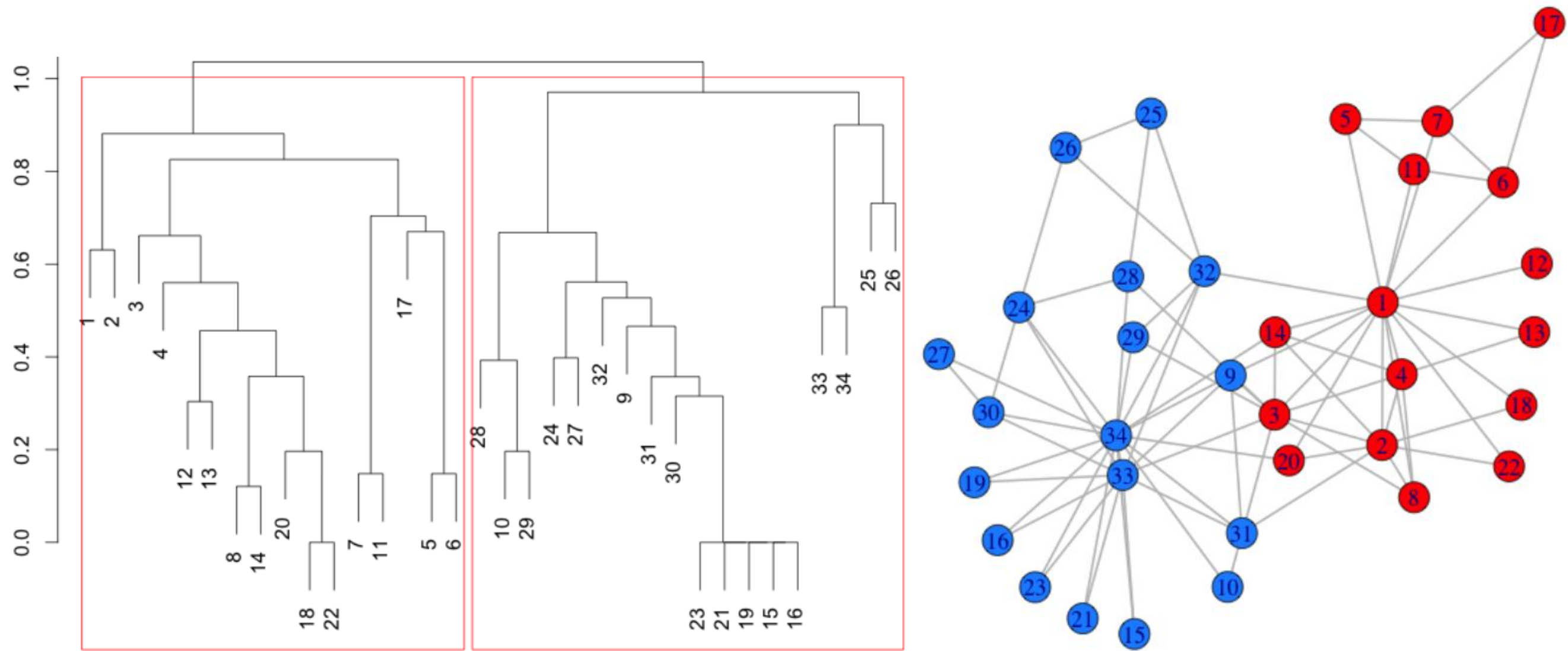
1. Разбиваем вершины на два множества
2. Выбираем ребро между множествами  
Пытаемся «перекинуть вершины», чтобы уменьшить число рёбер между сообществами

<http://networksciencebook.com/chapter/9>



Выделение сообществ: 1й способ

Обычная кластеризация с мерой схожести вершин



## Выделение сообществ: 1й способ

недостаток

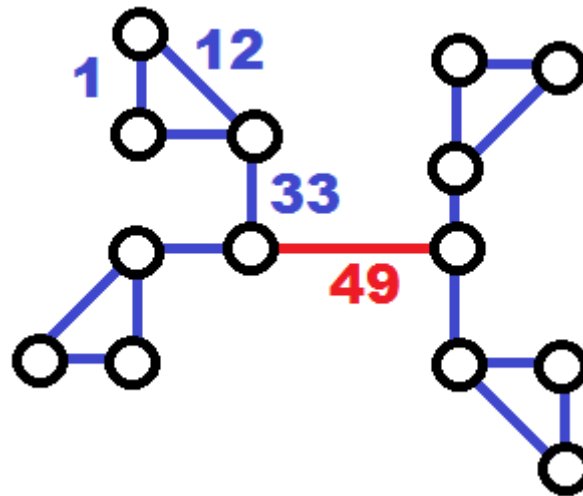
**формально не пытаемся выполнить условия «сообщности»:  
много рёбер внутри сообщества  
слабые связи между сообществами**



## Выделение сообществ: 2-й способ – Edge betweenness (Girvan-Newmann's method)

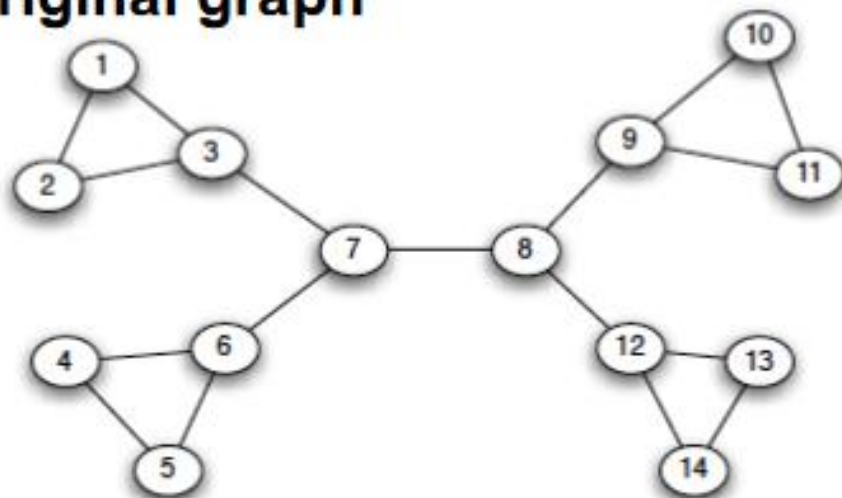
**Edge betweenness** – число кратчайших путей, проходящих через ребро

Повторять пока есть рёбра  
удаление ребра с максимальным значением EB  
Получаем иерархическое разложение графа

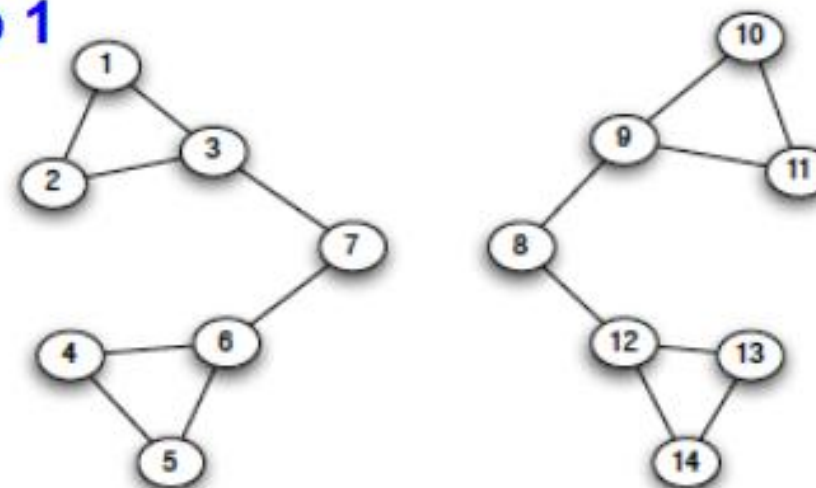


## Edge betweenness (Girvan-Newmann's method)

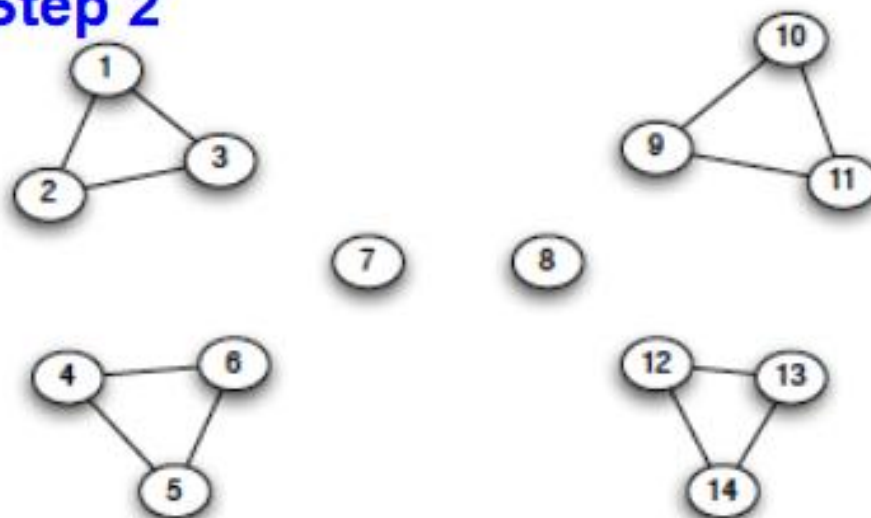
Original graph



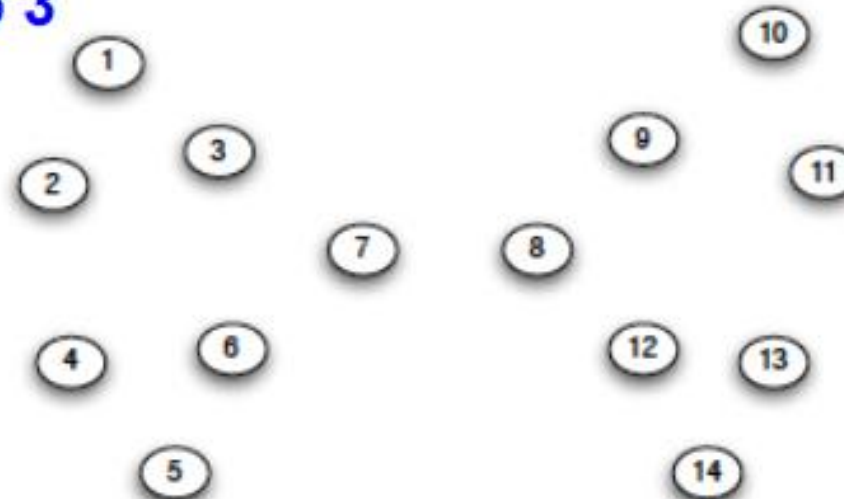
Step 1



Step 2



Step 3

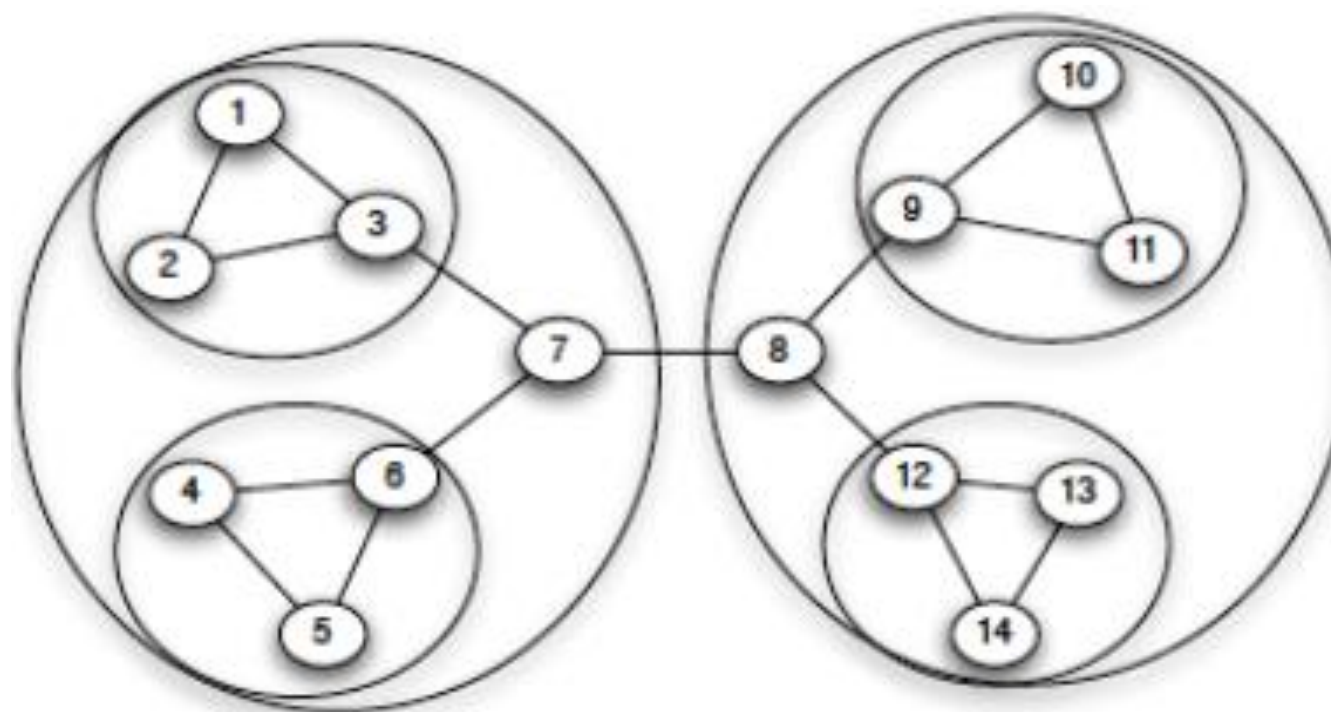


**Edge betweenness: когда останавливаться (в иерархическом делении)**

**Как в кластеризации:  
ввести функционал качества**

**Число рёбер в группе – ожидаемое число рёбер**

**Почему не оптимизировать этот функционал напрямую?**



## Выделение сообществ: 3-й способ – модулярность

тоже Girvan и Newman

**Сравниваем число рёбер в сообществе с ожидаемым числом рёбер**

$$Q = \frac{1}{2m} \sum_{ij} \left( a_{ij} - \frac{\deg(i) \deg(j)}{2m} \right) \cdot I[x_i = x_j]$$

$x_i$  – метка  $i$ -й вершины

**обоснование – след. слайд**

**как минимизируется**

- симуляция отжига
- спектральные методы и т.п.
- жадные алгоритмы
- попытки объединять/перетаскивать сообщества

## Обоснование модулярности

Уже был приём...

Есть матрица смежности  $A = \| a_{ij} \|_{n \times n}$

Если просуммировать – вектор степеней

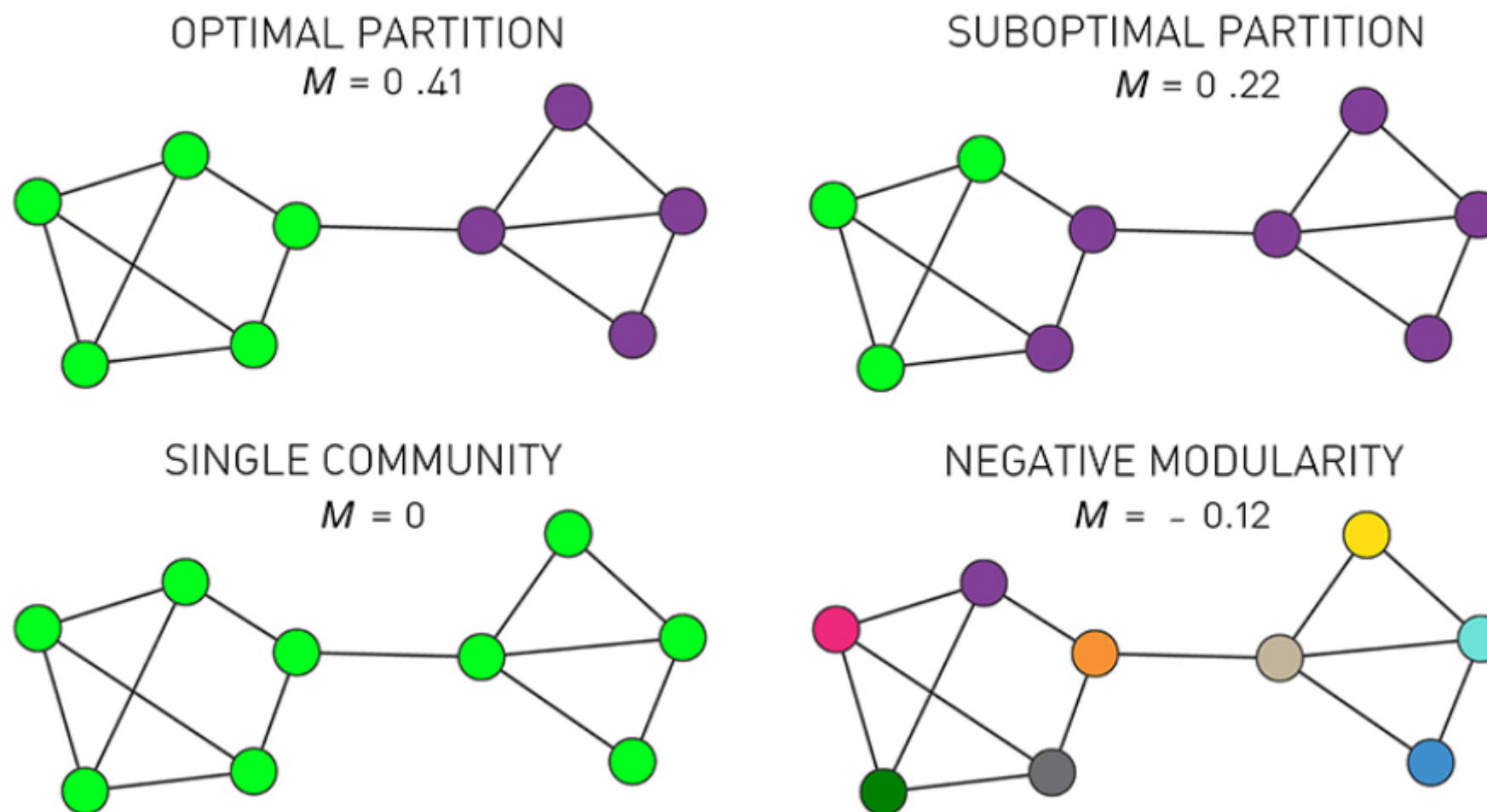
$$\text{sum}(A, \text{axis} = \text{any}) = d = (d_1, \dots, d_n)^T$$

