

Прикладные задачи анализа данных

АНАЛИЗ СОЦИАЛЬНЫХ СЕТЕЙ

Дьяконов А.Г.

**Московский государственный университет
имени М.В. Ломоносова (Москва, Россия)**

Исследование социальных сетей

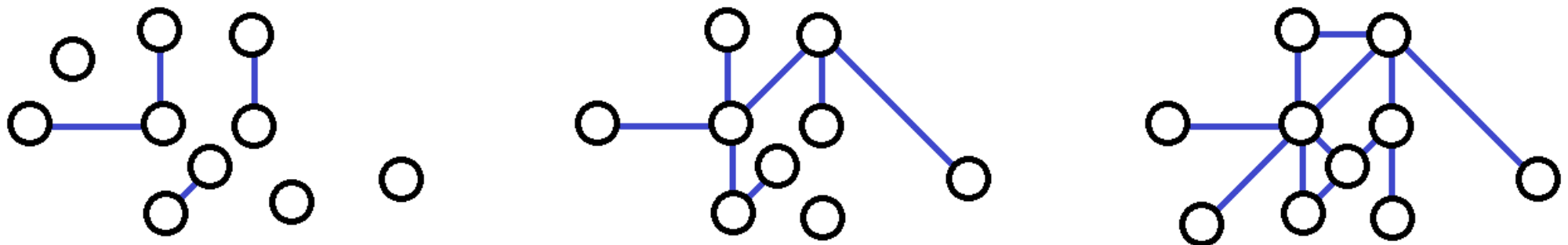


Социальная сеть – динамический граф (пример: мобильная сеть)

Вершины – пользователи (и группы)

Рёбра – дружба (членство) / связи, отношения

Кластеры – сообщества



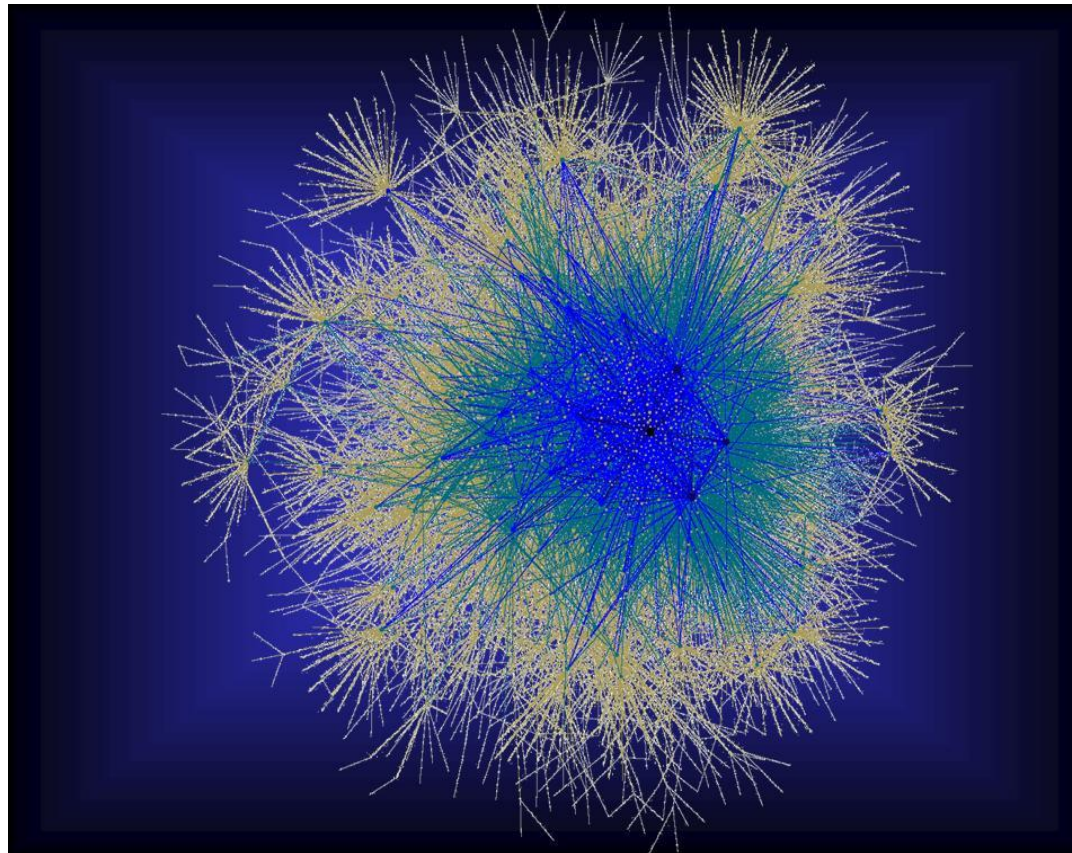
Примеры соцсетей:

- «классические» (Facebook, vk, Одноклассники)
- мобильные сети
- научные сообщества (связь по публикациям)
- почтовые (связь по отправке писем)
- интернет-магазин (связь по одинаковым купленным товарам)
- даже сам интернет

Какие здесь графы?

Какие задачи здесь актуальны (возможны)?

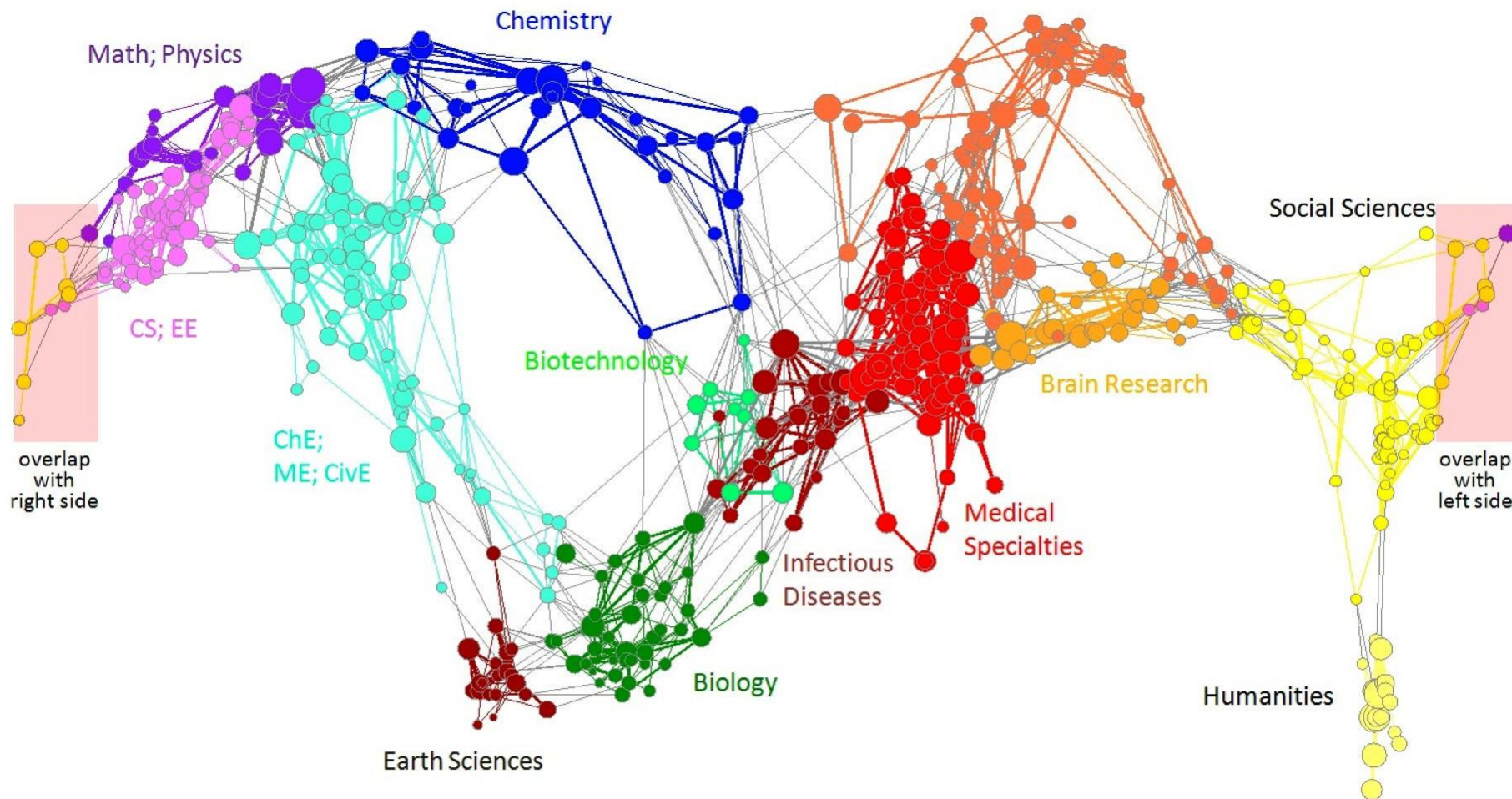
Картинки с графами



graph of the BGP (Gateway Protocol) web graph, consisting of major Internet routers (6400 вершин, 13000 рёбер)
Ross Richardson, Fan Chung Graham

Примеры графов

Граф цитирований



Börner и др.

Задачи с социальными сетями

- **Анализ поведения пользователей**

- выявление аккаунтов-дубликатов
- пользователей нарушающих, склонных нарушать правила, не похожих на других

- **Прогнозирование**

- поведения пользователей (когда будет пользоваться услугами, в какую группу вступит, с кем подружится)
- предсказание и предотвращение ухода пользователей
- предсказание трафика (в каком объёме будет скачивать/закачивать)

- **Рекомендация**

- предсказание эффективности действия рекламы для конкретного пользователя
- формирование таргетированных предложений (рекламы, по вступлению в группы, заполнению профиля и т.п.)

Задачи с социальными сетями

• Кластеризация

- разбиение пользователей на группы (для более корректного А/В-тестирования, разработки стратегий под группы, более тщательного анализа аудитории)
- выявление «кругов общения пользователей» (друзей, которых объединяет некоторая сущность, например «друзья по вузу»)
- выделение сообществ
- выделение базисов источников информации в блогосфере

• Взаимодействие с другими соцсетями/ресурсами

- матчинг сетей/графов (установление соответствия между пользователями одной сети и другой)
- использование данных соцсети для решения задач других заказчиков
 - скоринг (оценка заёмщика) - в банках
 - персональные рекомендации - в интернет-магазинах
 - таргетированная реклама - в рекламе, СМИ (таргетированные новости)

Задачи с социальными сетями

- **Анализ текстов**

- **обнаружение недопустимых текстов (оскорблений, рекламы, нарушения закона и т.п.)**
- **анализ общественного мнения по постам**
- **анализ лояльности к брендам по постам**

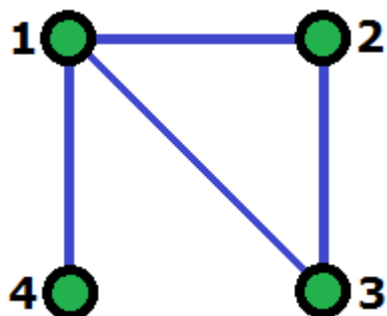
- **Визуализация**

- **поиск закономерностей в данных соцсети и их представление**
- **анализ общественного мнения по постам**
- **научные исследования графов соцсетей**

ДЗ: придумать свои задачи!

Основные понятия теории графов

Граф



матрица сопряжённости

	1	2	3	4
1		1	1	1
2	1		1	
3	1	1		
4	1			

как правило разреженная

диагональная матрица степеней

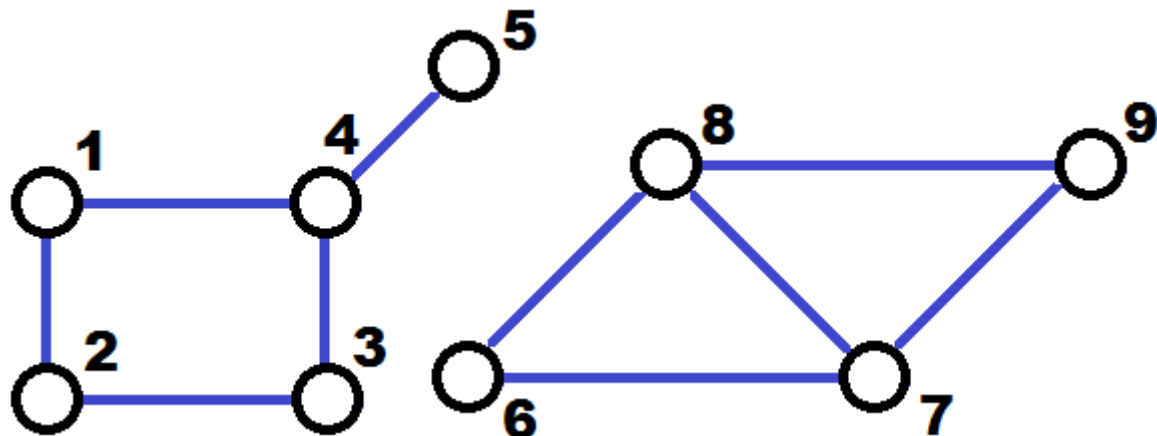
	1	2	3	4
1	2			
2		2		
3			2	
4				1

матрица Лапласа

	1	2	3	4
1	2	-1	-1	-1
2	-1	2	-1	
3	-1	-1	2	
4	-1			1

Основные понятия теории графов

Неориентированные

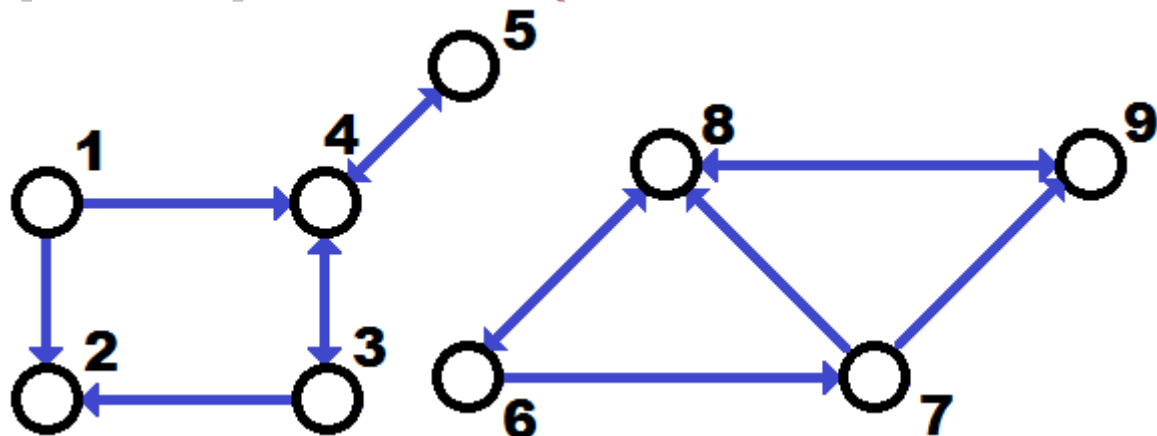


**соседство
окрестности**

степень

**входящая/исходящая
степень**

Ориентированные (где они возникают?)



связные компоненты

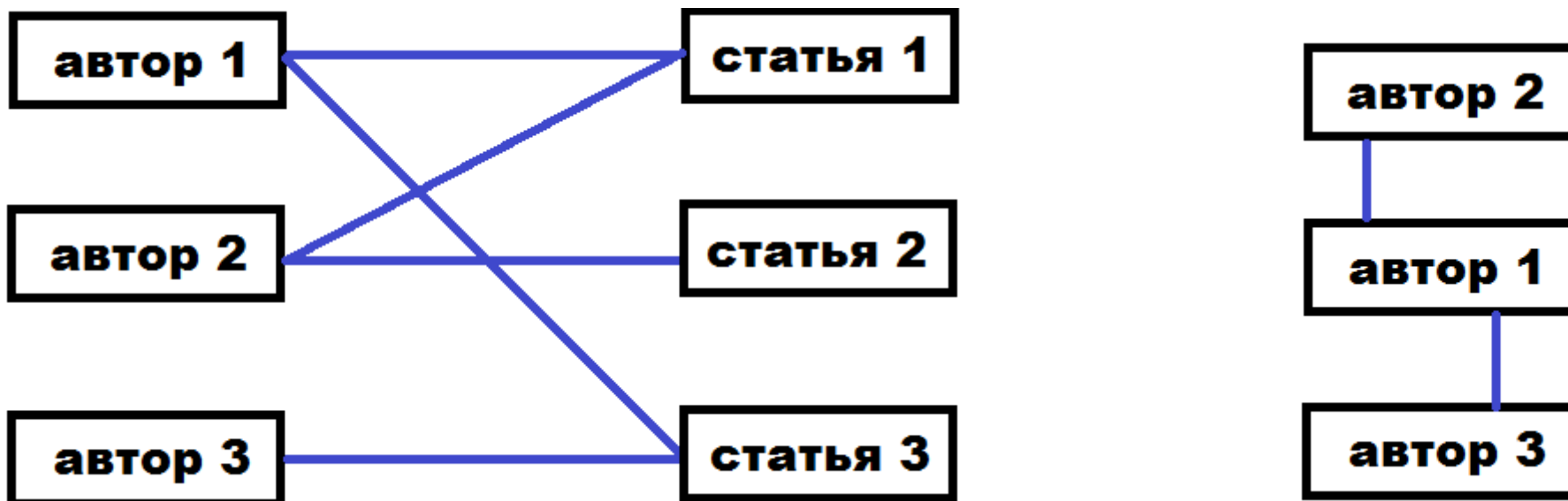
клика

максимальная клика

+ взвешенные графы

Основные понятия теории графов

Двудольные графы



Научные сообщества

Граф цитирования (ориентированный)

Граф соавторства (неориентированный/двудольный)

Граф сходства статей (с весами)

Понятие сложной сети

1. **Специальные распределения (степеней вершин)**
2. **Модель «малого мира» (малый диаметр и т.п.)**
3. **Высокий коэффициент кластеризации**
4. **Разреженность**

Complex network

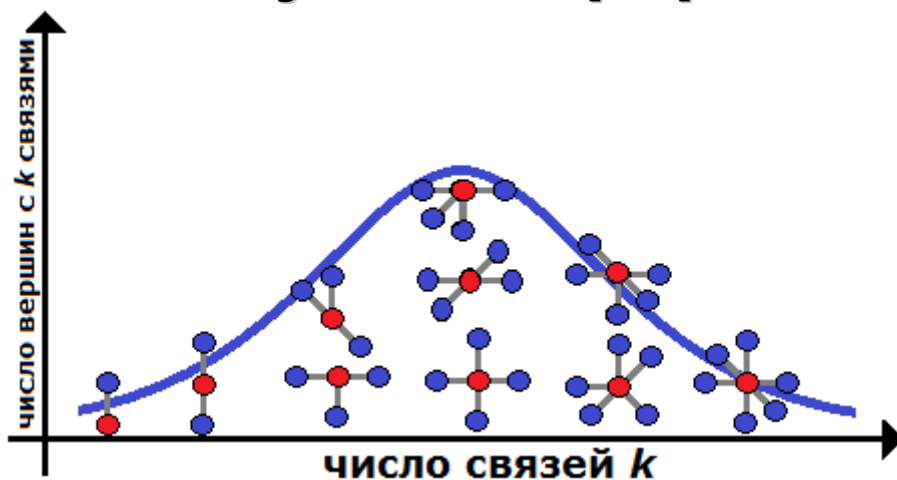
1. **Power law degree distribution**
2. **Small diameter and average path length: "small world"**
3. **High clustering coefficient**
4. **Sparcity**

1. Распределение вершин

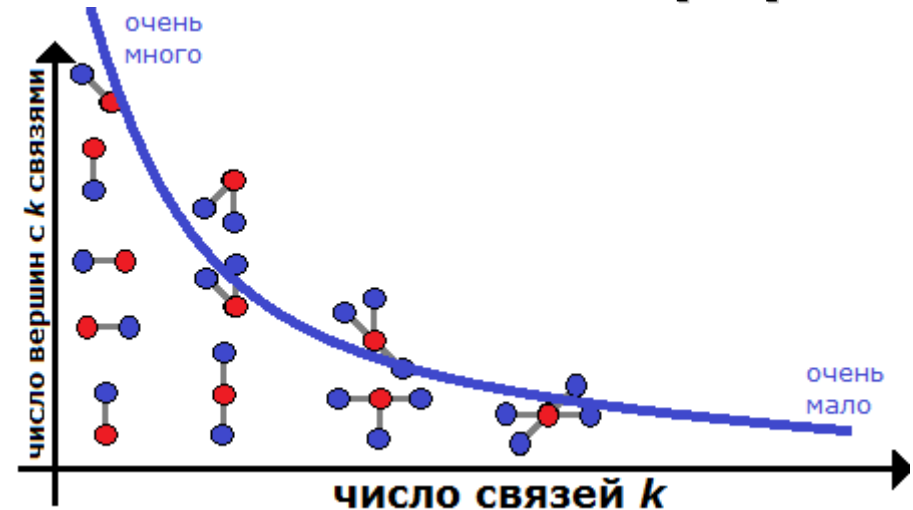
Безмасштабные (scale-free) сети – сети, в которых степени вершин распределены по **степенному закону**:

доля вершин с k связями $\sim k^{-\gamma}$, обычно $2 < \gamma < 3$.

