Прикладные задачи анализа данных

АНАЛИЗ СОЦИАЛЬНЫХ СЕТЕЙ
SOCIAL NETWORK ANALYSIS
ЛЕКЦИЯ №2

Дьяконов А.Г.

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

Добавка к лекции 1: для моделирования графов

motif (мотиф – кирпичик)

небольшой граф, слишком часто встречающийся как подграф в нашем графе

Subgraph ratio =
число вхождений / число вхождений в случайном графе

Subgraph concentration =
число вхождений / число вхождений всех подграфов такого размера

Добавка к лекции 1: эволюция графов

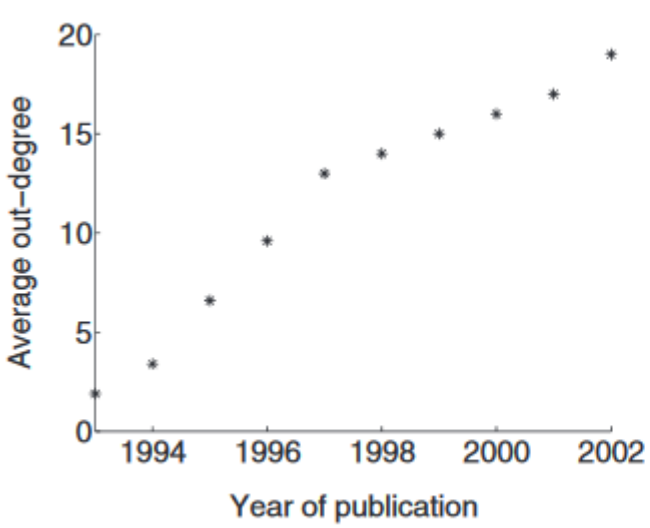
Динамика изменений графов

граф	размер / история	описание
ArXiv Citation Graph	n=29555 e=352807 01.1993 – 04.2003	2003 KDD Cup
Patents Citation Graph	n=3923922, e=16522438 01.1963(1975) – 12.1999	U.S. patent dataset
Autonomous Systems Graph	n≤6474, e≤26467 11.1997 – 01.2000	Граф коммункаций через интернет
Affiliation Graphs		arXiv ⇒ аффилиации (двудольный граф)

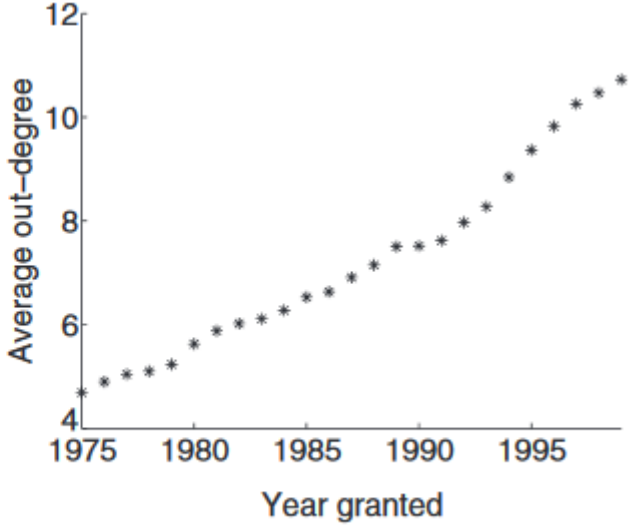
J. Leskovec, J. Kleinberg, C. Faloutsos «Graph evolution: Densification and shrinking diameters» ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1), 2