Хотим «случайную матрицу» вероятностей с такими же суммами:

$$\frac{d \cdot d^T}{\text{sum}(d)} = \frac{1}{2m} \| d_i d_j \|_{n \times n}$$

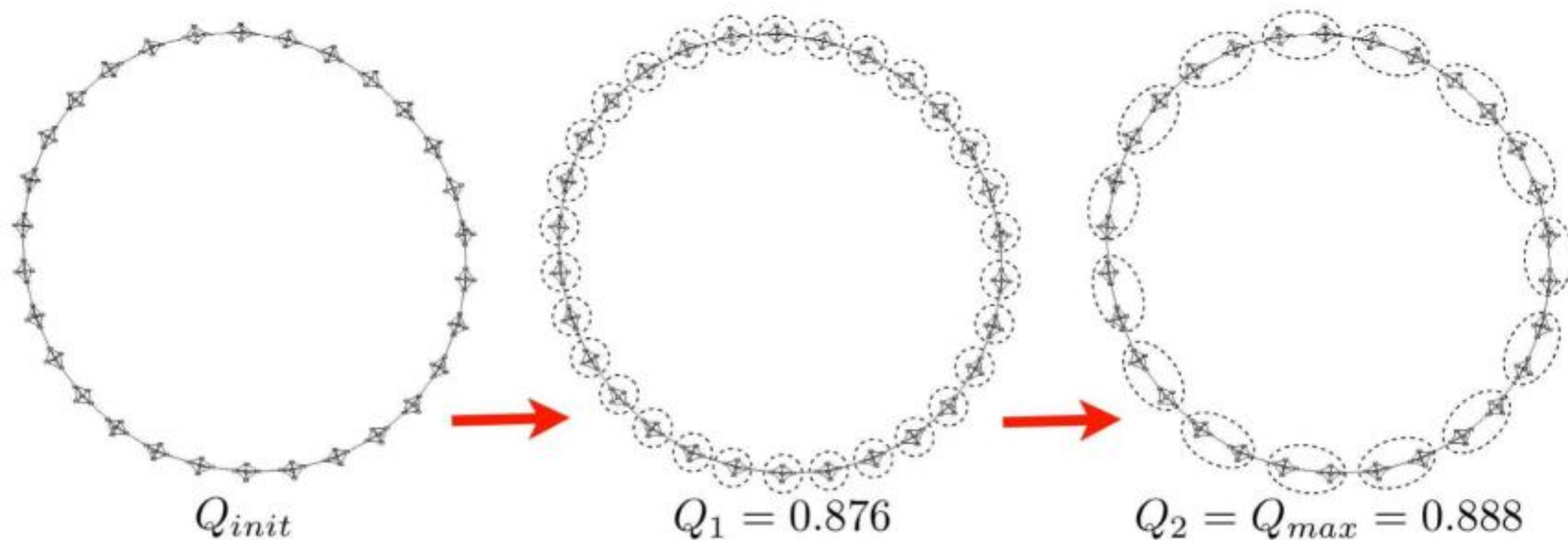
+ нормализация, чтобы была на отрезке  $[-1, +1]$

## Модулярность: иллюстрация



<http://networksciencebook.com/chapter/9>

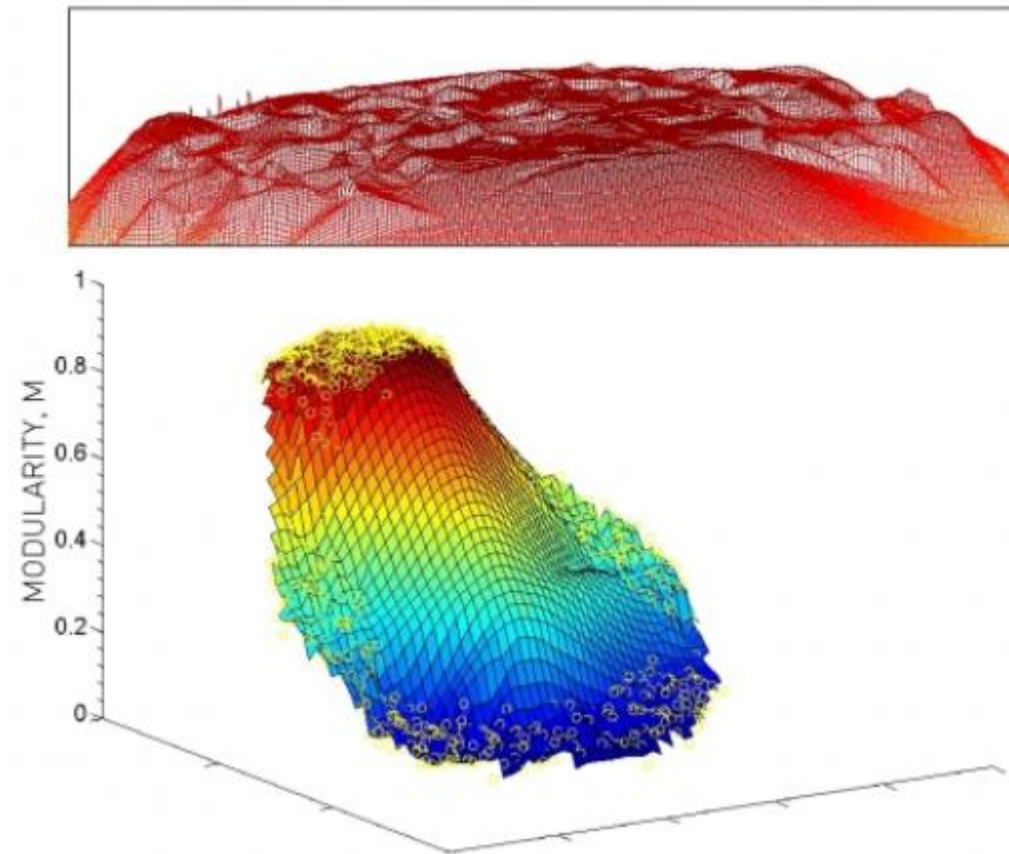
## Иногда модулярность подводит...



**У модулярности есть порог чувствительности: маленькие сообщества не обнаруживает!  $\leq \sqrt{2n_e}$**



## Иногда модулярность подводит...



**У модулярности нет чётко выраженного максимума  
~ плато с пиками**

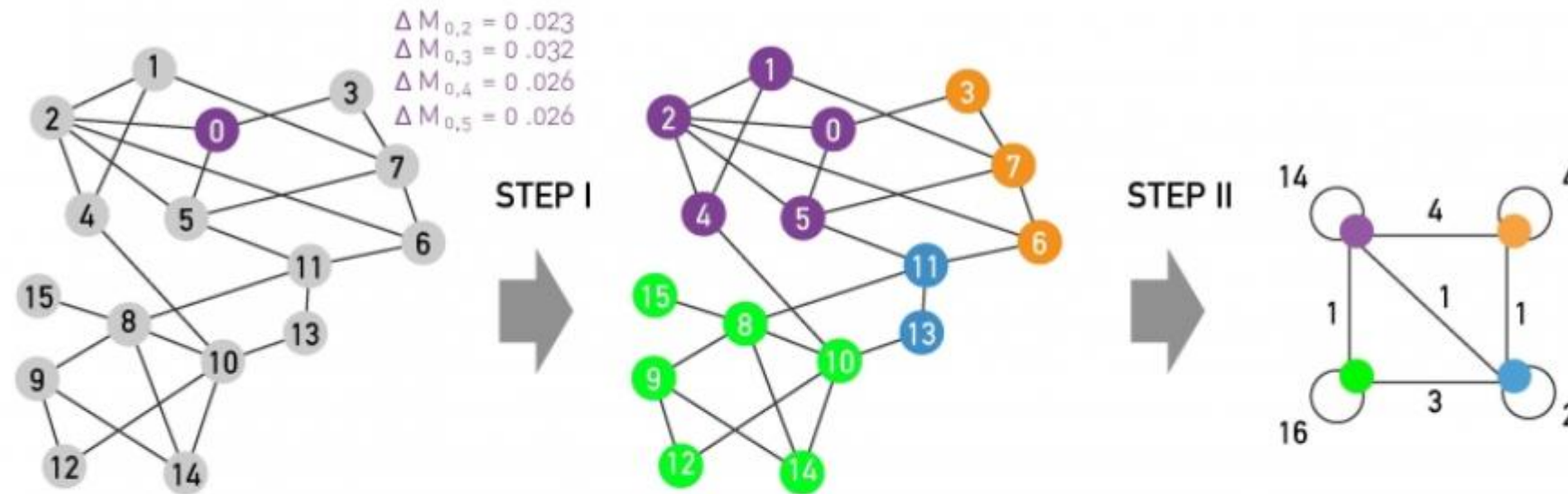
<http://networksciencebook.com/chapter/9>

## **Fast community unfolding: Louvain method / Multilevel**

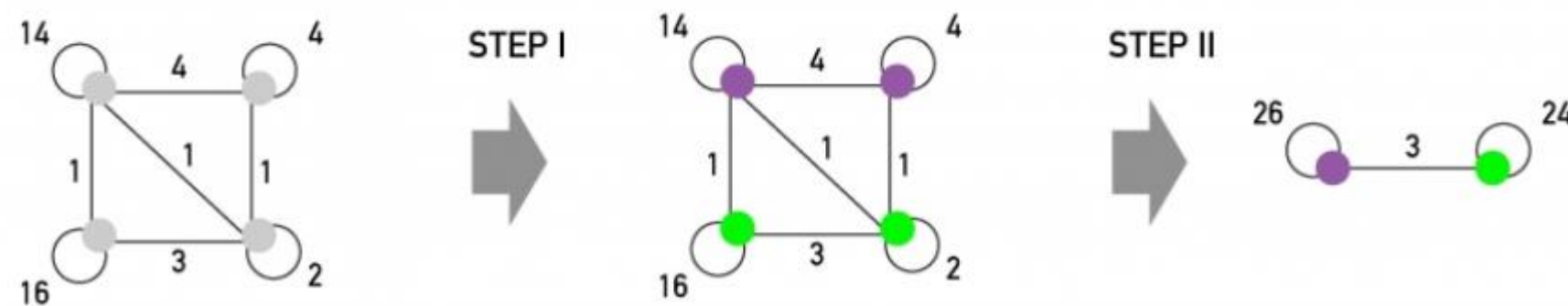
- 1. Каждая вершина приписывается в своё сообщество**
- 2. Пока возможно:**
  - а. Для каждой вершины – изменение модулярности при перемещении её в сообщество (каждого) соседа**
  - б. Максимальное изменение реализуем**
- 3. Пока увеличивается модулярность:**  
**вершины сообществ превращаем в мета-вершины**

## Fast community unfolding: Louvain method / Multilevel

1<sup>ST</sup> PASS



2<sup>ND</sup> PASS



<http://networksciencebook.com/chapter/9>

## **Fast community unfolding: Louvain method / Multilevel**

**Такая жадная оптимизация обычно, когда граф очень большой**

**Все вершины нумеруются и проходятся в порядке номеров  
(пытаемся отнести каждую в соседнее сообщество)**

**Весы при сворачивании – сумма весов рёбер между сообществами**

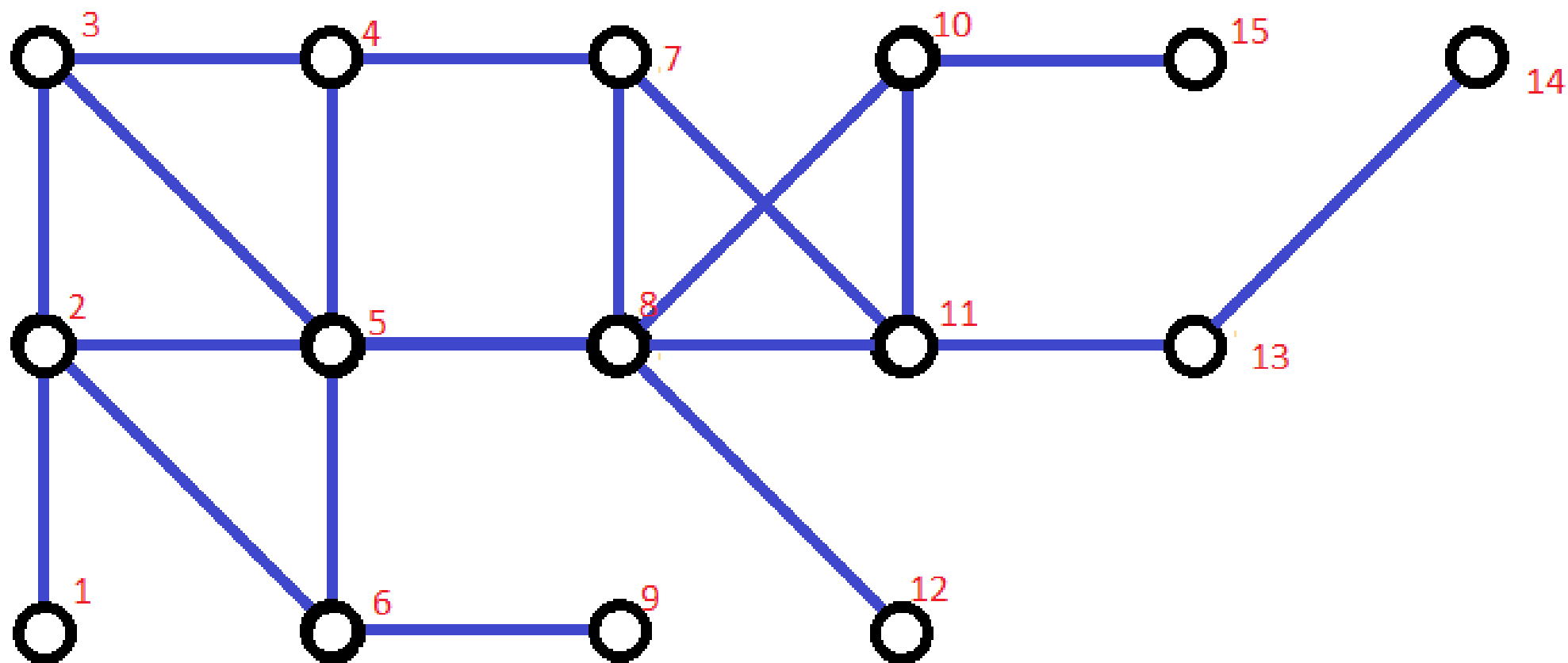
**На практике высокая скорость  $O(n)$**

**V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., 2008.**

**<https://sites.google.com/site/findcommunities/>**

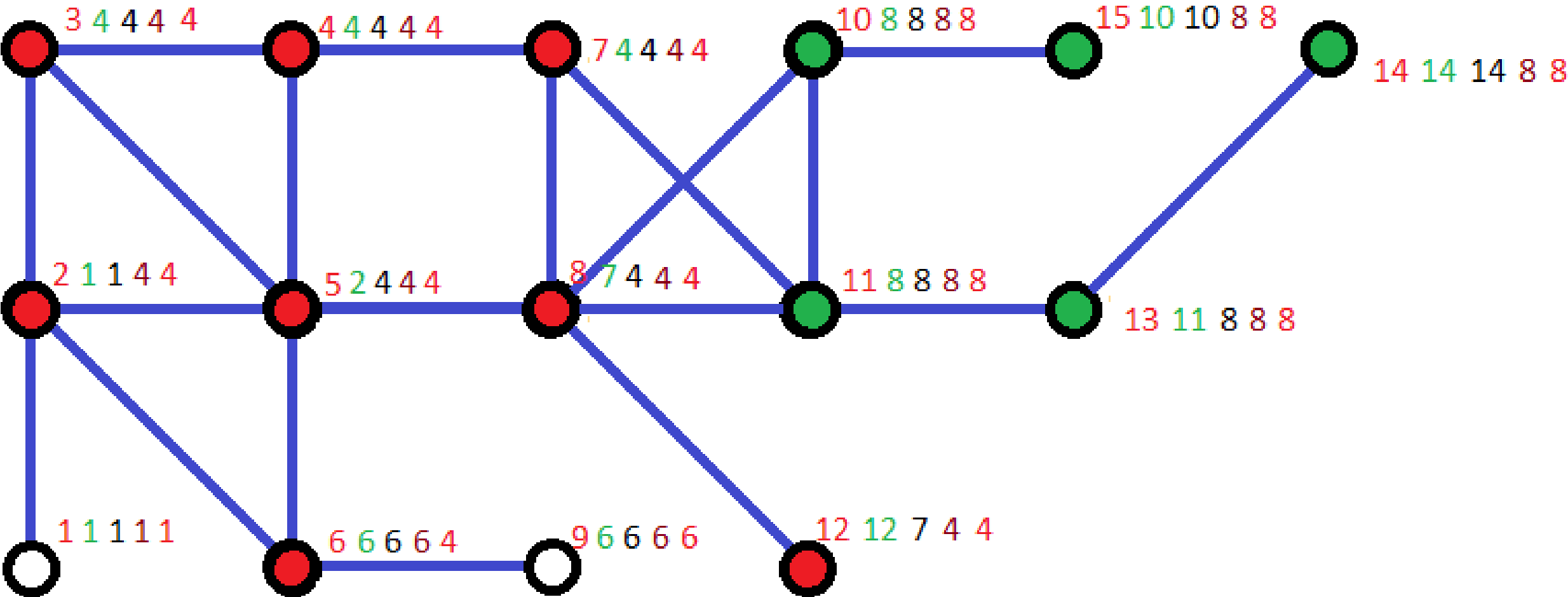
## Выделение сообществ: 4-й способ – Label Propagation

1. Случайно приписать метки вершинам
2. Цикл по вершинам (в случайном порядке)
  - а. Метка вершины заменяется на самую частую метку соседей



Zhu, Xiaojin. «Learning From Labeled and Unlabeled Data With Label Propagation». CiteSeerX 10.1.1.14.3864