Случайный граф

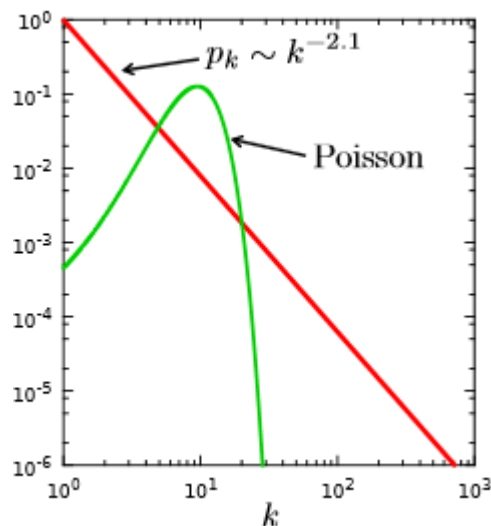


Безмасштабный граф



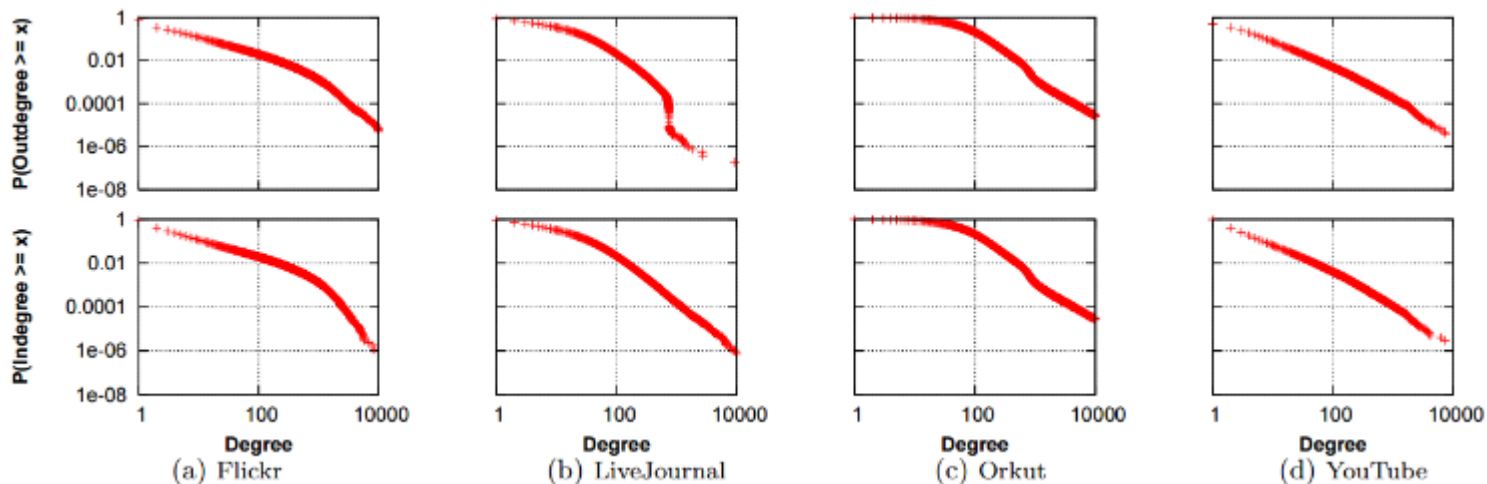
M.E.J. Newman Power laws, Pareto distributions and Zipf's law // Contemporary Physics, pages 323–351, 2005

1. Распределение вершин



<http://barabasi.com/networksciencebook/>

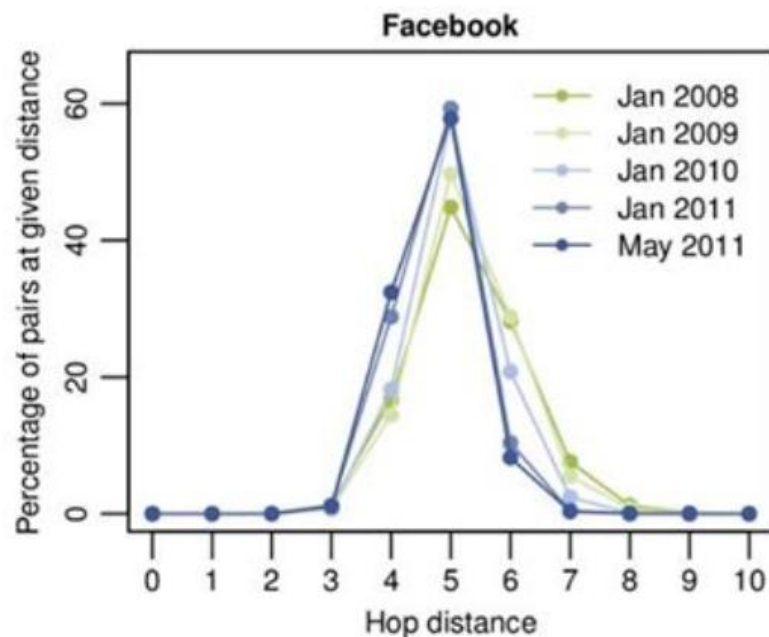
В логарифмической шкале



ИСТОЧНИК?

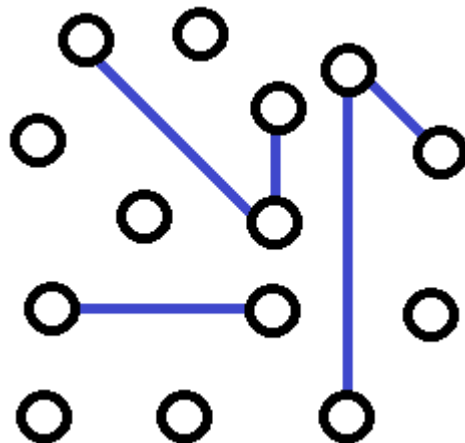
2. Модель малого мира

Граф	Среднее расстояние между вершинами
Граф почтовых рассылок (D. Watts, 2001, 48000 вершин)	6
Граф сообщений в MSN Messenger (J. Lescoves и др. 2007, 240 млн. вершин)	6.6
Граф Фейсбука (L. Backstrom и др. 2012, 720 млн. вершин)	4.74

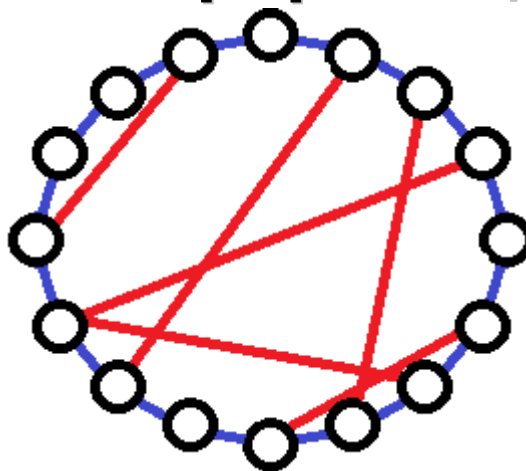


2. Модель малого мира

Генерация случайных графов: модель Эрдёша–Реньи



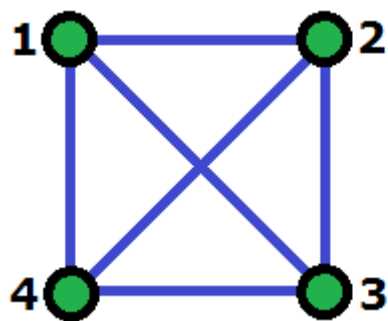
Генерация случайных графов: модель малого мира



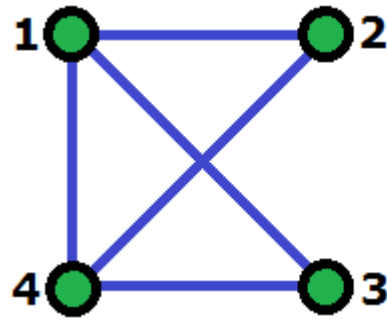
3. Коэффициент кластеризации (полноты)

1. Глобальный

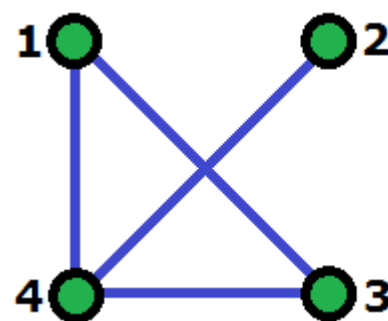
1.1. число треугольников / возможное число (число линий из трёх точек)



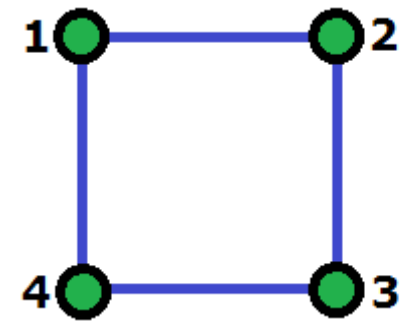
4/4



2/4



1/4



0/4

1.2. Среднее локальных коэффициентов

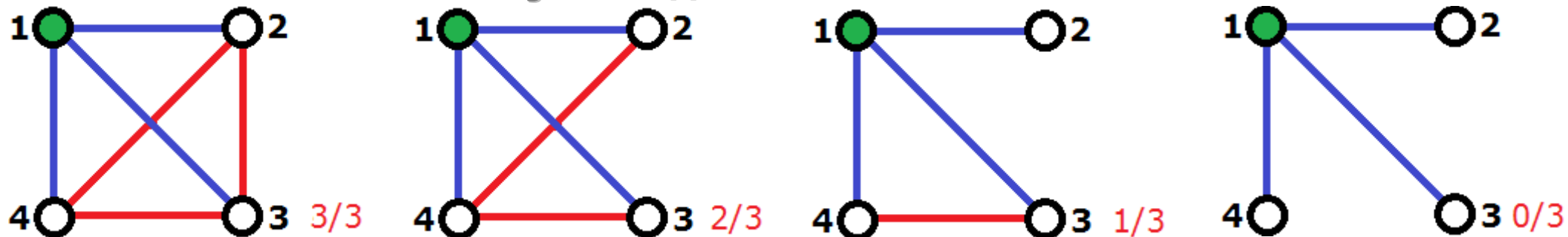
Внимание! Это признак;

3. Коэффициент кластеризации (**clustering coefficient**)

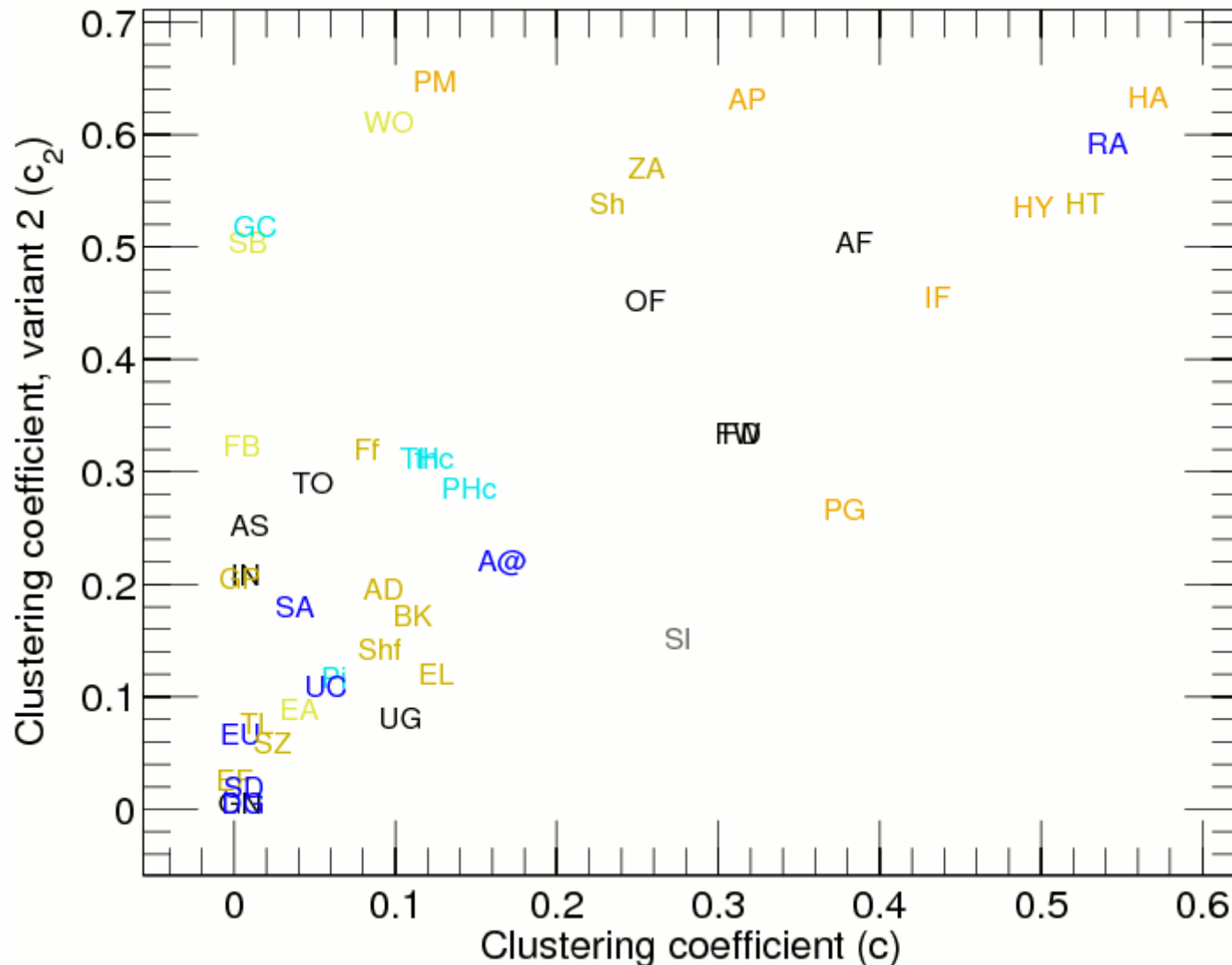
2. Локальный

для вершины = насколько её соседи близки к образованию клики

число связей у соседей / число возможных связей



3. Коэффициент кластеризации



Два способа определения коэффициента кластеризации

<https://networkscience.wordpress.com/>

4. Разреженность

Большинство реальных графов – разреженные (sparse).

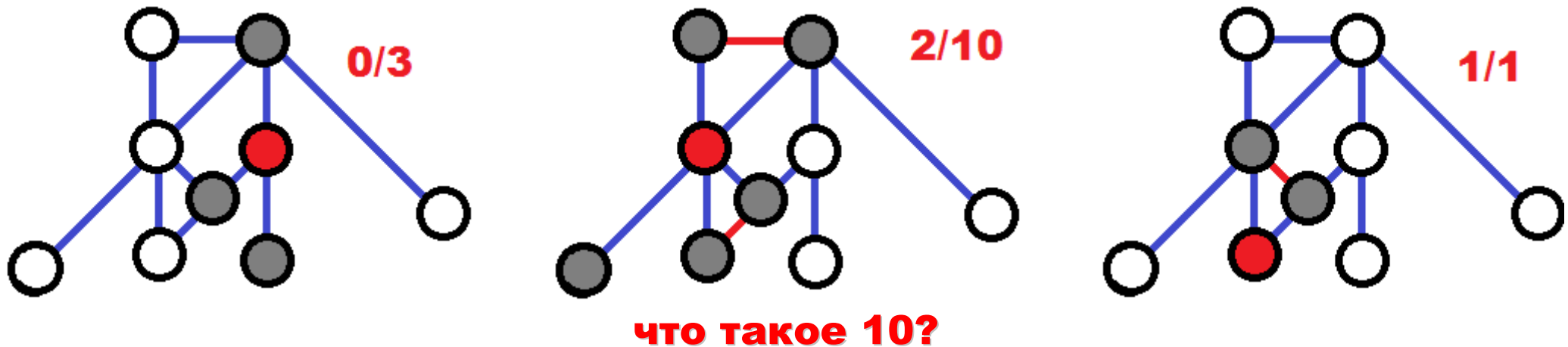
Данные	Число вершин	Средняя степень
WWW (Stanford-Berkeley)	319,717	9.65
Social networks (LinkedIn)	6,946,668	8.87
Communication (MSN IM)	242,720,596	11.1
Coauthorships (DBLP)	317,080	6.62
Internet (AS-Skitter)	1,719,037	14.91
Roads (California)	1,957,027	2.82
Proteins (S. Cerevisiae)	1,870	2.39

из Leskovec et al., Internet Mathematics, 2009

Часто графы просто погружают в признаковое пространство... и граф превращается в вектор

Пример признака (уже был)

Коэффициент полноты (clustering coefficient)



характеризует полноту его-графа одной вершины
(~ окрестность первого порядка)

Как интерпретировать?

В чём недостаток?

Как исправить?

Недостатки

**лучше использовать в сочетании с другими признаками
(например, число соседей)**

Это типично для признаков на графе!

**Как придумать признак для всего графа
(а не отдельной вершины)?**

Как придумать признаки для всего графа

Признак графа – функция от признаков вершин (рёбер, ...)

Любая функция!

- **сумма**
 - **среднее**
 - **максимум**
 - **минимум**
 - **медиана**
 - **сумма квадратов**
- и т.п.**

Сходство вершин

Часто надо измерить сходство двух вершин/рёбер/подграфов

Какие бывают похожести?

Что значит, что вершины похожи?

Важность вершин

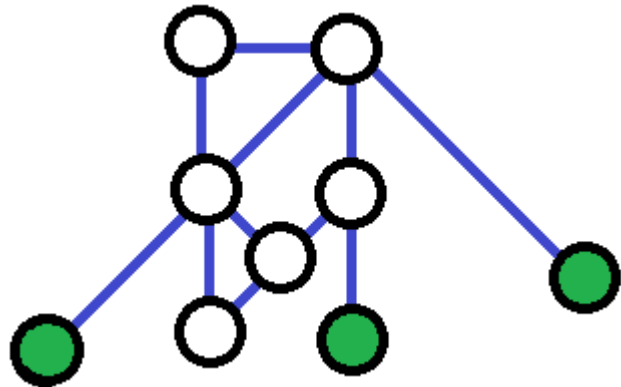
Часто надо измерить особенность вершины/ребра/подграфа

Например, для поиска непохожих вершин, влиятельных блогеров

Какие вершины считать «важными»?

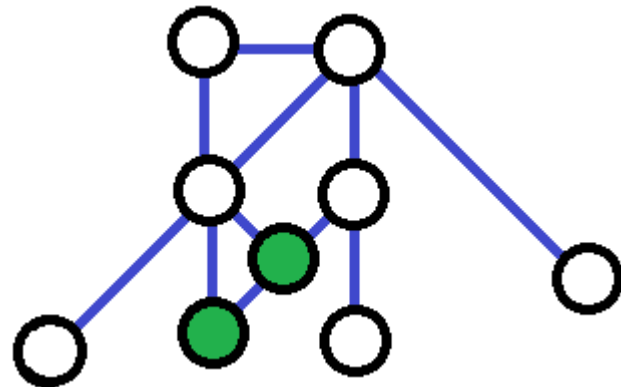
Сходство вершин

1. Формальная (по характеристикам)



По информации о членах
соцсети: в одной группе
института, одни интересы,
участвовали в одном мероприятии

2. По близости

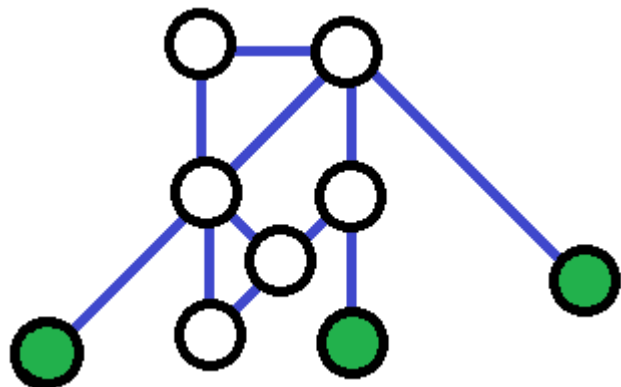


Два близких друга,
близнецы

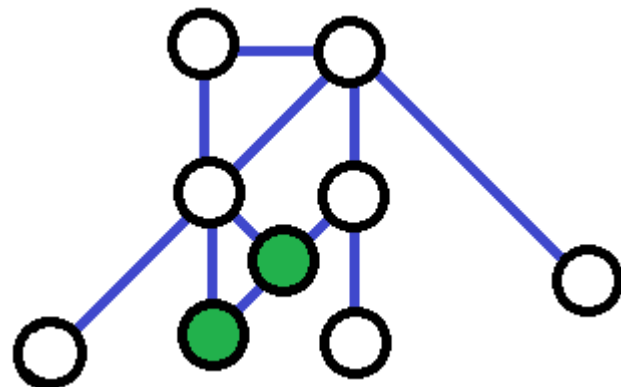
Как определить эти похожести на практике?

Сходство вершин

1. Формальная (по характеристикам)



2. По близости



Как измерить?

Погружение в признаковое пространство

Вычисление сходства в нём

Оценка расстояния на графе

Важность

Какие вершины считать важными?

- По отдельным признакам (например, много соседей)
- По рекурсии (важная вершина соединена с важными)

Пример важности – центральность вершины (сейчас рассмотрим)

Кстати, а что такое граф? С точки зрения реализации

Очень полезно

**Любой объект имеет много представлений
(подпространство, многогранник и т.п.)**

1. С точки зрения определения

2. С точки зрения реализации

Разреженная матрица

Объекты (пользователи) – строки/столбцы

Аппарат линейной алгебры

3. С точки зрения сути

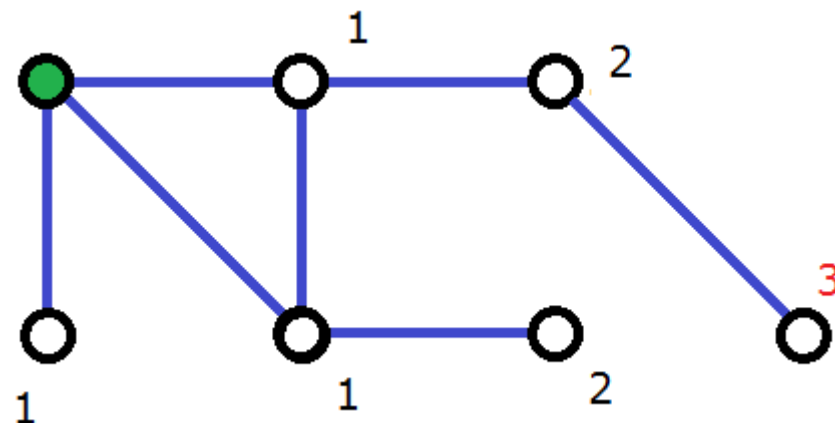
Это формализация отношений

Важны окрестности большого порядка, их свойства, связи,
не всё может быть отражено в графе

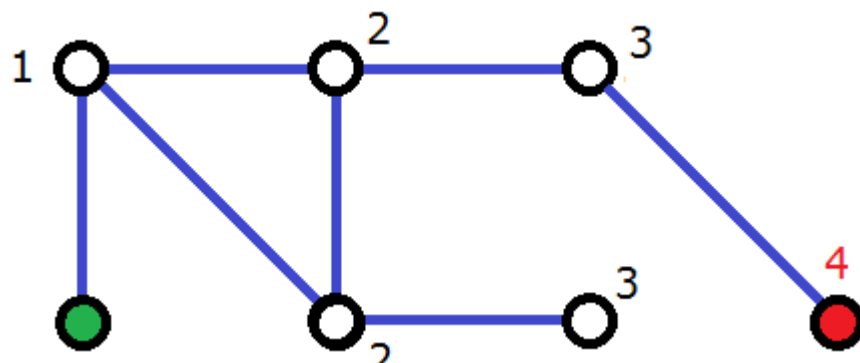
Центральность вершины в графе

Эксцентриситет – вершины v

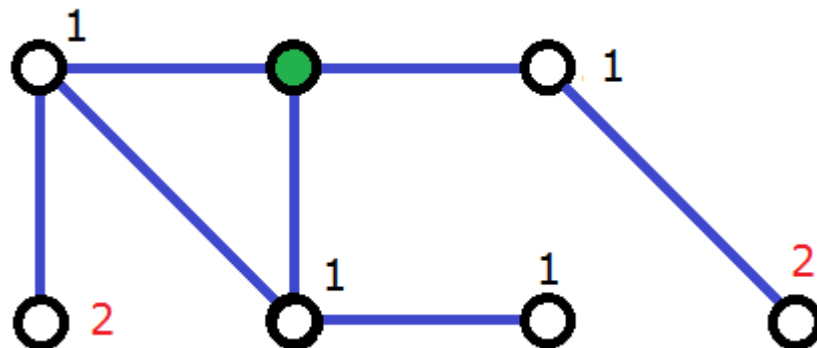
$$\varepsilon(v) = \max_{u \in V} d(u, v)$$



Диаметр – максимальный эксцентриситет



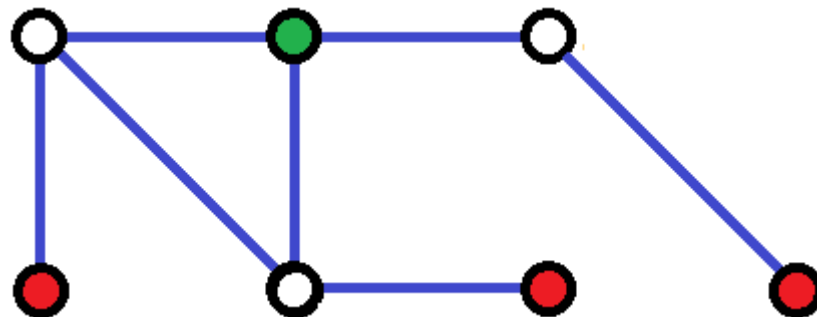
Радиус – минимальный эксцентриситет



**Вершина графа центральная,
если её эксцентриситет равен радиусу графа.**

Центр – множество центральных точек

Периферия – множество точек с максимальным эксцентриситетом



Интересная терминология

Степенная центральность (Degree centrality) – число соседей

Центральность по близости (Closeness centrality) – $\sum_{u \neq v} \frac{1}{d(u, v)}$

Центральность по путям (Betweenness centrality) – число кратчайших путей, проходящих через эту вершину

Собственная центральность (Eigenvector centrality) – центральность вершины зависит от центральности соседей

$$c_i = \sum_j a_{ij} c_j$$

собственный вектор ~ max с.з. A ???