https://www.cs.cmu.edu/~jure/pubs/powergrowth-tkdd.pdf

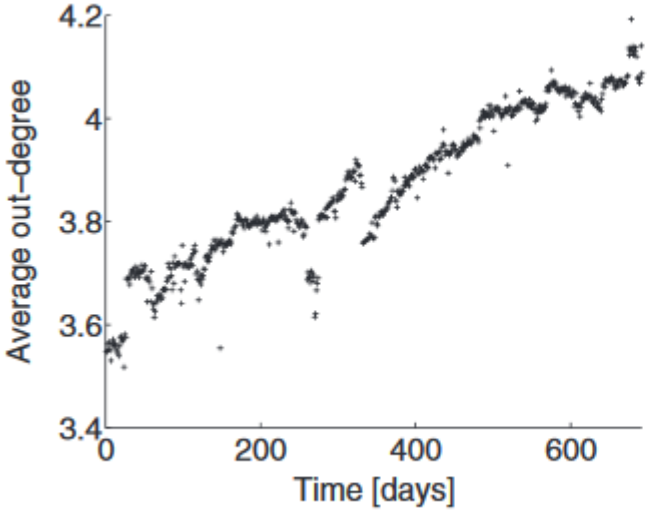
Добавка к лекции 1: эволюция графов



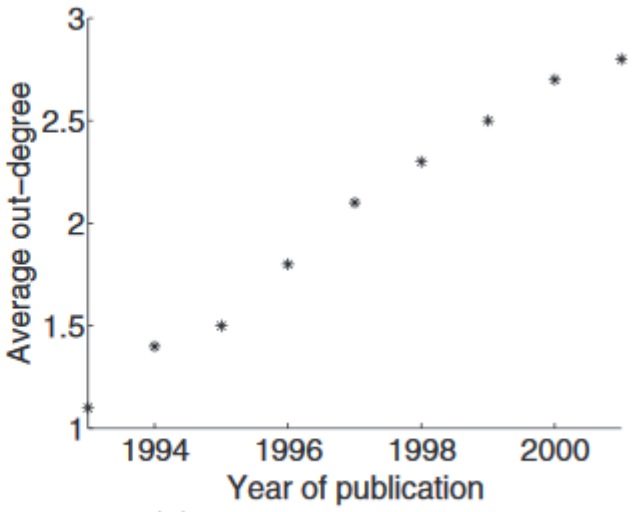
(a) arXiv



(b) Patents



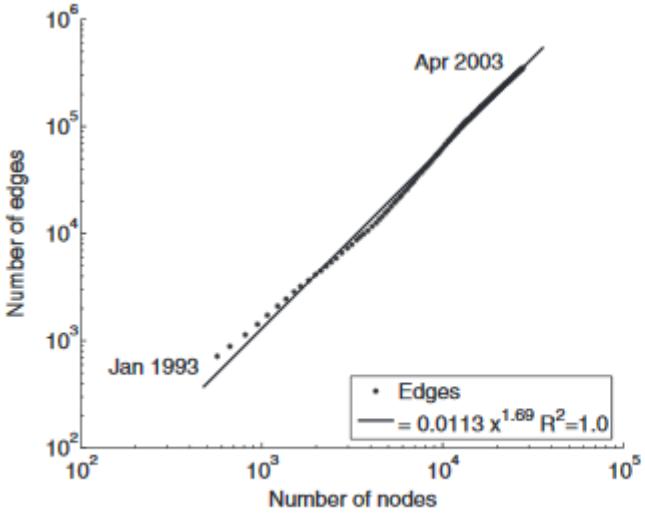
(c) Autonomous Systems



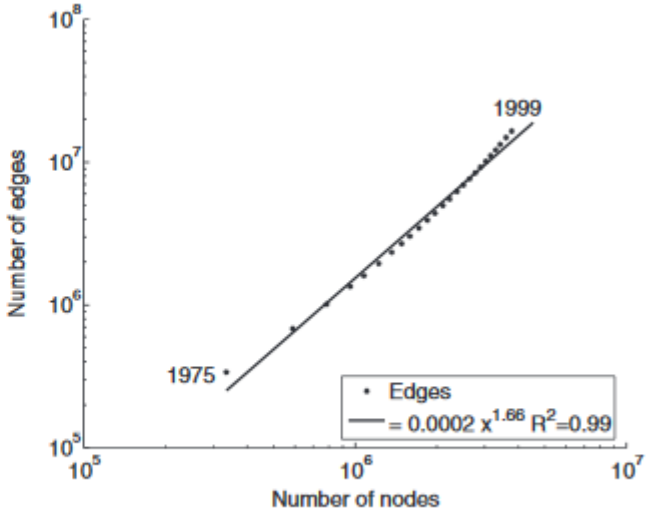
(d) Affiliation network

средняя степень вершин / время (считалось const)