Выделение сообществ: 4-й способ – Label Propagation



## Выделение сообществ: 5-й способ – Walktrap

1. Приписать каждую вершину к своему сообществу
2. Пока можно: слить 2 самых ближайших сообщества

**Как измеряется близость сообществ**

$$r_{A,B}(t) = \sqrt{\sum_{i=1}^n \frac{(P_{A,i}^t - P_{B,i}^t)^2}{\deg(i)}} = \| D^{-0.5} P_A^t - D^{-0.5} P_B^t \|,$$

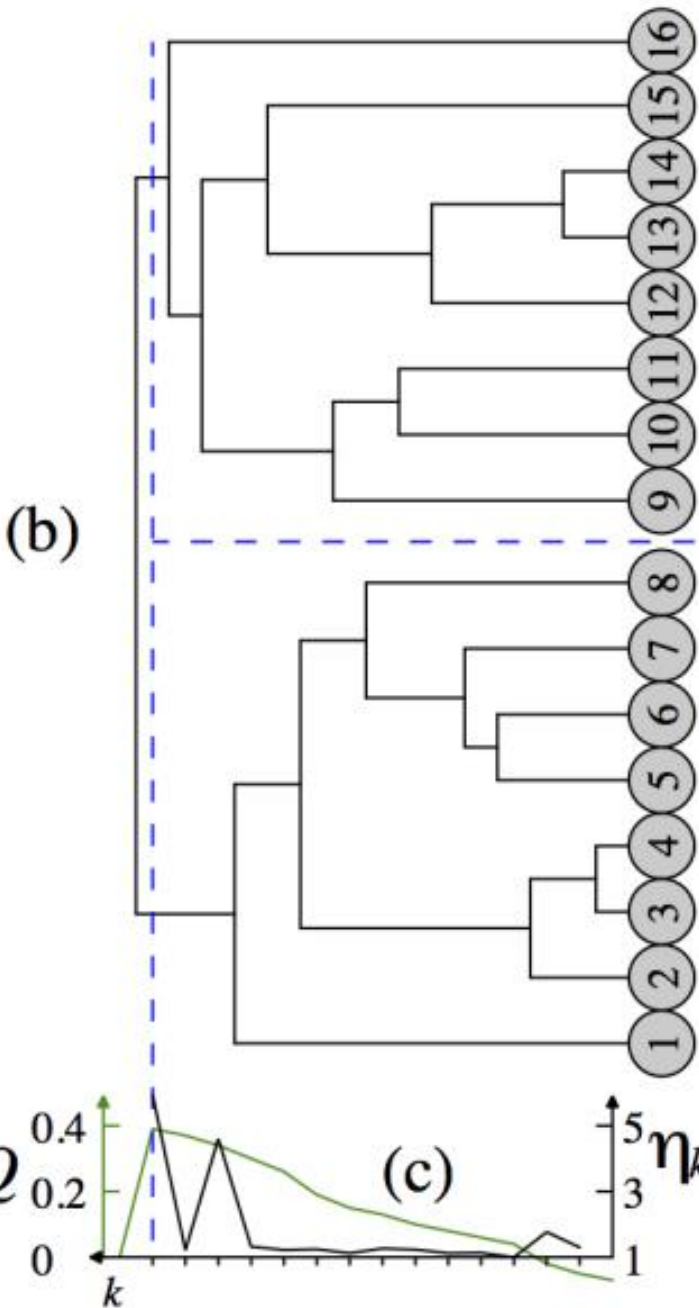
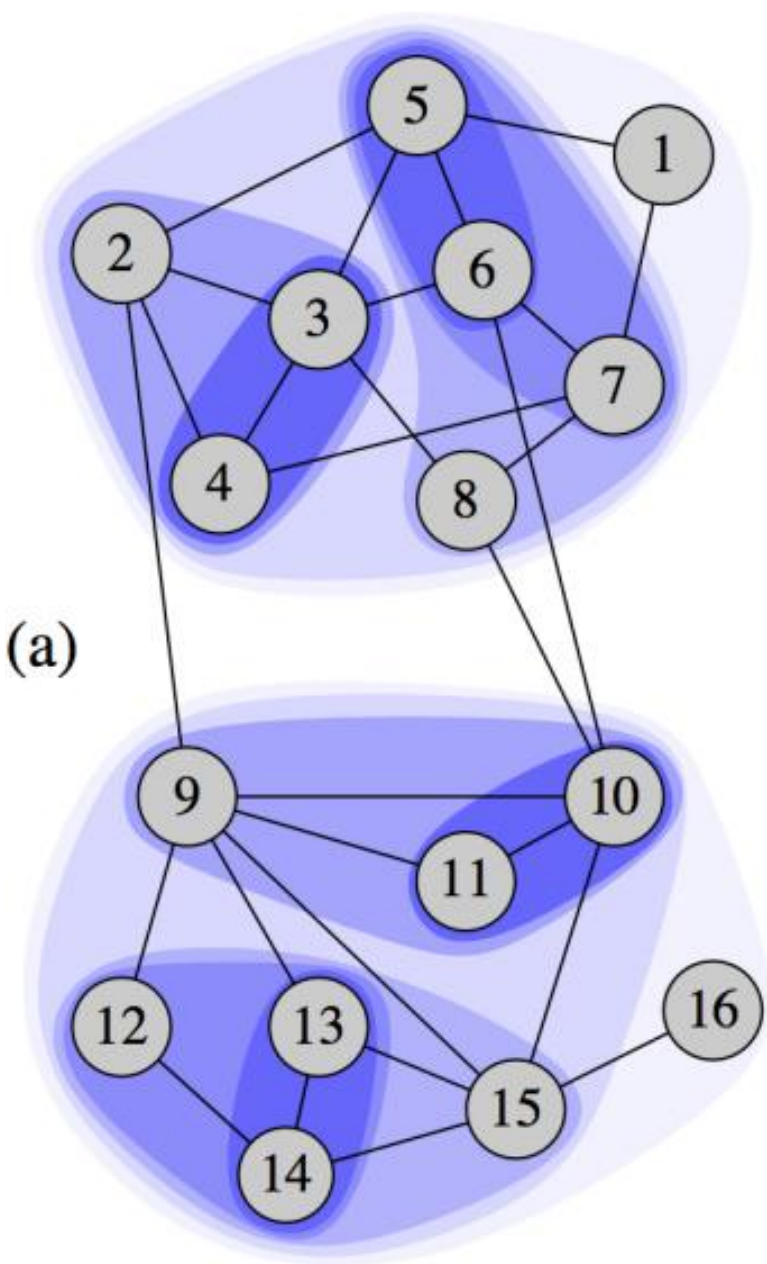
$$P_{A,i}^t = \frac{1}{|A|} \sum_{j \in A} P_{ij}^t$$

$P_{ij}^t$  – вероятность попасть из  $i$  в  $j$  за  $t$  шагов

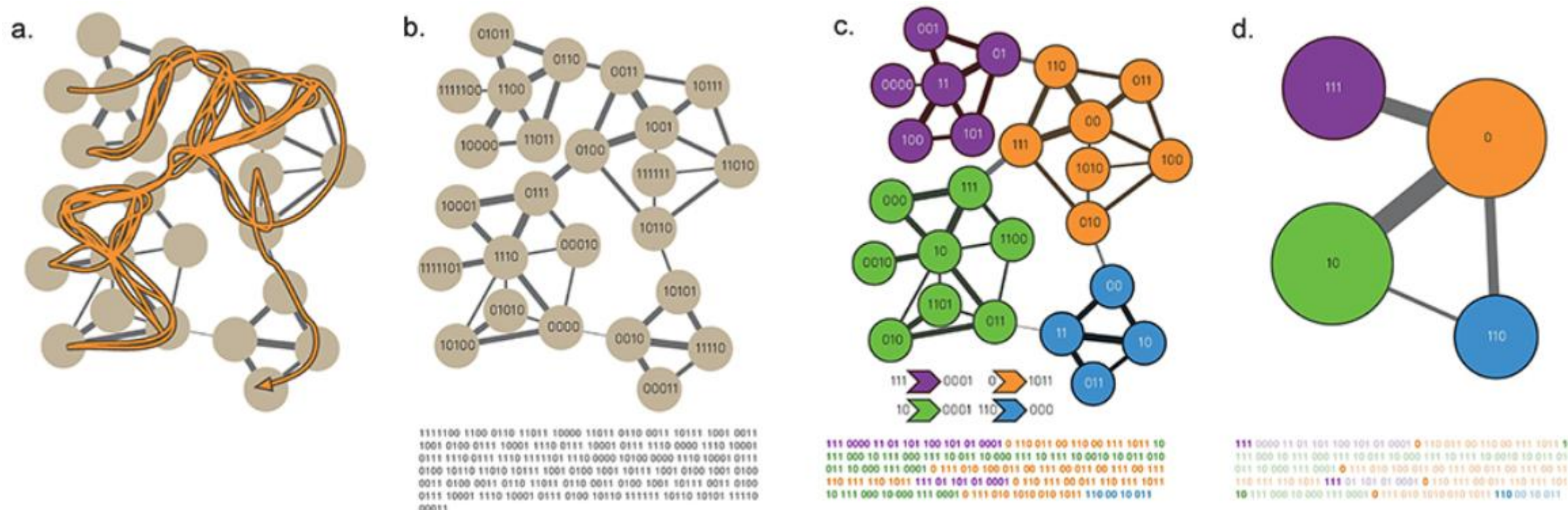
(можно вычислить приближённо – случайными блужданиями)



Walktrap



## Выделение сообществ: 6-й способ – Выделение сообществ: Infomar



**Идея: когда случайно блуждаем (a) надо иметь возможность эффективно кодировать маршрут (d). Для этого кодируются сообщества, вершины в них (эти коды могут пересекаться) + спецсимвол смены сообщества**

**Как и раньше, специальной оптимизацией минимизируем среднюю длину кода**

**Map Equation:**

$$L = qH(Q) + \sum_{c=1}^{n_c} p_c^c H(P_c)$$

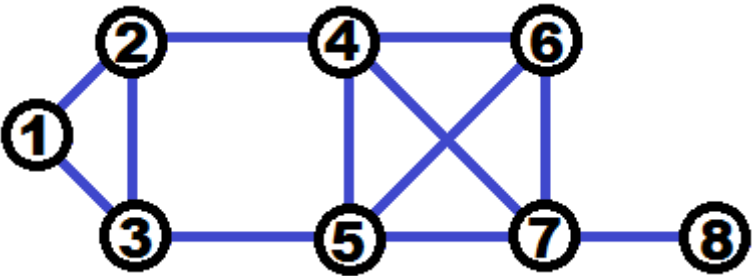
## Другая идея выделения сообществ

**Разбиение графа!**

Выделение сообществ: 7-й способ – спектральная теория графов

Матрица смежности

	1	2	3	4	5	6	7	8
1		1	1					
2	1		1	1				
3	1	1			1			
4		1			1	1	1	
5			1	1		1	1	
6				1	1		1	
7				1	1	1		1
8							1	

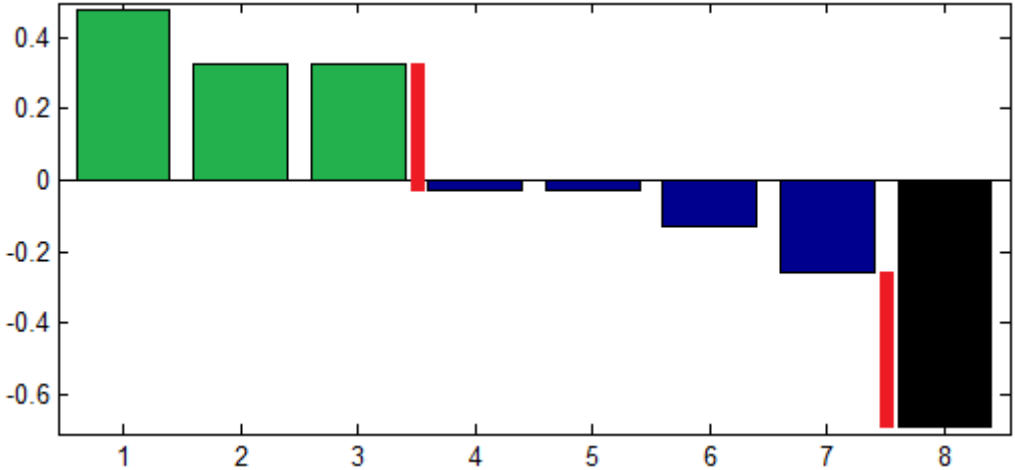


Матрица Лапласа

	1	2	3	4	5	6	7	8
1	2	-1	-1					
2	-1	3	-1	-1				
3	-1	-1	3		-1			
4		-1		4	-1	-1	-1	
5			-1	-1	4	-1	-1	
6				-1	-1	3	-1	
7				-1	-1	-1	4	-1
8							-1	1

```
L = full(diag(sum(S))-S);
[X,Y] = eig(L);
bar(X(:,2))
```

-0.3536	0.4758	0.4032	0.6744	0.0000	0.1498	-0.0938	-0.0000
-0.3536	0.3271	0.1388	-0.4363	0.6015	-0.1862	0.1540	-0.3717
-0.3536	0.3271	0.1388	-0.4363	-0.6015	-0.1862	0.1540	0.3717
-0.3536	-0.0261	-0.3076	-0.1099	0.3717	0.3132	-0.4117	0.6015
-0.3536	-0.0261	-0.3076	-0.1099	-0.3717	0.3132	-0.4117	-0.6015
-0.3536	-0.1307	-0.4737	0.3524	0.0000	-0.7131	0.0292	0.0000
-0.3536	-0.2583	-0.1846	0.1162	0.0000	0.4336	0.7568	0.0000
-0.3536	-0.6889	0.5926	-0.0506	-0.0000	-0.1244	-0.1767	-0.0000

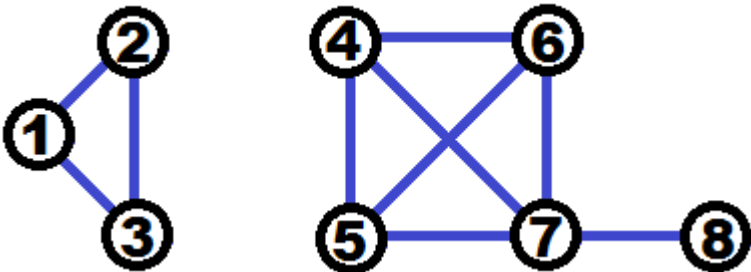


Всё содержится в одном векторе! И на одном слайде!  
Потом – теоретическое обоснование

Спектральная теория графов

Первый с.в. – константный  
Второй с.в. – отражает разбиение графа

Но когда граф несвязный...



```
L =
0.5774      0      0      0.2673      0.7715      0      0      0
full(diag(sum(S)) -
S);
0.5774      0      0      -0.8018     -0.1543      0      0      0
0.5774      0      0      0.5345     -0.6172      0      0      0
0      -0.4472     -0.2887      0      0      0.1274     -0.8065      0.2236
[X,Y] = eig(L);
0      -0.4472     -0.2887      0      0      0.6348      0.5136      0.2236
0      -0.4472     -0.2887      0      0      -0.7621      0.2929      0.2236
0      -0.4472      0.0000      0      0      0      0     -0.8944
0      -0.4472      0.8660      0      0      0      0      0.2236

diag(Y)' =  -0.0000      0.0000      1.0000      3.0000      3.0000      4.0000      4.0000      5.0000
```

Теперь два «константных» вектора!

**Проблема разбиения графа [не совсем из теоретической части]**

$$x^T L x = \sum_{(i,j)} (x_i - x_j)^2 \rightarrow \min_x,$$

**если**  $x = (x_1, \dots, x_n) \in \{\pm 1\}^n$ , **то минимизация логична для разбиения (надо избежать очевидного константного решения).**

**Но это сложная переборная задача, поэтому вместо**  $x = (x_1, \dots, x_n) \in \{\pm 1\}^n$   
**решают вещественную задачу с ограничениями**

$$\tilde{1}^T x = 0, \|x\| = 1$$

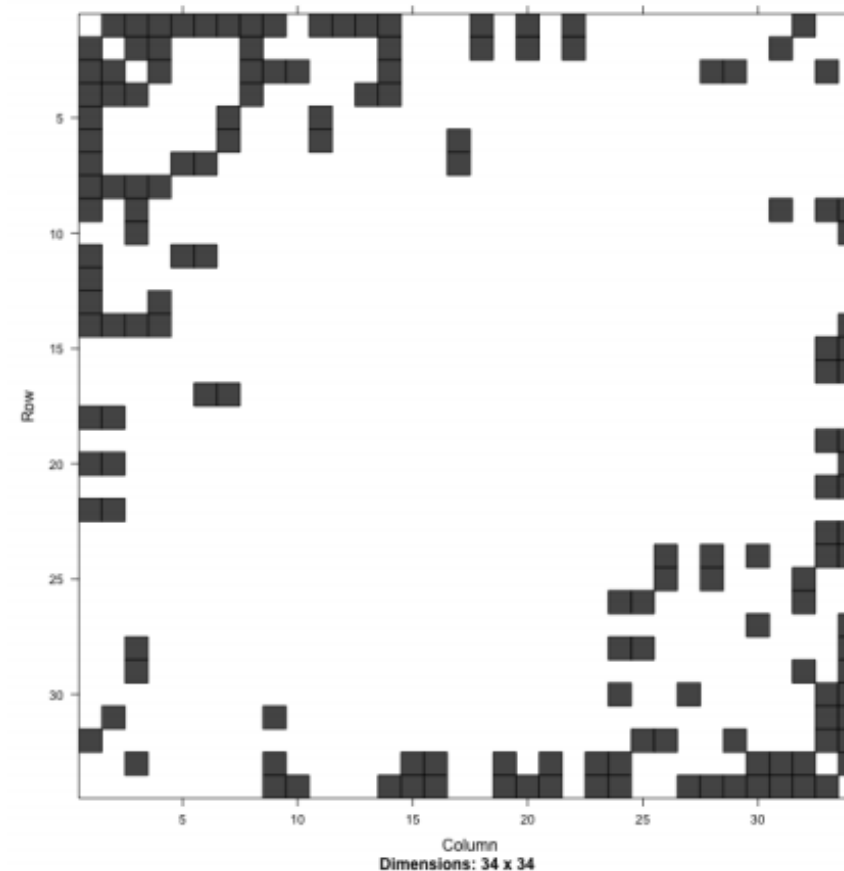
**(ортогональность, чтобы не подходила константа).**

**Решение – собственный вектор, соответствующий второму по величине с.з. матрицы Лапласа.**

**Потом**  $(\text{sgn}(x_1), \dots, \text{sgn}(x_n))$ .

## Совмещение идей

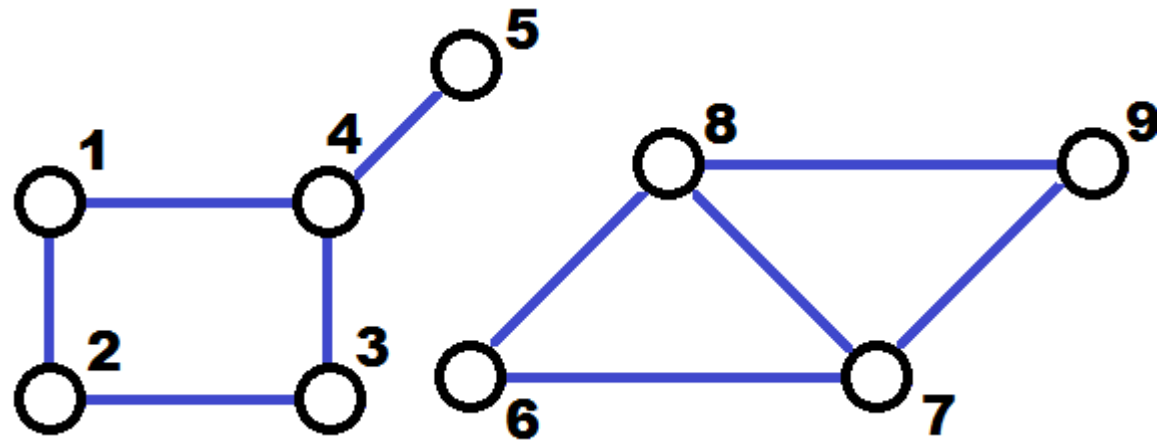
1. Найти второй собственный вектор
2. По его значениям упорядочить вершины



3. Как именно делить решаем по отдельному функционалу (ex: модулярность), надо перебрать всего  $n-1$  деление.