Метод:

- вычисление собственных векторов
- взятие вектора с максимальным собственным значением
- его значения – центральность вершины

Katz

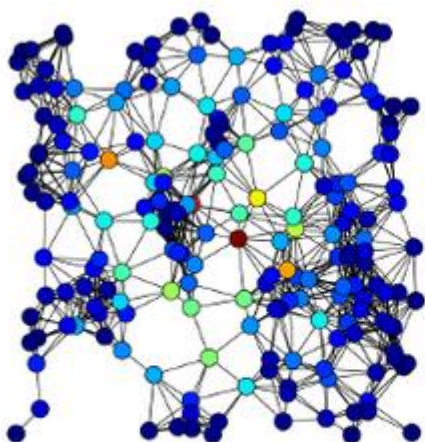
взвешенная сумма путей, приходящих в вершину.

Путь длины k берём с коэффициентом β^k , $\beta \in [0, 1]$

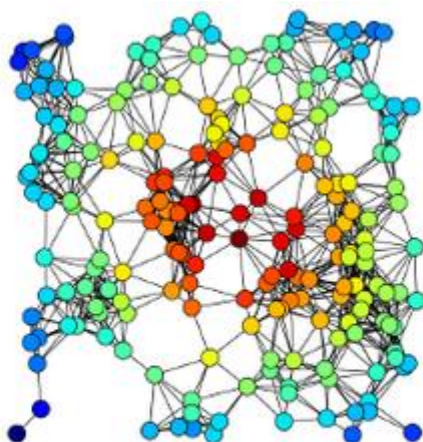
$$\begin{aligned}
 & (\beta A + \beta^2 A^2 + \beta^3 A^3 + \dots) \tilde{\mathbf{1}} = \\
 & (\beta A + \beta^2 A^2 + \beta^3 A^3 + \dots)(I - \beta A)(I - \beta A)^{-1} \tilde{\mathbf{1}} = \\
 & (\beta A + \beta^2 A^2 + \beta^3 A^3 + \dots - \beta^2 A^2 - \beta^3 A^3 - \dots)(I - \beta A)^{-1} \tilde{\mathbf{1}} = \\
 & \beta A(I - \beta A)^{-1} \tilde{\mathbf{1}}
 \end{aligned}$$

На основе этого вычисляется центральность.

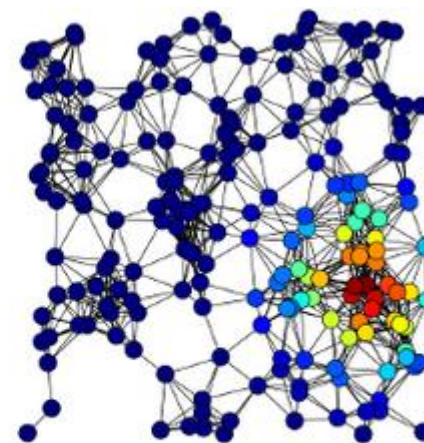
Разные виды центральности



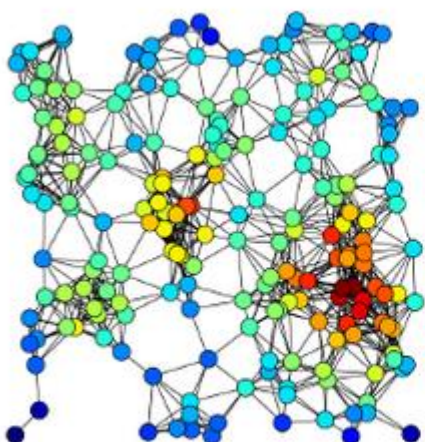
Betweenness centrality



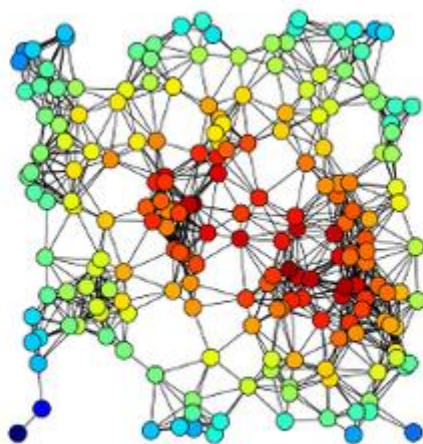
Closeness centrality



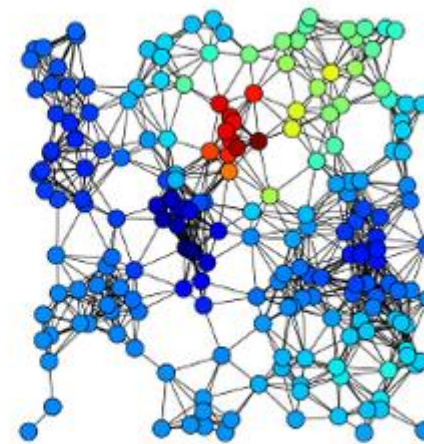
Eigenvector centrality



Degree centrality



Harmonic centrality



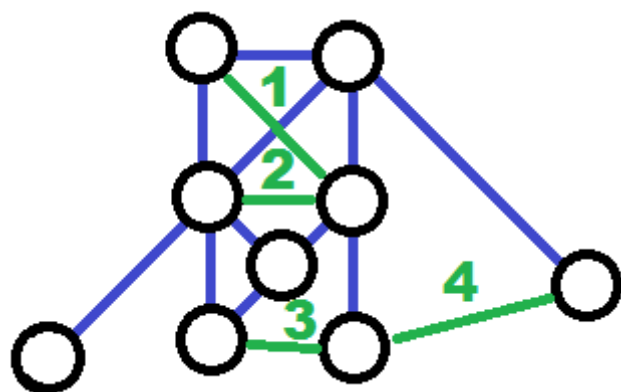
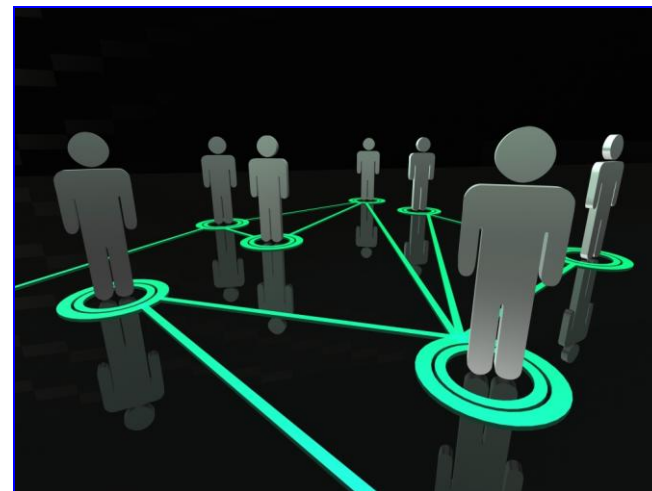
Katz centrality

Прогнозирование появления ребра в динамическом графе (Link Prediction Problem)

Международное соревнование «IJCNN
Social Network Challenge»

<http://www.kaggle.com/c/socialNetwork/>

Приложения: социальные сети, сотовые операторы, мобильные операторы и т.д.



Дан граф,
Список потенциальных рёбер
Необходимо ранжировать список
по вероятности появления

Как решать?

Методы решения LPP

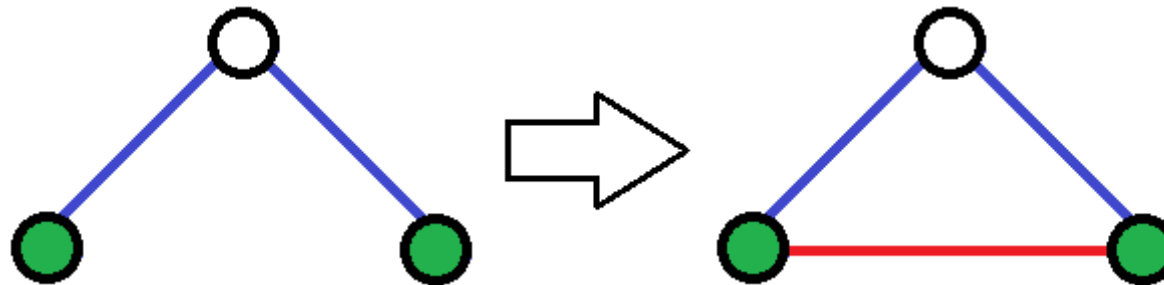
Формирование признакового пространства

Признаки строятся для пар вершин (i, j)

Все признаки «логичные» + описательные

Первая идея – принцип «друг моего друга»:

**если Иван дружит с Сергеем, а Сергей с Петром,
то Иван подружится (дружит) с Петром**



**если (x, z) – ребро, (z, y) – ребро,
то (x, y) – ребро или станет ребром.**

Пример признака на этом принципе? В его чём недостатки?

признак №1 – число соседей

Чем больше общих друзей имеют Иван и Пётр, тем более вероятней, что они подружатся.

**$|\Gamma(x) \cap \Gamma(y)|$ – хорошая мера сходства вершин,
где $\Gamma(x)$ – множество соседей вершины x**

признак №2

$|\Gamma(x)| \cdot |\Gamma(y)|$ – коэффициент предпочтительности

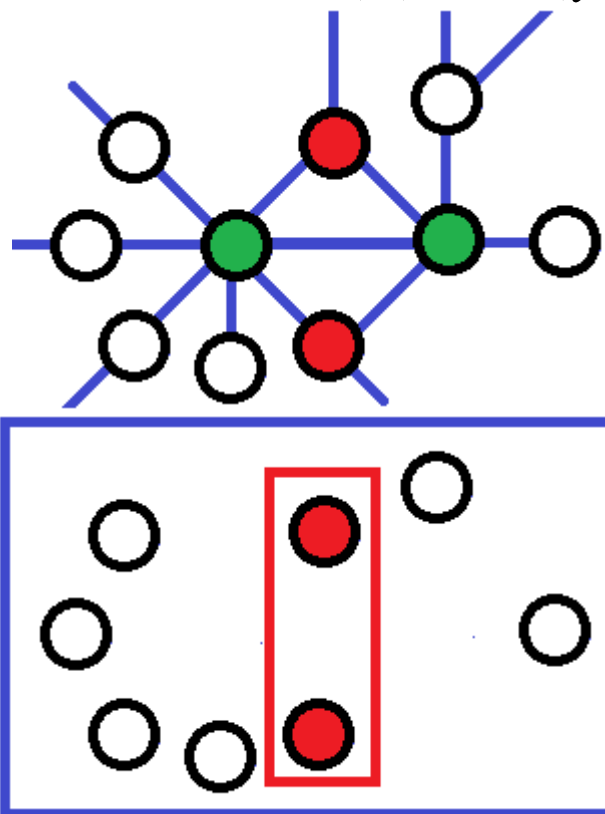
Чем более общительны, тем скорее подружатся

признак №3

Или наоборот: чем больше процент общих друзей

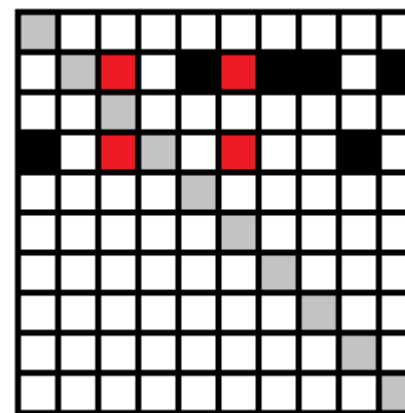
$$\frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

– коэффициент Жаккара



обычные признаки
для сравнения множеств

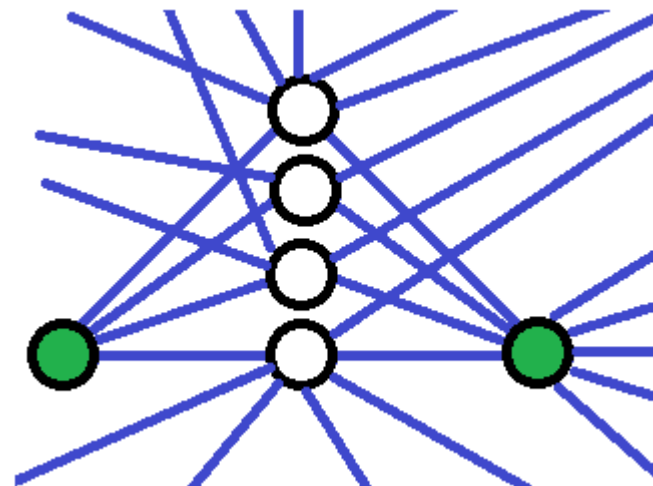
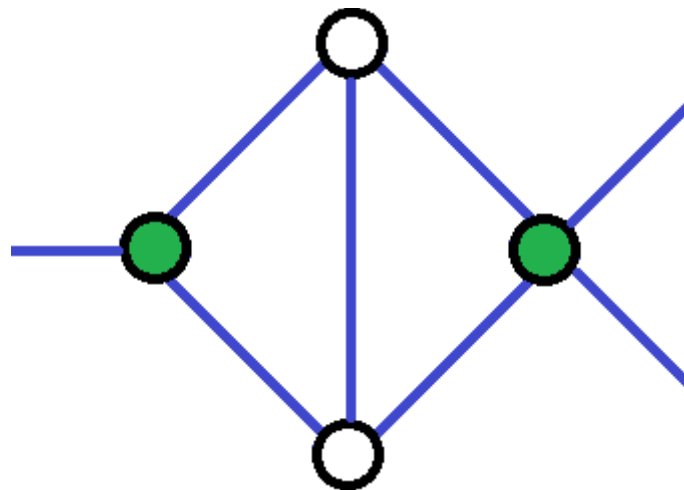
просто сравнение строк матрицы
смежности



Полезно: разный подход к описанию смысла (множества, строки)

признак №4

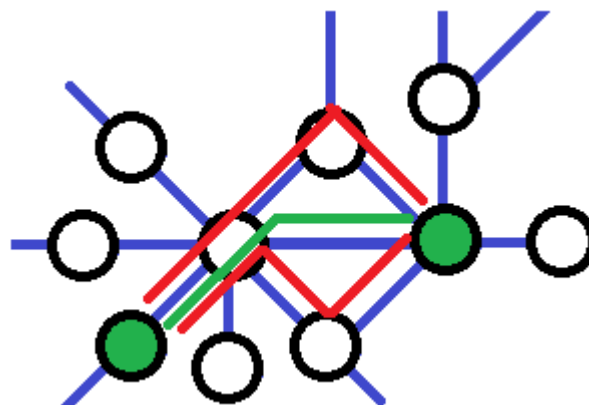
не все друзья одинаковые!



$$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} - \text{коэффициента Адамик/Адара}$$

признак №5

Учитывать целые цепочки друзей-друзей



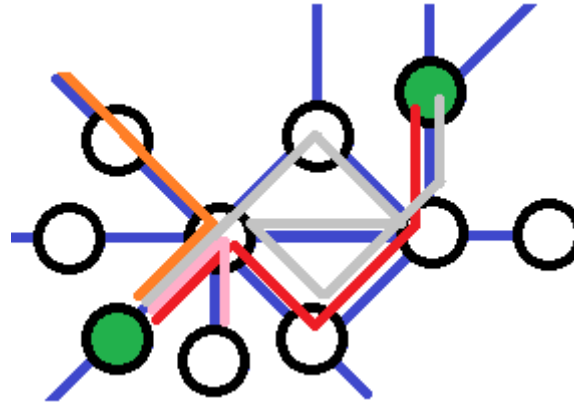
$$\sum_{l=1}^{\infty} \beta^l \text{path}_l(x, y) - \text{признак Katz}$$

равен ху-му элементу матрицы

$$(I - \beta M)^{-1} - I,$$

Признаки на основе случайных блужданий

Вершины близки, если из одной легко попасть во вторую



Пример: среднее время достижения вершины

Часто используют не матрицу смежности, а её k-SVD-аналог

Признаки на основе рекуррентных вычислений

$$\text{sim}(x, y) = \frac{\gamma}{|\Gamma(x)| \cdot |\Gamma(y)|} \sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} \text{sim}(a, b)$$

Вероятностные методы

Пусть вершина i порождается с вероятностью $P(i)$

По ней порождается латентный класс с вероятностью $P(z | i)$

По нему порождается ребро с вероятностью $P(j | z)$

Вероятность появления ребра =

$$P(i)P(z | i)P(j | z)$$

**– это ответ, вероятности здесь оцениваются EM-алгоритмом,
максимизируя логарифм правдоподобия**

$$\sum_{\{i,j\} \in E} \log(P(i, j))$$

Алгоритм PageRank (подробнее про случайные блуждания)

Что такое важные страницы в интернете

- 1. На них ссылаются (есть входящие ссылки)**
- 2. На них ссылаются важные страницы**

Алгоритм PageRank

Если страница j с важностью r_j имеет n выходных ссылок, каждая ссылка «передает» важность r_j / n

Важность страницы = сумма всех входных ссылок

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{\deg_{\text{out}}(i)}$$

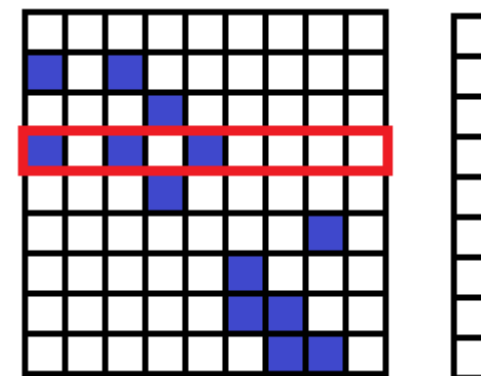
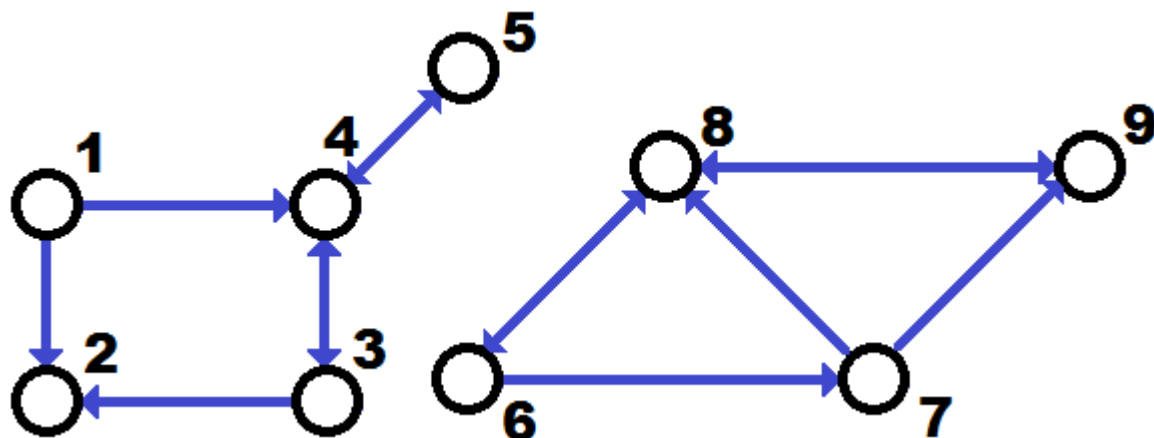
пронормируем матрицу смежности ($m_{ij} = 1 \Leftrightarrow (j \rightarrow i)$):

$$\text{sum}(M, 1) = (1, \dots, 1),$$

тогда вектор важности рекурсивно записывается как

$$r = Mr$$

Алгоритм PageRank



$$r_4 = \frac{r_1}{2} + \frac{r_3}{2} + \frac{r_5}{1}$$

Внимание на построение матрицы смежности!

Решаем задачу на собственные значения

$$Mr = \lambda r$$

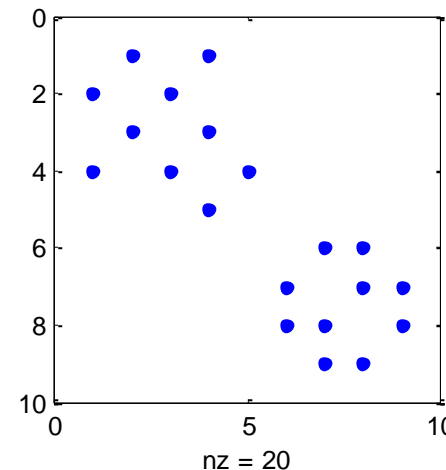
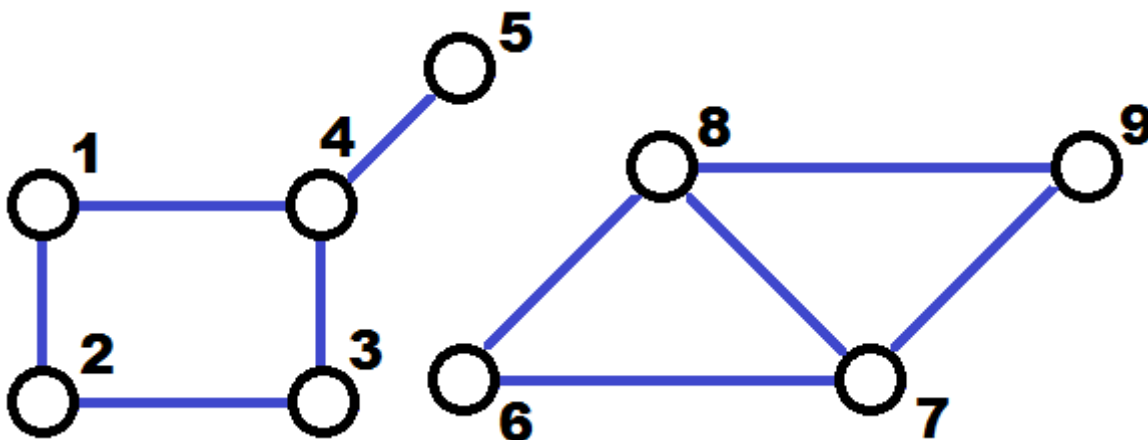
Наибольшее с.з. = 1

Берём его собственный вектор!