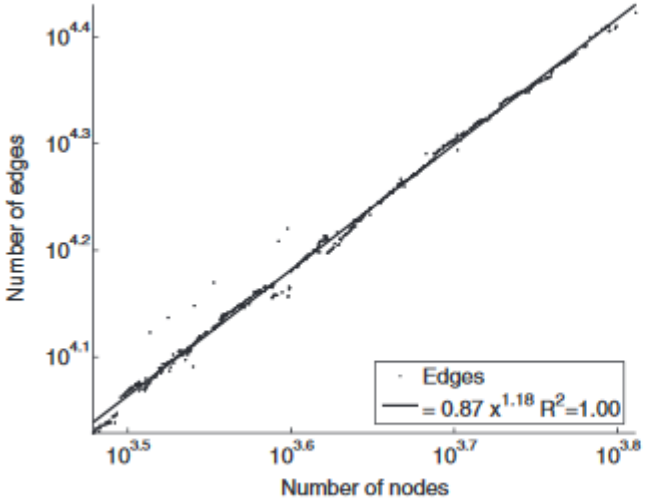
Добавка к лекции 1: эволюция графов



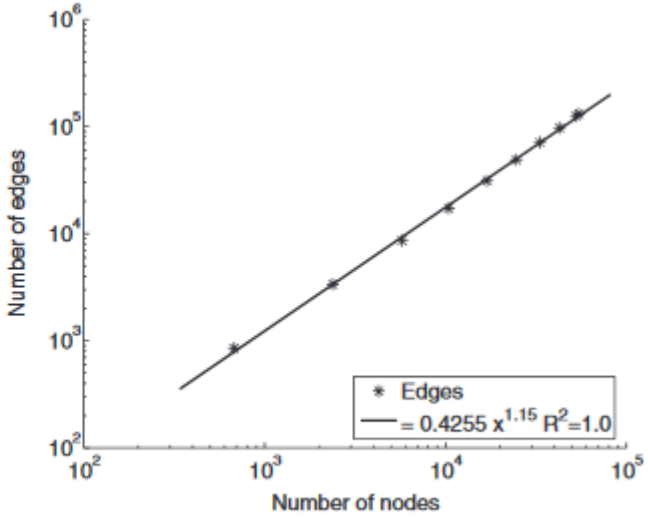
(a) arXiv



(b) Patents



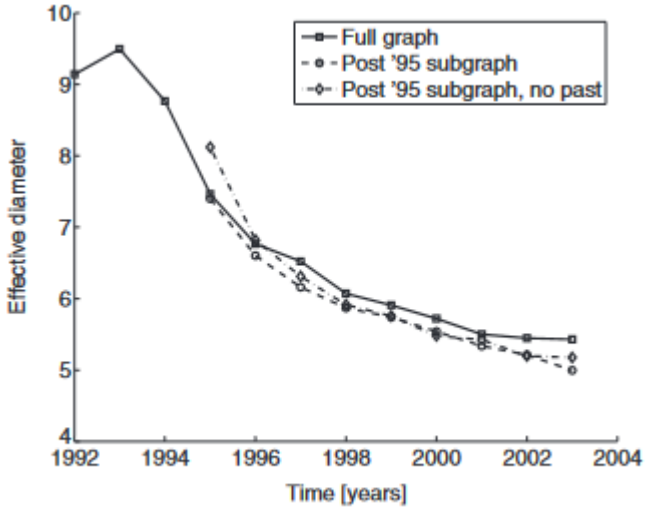
(c) Autonomous Systems



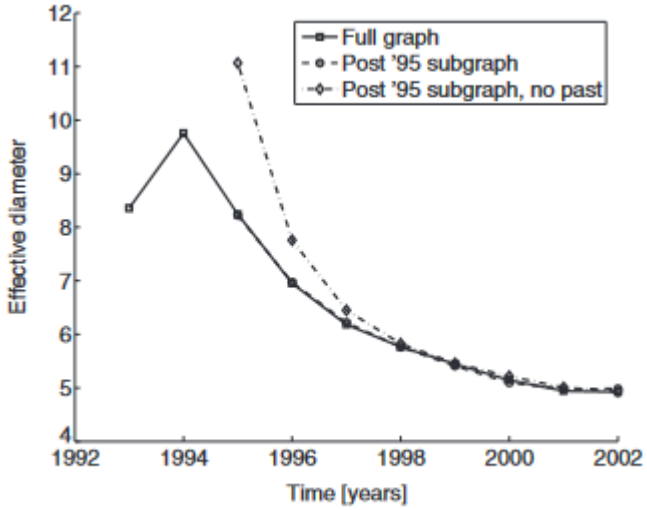
(d) Affiliation network

рёбра / вершины

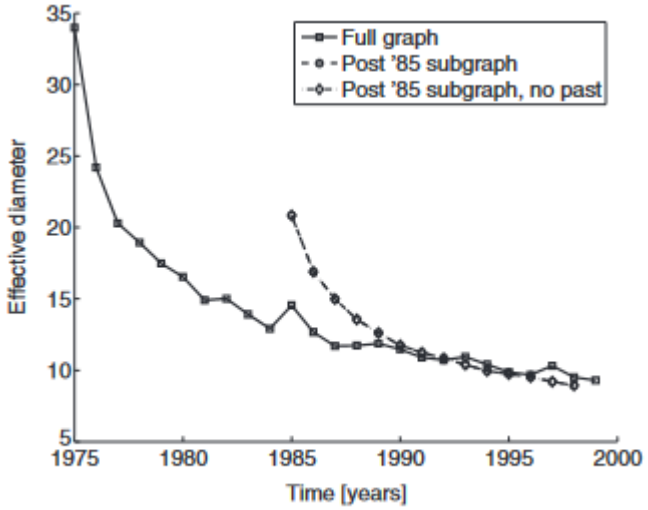
Добавка к лекции 1: эволюция графов



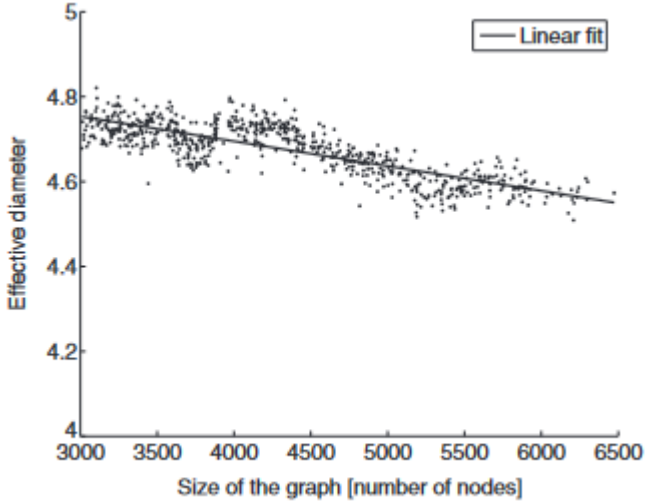
(a) arXiv citation graph



(b) Affiliation network



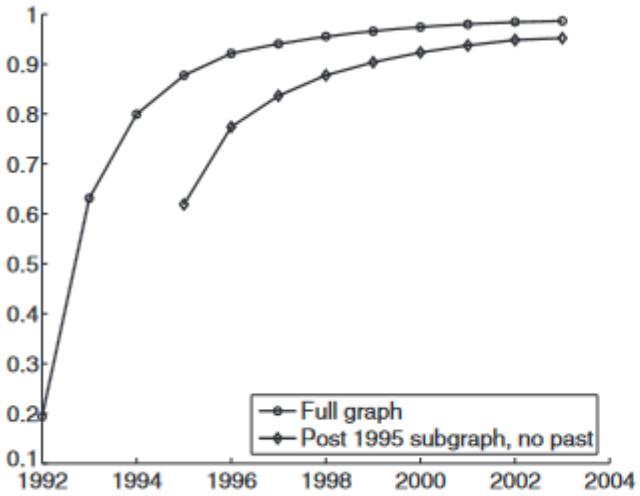
(c) Patents citation graph



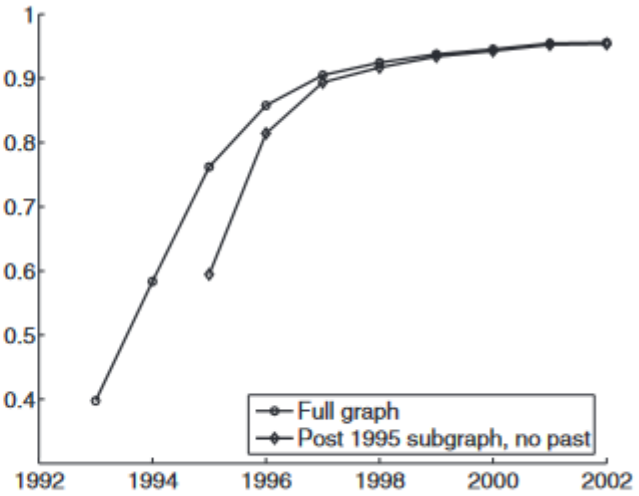
(d) Autonomous Systems

эффективный диаметр (90% на расстоянии не выше этого)

Добавка к лекции 1: эволюция графов



(a) arXiv citation graph



(b) Affiliation network

Доля вершин, входящих в большую компоненту связности «giant connected component»

В статье есть модели построения динамических графов «The Forest Fire Model»

есть сокращённая версия

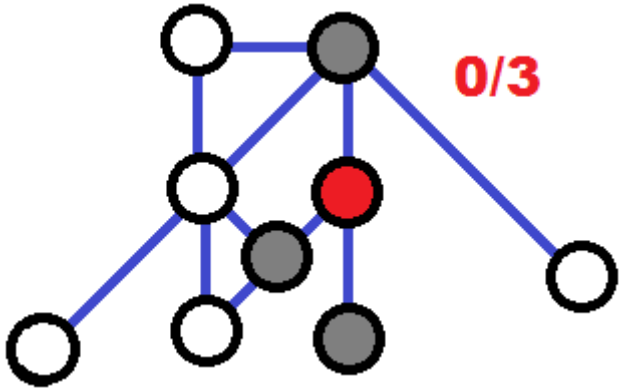
https://www.cs.cornell.edu/home/kleinber/kdd05-time.pdf