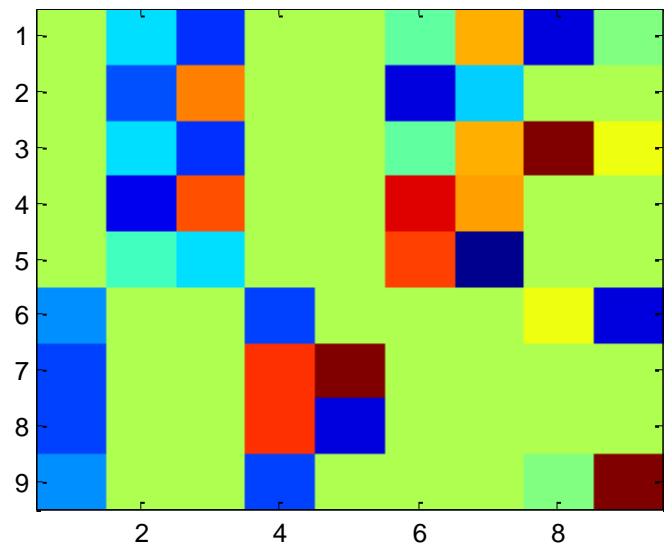
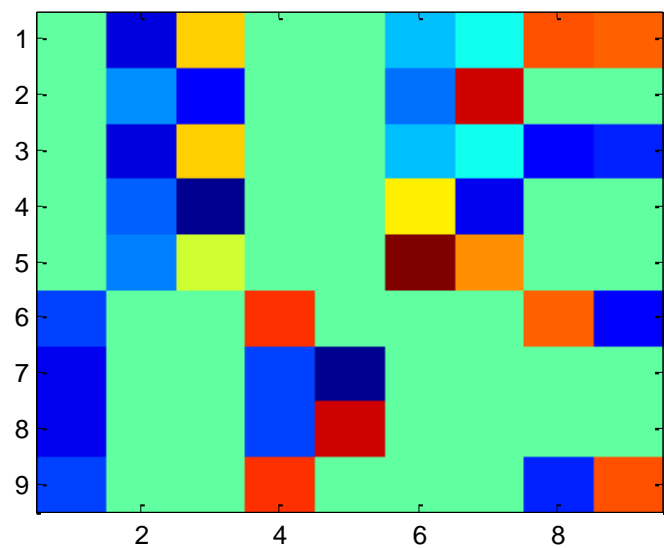


SVD над матрицей смежности

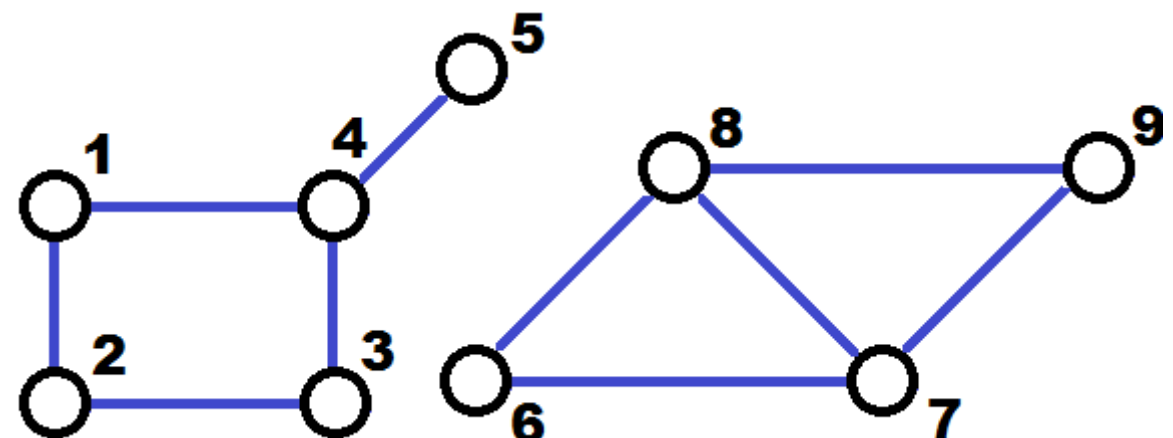


```
S = sparse([1 1 2 2 3 3 4 4 6 6 7 7 8 8 8 9 9 5 4 7], ...
           [2 4 1 3 2 4 1 5 7 8 8 9 6 7 9 8 7 4 3 6], 1)
[U L V] = svds(S,9);
disp(U)
disp(V)
disp(diag(L) ')
```

0.0000	-0.5295	-0.3893	0.0000	0.0000	-0.2441	0.0923	-0.2743	0.6518
0.0000	0.3646	-0.4958	-0.0000	-0.0000	-0.2787	-0.7373	-0.0000	-0.0000
0.0000	-0.5295	-0.3893	0.0000	-0.0000	-0.2441	0.0923	0.2743	-0.6518
0.0000	0.4669	-0.6350	0.0000	0.0000	0.2176	0.5757	0.0000	0.0000
0.0000	-0.2973	-0.2186	-0.0000	-0.0000	0.8694	-0.3286	-0.0000	0.0000
-0.4352	0.0000	-0.0000	-0.5573	0.0000	0.0000	0	0.6518	0.2743
-0.5573	-0.0000	-0.0000	0.4352	-0.7071	0.0000	-0.0000	0.0000	-0.0000
-0.5573	0.0000	-0.0000	0.4352	0.7071	0	0.0000	-0.0000	-0.0000
-0.4352	0	-0.0000	-0.5573	0.0000	-0.0000	0	-0.6518	-0.2743
0.0000	0.3893	-0.5295	0.0000	0.0000	-0.0923	-0.2441	-0.7068	-0.0208
0.0000	-0.4958	-0.3646	-0.0000	-0.0000	-0.7373	0.2787	0.0000	-0.0000
-0.0000	0.3893	-0.5295	-0.0000	0.0000	-0.0923	-0.2441	0.7068	0.0208
0.0000	-0.6350	-0.4669	0.0000	0.0000	0.5757	-0.2176	0.0000	-0.0000
-0.0000	0.2186	-0.2973	-0.0000	-0.0000	0.3286	0.8694	-0.0000	-0.0000
-0.4352	0	-0.0000	0.5573	-0.0000	0.0000	0	-0.0208	0.7068
-0.5573	0	-0.0000	-0.4352	0.7071	0.0000	0.0000	0.0000	0.0000
-0.5573	-0.0000	-0.0000	-0.4352	-0.7071	-0.0000	-0.0000	0.0000	0.0000
-0.4352	0	-0.0000	0.5573	-0.0000	0	0	0.0208	-0.7068
2.5616	2.1358	2.1358	1.5616	1.0000	0.6622	0.6622	0.0000	0.0000

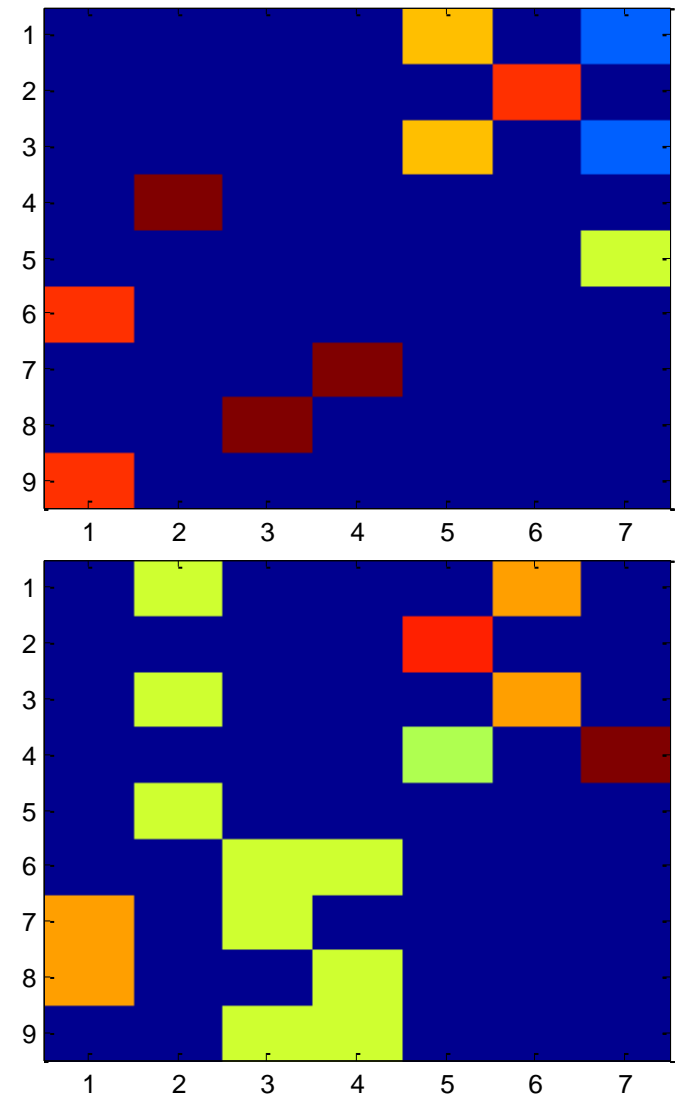


Неотрицательные матричные разложения



```
S = sparse([1 1 2 2 3 3 4 4 6 6 7 7 8 8 8 9 9 5 4 7], ...
           [2 4 1 3 2 4 1 5 7 8 8 9 6 7 9 8 7 4 3 6], 1)
[U,V] = nnmf(S,7);
disp(U)
disp(V')
```

0	0	0.0000	0	1.1234	0.0000	0.4880
0	0	0.0000	0.0000	0	1.4142	0.0000
0	0	0.0000	0	1.1234	0.0000	0.4880
0	0	0.0000	1.7070	0.0000	0.0308	0
0.0000	0.0000	0	0	0	0	1.0000
0.0006	1.4145	0	0	0.0000	0	0
2.8290	1.4145	0	0.0000	0.0000	0	0
0.0000	0.0000	1.7321	0	0	0	0.0000
0.0006	1.4145	0	0	0.0000	0	0
0.0000	0	0	0.5731	0.0000	0.7071	0
0.0000	0	0	0	0.8901	0.0000	0.0000
0.0000	0	0	0.5731	0.0000	0.7071	0
0.0000	0	0	0	0.4557	0	1.0000
0	0	0	0.5858	0	0	0
0.7071	0	0.5774	0	0.0000	0	0
0	0.7072	0.5774	0	0	0	0
0.0000	0.7070	0	0	0.0000	0.0000	0
0.7071	0	0.5774	0	0.0000	0	0



**Spectral modularity maximization [Newman, 2006]****Если  $x_i \in \{\pm 1\}$ , то**

$$Q = \frac{1}{2n} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) (x_i x_j + 1), \text{ тогда}$$

$$\frac{1}{2n} \sum_{ij} \underbrace{\left( A_{ij} - \frac{k_i k_j}{2m} \right)}_{B_{ij}} x_i x_j \rightarrow \min .$$

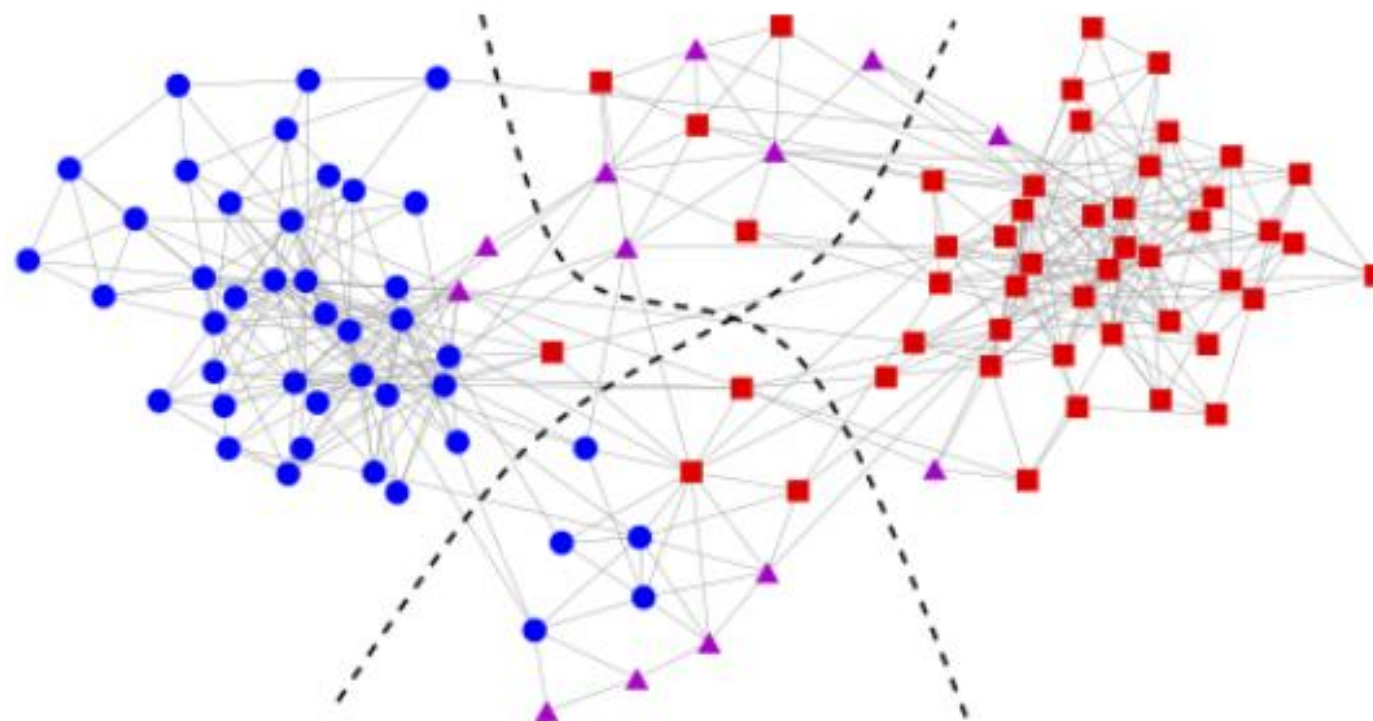
**Вычислить  $k = \deg(A)$ ,**

$$B = A - \frac{1}{2m} k k^T,$$

**Найти max с.в.  $Bv = \lambda v$   
 $\text{sgn}(v)$** **т.е. в задаче на с.з. используют разные матрицы...**

## Тестирование разных методов

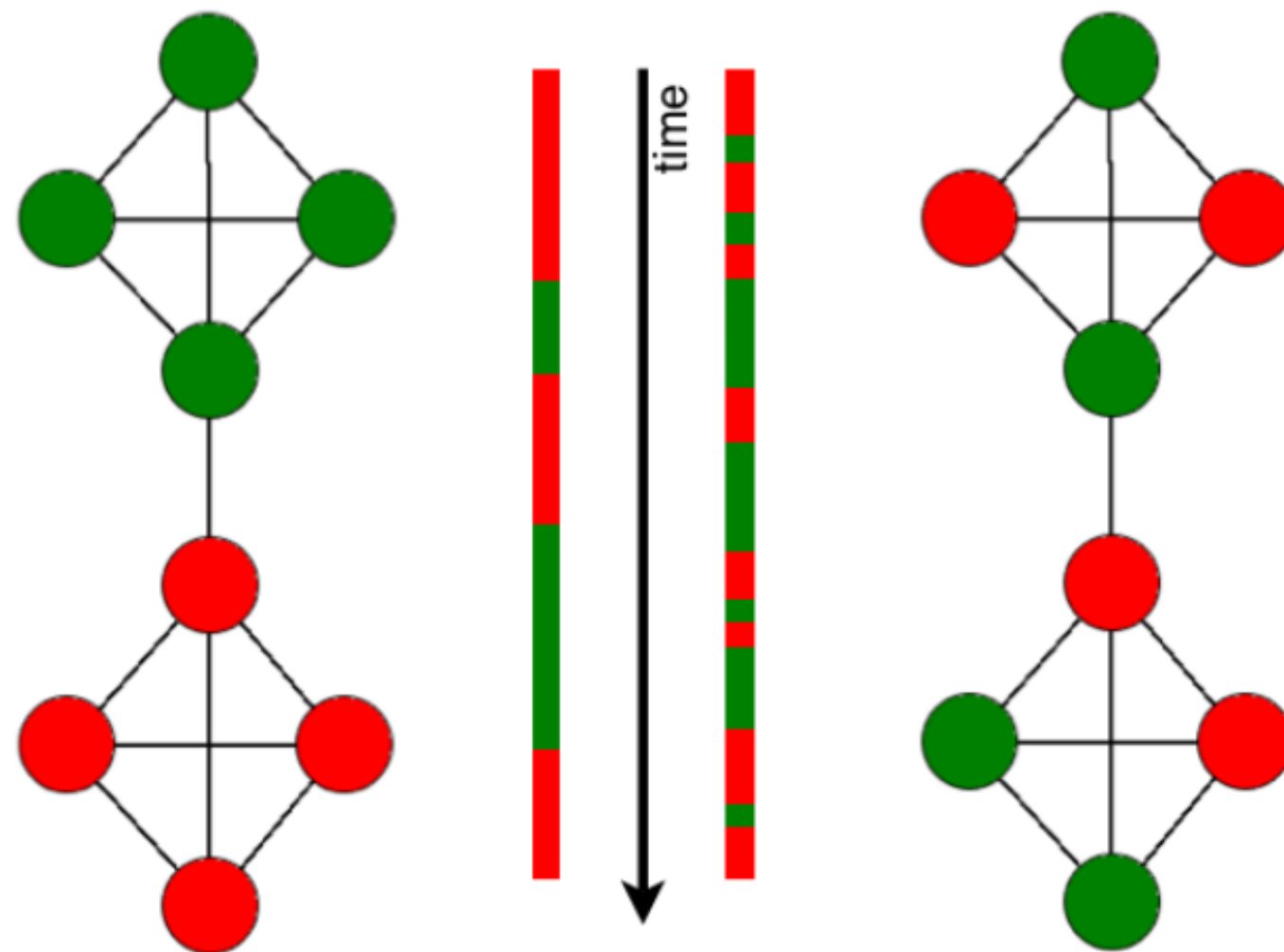
**1) Можно на реальных графах,  
где есть мета-информация о реальных сообществах**