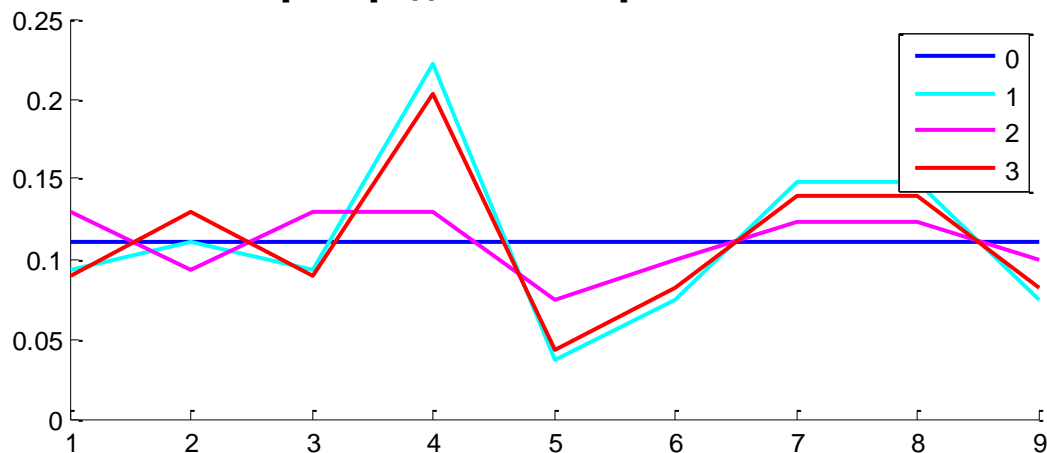
Итерационный метод $r = Mr$

это и находит

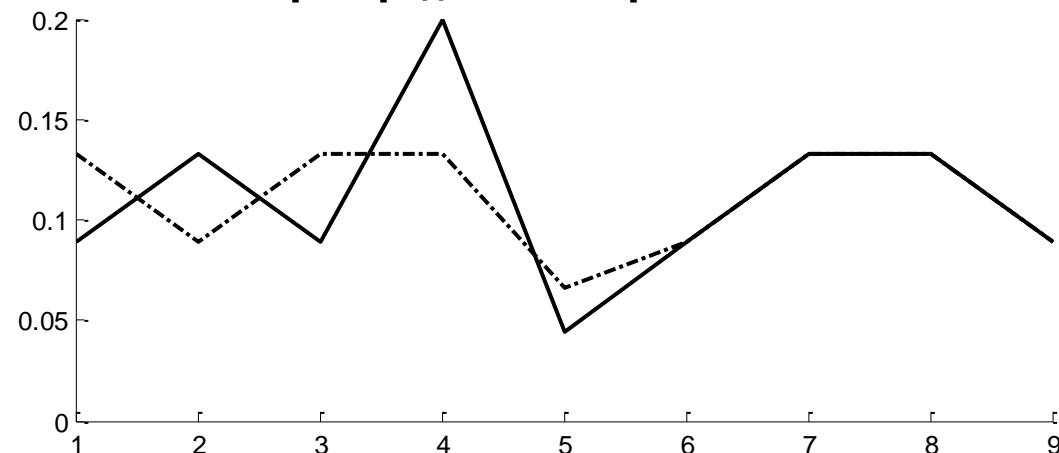
Проблема: на практике не всегда получается



распределение вероятностей



распределение вероятностей

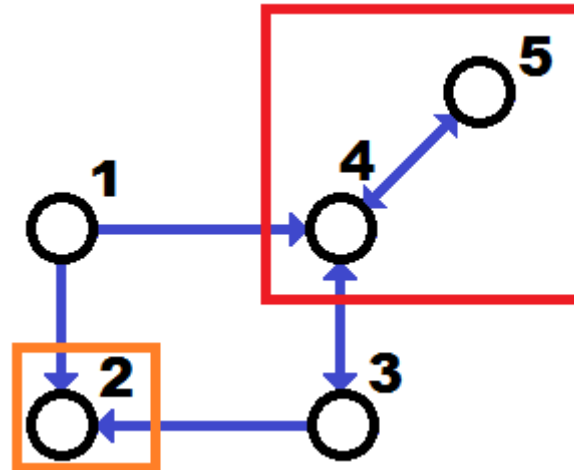


Почему?

Два типа проблем

1. Циклы

2. Мёртвые вершины

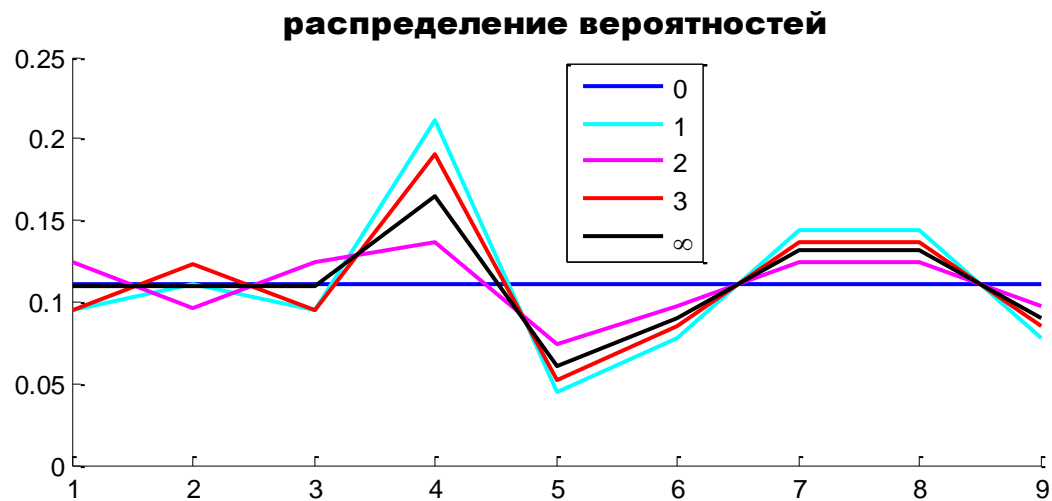


Решение: в итерационном алгоритме с вероятностью 0.1-0.2 прыгать в случайную вершину графа (~5 шагов)

Брин, Пейдж:

$$r_j = \beta \sum_{i \rightarrow j} \frac{r_i}{\deg_{\text{out}}(i)} + (1 - \beta) \frac{1}{N}$$

В результате



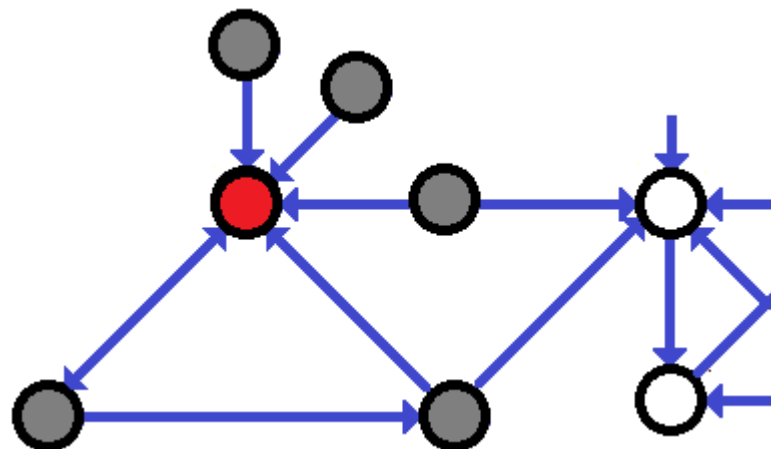
Практические аспекты

Переход не в произвольную вершину, а

- **в похожую,**
 - **из этого топика,**
 - **из доверительного множества (анти-спам: *.edu),**
 - **в эту вершину (SimRank)**
- и т.п.**

Зачем?

Ответ: в случае спама – борьба с фермами спама

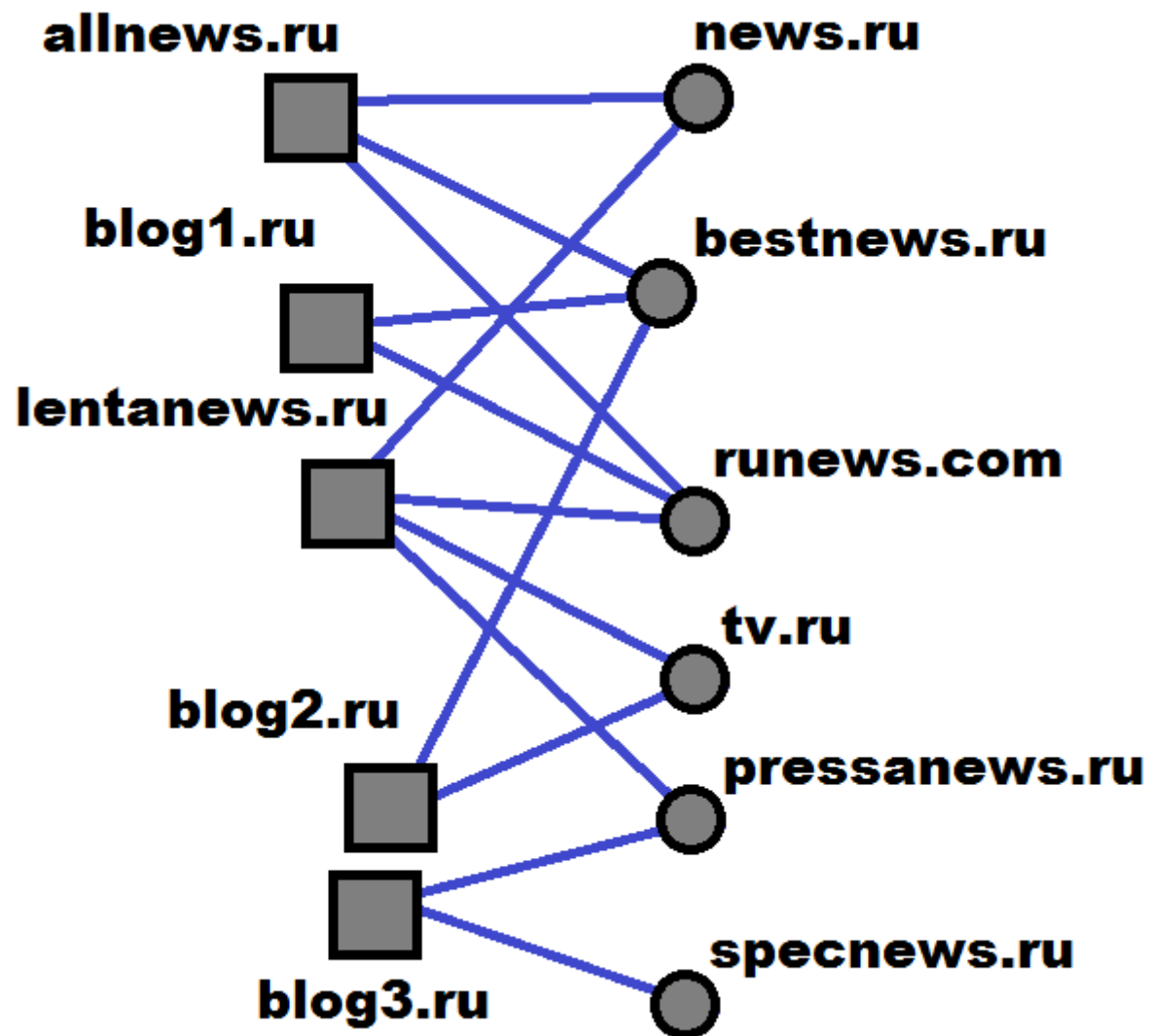


- – доверенная зона,
- – ферма спама,
- – спам

Для формирования доверенной зоны можно использовать эксперта

Ещё итерационные алгоритмы поиск ценных источников информации

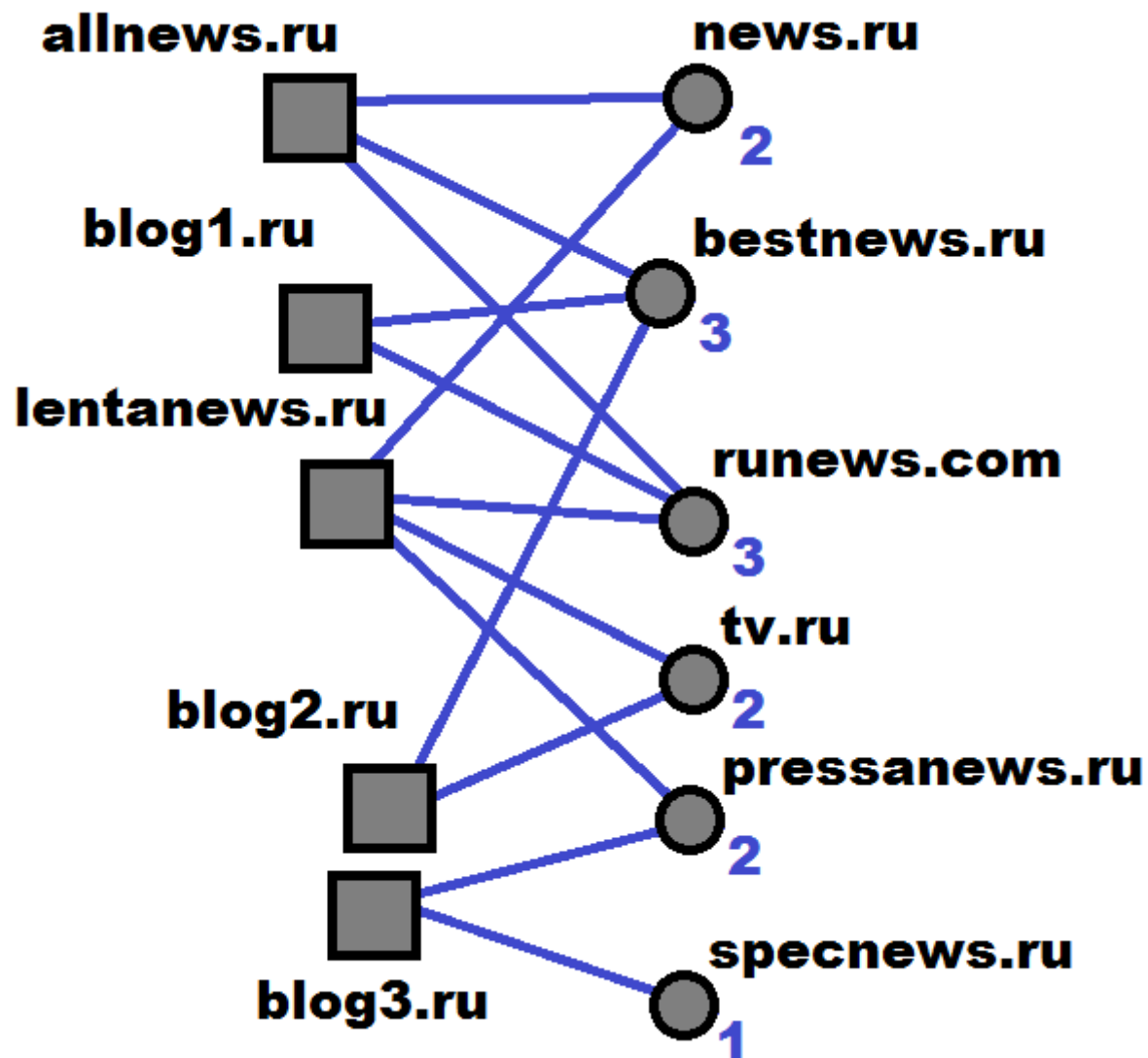
Агрегаторы



Новостные
сайты

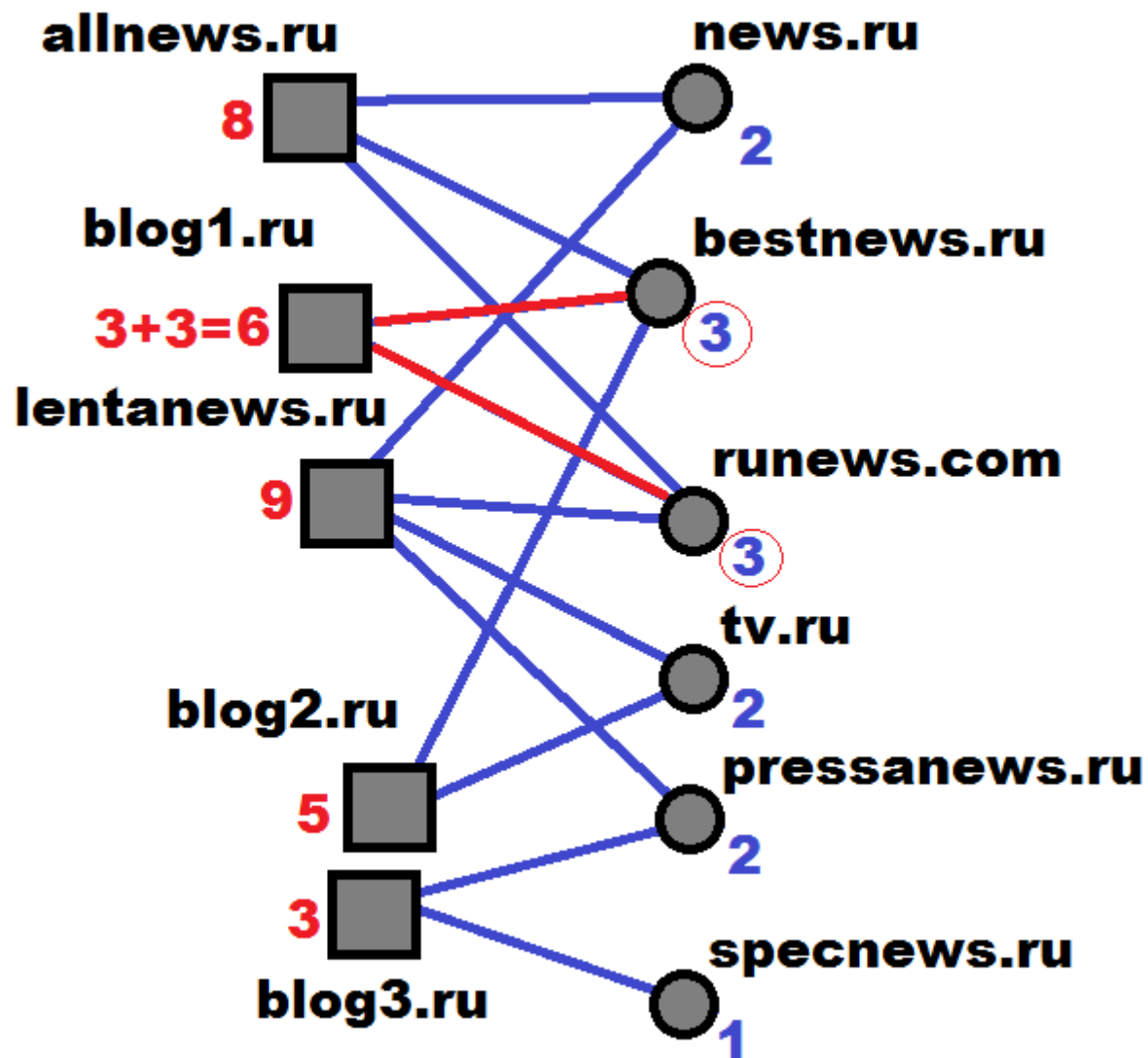
Ещё итерационные алгоритмы

Ценное то – на что ссылаются



Ещё итерационные алгоритмы

Ценное то – что ссылается на ценное



Ещё итерационные алгоритмы

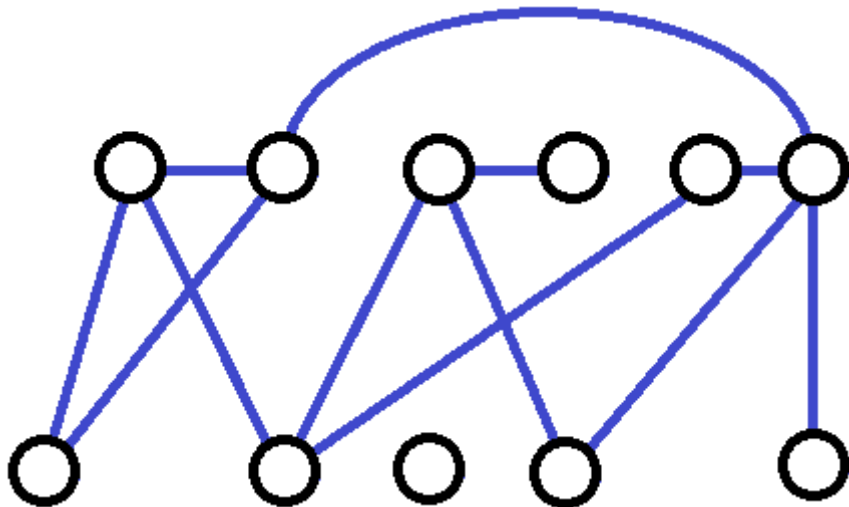
Дальше идея понятна...

К решению какого матричного уравнения всё сводится?

Какая задача здесь возникает?

Соревнование «IJCNN Social Network Challenge»

**Задача не в стандартной постановке –
граф почти двудольный, ориентированный!**



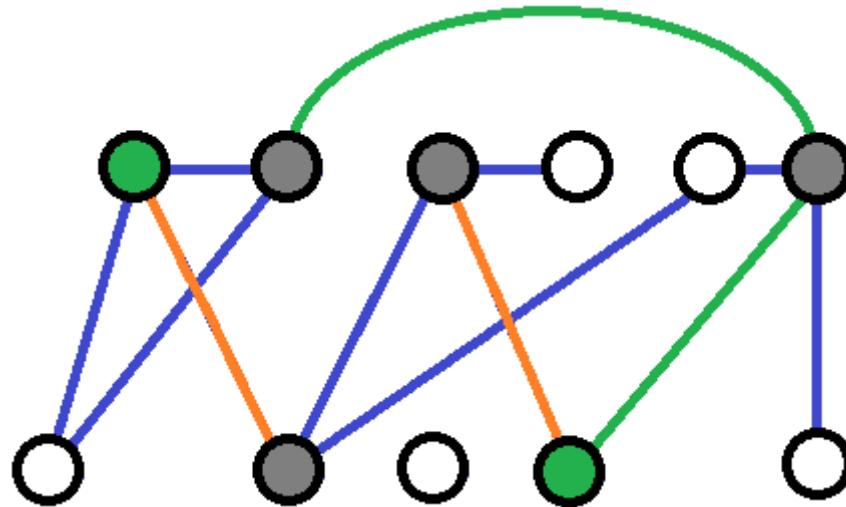
вершин = 1'100'000

рёбер = 7'200'000

Сеть Flickr

**Тест = 4480+4480
потенциальных рёбер**

Описанные признаки легко обобщаются на двудольный случай



Кстати, тонкости в задаче –

как выбрать обучающую выборку (надо знать как делал заказчик)!