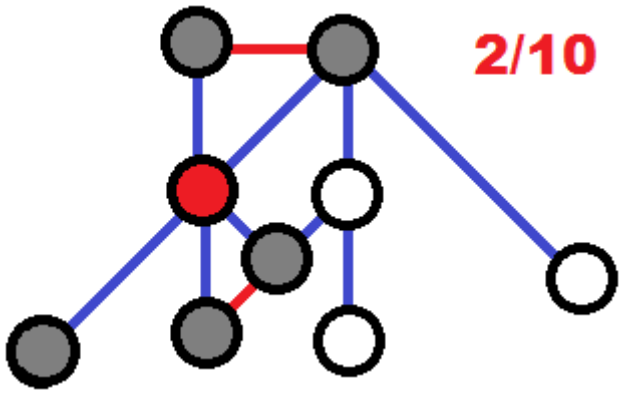
Часто графы просто погружают в признаковое пространство…

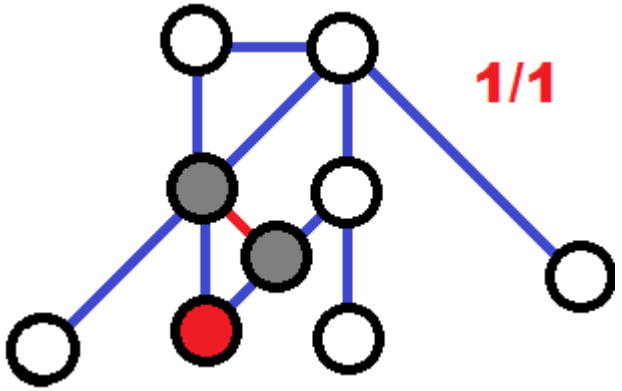
и граф превращается в вектор

Пример признака (уже был)

Коэффициент полноты (clustering coefficient)







что такое 10?

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! Любая статистика!

- 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

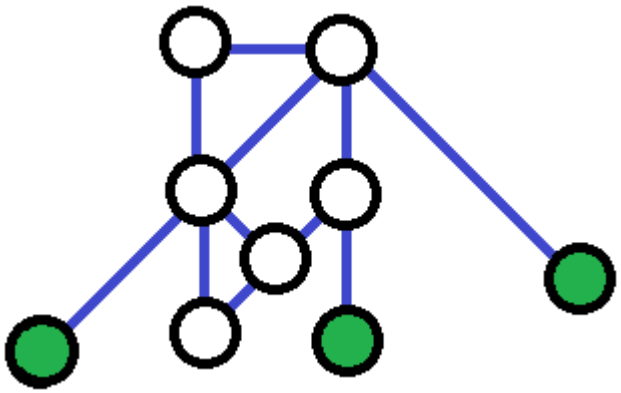
Часто надо измерить особенность вершины/ребра/подграфа

Например, для поиска непохожих вершин, влиятельных блогеров

Какие вершины считать «важными»?

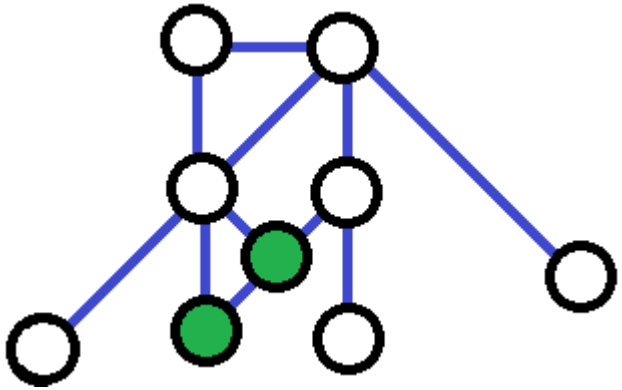
Сходство вершин

1. Формальная (по характеристикам)



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

2. По близости

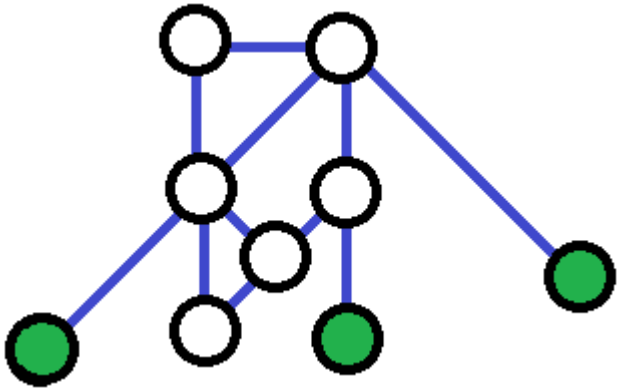


Два близких друга, близнецы

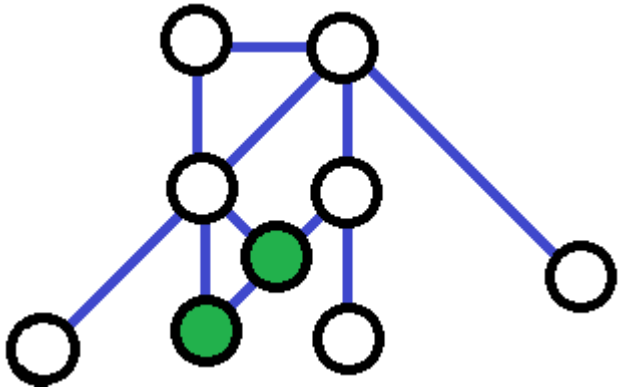
Как определить эти похожести на практике?