**сеть книг о политике [источник?]**

**2) Можно на модельных графах**

## Стабильность разбиения

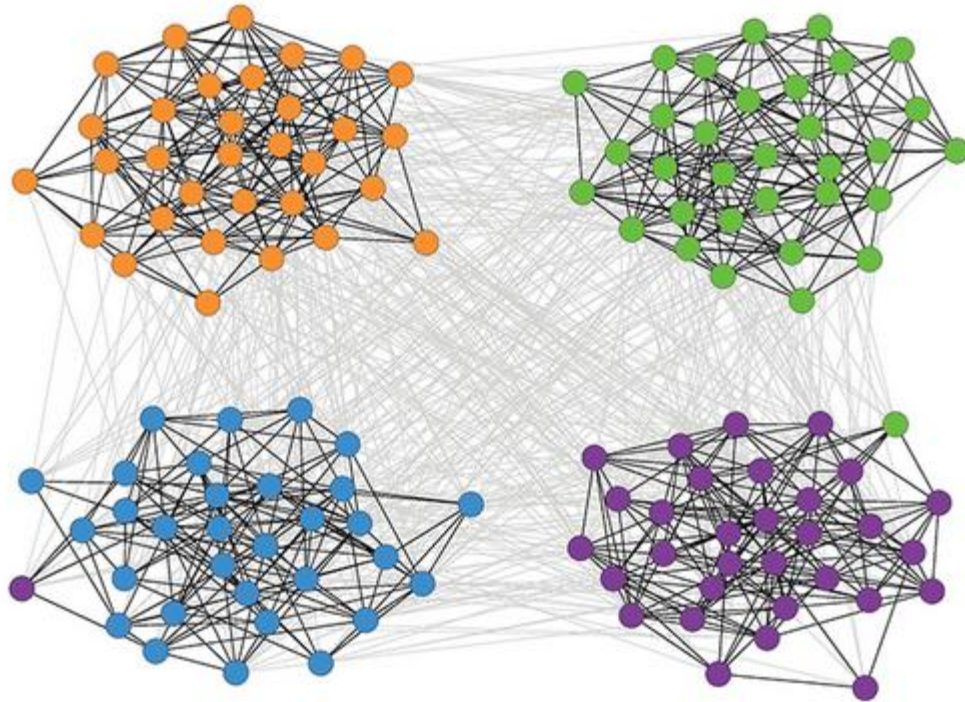


**Если блуждаем по графу,  
то должны долго быть в пределах одного сообщества**

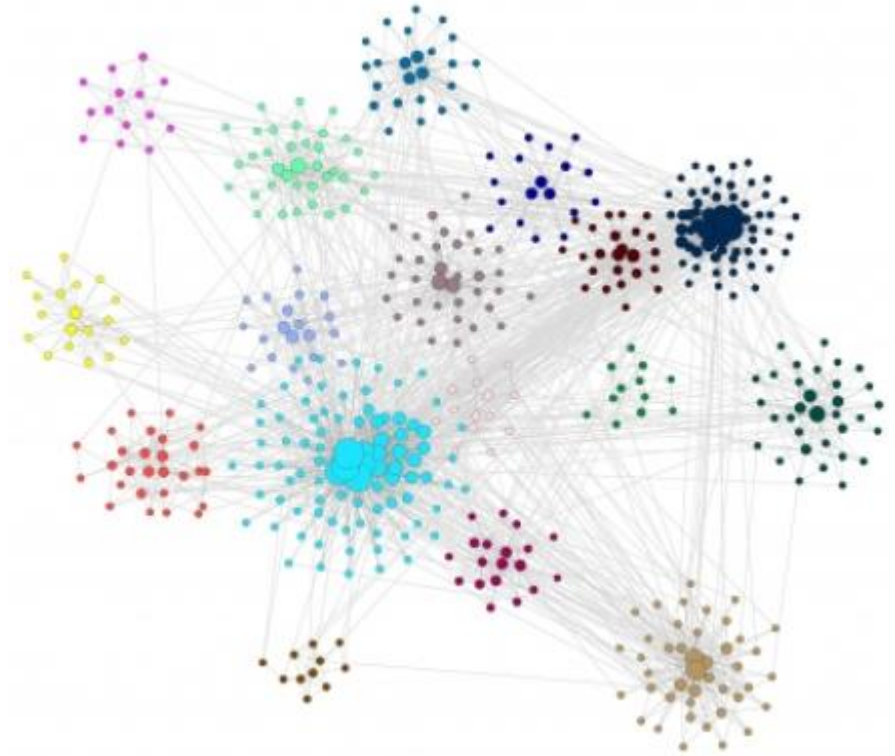
J.-C. Delvenne, S. Yaliraki & M. Barahona, *Stability of graph communities across time scales*. arXiv:0812.1811.

## Пример модельной задачи

### Girvan-Newman (GN) Benchmark



### Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

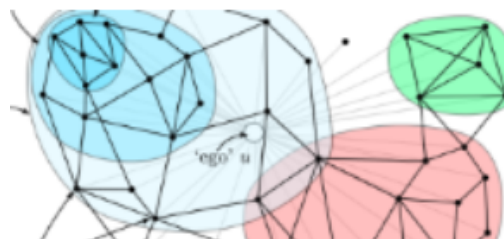


<http://networksciencebook.com/chapter/9>



## Задача

# Выделение кругов пользователей в эго-подграфах графов социальной сети



Knowledge • 122 teams

## Learning Social Circles in Networks

Tue 6 May 2014

Enter/Merge by

Tue 28 Oct 2014 (27 days to go)

### Dashboard

Home



Data



Make a submission



Information



Description

Evaluation

Rules

FAQ

Timeline

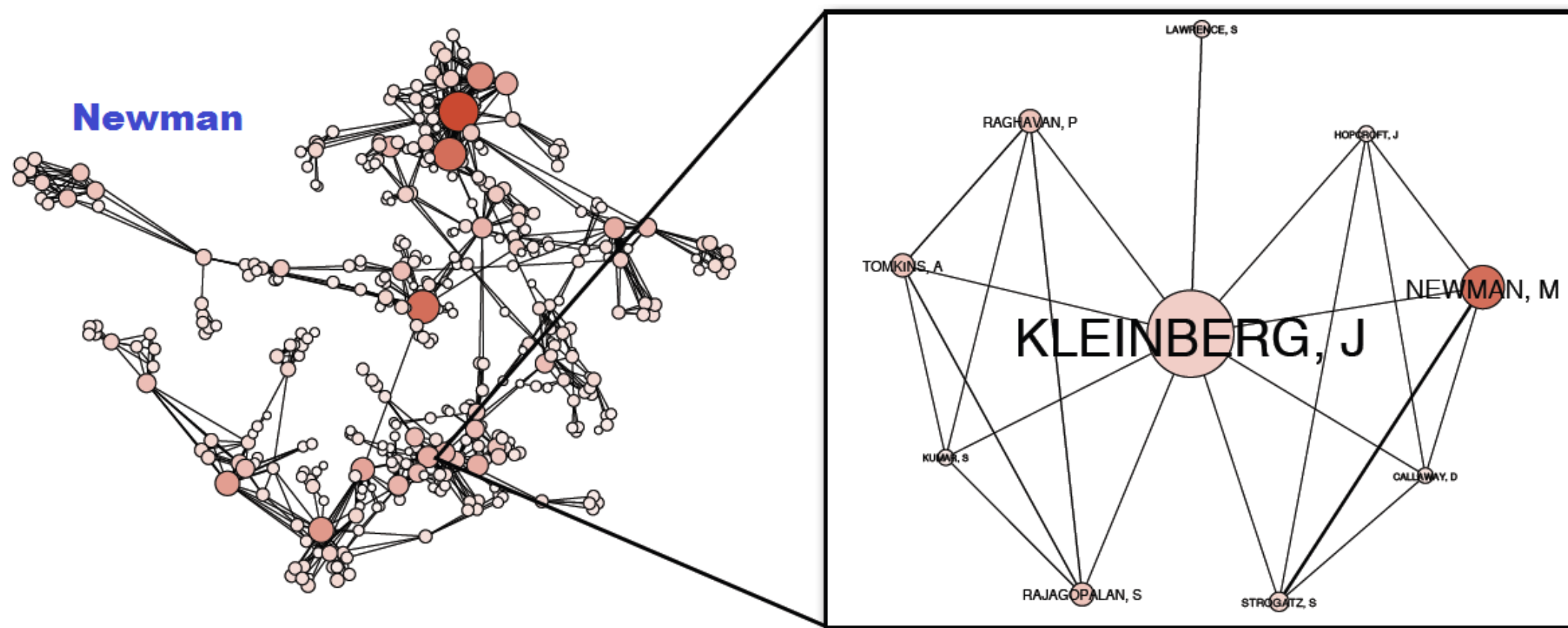
Forum

[Competition Details](#) » [Get the Data](#) » [Make a submission](#)

## Model friend memberships to multiple circles

Social Circles help users organize their personal social networks. These are implemented as "circles" on Google+, and as "lists" on Facebook and Twitter. Each circle consists of a subset of a particular user's friends. Such circles may be disjoint, overlap, or be hierarchically nested.

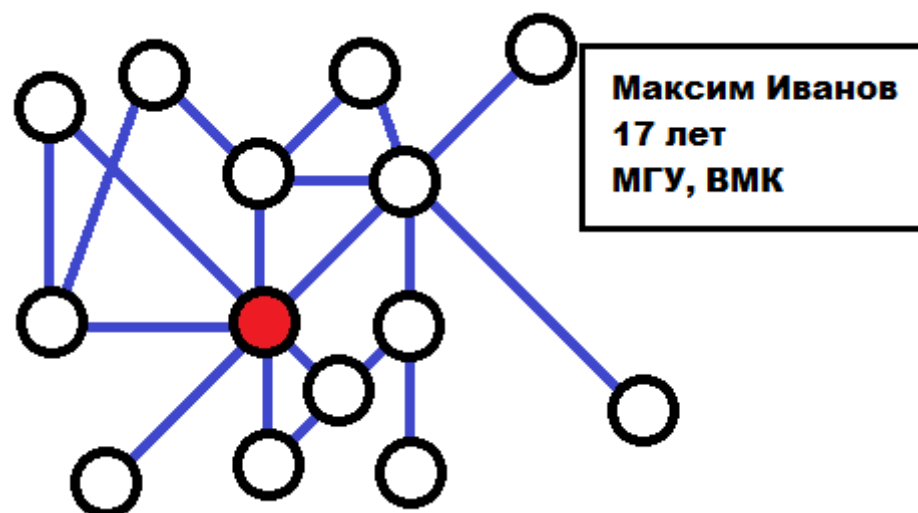
## Эго-подграфы



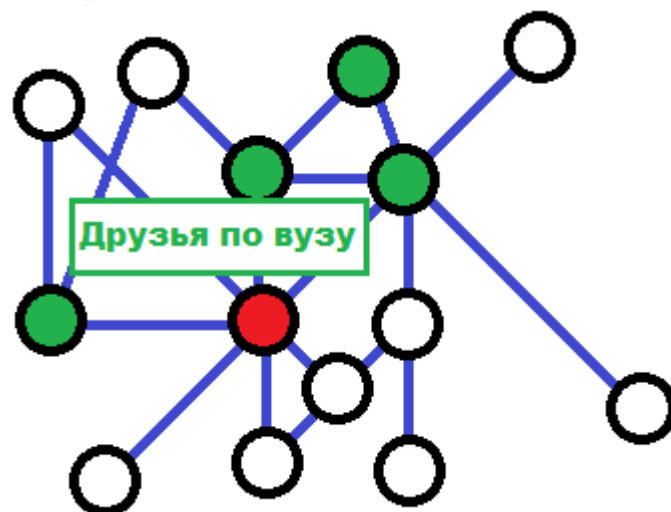
**окрестность порядка 1**  
**(не обязательно связный граф – без порождающей вершины)**



## Задача определения кругов



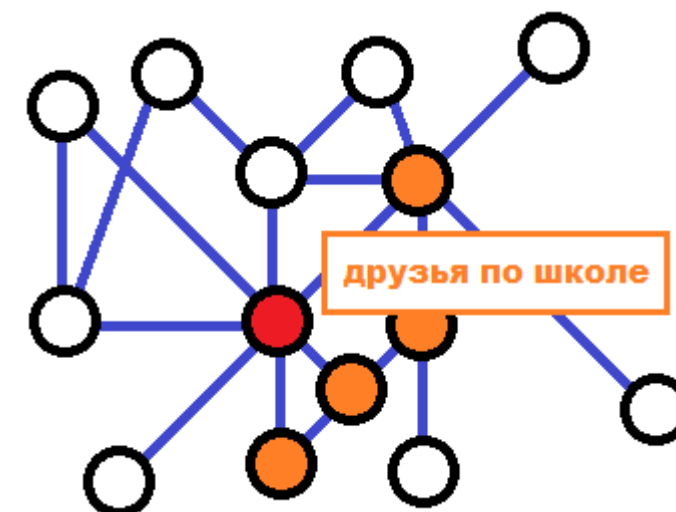
**Здесь:** соцсеть =  
граф + признаки вершин



**Круг – подмножество друзей**  
**Определяет пользователь**  
**Себя в круг не включает**

**Круги могут пересекаться**  
**Не все друзья в кругах**

**Что в данных говорит о круге?**



## Обучение

для 60 пользователей – круги

всего: 110 эго-сетей

всего: 27520 пользователей (основных + друзей + друзей друзей)

57 признаков для описания этих пользователей

## Контроль

50 пользователей

## Файл ответа

```
UserId,Predicted
25708,25709 25710;25711 25712
2473,2474 2475 2476 2477;2478 2479
...
```

## Качество

«редакторское расстояние»

Качество – редакторское расстояние

операции (стоимость = 1)

добавление к кругу

создание круга с одним «юзером»

удаление из круга

удаление круга с одним «юзером»

1 2 3;4 5;6  
1 2 3; 4 5 [delC]  
2 3; 4 5 [del]  
2 3; 4 5; 1 [insC]  
2 3; 4 5 6; 1 [ins]

4 операции = 1 + 1 + 2

```
% редакторское расстояние
function cost = myeditloss(list1,list2)

n = max(length(list1),length(list2));
M = zeros(n); % матрица отличий кругов

for i = 1:n
    if i<=length(list1)
        set1 = list1{i};
    else
        set1 = [];
    end;
    for j = 1:n
        if j<=length(list2)
            set2 = list2{j};
        else
            set2 = [];
        end;
        M(i,j) = length(setxor(set1, set2));
    end;
end;

% венгерский алгоритм
[assignment,cost] = munkres(M);
```

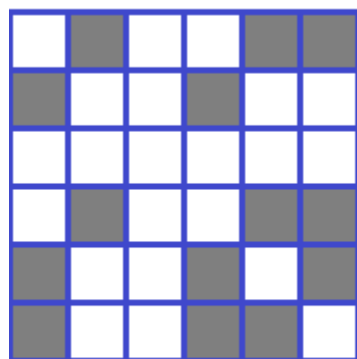
	2 3	4 5 6	1
1 2 3	1	6	2
4 5	4	1	3
6	3	2	2