Если не-рёбра = случайные не рёбра,
то задача лёгкая, обобщения нет

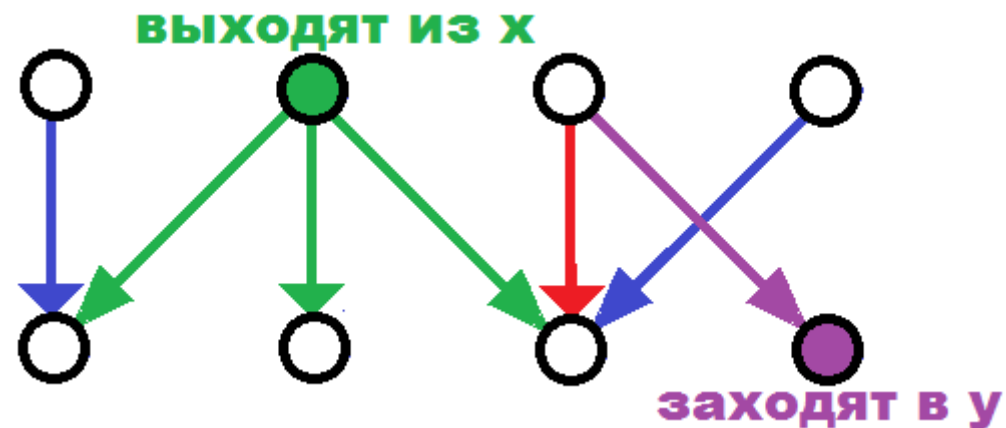
Если не-рёбра = почти рёбра,
то они могут скоро стать рёбрами... а этому мы и должны научиться

Первый подход

друг друга

$$\frac{|(\Gamma(x,*) \times \Gamma(*,y)) \cap E|}{|\Gamma(x,*)| \cdot |\Gamma(*,y)| + 1}$$

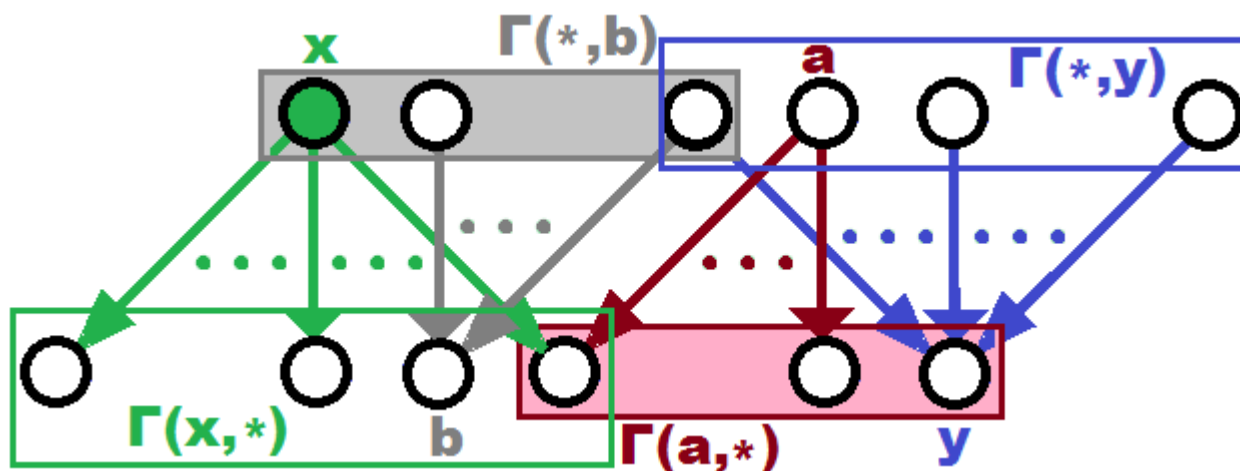
$$\Gamma(x,*) = \{y \in V \mid (x, y) \in E\}$$



Улучшение качества при таком признаке

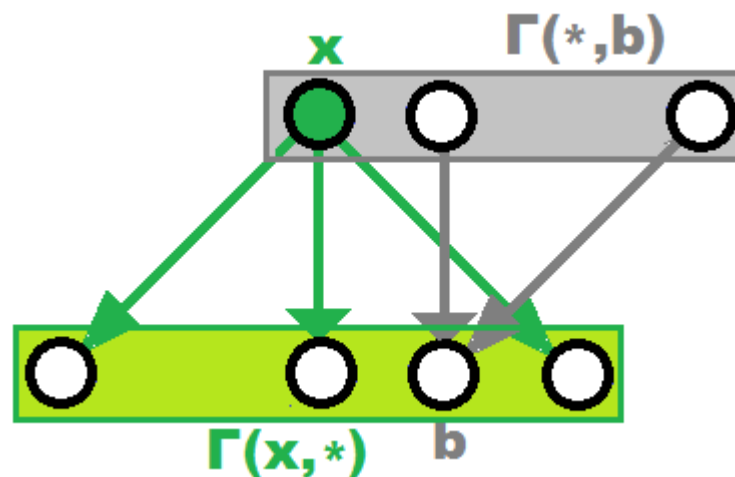
$$\frac{\sum_{\substack{a \in \Gamma(*,y) \\ b \in \Gamma(x,*)}} \frac{|\Gamma(a,*) \cap \Gamma(x,*)| \cdot |\Gamma(*,b) \cap \Gamma(*,y)|}{\sqrt{|\Gamma(a,*)| \cdot |\Gamma(*,b)|}}}{|\Gamma(x,*)| \cdot |\Gamma(*,y)| + 1}$$

Какой смысл этого признака?



Признак №2

$$\frac{1}{|\Gamma(x,*)|} \sum_{b \in \Gamma(x,*)} \frac{|(\Gamma(*,b) \cap \Gamma(x,*)) \cap E|}{|\Gamma(*,b)| \cdot |\Gamma(x,*)| + 1}$$



**насколько дружелюбны друзья x
(не зависит от y , хорош в комбинации)**

Второй подход

вершины соединены, если соединены похожие

$$\frac{|(X \times Y) \cap E|}{|X| \cdot |Y| + 1}$$

X – вершины похожие на x ,

Y – вершины похожие на y .

Что такое похожие?

сравниваем как строки в матрице смежности

Лучшее – скалярное произведение с довеском:

$$|\Gamma(x,*) \cap \Gamma(a,*)| - \frac{1}{2 + |\Gamma(a,*)| - |\Gamma(x,*) \cap \Gamma(a,*)|}$$

Оптимальные множества: $|X| = 9, |Y| = 40$

При разных метриках – некоррелированные признаки

Как учитывать похожесть?

Вместо $\frac{|(X \times Y) \cap E|}{|X| \cdot |Y| + 1}$

весовую схему

$$\frac{1}{|X| \cdot |Y| + 1} \sum_{a \in A} \sum_{b \in B} w(a)w'(b)$$

Блендинг

$$\text{I} = 87.5$$

$$\text{I} + \text{II} = 90.7$$

$$\text{III} = 90.7$$

$$\text{I} + \text{II} + \text{III} = 92.6$$

$$\text{PR} = 93.0$$

$$\text{I} + \text{II} + \text{III} + \text{PR} = 95.0$$

Сообщество в графе

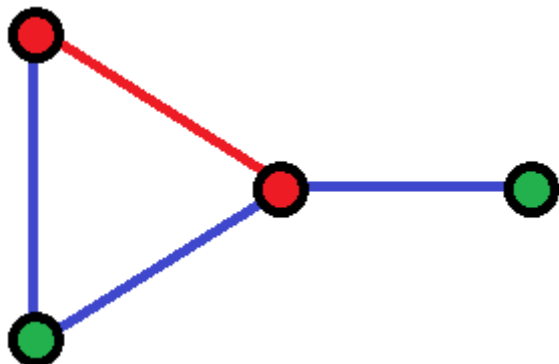
нет определения
рёбер внутри сообщества много,
рёбер соединяющих сообщество с остальными вершинами мало
малый радиус сообщества

Какие бывают определения:

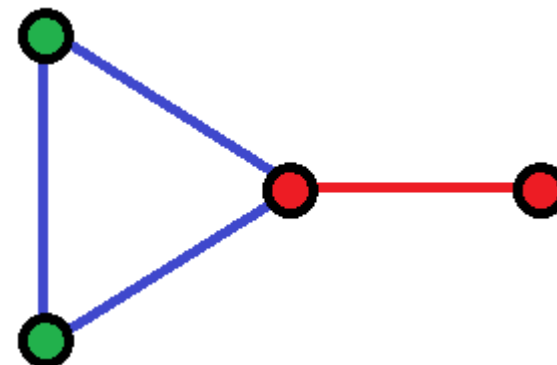
- 1. Чёткие**
- 2. Нечёткие (не определения, см. выше)**
- 3. Алгоритмические (то что получается в результате действия алгоритма)**

Сообщество в графе

Идеальный кандидат – клика

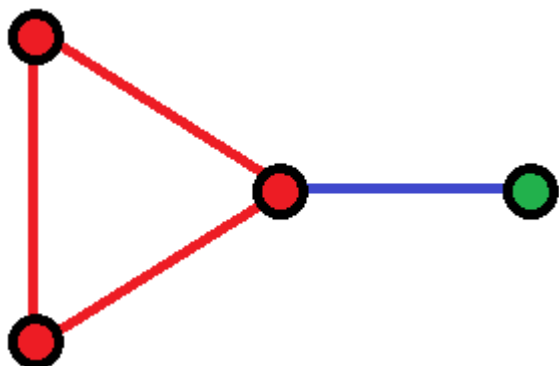


Клика



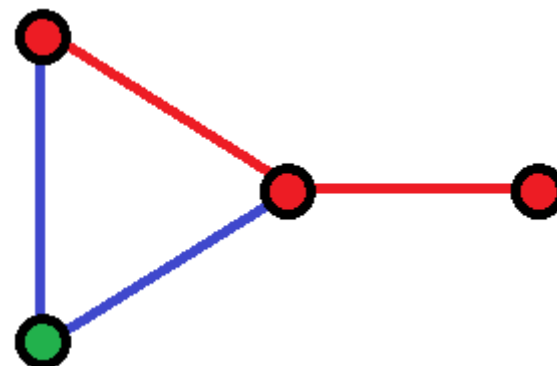
Максимальная клика

Не может быть расширена



Наибольшая клика

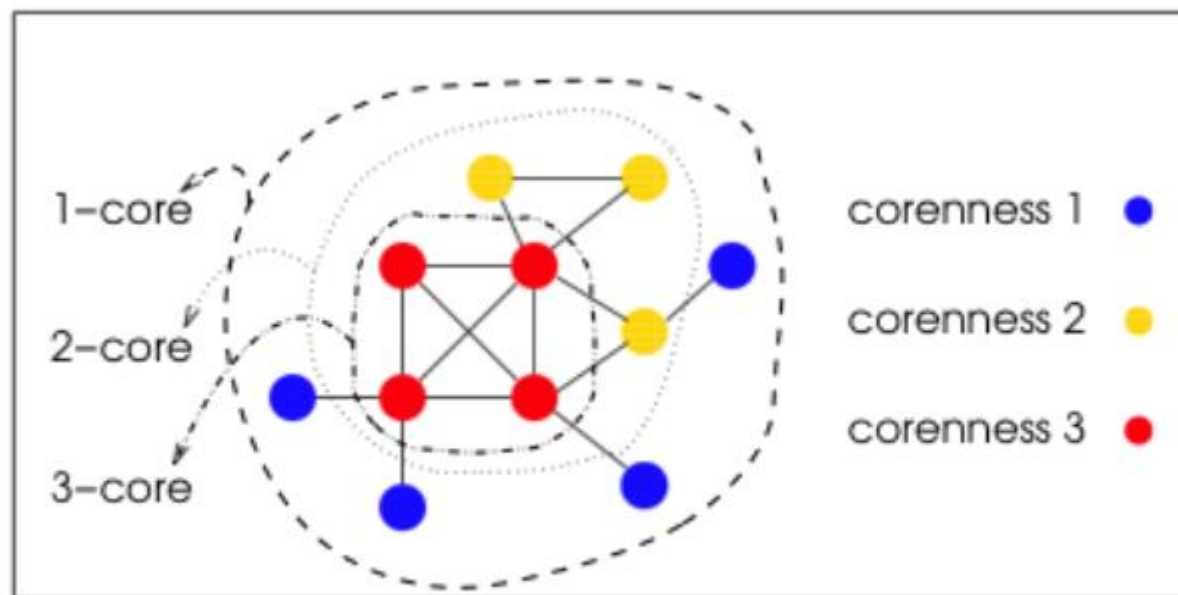
Клика наибольшего размера



Не клика

Но вычислительные сложности...

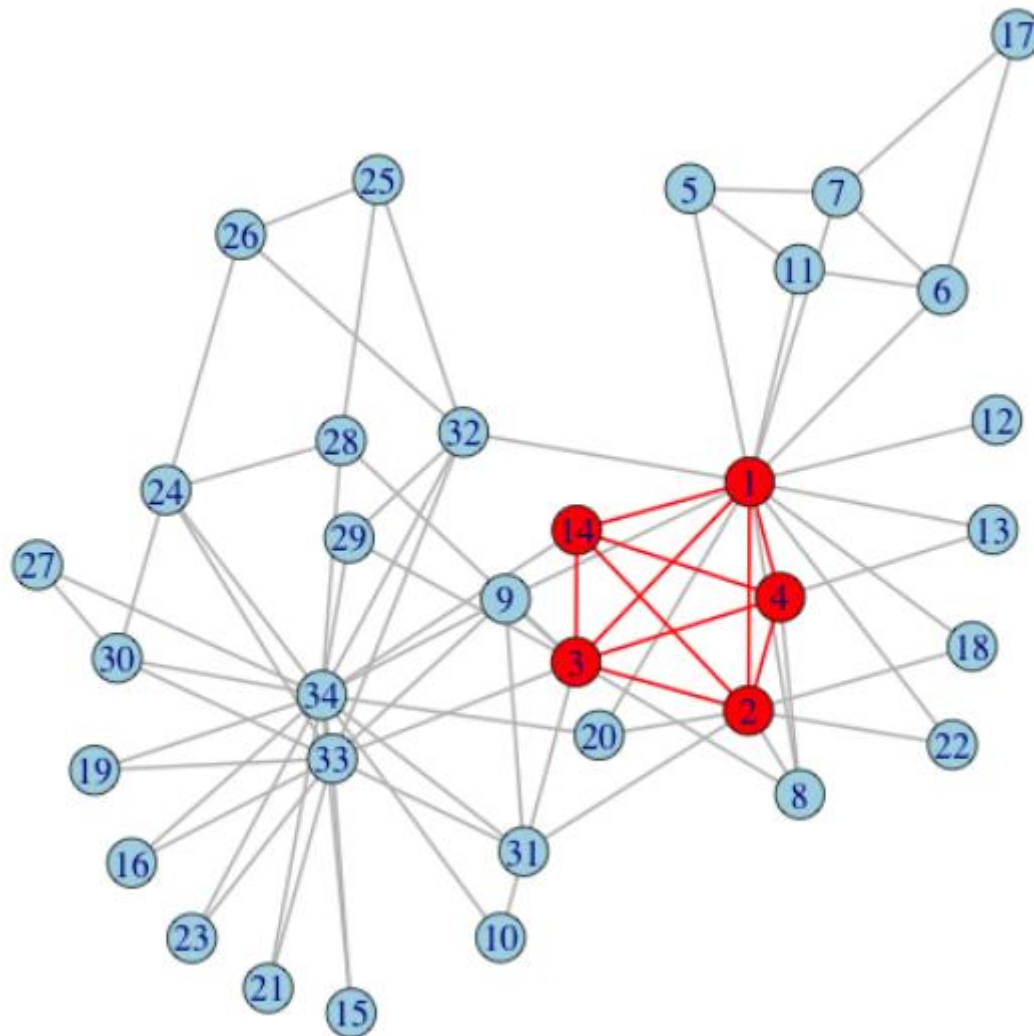
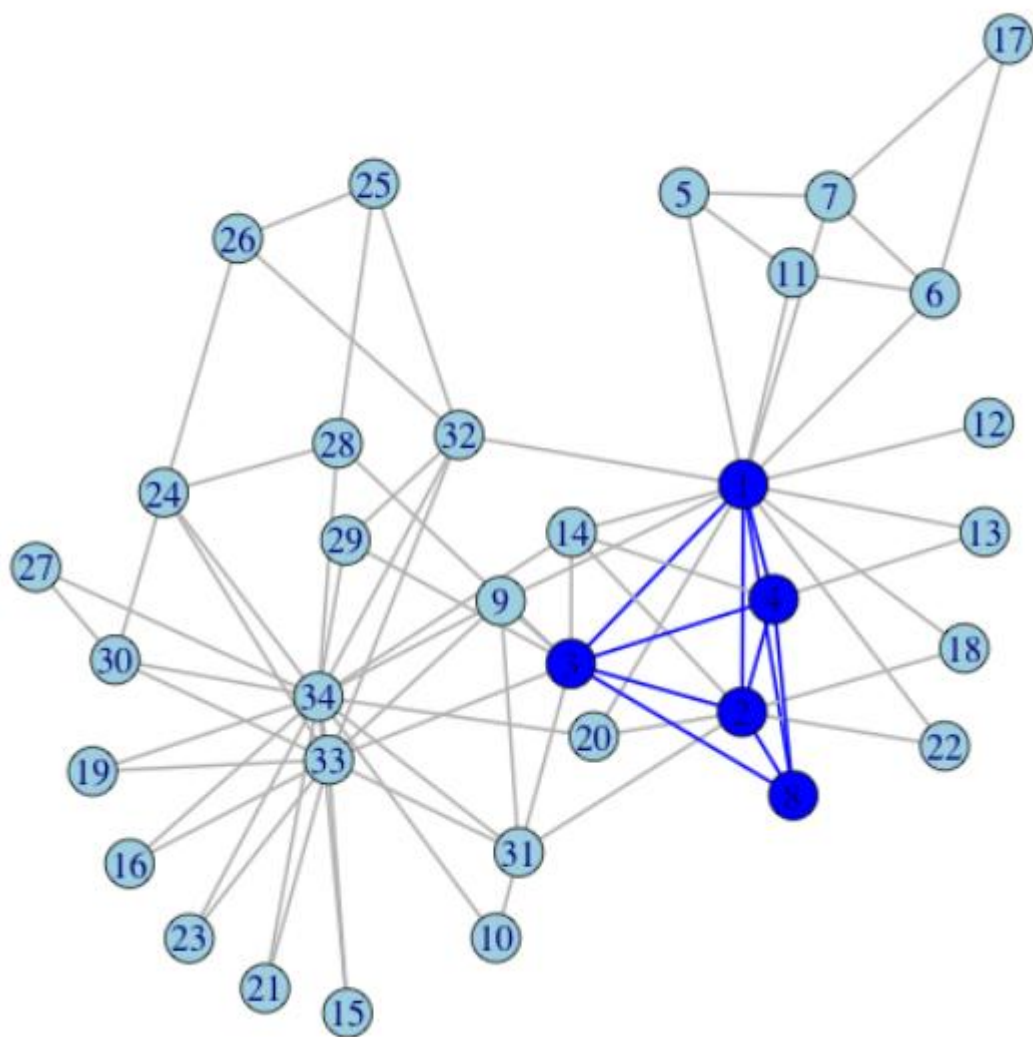
к-ядра



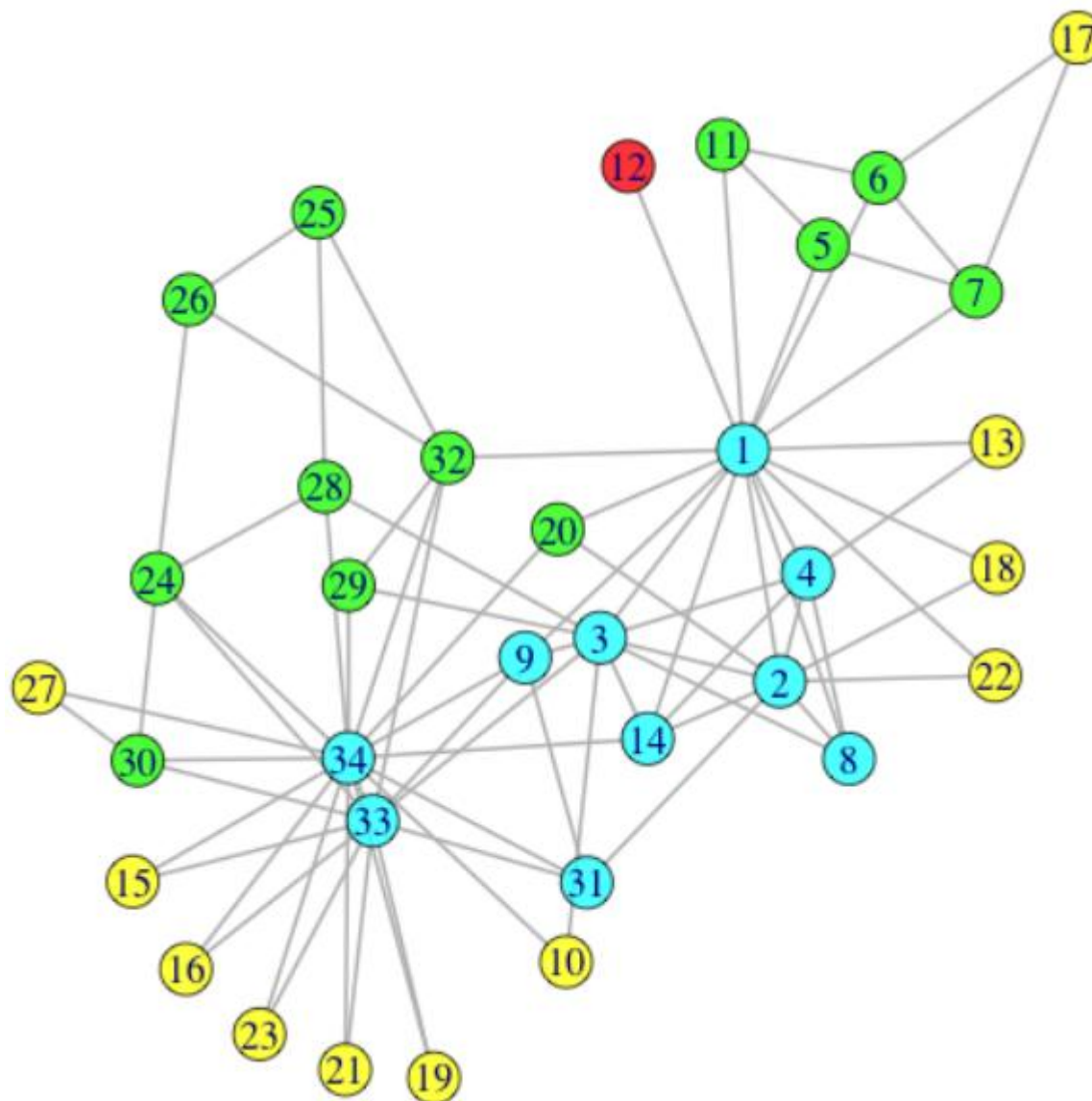
Alvarez-Hamelin et.al., 2005

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

Наибольшие клики (Карате клуб)