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**

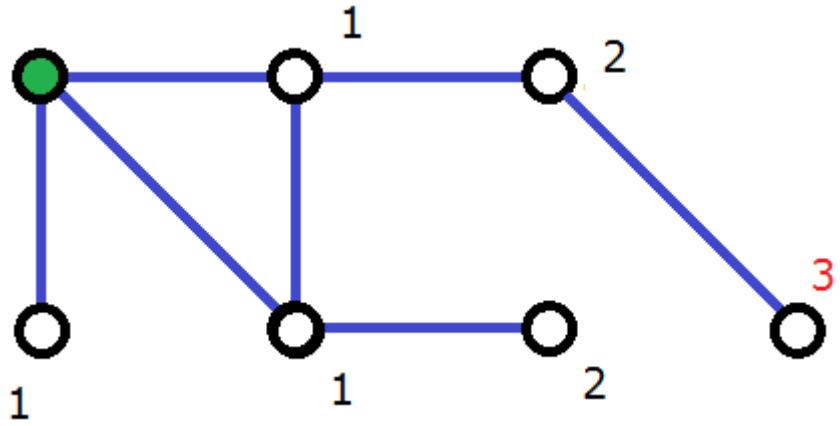
Это формализация отношений

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

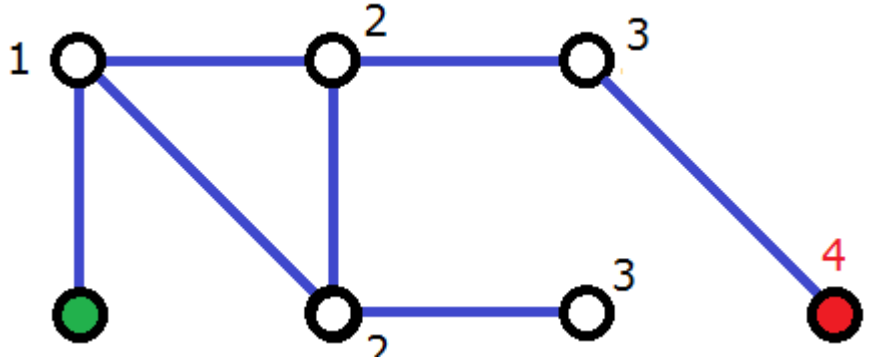
Центральность вершины в графе

Эксцентриситет – вершины v

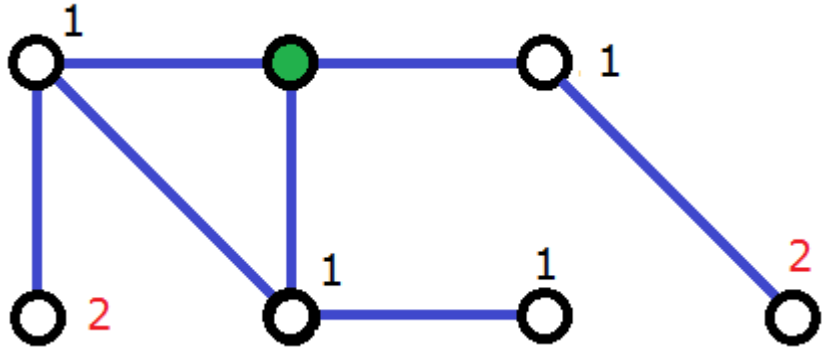
$$\varepsilon(v) = \max_{u \in V} d(u,v)$$



Диаметр – максимальный эксцентриситет



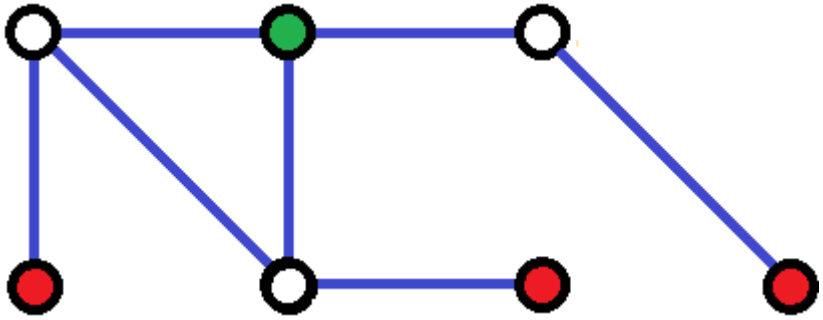
Радиус – минимальный эксцентриситет



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

$$d_{\text{out}} = A\tilde{1}$$
$$d_{\text{in}} = A^{\text{T}}\tilde{1}$$
$$a_{ij} \sim (i \to j)$$

ij-й элемент ~ дуга из i в j

Быстрое вычисление: $O(1)$

Центральность по близости (Closeness centrality) –

$$\sum_{u \neq v} \frac{1}{d(u,v)}$$

нужны все попарные расстояния

алгоритм Дейкстра

$$O(n_v^2 \log n_v + n_v n_e)$$

предполагается связность графа

Центральность по путям (Betweenness centrality)

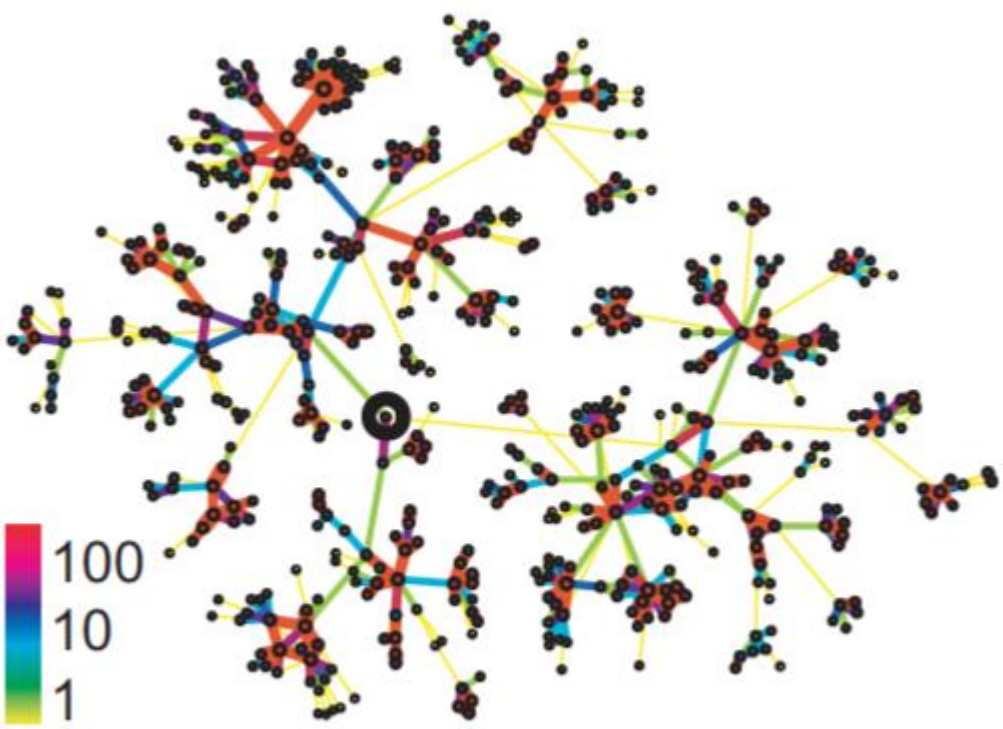
– число (доля) кратчайших путей, проходящих через эту вершину