**Описание метода решения –**  
**сингулярное разложение матрицы смежности**

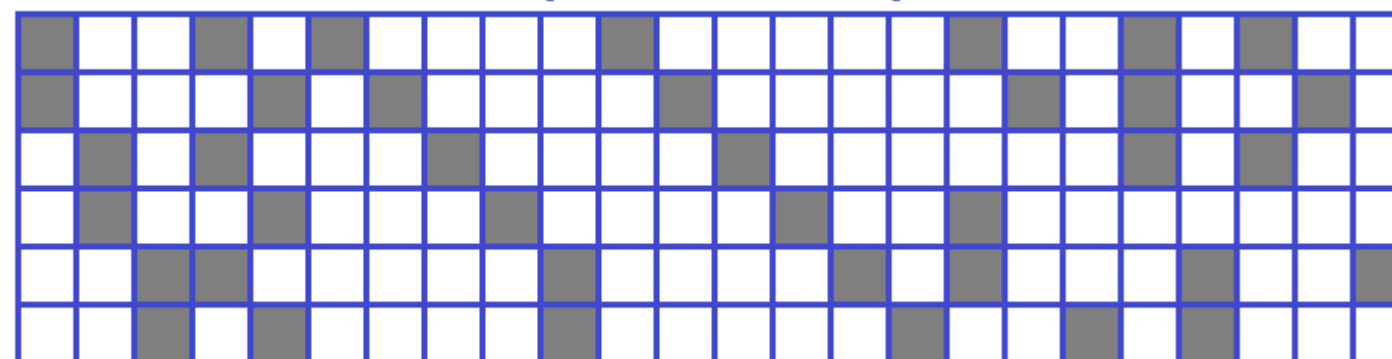
**Есть возможность использовать признаковые описания**

**Просто добавляется признаковая матрица**

матрица смежности



признаковая матрица



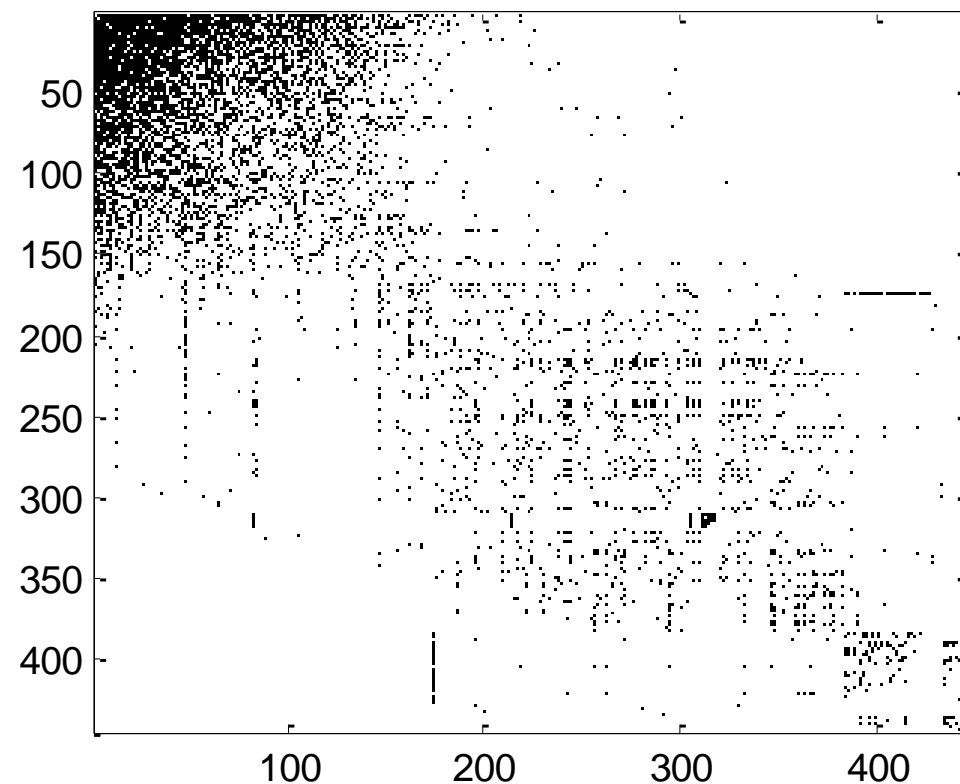
один категориальный признак

$$[U \ L \ V] = \text{svds}(M * M' + \alpha * X * X', k\_svd);$$

**К сожалению, нет хорошего эффекта...**

**Вопрос: какую матрицу раскладывать,  
смежности, Лапласа, с нормировками...**

## Оправдание алгоритма

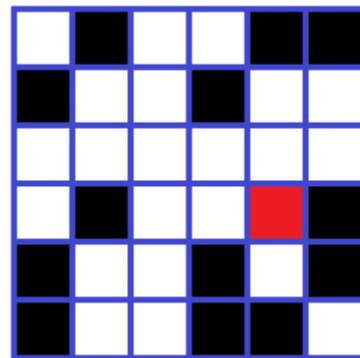


**Матрица смежности (упорядоченность вершин по первой компоненте)  
действительно, есть факторизация**

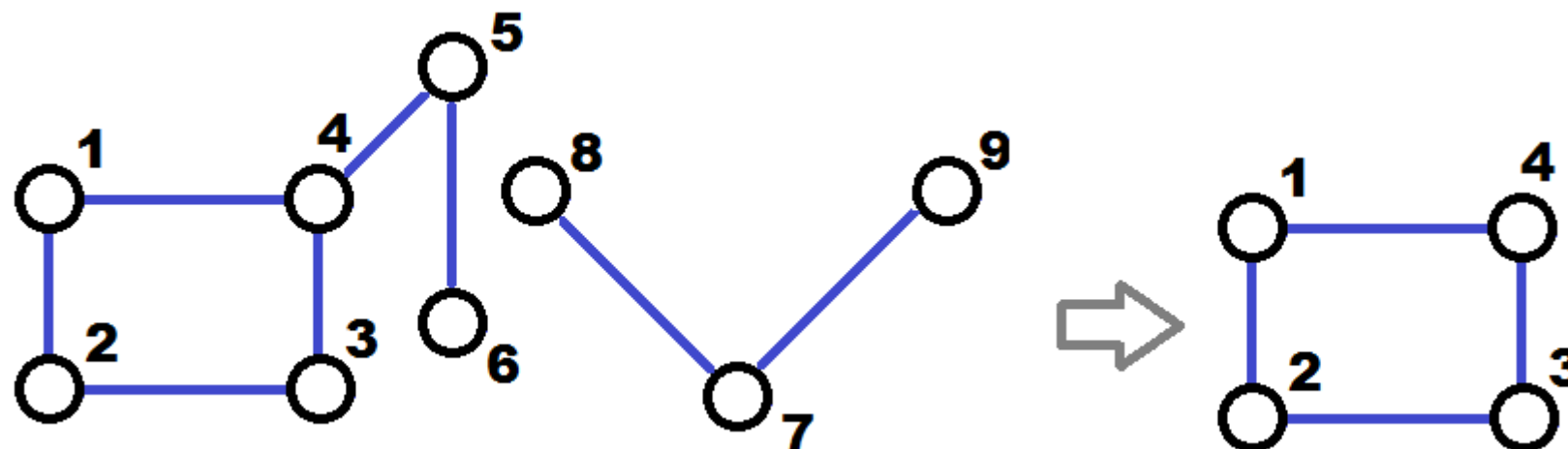
**Идея:** ввести рейтинг принадлежности к компоненте  
– значение в векторе сингулярного разложения

## Этапы алгоритма

### 1. Получение матрицы смежности (симметризация) не все матрицы были симметричными



### 2. Удаление висячих вершин

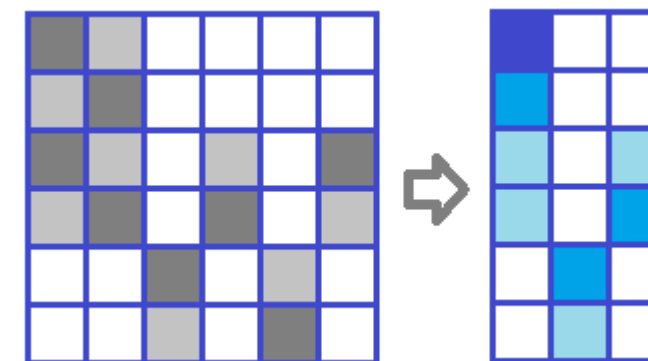


## Этапы алгоритма

### 3. SVD разложение, k=90

```
[U, ~, ~] = svds(M, min(min(size(M)), ksvd));
U = abs(U);
U = bsxfun(@rdivide, U, sqrt(sum(U.^2)));
RU = U' * U;
RUp = (RU > pcorr);

ans1 = {};
for i=1:size(U,2)
    Irup = RUp(i,:);
    if any(Irup)
        x = mean(U(:, Irup), 2);
        circ_4ans = getcircleit2(M, x, fI, gc1, gc2, gc3);
        [ans1, isadd] = addcircle2ans(ans1, circ_4ans, padd);
        RUp(:, Irup) = false;
    end;
end;
ans1 = delintersects(ans1);
```



**объединяем похожие компоненты, корреляция > порога = 0.44**



## Этапы алгоритма

### 4. Добавление круга

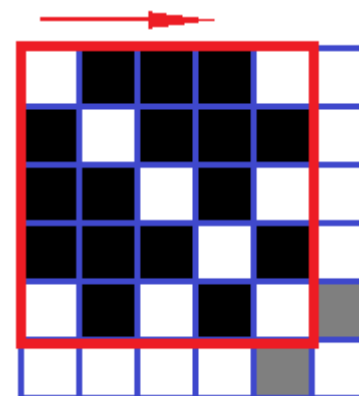
Принадлежность круга  $>$  порога = 0.04

Идём по убыванию рейтинга, пока  
связь с предыдущими вершинами  $>$  порог = 0.15

```
x(x<q) = -Inf;
[my, c] = max(x);
if isinf(my)
    c = [];
    return;
end;

while true
    y = alpha*sum(M(:,c),2) + x;
    y(c) = -Inf;
    [my,j] = max(y);
    if (isinf(my))
        break;
    end;
    if mean(M(c,j))<p
        break;
    end;
    c = [c, j];
end;

c = fI(c);
```



## Этапы алгоритма

**Рейтинг = лк числа связей с предыдущими вершинами + SVD-коэффициенты**

### 5. Окончательное добавление

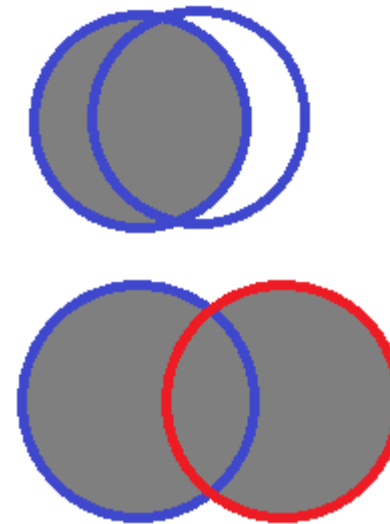
**Если большое пересечение с уже существующим – не добавлять**

```
function [anss, isadd] = addcircle2ans(anss, circle, p)

if isempty(circle)
    isadd = false;
    return;
end;

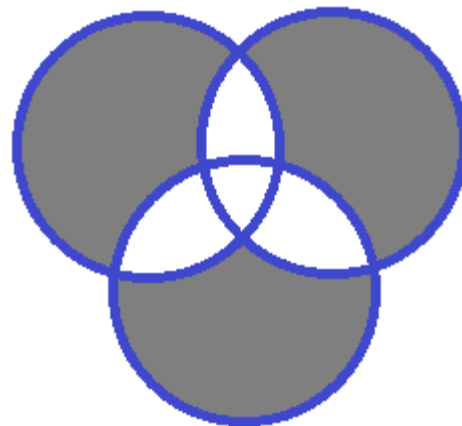
for j=1:length(anss)
    a = anss{j};
    p_jac = length(intersect(a,circle))/length(union(a,circle));
    if p_jac > p
        isadd = false;
        return;
    end
end

anss{end+1} = circle;
isadd = true;
```



## Этапы алгоритма

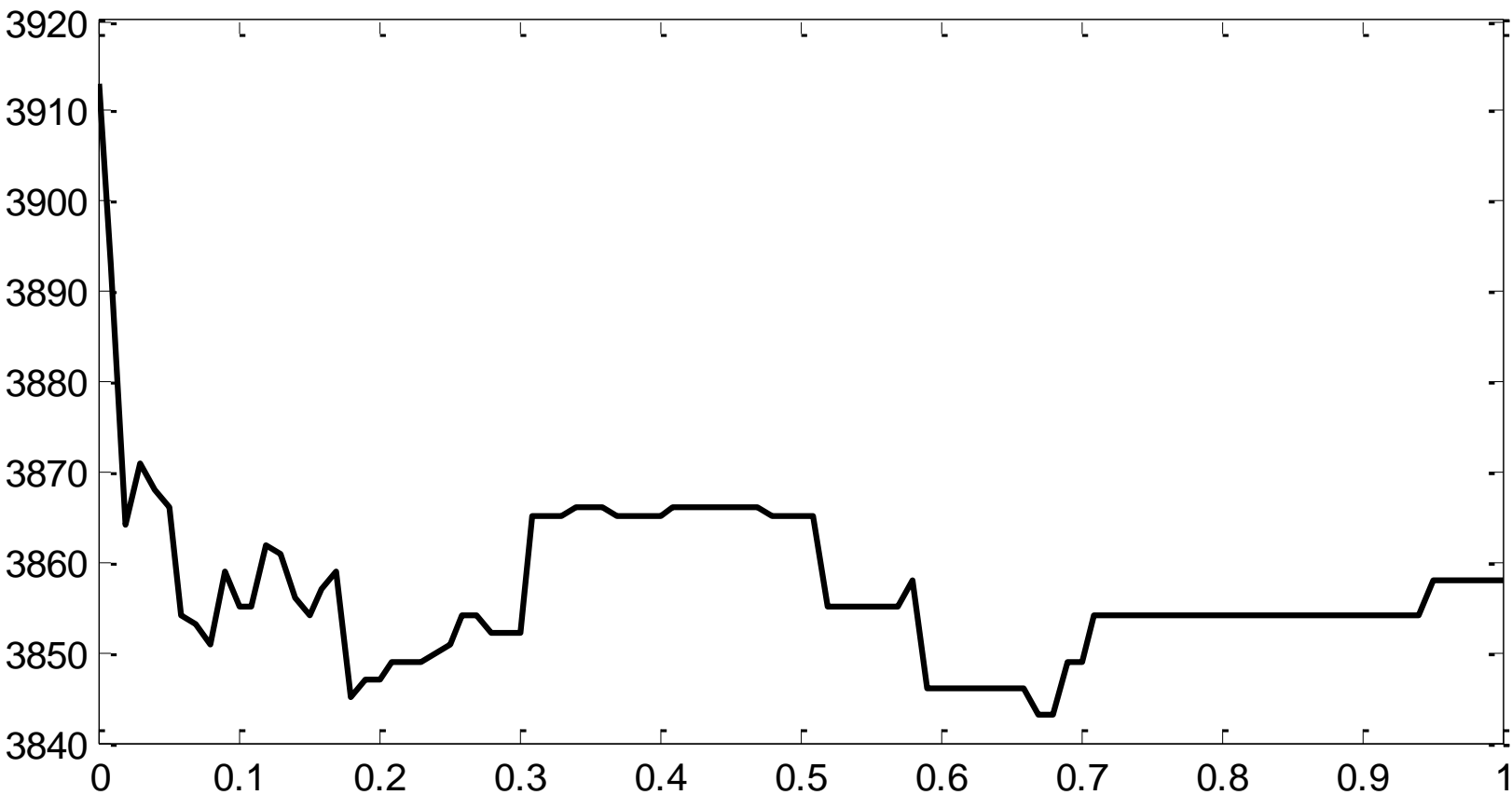
### 6. Удаление пересечений



**следует из функционала качества**

1) Настройка параметров

Типичная картинка

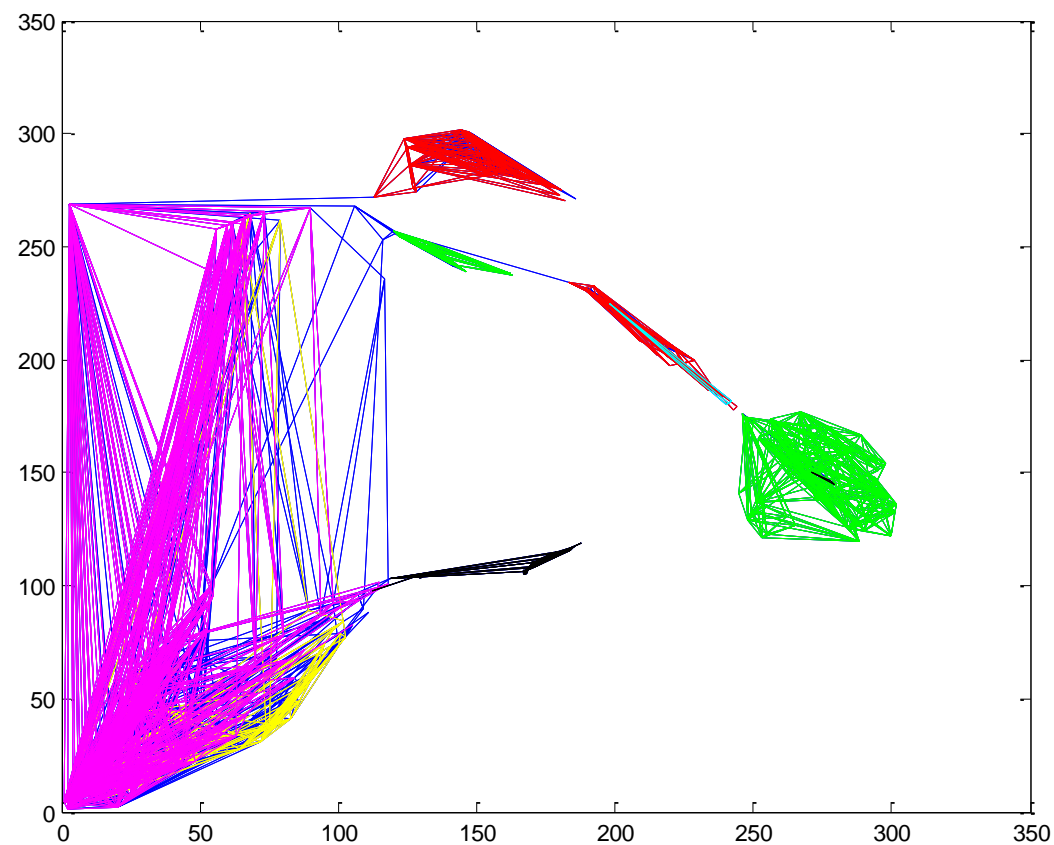


порог в добавлении кругов.