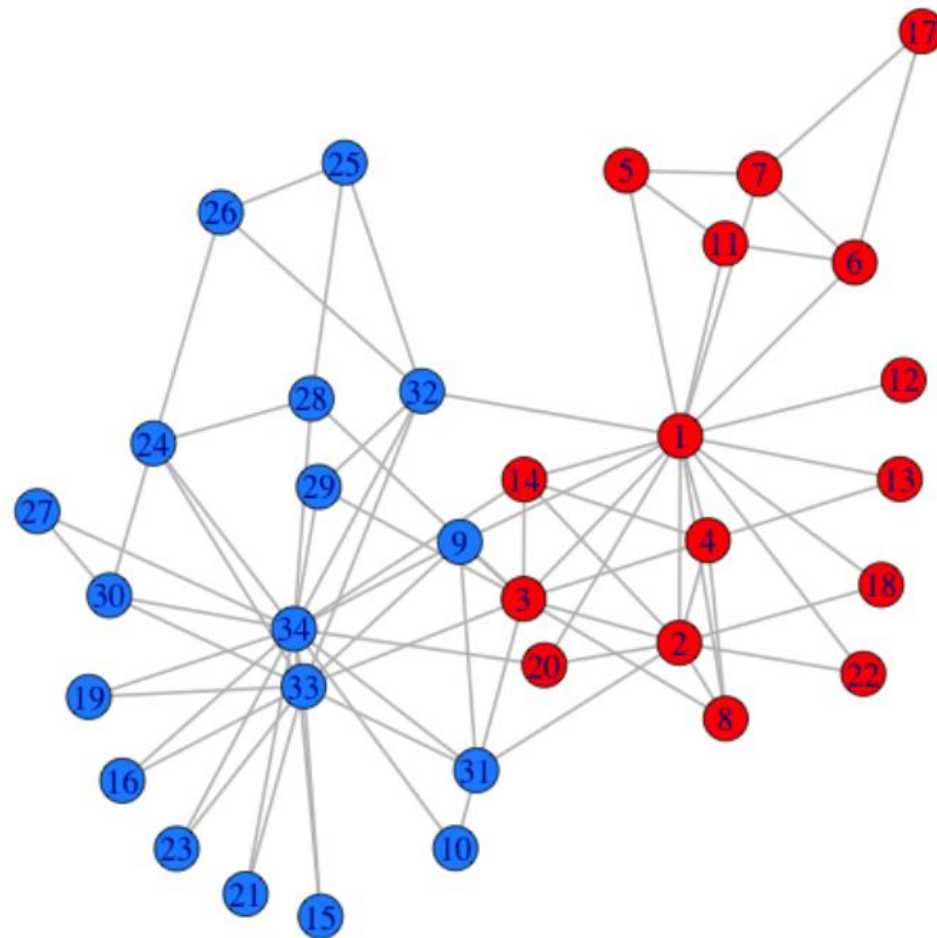
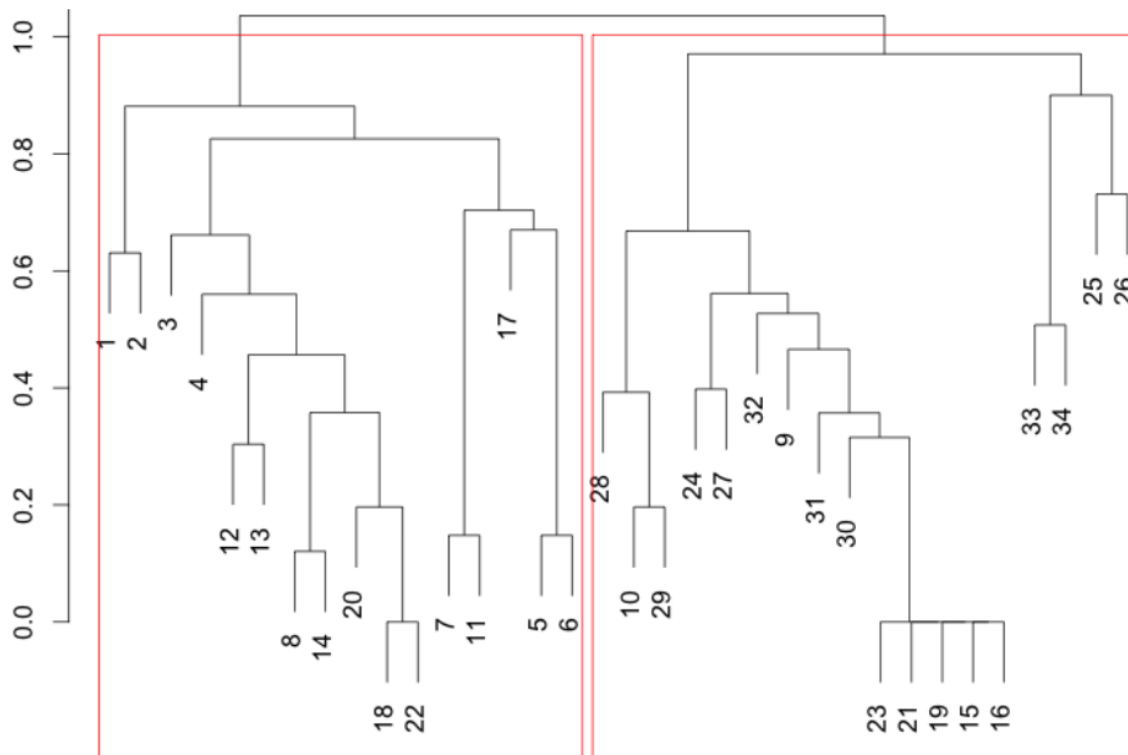
Ядра (Карате клуб)



Выделение сообществ

1й способ

Обычная кластеризация с мерой схожести вершин



Выделение сообществ

1й способ - недостатки

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

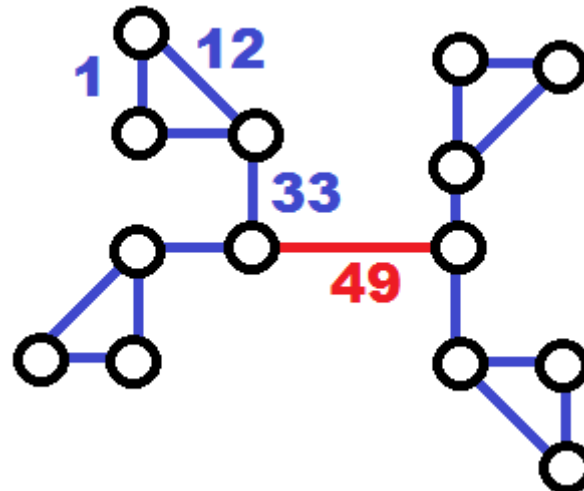
Выделение сообществ

2-й способ (Girvan-Newman)

Edge betweenness – число кратчайших путей, проходящих через ребро

**Повторять пока есть рёбра
удаление ребра с максимальным значением EB**

Получаем иерархическое разложение графа



На каком этапе останавливаться (в иерархическом делении)

**Как в кластеризации:
ввести функционал качества**

Число рёбер в группе – ожидаемое число рёбер

Почему не оптимизировать этот функционал напрямую?

3-й способ (модулярность, тоже Girvan и Newman)

Сравниваем число рёбер в сообществе с ожидаемым числом рёбер

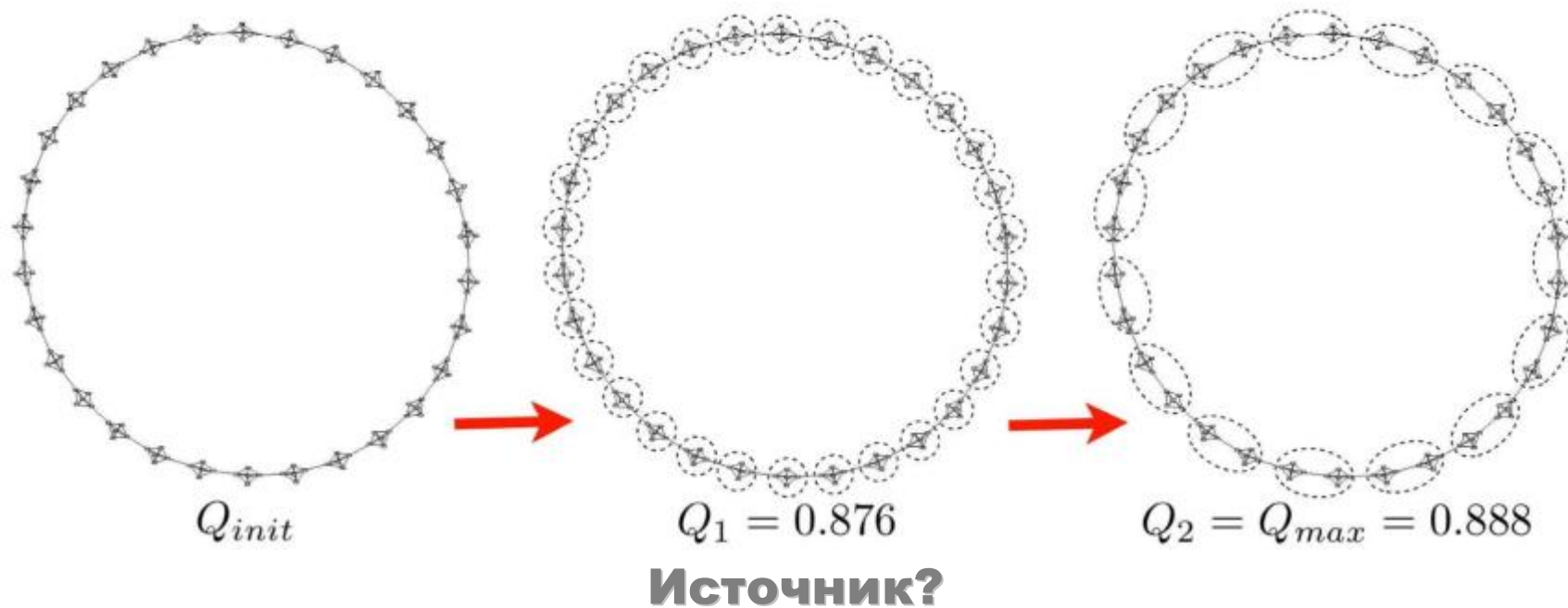
$$Q = \frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{\deg(i) \deg(j)}{2m} \right) \cdot I[x_i = x_j]$$

x_i – метка i -й вершины

как минимизируется

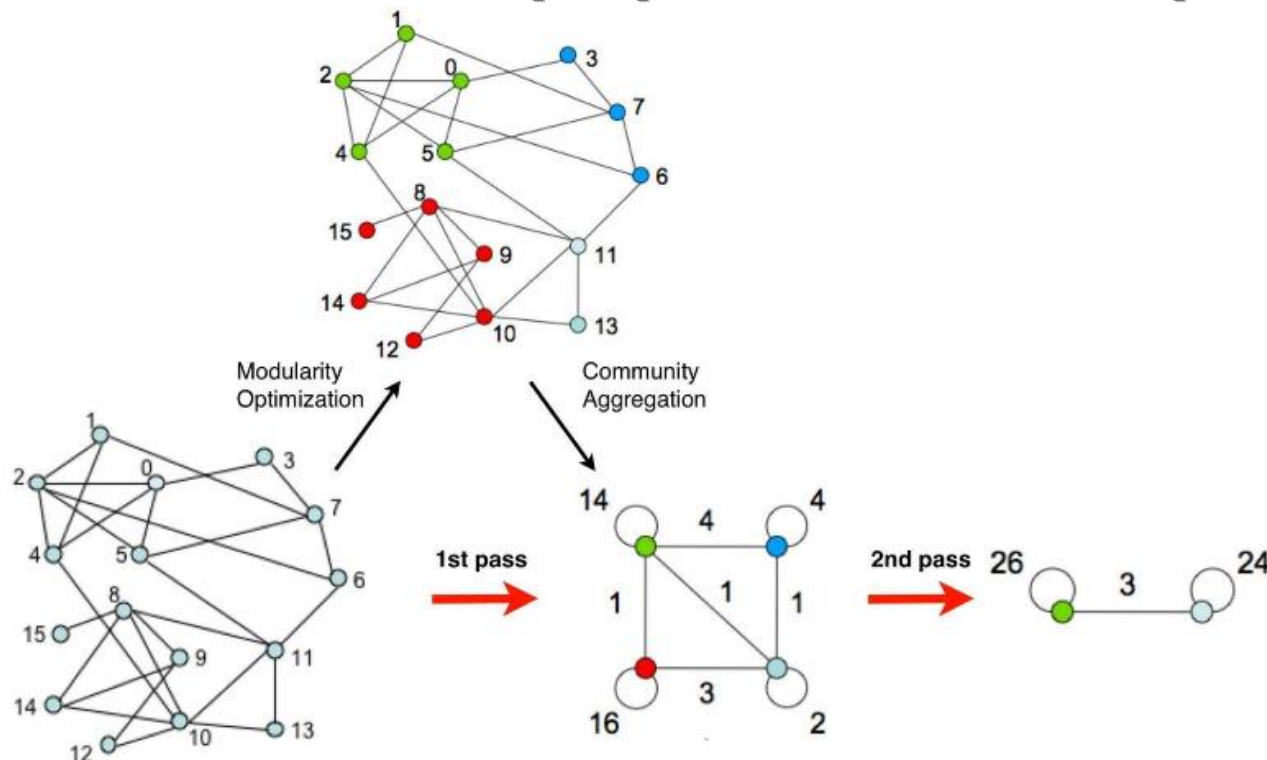
- симуляция отжига
- спектральные методы и т.п.
- жадные алгоритмы
- попытки объединять/перетаскивать сообщества

Иногда модулярность подводит...



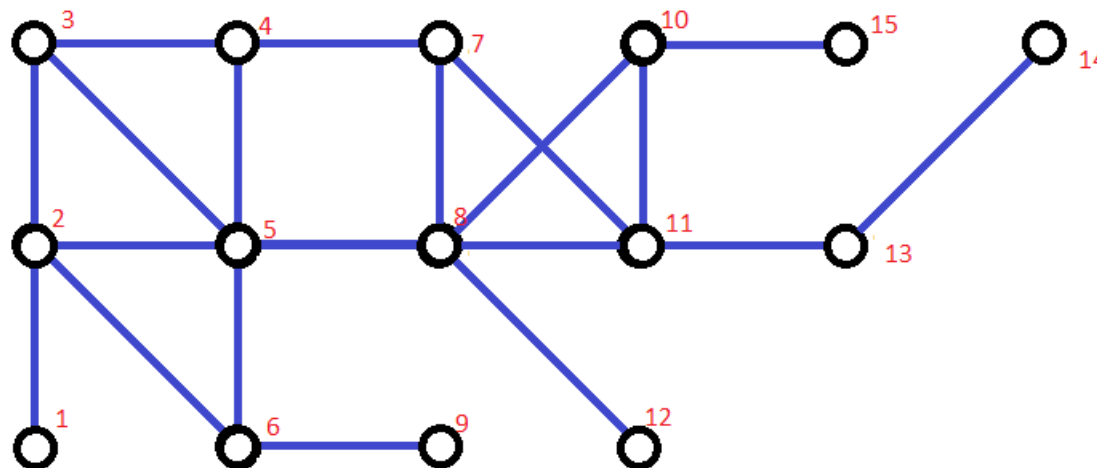
Fast community unfolding [Multilevel]

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

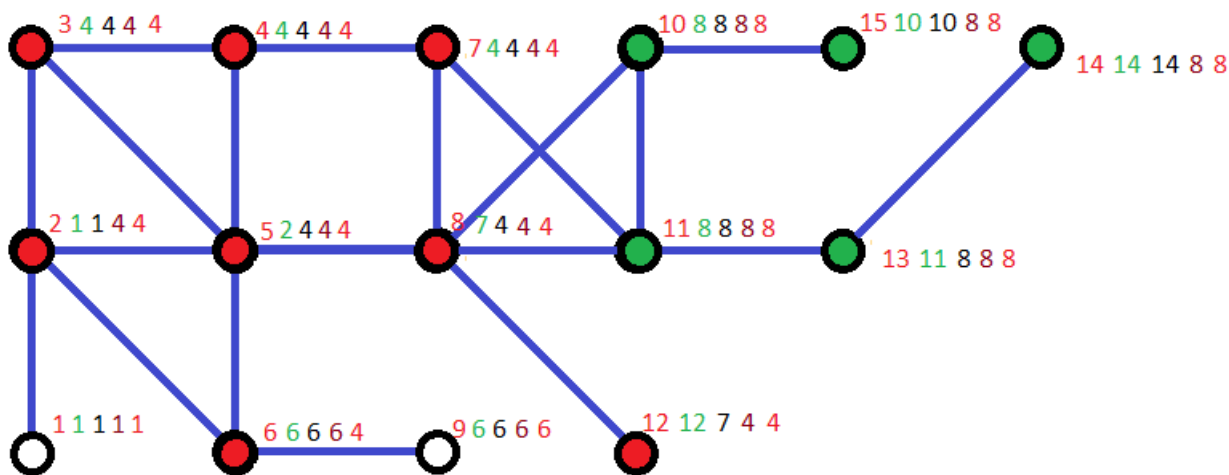


4-й способ: Label Propagation

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



Label Propagation



5й способ: Walktrap

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

Как измеряется близость сообществ

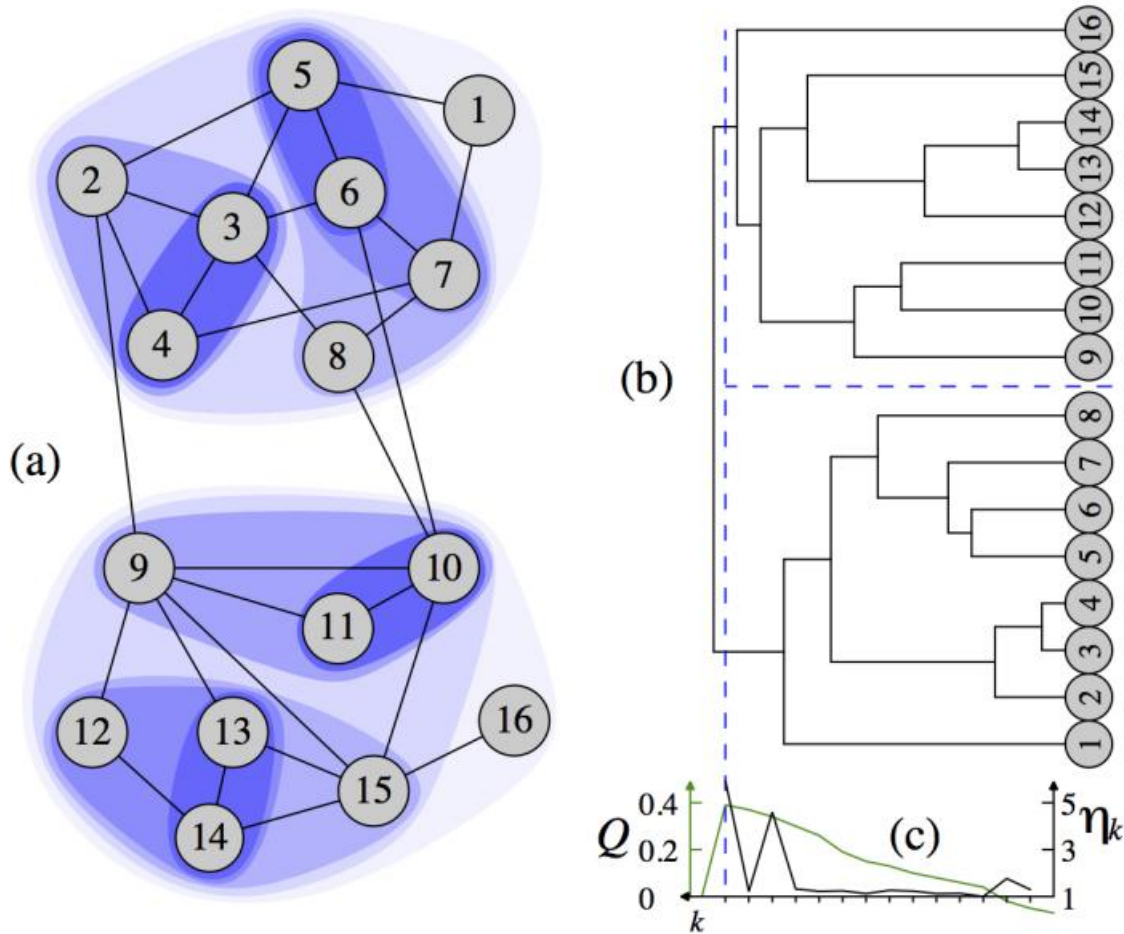
$$r_{A,B}(t) = \sqrt{\sum_{i=1}^n \frac{(P_{A,i}^t - P_{B,i}^t)^2}{\deg(i)}} = \| D^{-0.5} P_A^t - D^{-0.5} P_B^t \|,$$

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

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

(можно вычислить приближённо – случайными блужданиями)

Walktrap



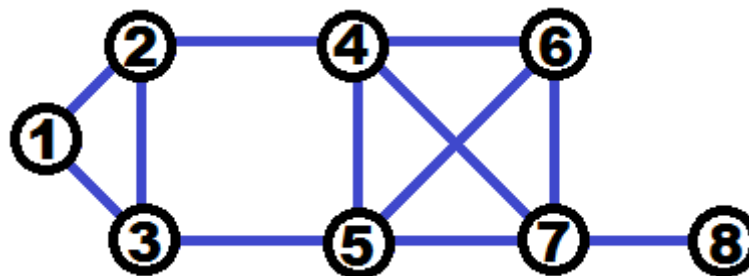
Другая идея выделения сообществ

Разбиение графа!

6-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	

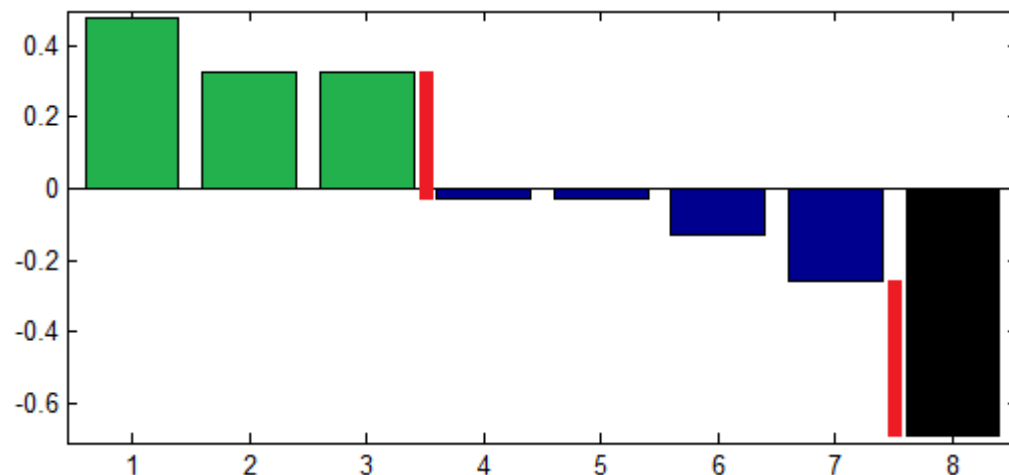


Матрица Лапласа

	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



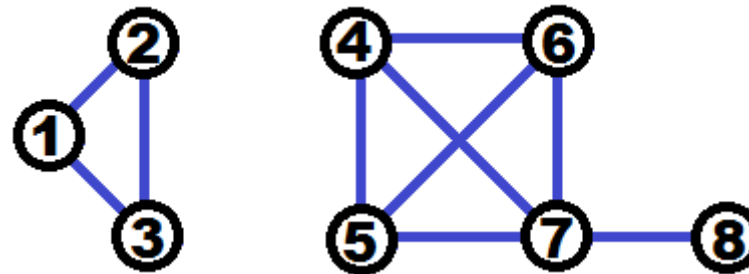
Всё содержится в одном векторе! И на одном слайде!

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
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
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]

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

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

Избежать очевидного константного решения: $\tilde{1}^T x = 0$.