Центральность ~ если ходить по графу,
то часто посещаешь эту вершину

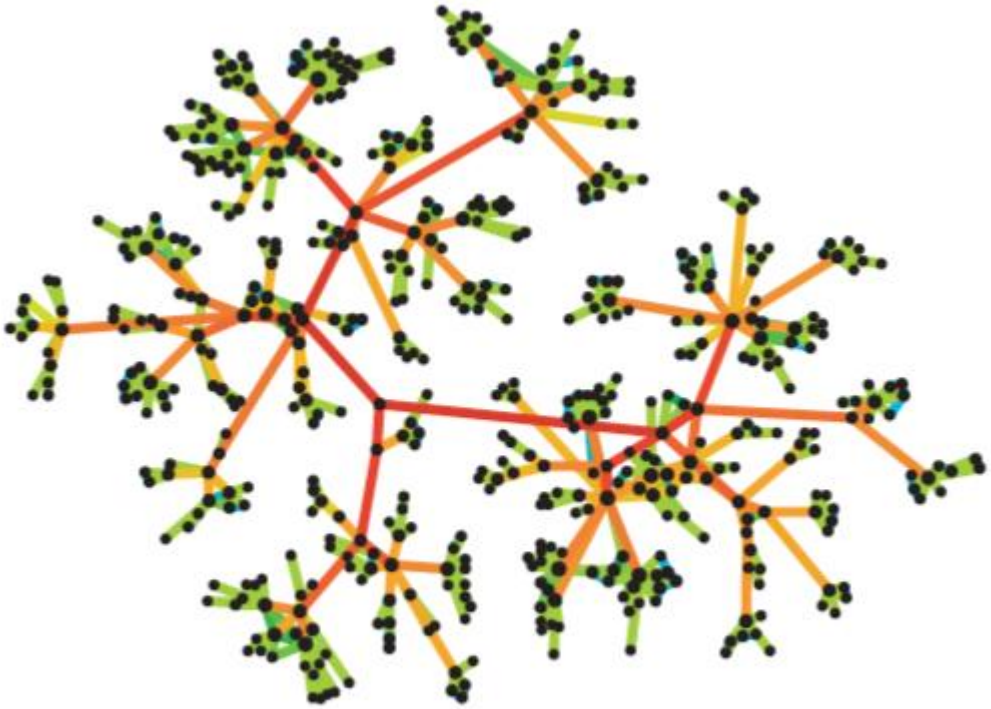
Есть $O(n_v n_e)$ алгоритм (для графа без весов)

U. Brandes «A faster algorithm for betweenness centrality» // Journal of Mathematical Sociology, vol. 25, no. 2, pp. 163-177, 2001

Центральность по путям (Betweenness centrality)



Edge strength



Edge betweenness

http://www2.ece.rochester.edu/~gmateosb/ECE442/Slides/block_4_sampling_modeling_inference_part_a.pdf

Собственная центральность (Eigenvector centrality) –

центральность вершины зависит от центральности соседей

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

$$N = D^{-1} A$$

$$N^{\mathrm{T}} x = x$$

max с.з. = 1

собственный вектор ~ max с.з.

Метод:

- **вычисление собственных векторов**
- **взятие вектора с максимальным собственным значением**
- **его значения – центральности вершин**

дальнейшая модификация ~ см. PageRank

Katz –

взвешенная сумма путей, приходящих в вершину

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

$$(\beta A + \beta^2 A^2 + \beta^3 A^3 + \ldots)\tilde{1} =$$
$$(\beta A + \beta^2 A^2 + \beta^3 A^3 + \ldots)(I - \beta A)(I - \beta A)^{-1}\tilde{1} =$$
$$(\beta A + \beta^2 A^2 + \beta^3 A^3 + \ldots - \beta^2 A^2 - \beta^3 A^3 - \ldots)(I - \beta A)^{-1}\tilde{1} =$$
$$\beta A (I - \beta A)^{-1}\tilde{1}$$

(тут если по-другому строить матрицу смежности)

На основе этого вычисляется центральность.

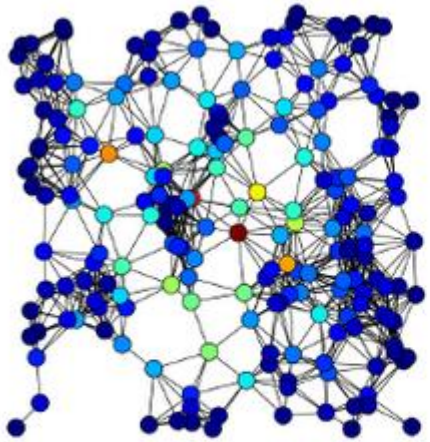
Эксцентриситетная центральность (Eccentricity centrality)

$$e(v) = \frac{1}{\max\limits_{u} d(u,v)}$$

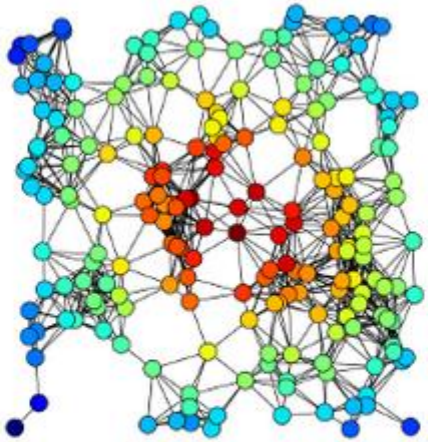
Сложность как в центральности по близости (Closeness centrality)