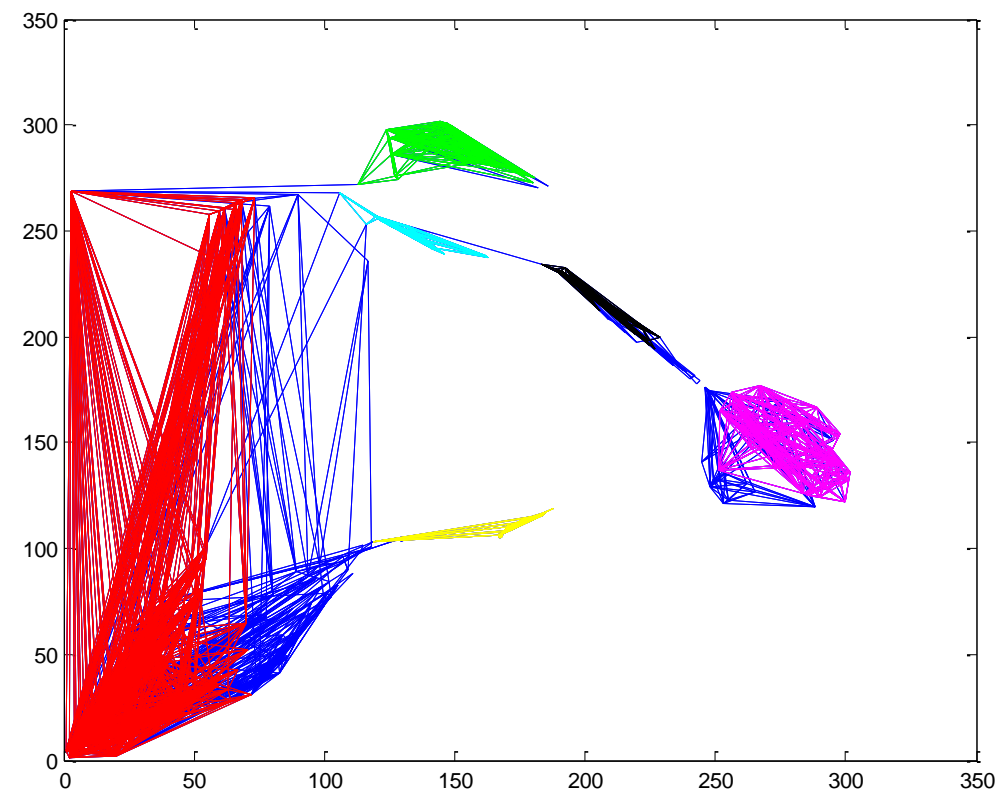
Уже по картинке видно: **Мало статистики!!!**

## Работа алгоритма

## Визуализация по 1й и 2й SVD-компоненте



**правильный ответ**

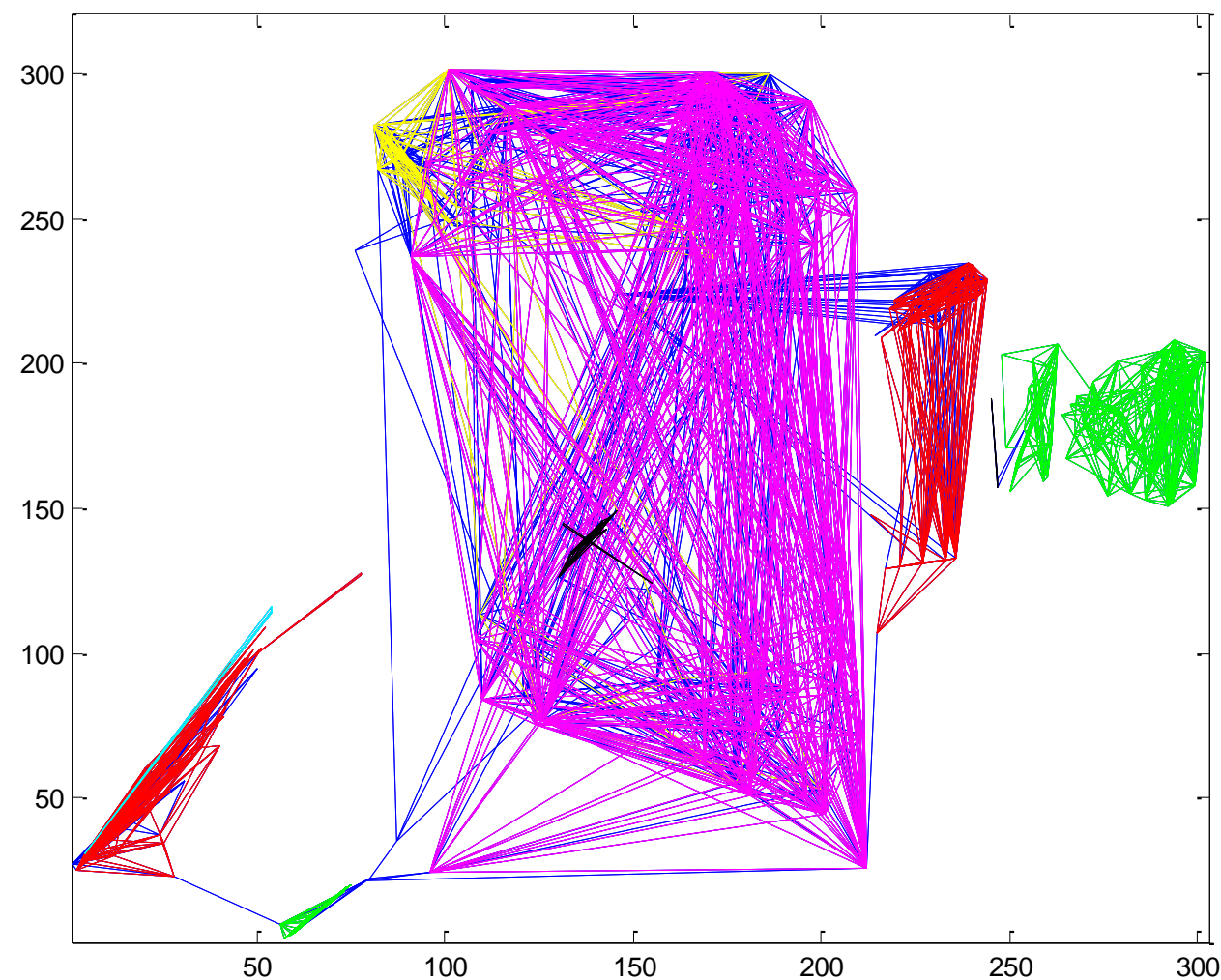


**ответ алгоритма**

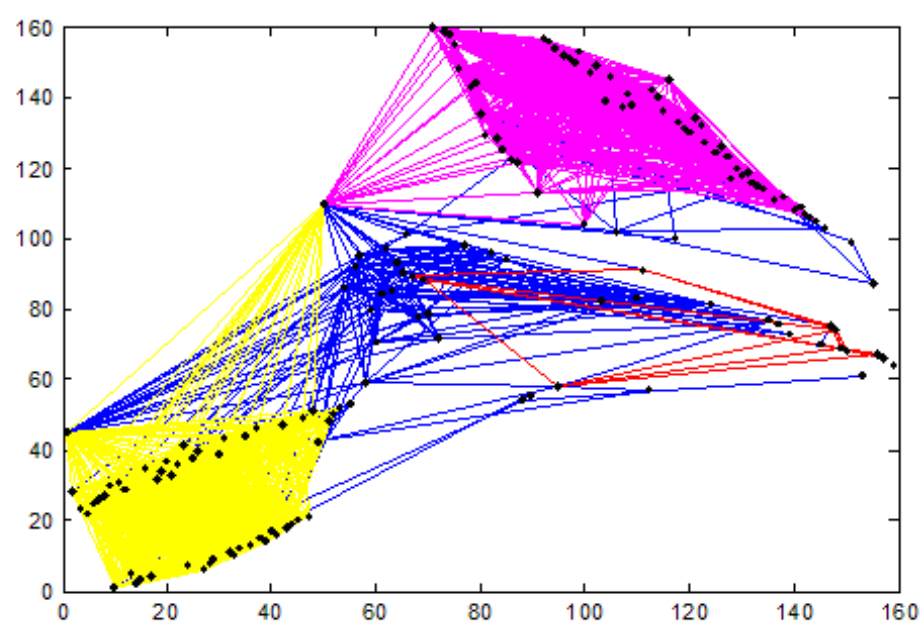
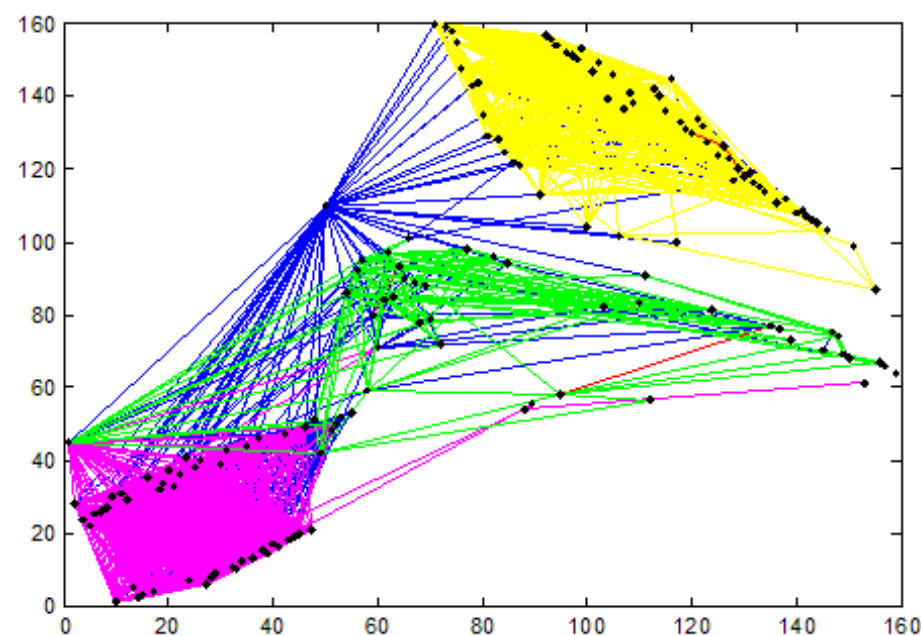
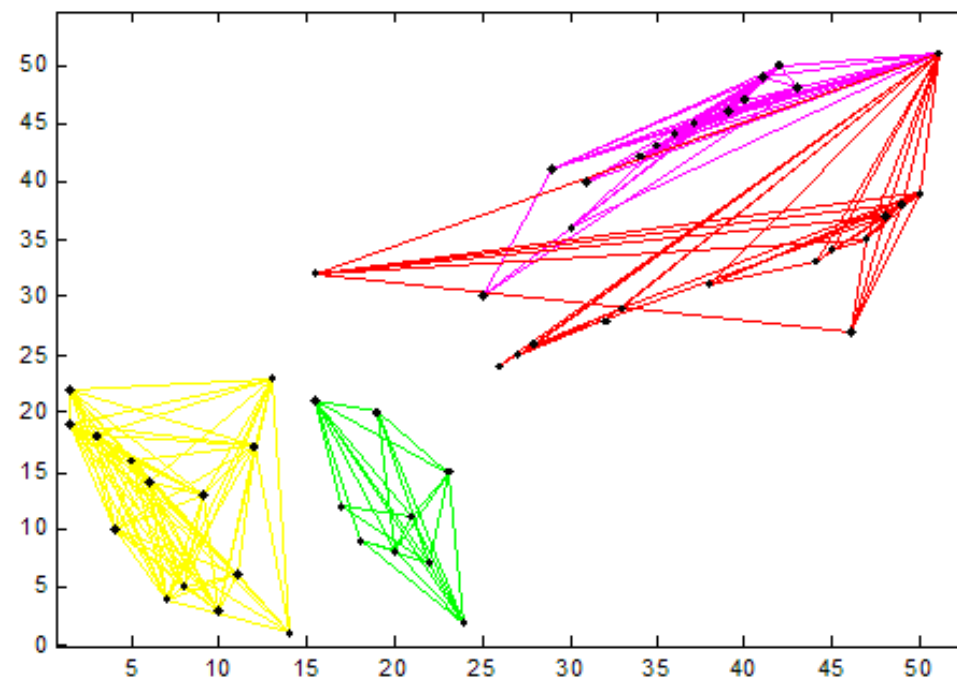
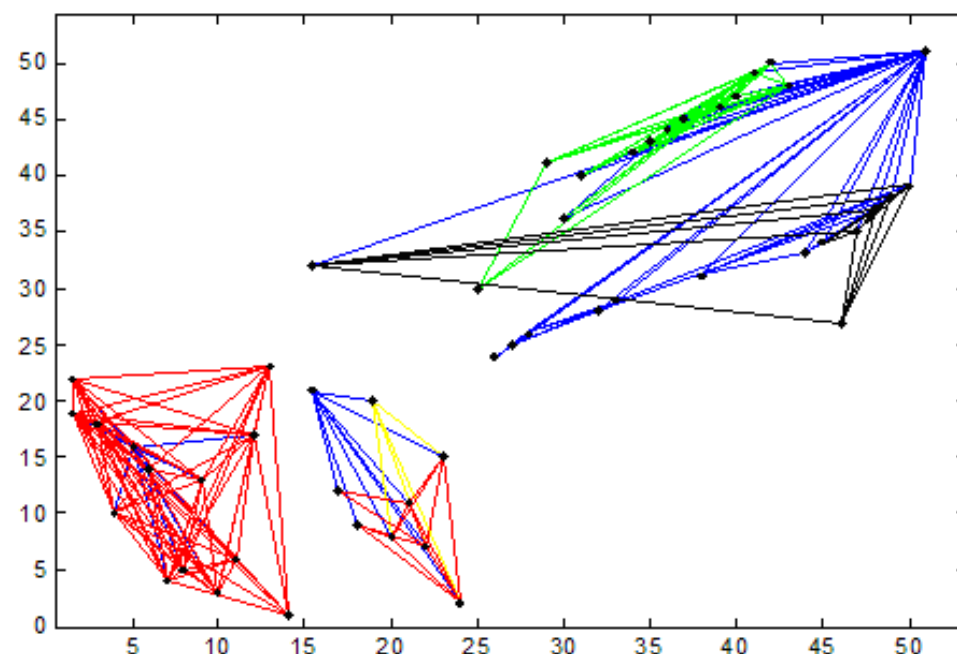
**Хитрость:** координаты – не значения компонент, а **tiedrank...**

## Работа алгоритма

### Визуализация по 3й и 4й SVD-компоненте



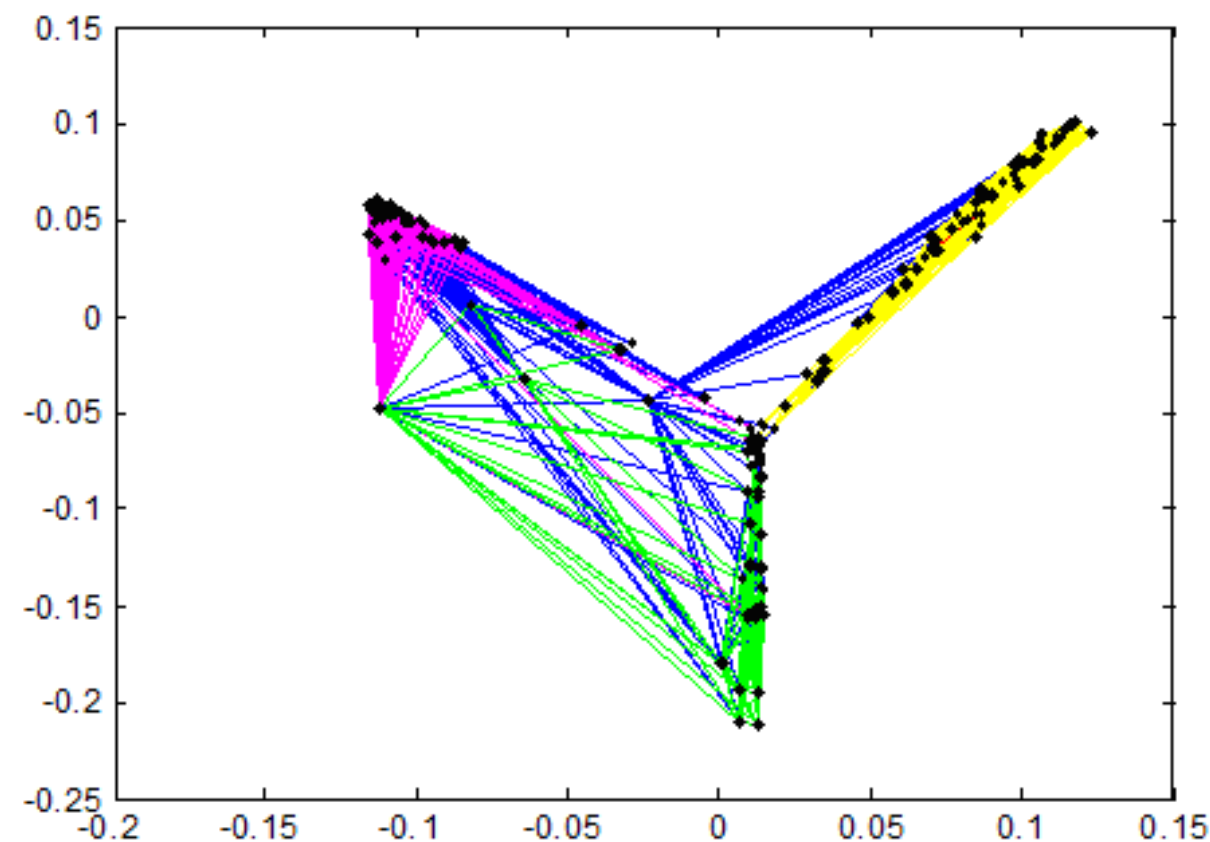
## Работа алгоритма





## MDS

**Можно проецировать граф на плоскость с сохранением расстояний**



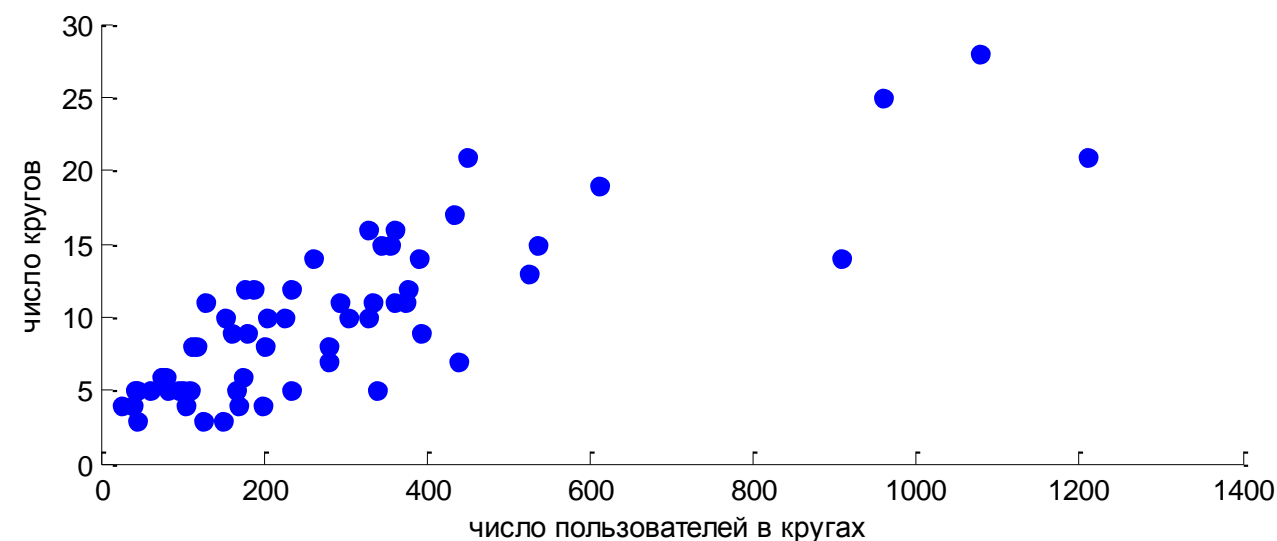
**Но получается не очень информативно**

## Что можно было сделать ещё...

**1) кластеризация в пространстве первых компонент SVD**  
(испугался трудоёмкости и неочевидности)

**2) грамотное выделение кластеров**  
(шёл от самой рейтинговой вершины – на модельных примерах может быть провальной стратегией)