Но это сложная переборная задача, поэтому вместо

$$x = (x_1, \dots, x_n) \in \{\pm 1\}^n, \tilde{1}^T x = 0,$$

Решают вещественную задачу с ограничениями

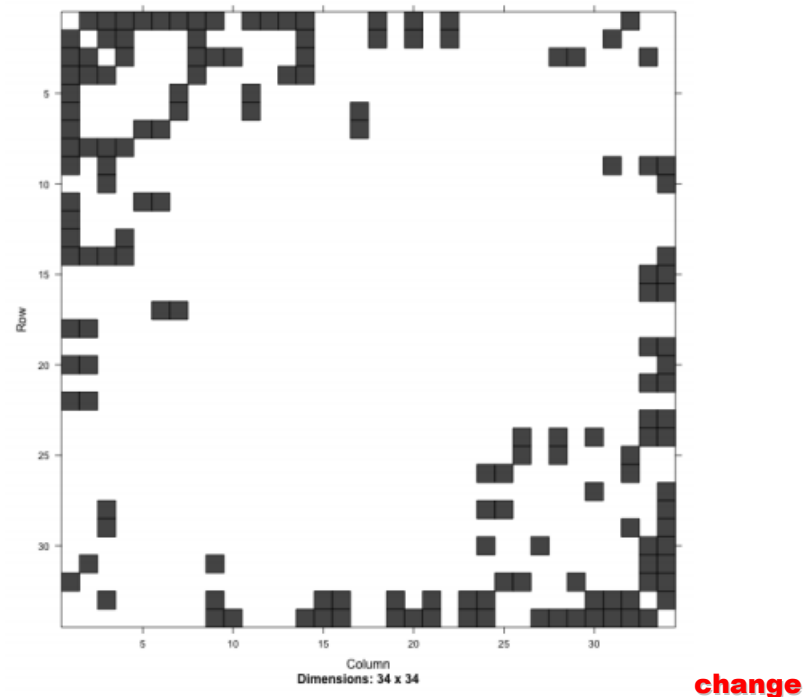
$$\tilde{1}^T x = 0, \|x\| = 1.$$

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

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

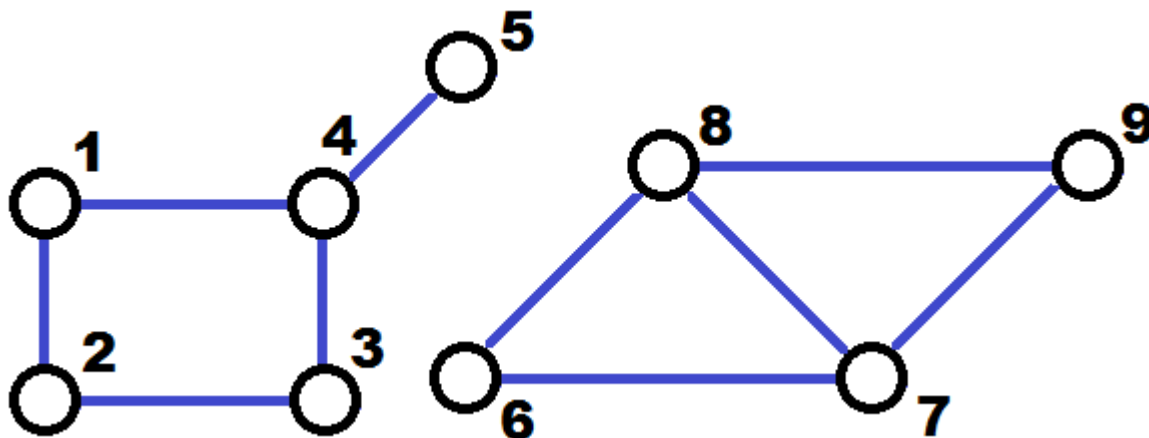
Совмещение идей

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.

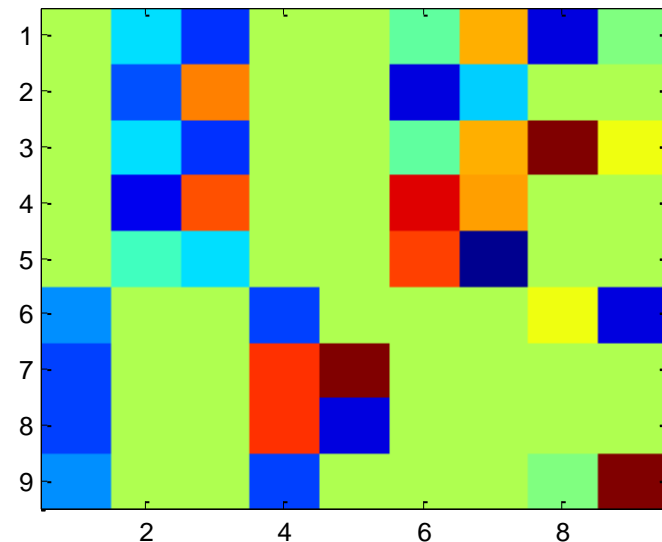
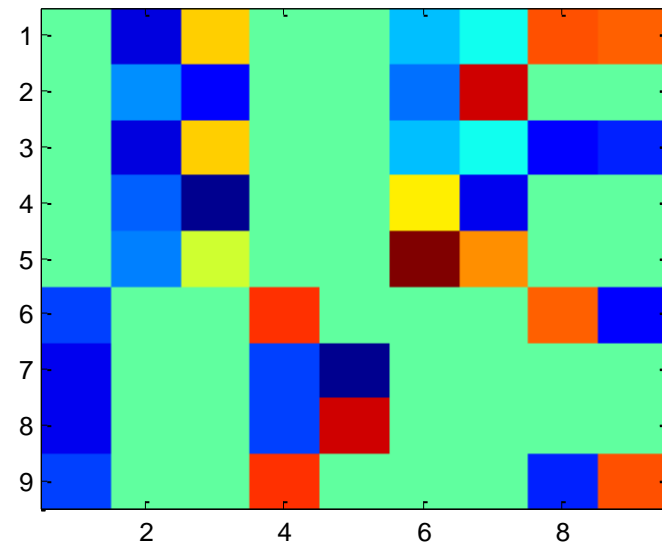
SVD над матрицей смежности



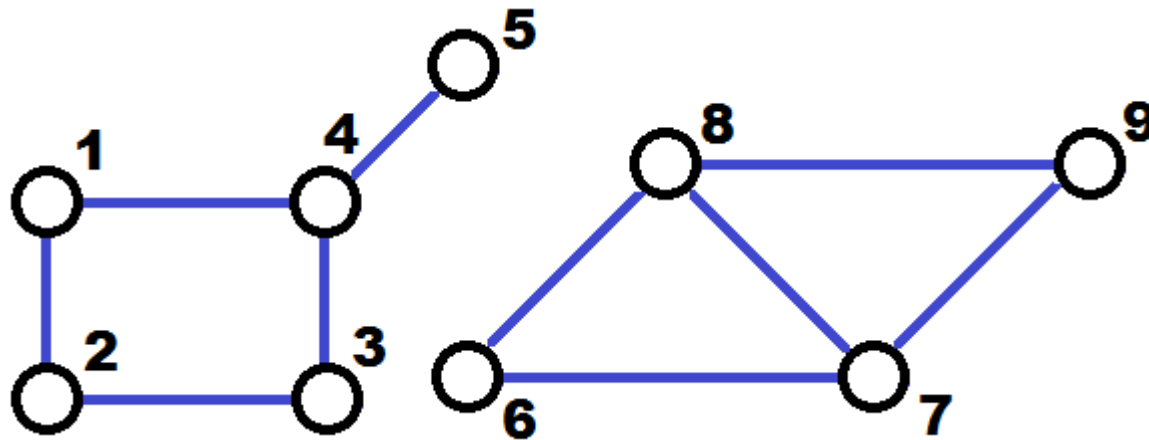
```
S = sparse([1 1 2 2 3 3 4 4 6 6 7 7 8 8 8 9 9 5 4 7], ...
          [2 4 1 3 2 4 1 5 7 8 8 9 6 7 9 8 7 4 3 6], 1)
```

```
[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



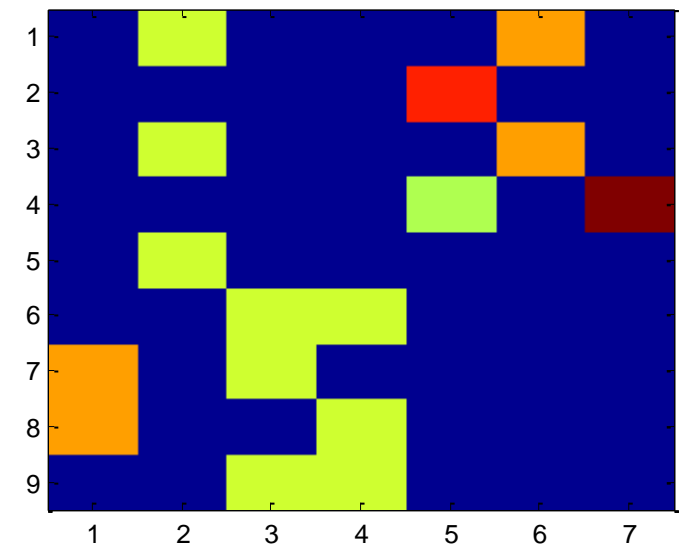
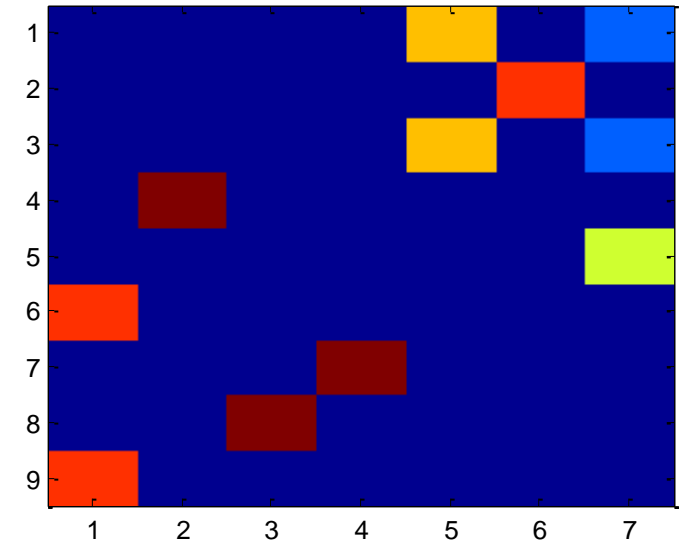
Неотрицательные матричные разложения



```
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 $k = \text{deg}(A)$,

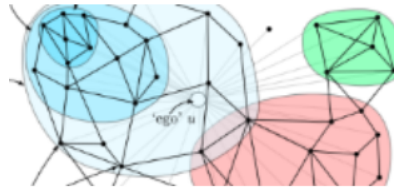
$$B = A - \frac{1}{2m} k k^T,$$

**Найти max с.в. $Bv = \lambda v$
 $\text{sgn}(v)$**

т.е. в задаче на с.з. используют разные матрицы...

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

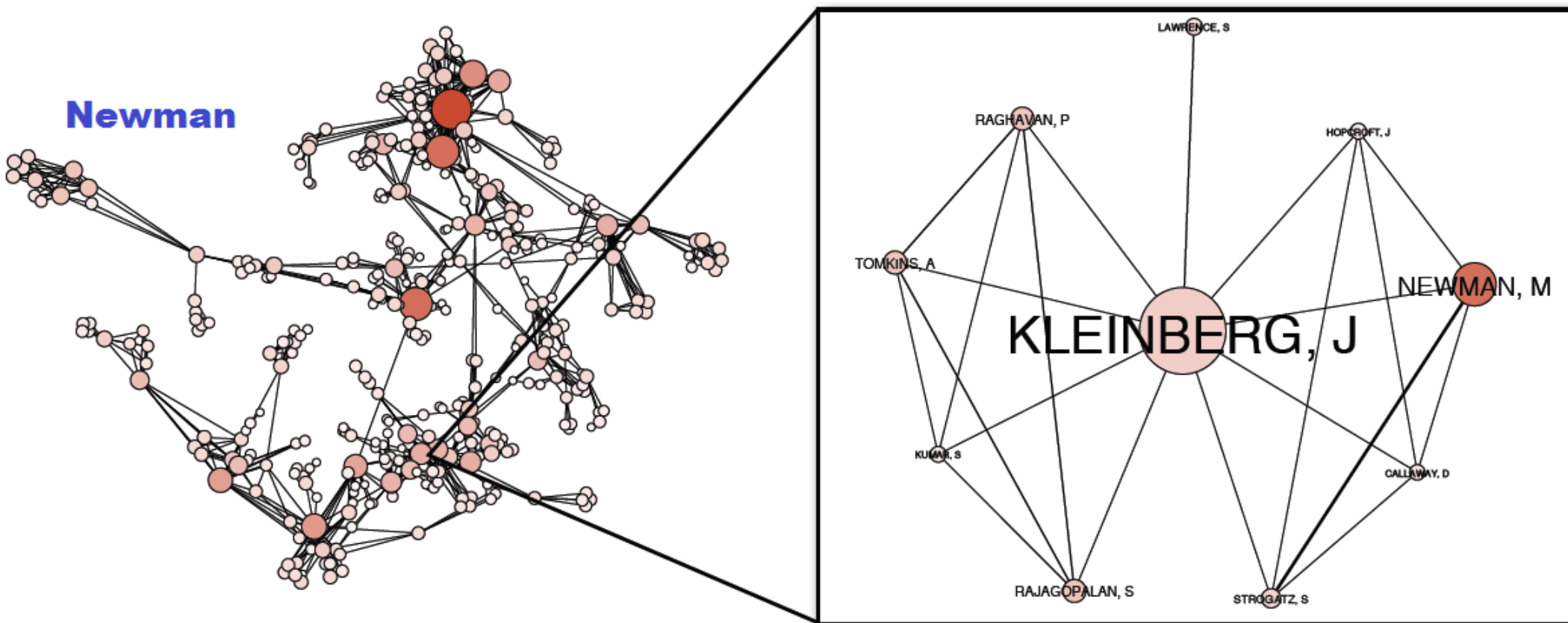


Competition Details » [Get the Data](#) » [Make a submission](#)

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.

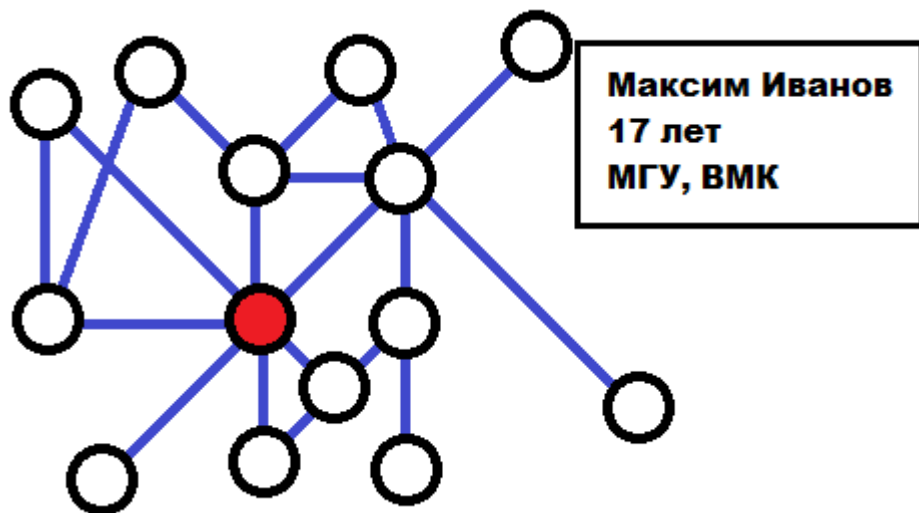
Эго-подграфы



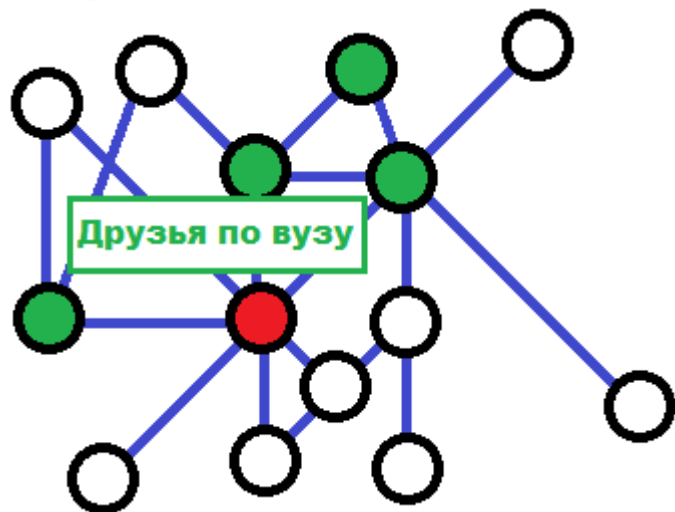
окрестность порядка 1

(не обязательно связный граф – без порождающей вершины)

Задача определения кругов



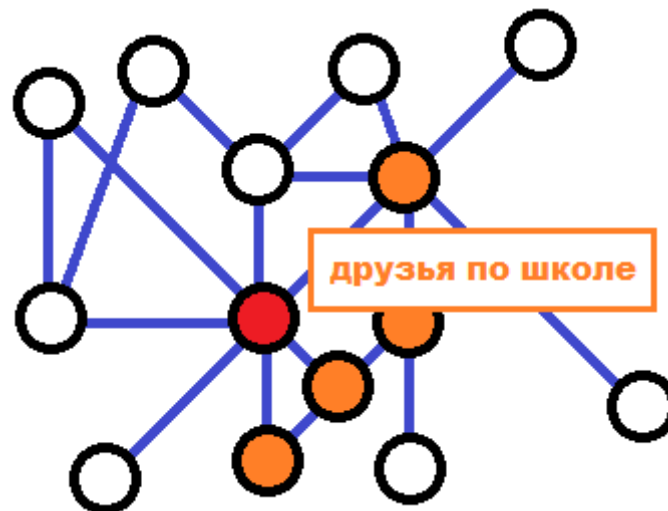
**Здесь: соцсеть =
граф + признаки вершин**



**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

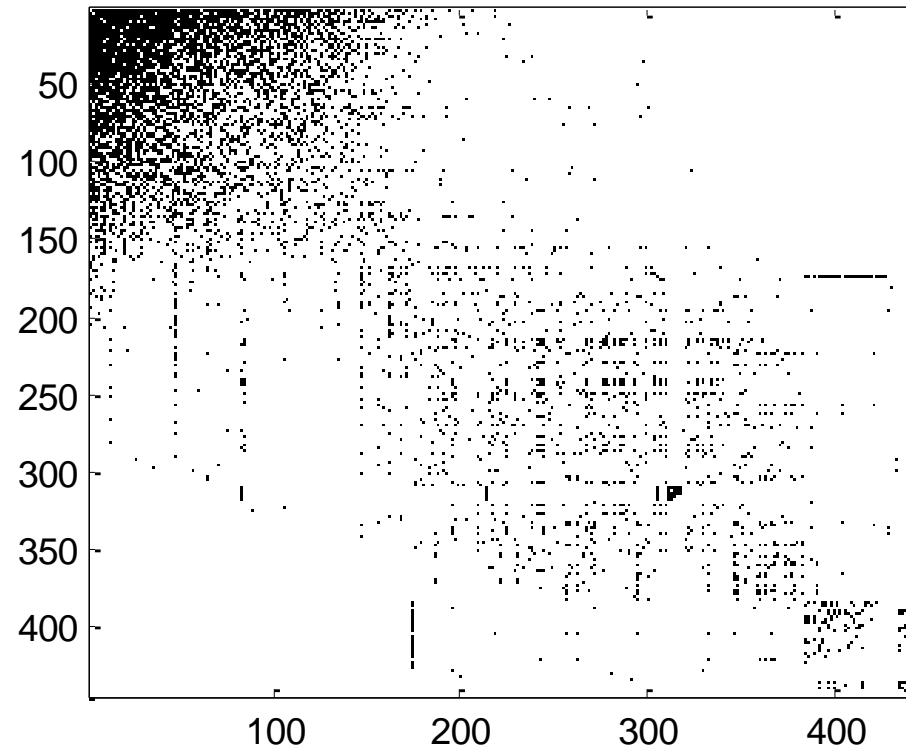


$$[U \ L \ V] = \text{svds}(M * M' + \text{alpha} * X * X', k_svd);$$

К сожалению, нет хорошего эффекта...