F.W. Takes and W.A. Kosters, Computing the Eccentricity Distribution of Large Graphs, Algorithms, vol. 6, nr. 1, pp. 100-118, 2013

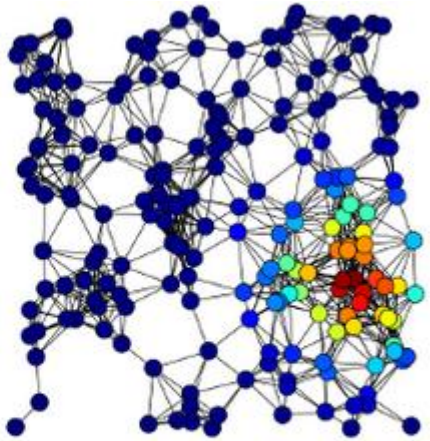
Разные виды центральности



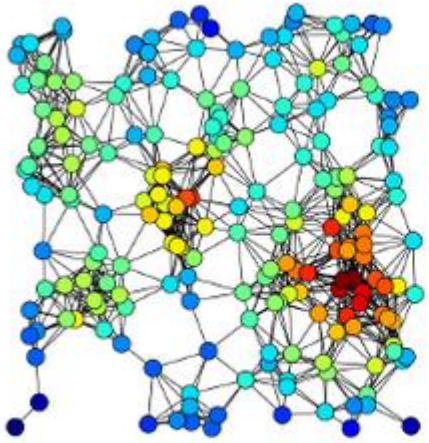
Betweennes centrality



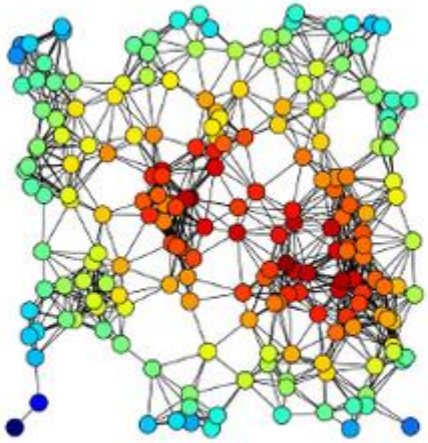