**3) можно было попробовать восстанавливать число кругов...**

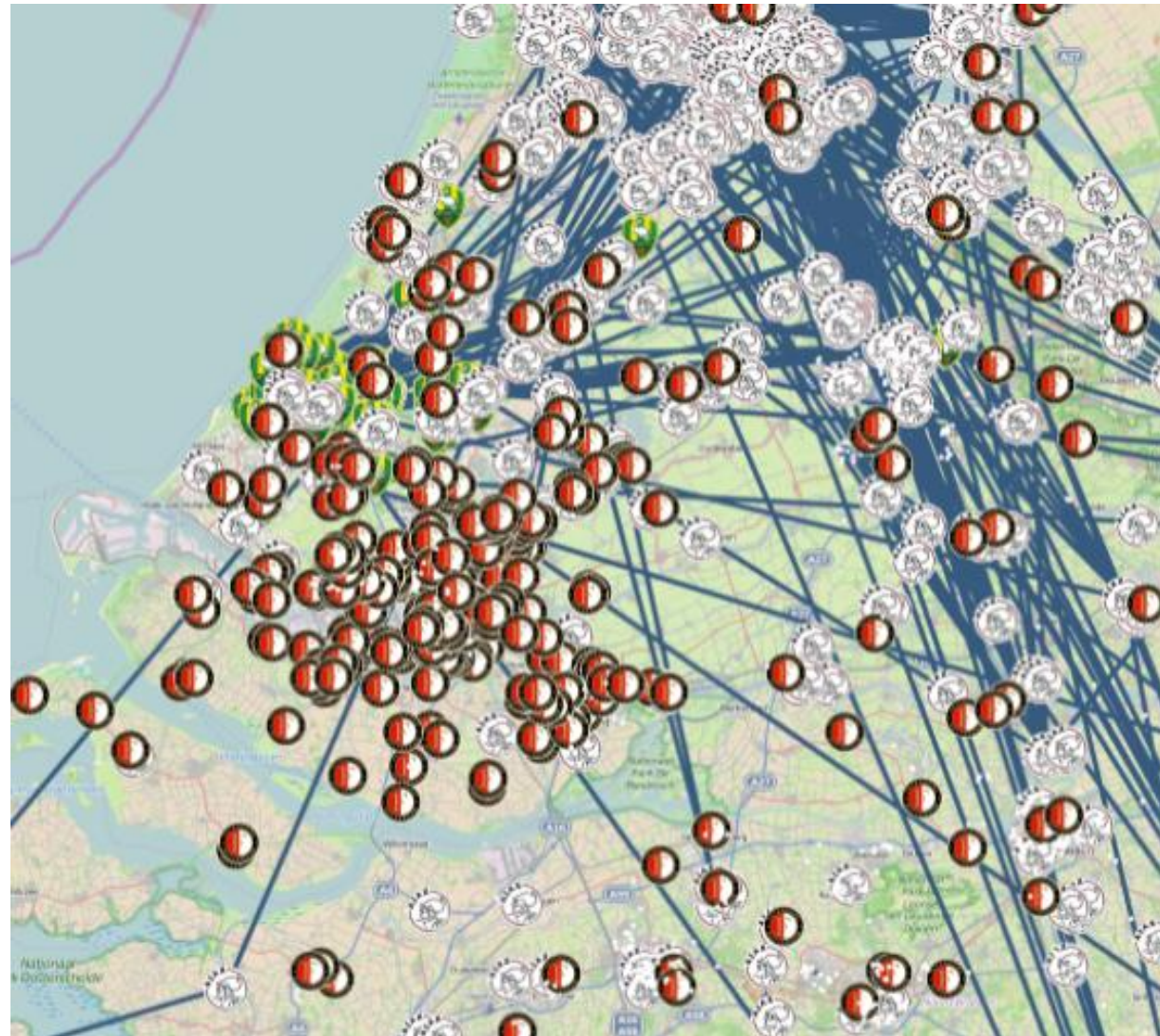


но, как правило, это не работает!

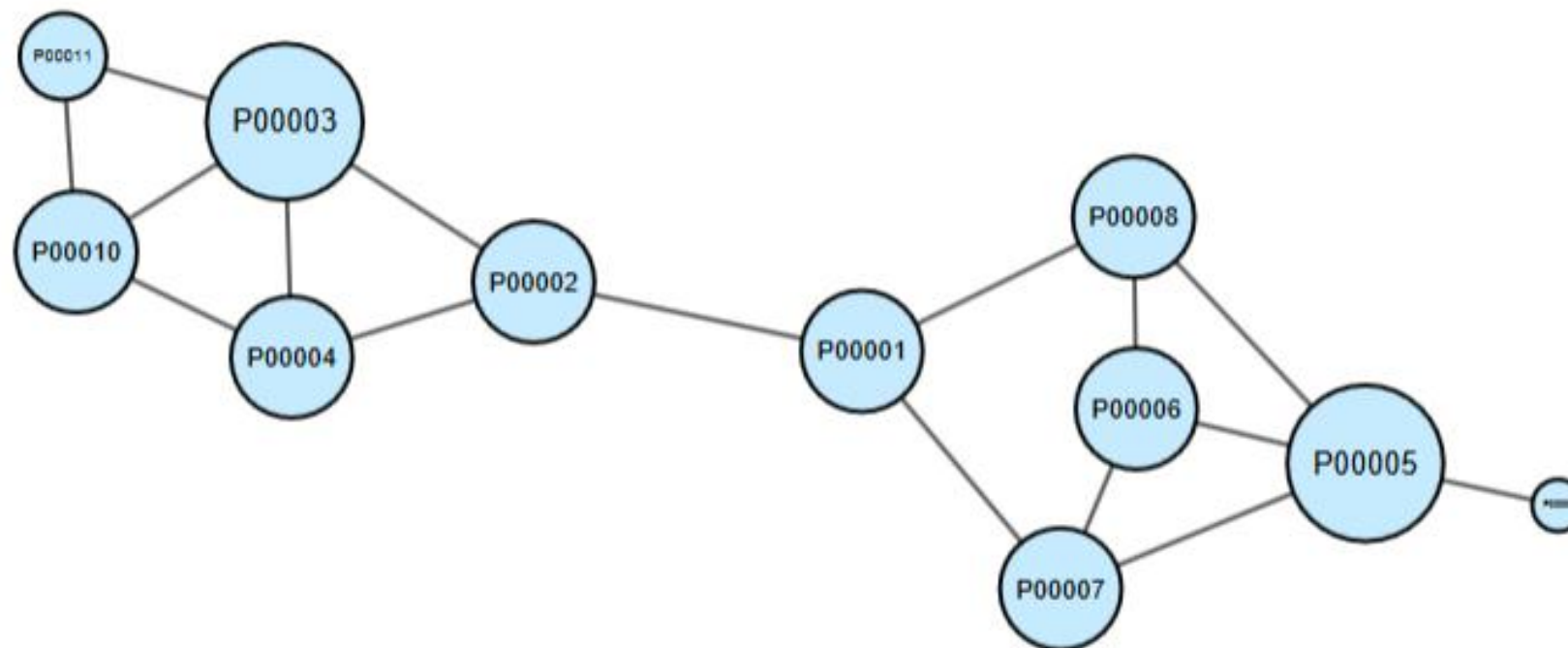
**4) объединение ответов кластеризаторов**  
(собственно, уже делал через SVD – хорошая тема)

## Case: анализ фанатских сообществ

### Проект Dutch National Police

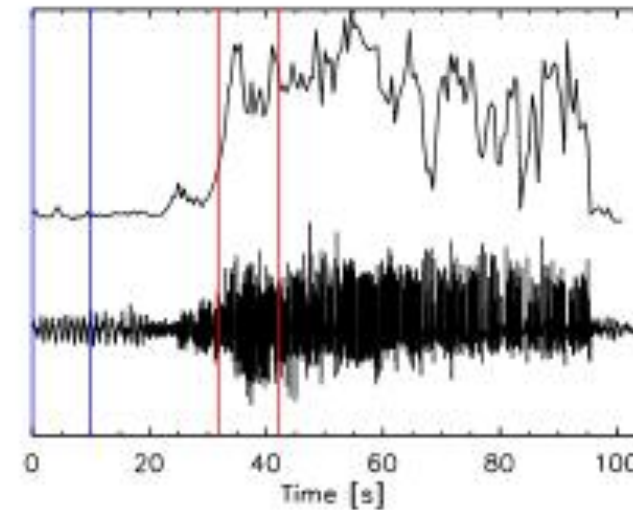
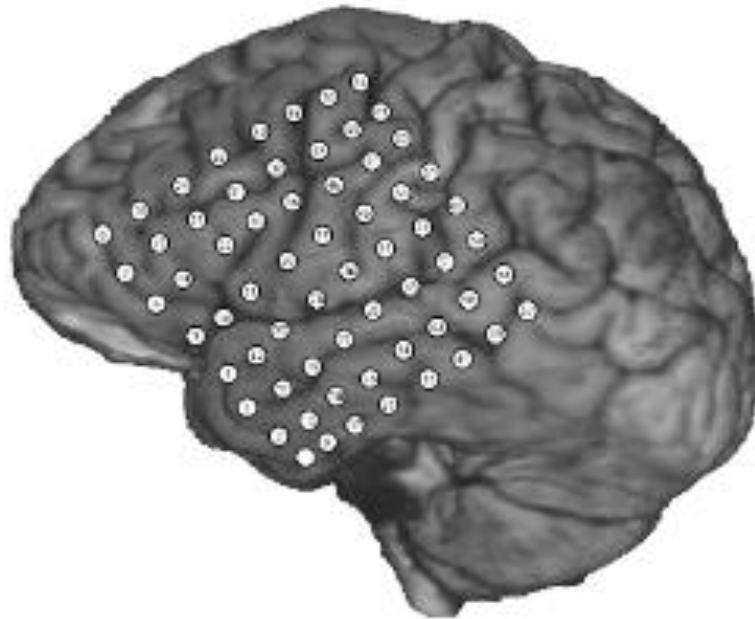


**Статистика преступлений футбольных фанатов**

**Case: анализ фанатских сообществ**

**Строим граф фанатов  
находим самые важные вершины**

<http://liacs.leidenuniv.nl/~takesfw/SNACS/lecture3.pdf>

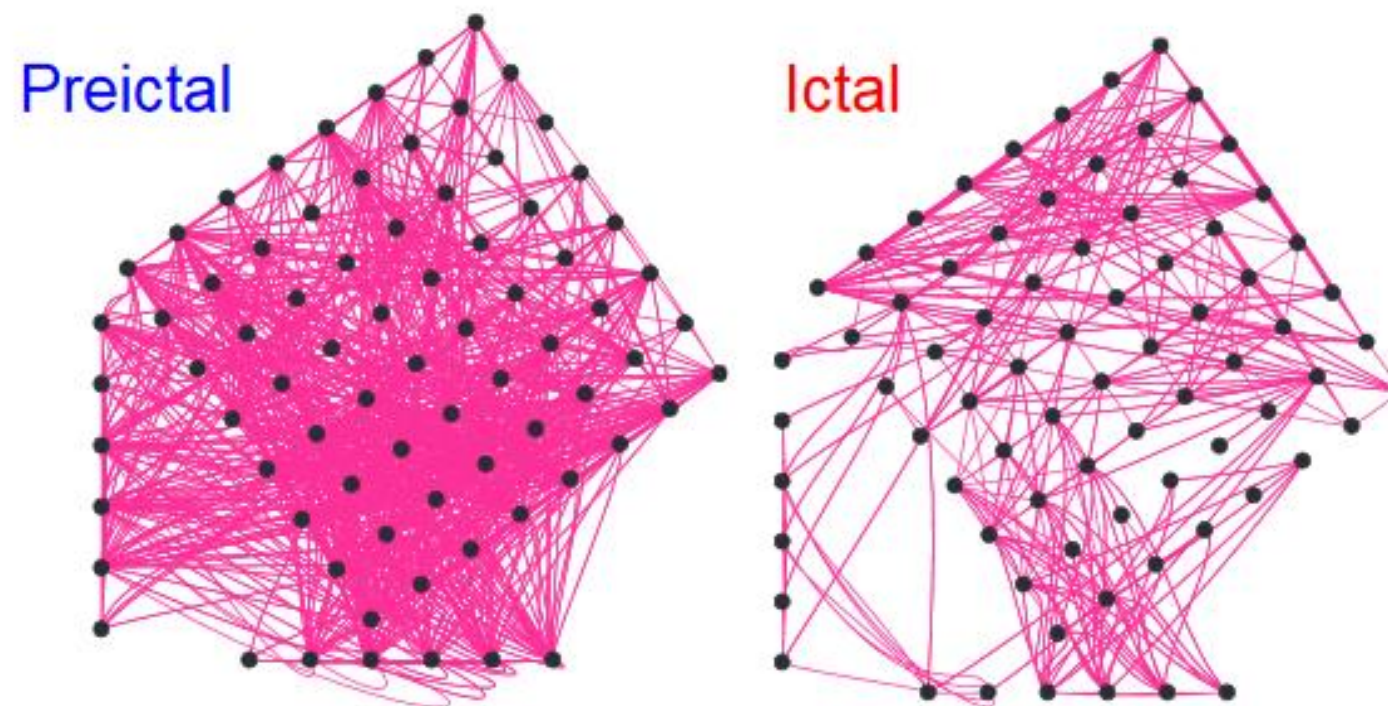
**Case: детектирование эпилепсии****Приступы ~ ненормальная нейронная активность****Electrocorticogram (ECoG)**

**M. A. Kramer et al, «Emergent network topology at seizure onset in humans» // Epilepsy Res., vol. 79, pp. 173-186, 2008**



## Case: детектирование эпилепсии

**Два 10-сек периода: до эпилепсии, после начала**  
**Граф: корреляция сигналов > порога**



**Хорошо различаются графы в признаковом пространстве**  
**(степень, центральность, коэф. кластеризации и т.п.)**

## Итог

**выделение сообществ – задача похожа на кластеризацию,  
но есть специфика графов  
Есть возможность проверять решение...**

**Опять много методов...**



## Что полезно: программирование

**igraph – The network analysis package**

<http://igraph.org/>

**NetworkX: Python software for network analysis (v1.5)**

<http://networkx.lanl.gov>

**Gephi: Java interactive visualization platform and toolkit**

<http://gephi.org>

## Что полезно: курсы

### Классная курс-книга

<http://networksciencebook.com/>

### Очень хороший

#### **Hadi Amiri «Social Media Computing - CMSC 498J»**

<http://legacydirs.umiacs.umd.edu/~hadi/cmssc498j/syllabus.html>

### Очень хороший

#### **Gonzalo Mateos «Network Science Analytics»**

<http://www2.ece.rochester.edu/~gmateosb/ECE442.html>

#### **Л.Жуков «Structural Analysis and Visualization of Networks» в ВШЭ**

<http://leonidzhukov.net/hse/2015/socialnetworks/>

### Неплохой курс

#### **Frank Takes «Social Network Analysis for Computer Scientists»**

<http://liacs.leidenuniv.nl/~takesfw/SNACS/>

## Что полезно: книги

**David Easley, Jon Kleinberg «Networks, Crowds, and Markets:  
Reasoning About a Highly Connected World»**

**<https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf>**



**Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman «Mining of  
Massive Datasets»**

**<http://infolab.stanford.edu/~ullman/mmds/book.pdf>**



**Eric D. Kolaczyk «Statistical Analysis of Network Data: Methods and Models»**  
**M. E. J. Newman «Networks: An Introduction» Oxford U. Press**

**ДЗ****Исследовать свою социальную сеть**

**Цель-максимум: изучить все-все-все понятия, которые успели пройти**

- Распределение степеней
- Является ли «малым миром»
- Коэффициенты кластеризации
- Разреженность, сильные/слабые связи
  - Разбиение на сообщества
  - Найти центральные вершины
- + ) попытка поставить и решить задачу появления рёбер