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

Оправдание алгоритма

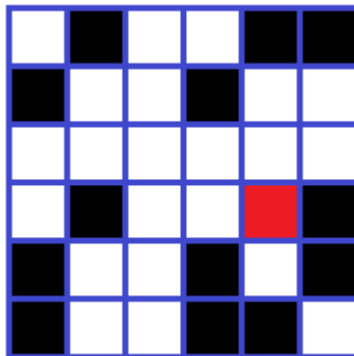


**Матрица смежности (упорядоченность вершин по первой компоненте)
действительно, есть факторизация**

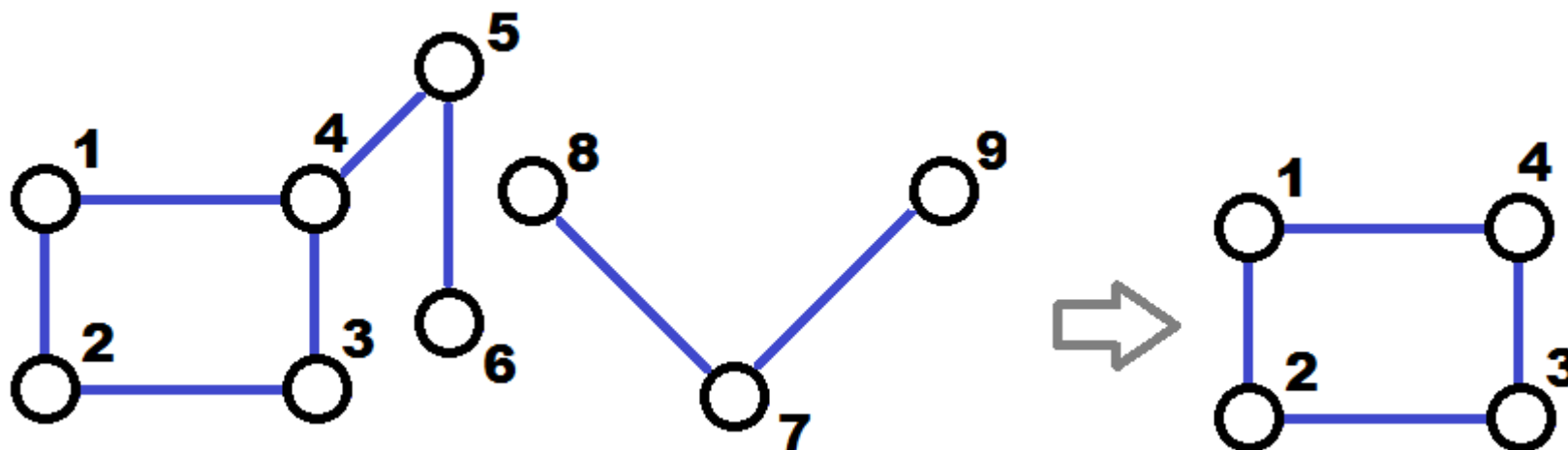
**Идея: ввести рейтинг принадлежности к компоненте – значение в
векторе сингулярного разложения**

Этапы алгоритма

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



2. Удаление висячих вершин

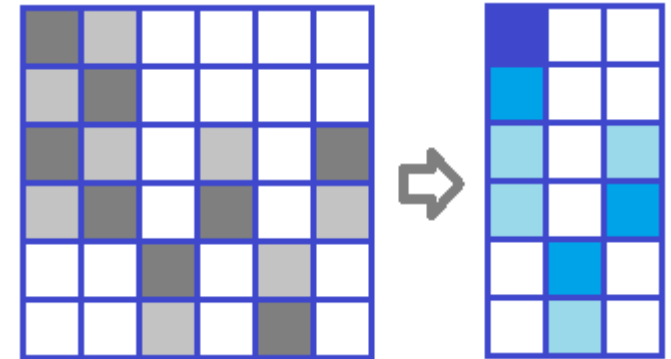


Этапы алгоритма

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

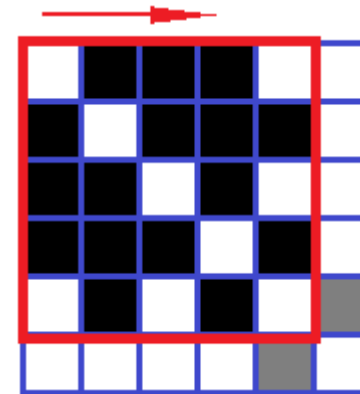
Идём по убыванию рейтинга, пока

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

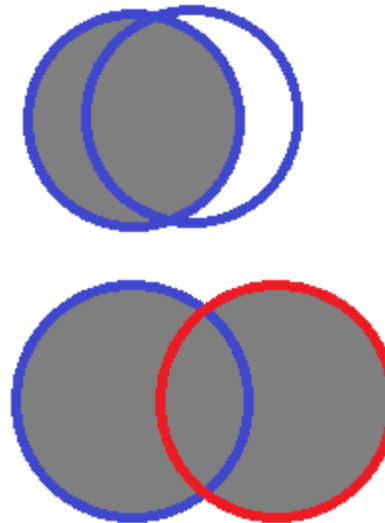
Если большое пересечение с уже существующим – не добавлять

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

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

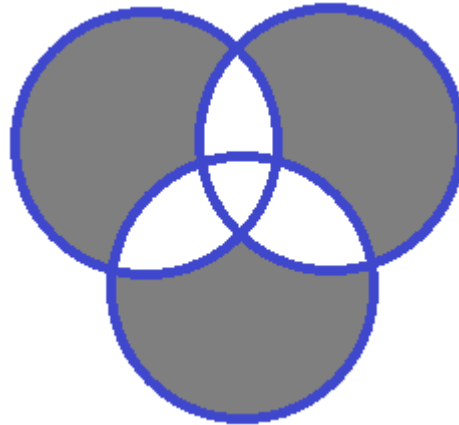
for j=1:length(anss)
    a = anss{j};
    p_jac = length(intersect(a,circle))/length(union(a,circle));
    if p_jac > p
        isadd = false;
        return;
    end
end

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



Этапы алгоритма

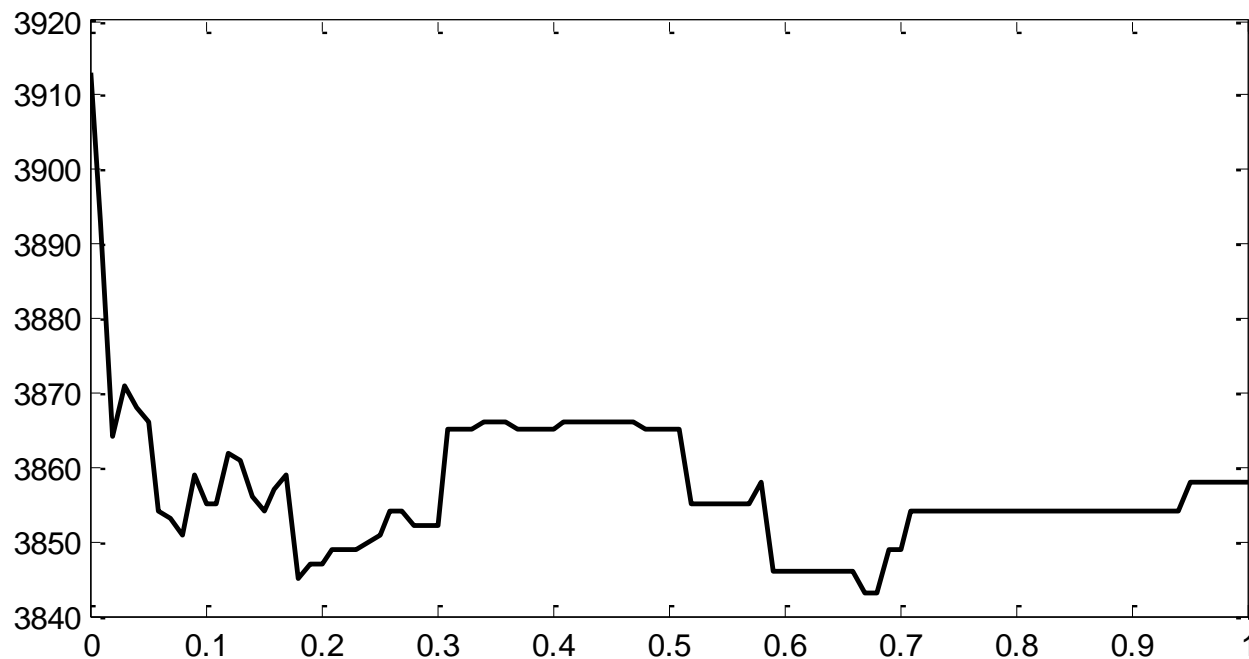
6. Удаление пересечений



следует из функционала качества

1) Настройка параметров

Типичная картинка



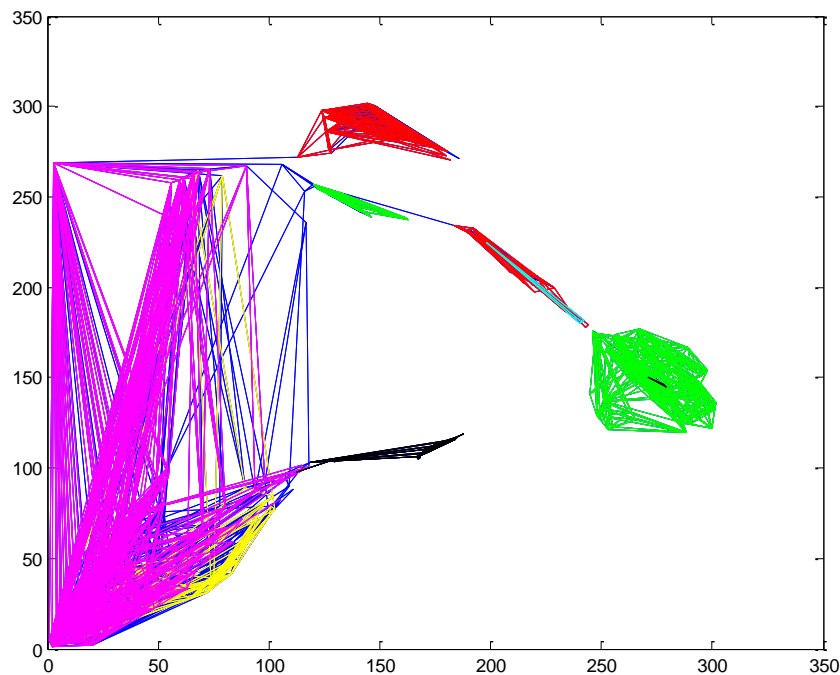
порог в добавлении кругов.

Уже по картинке видно:

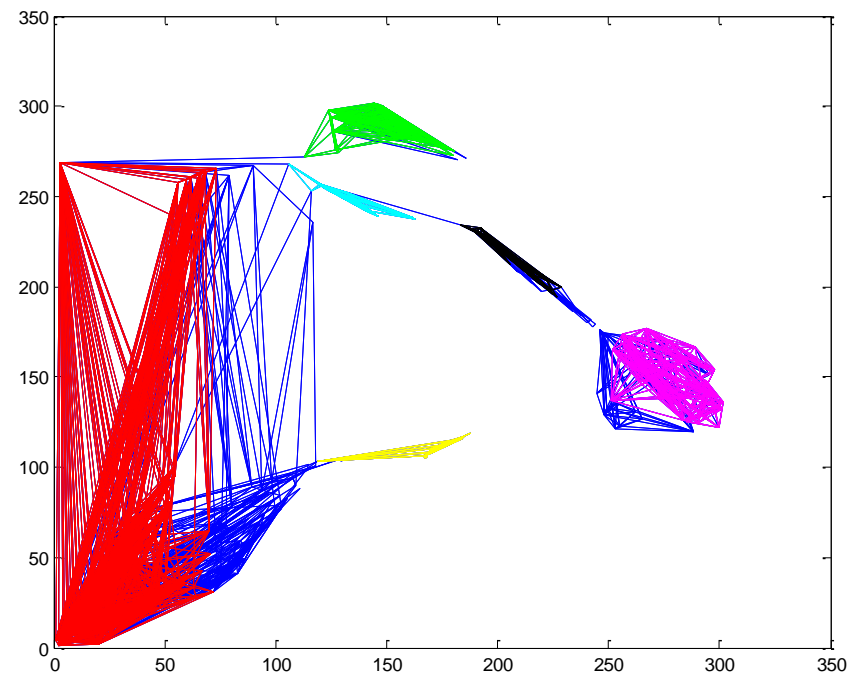
Мало статистики!!!

Работа алгоритма

Визуализация по 1й и 2й SVD-компоненте



правильный ответ

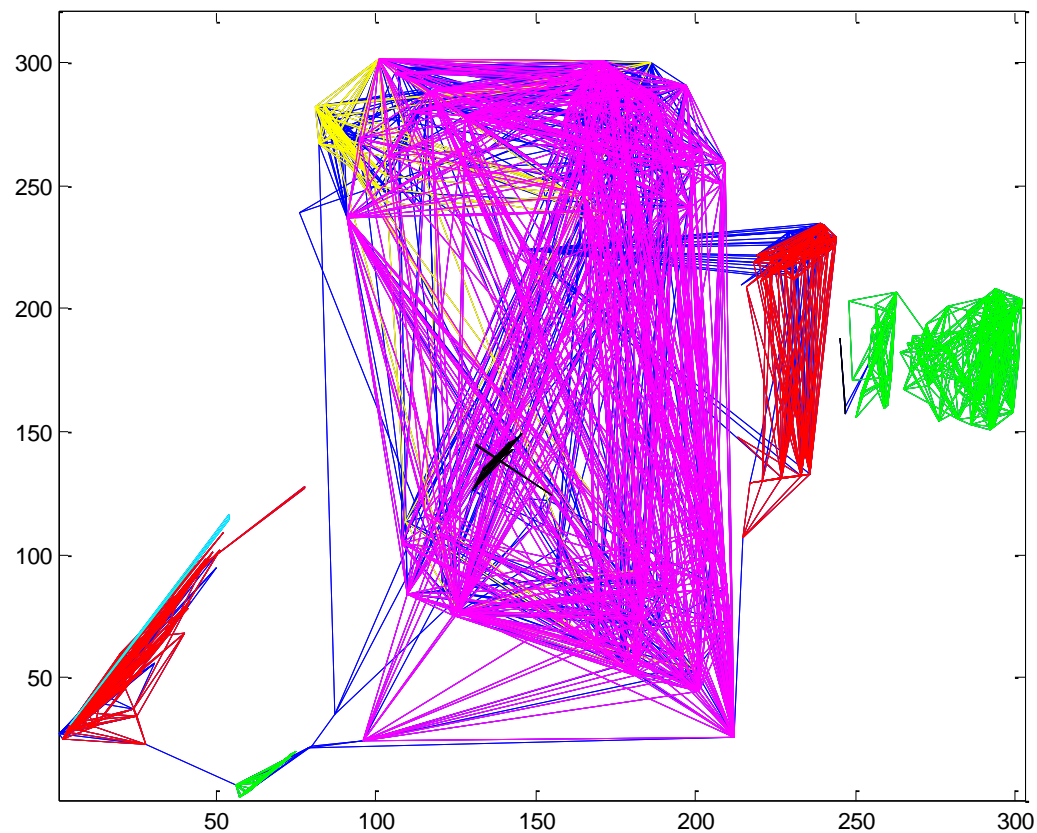


ответ алгоритма

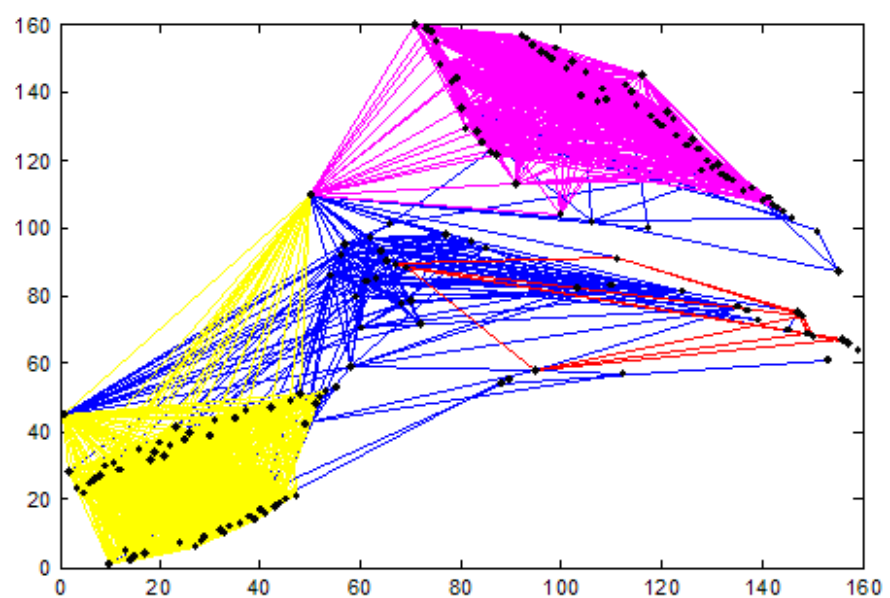
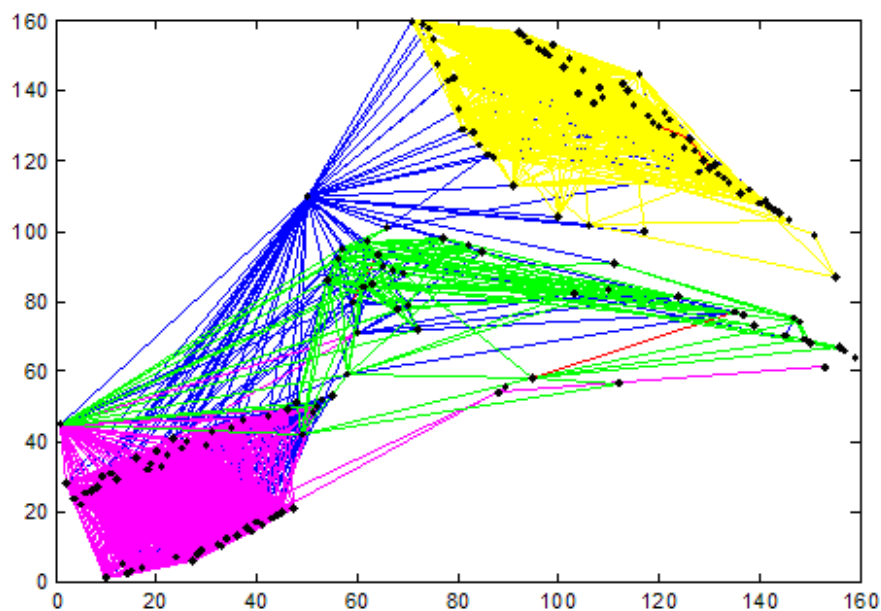
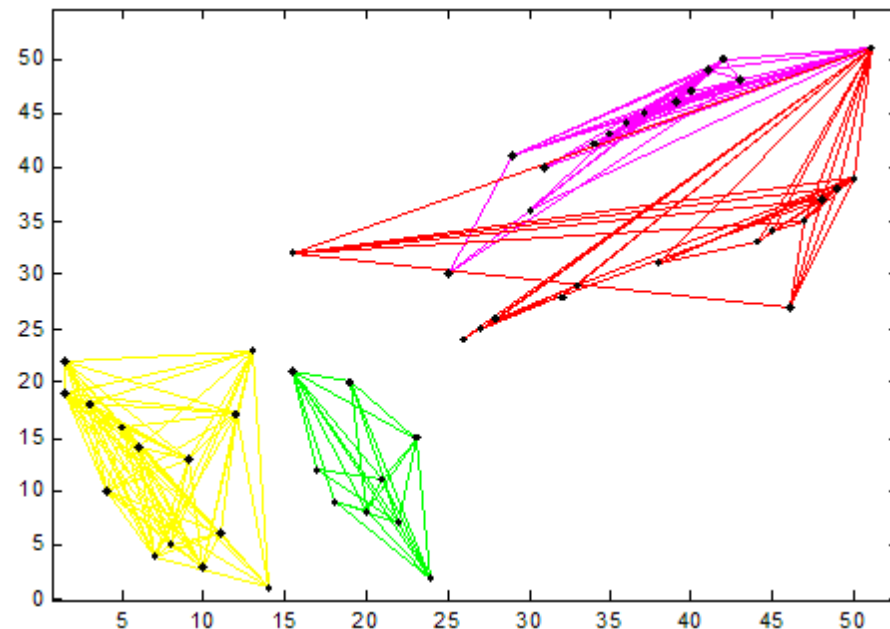
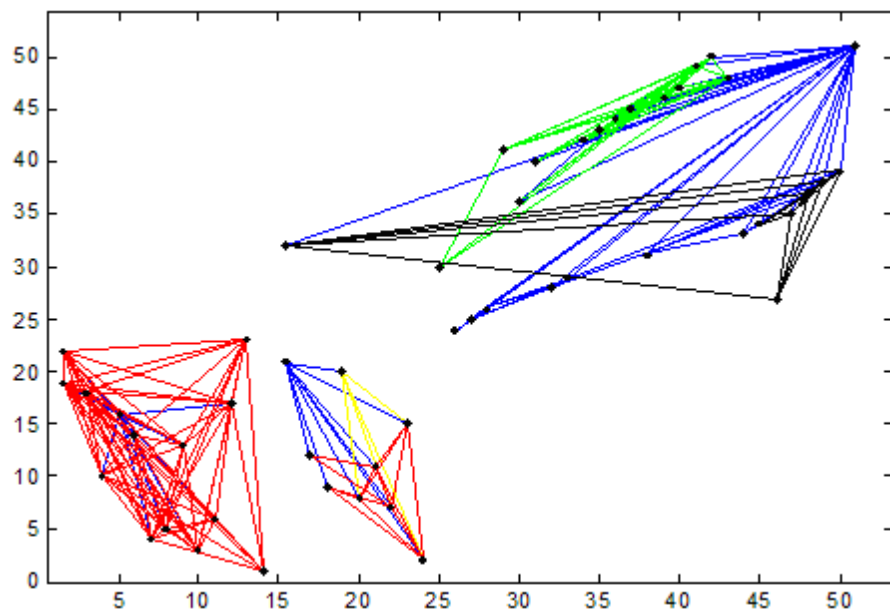
Хитрость: координаты – не значения компонент, а tiedrank...

Работа алгоритма

Визуализация по 3й и 4й SVD-компоненте

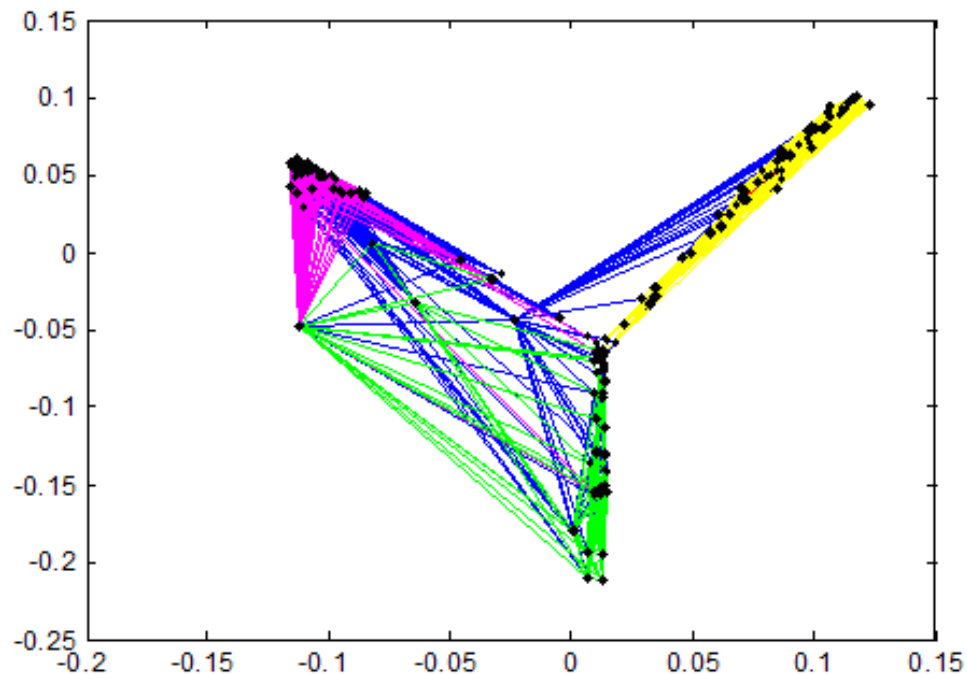


Работа алгоритма



MDS

Можно проецировать граф на плоскость с сохранением расстояний



Но получается не очень информативно

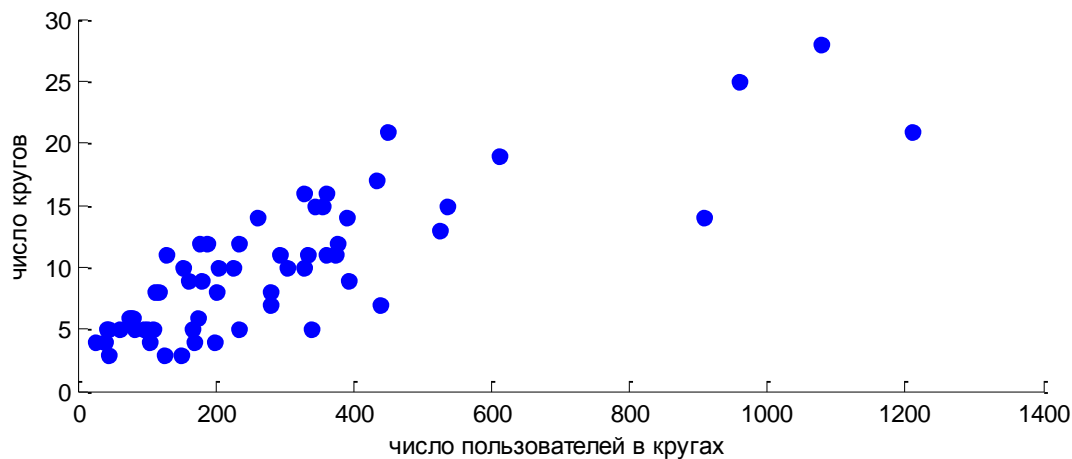
Что можно было сделать ещё...

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)

Что полезно

igraph – The network analysis package

<http://igraph.org/>

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

<http://networkx.lanl.gov>

3

Gephi: Java interactive visualization platform and toolkit

<http://gephi.org>

Л.Жуков курс Structural Analysis and Visualization of Networks в ВШЭ

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