Closeness centrality



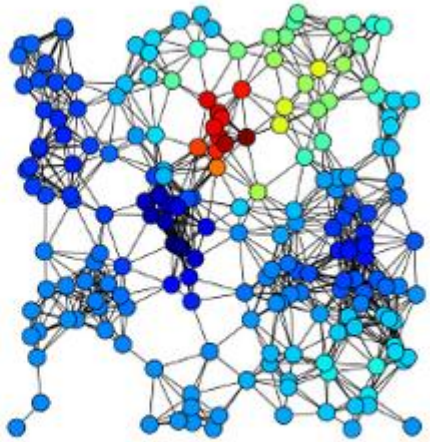
Eigenvector centrality



Degree centrality

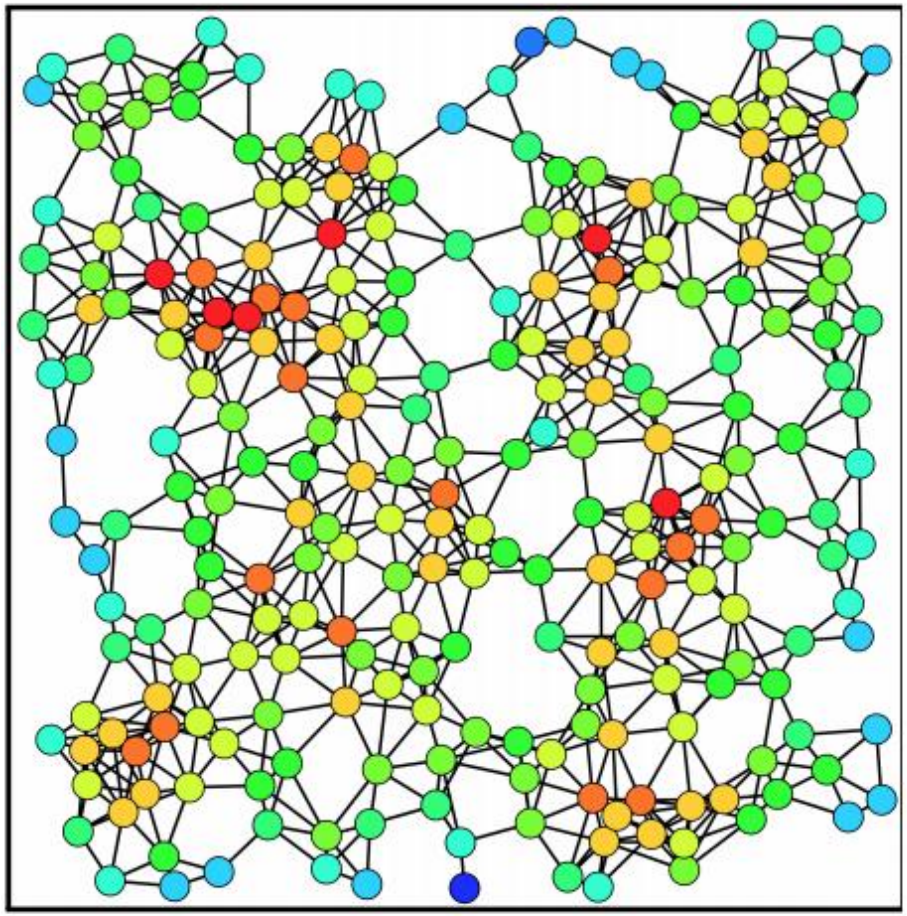


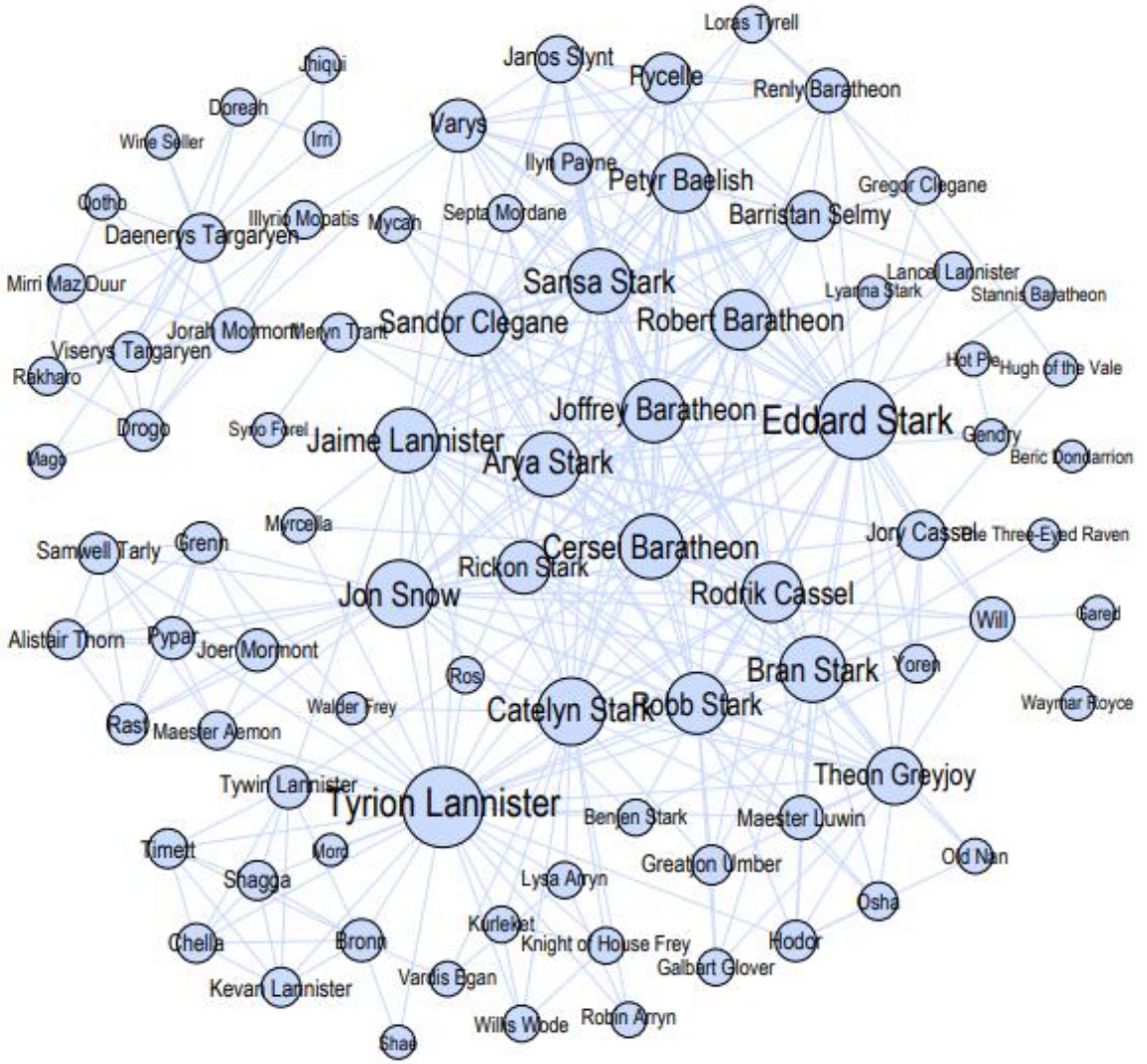
Harmonic centrality



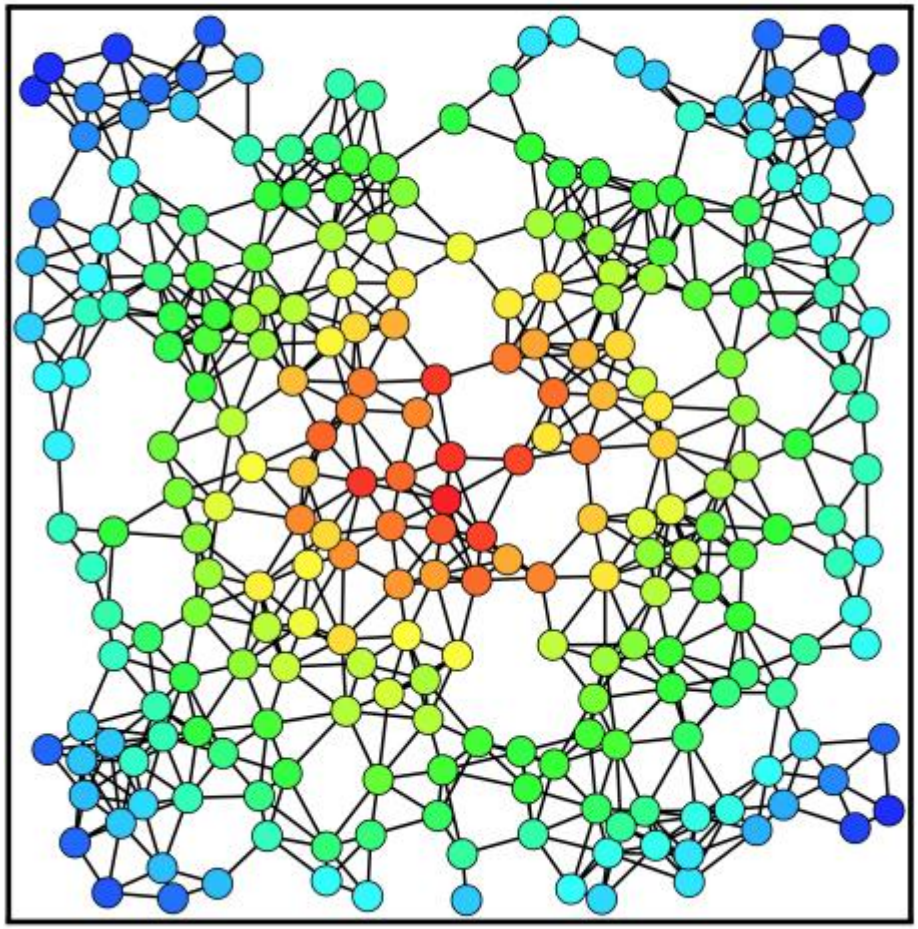
Katz centrality

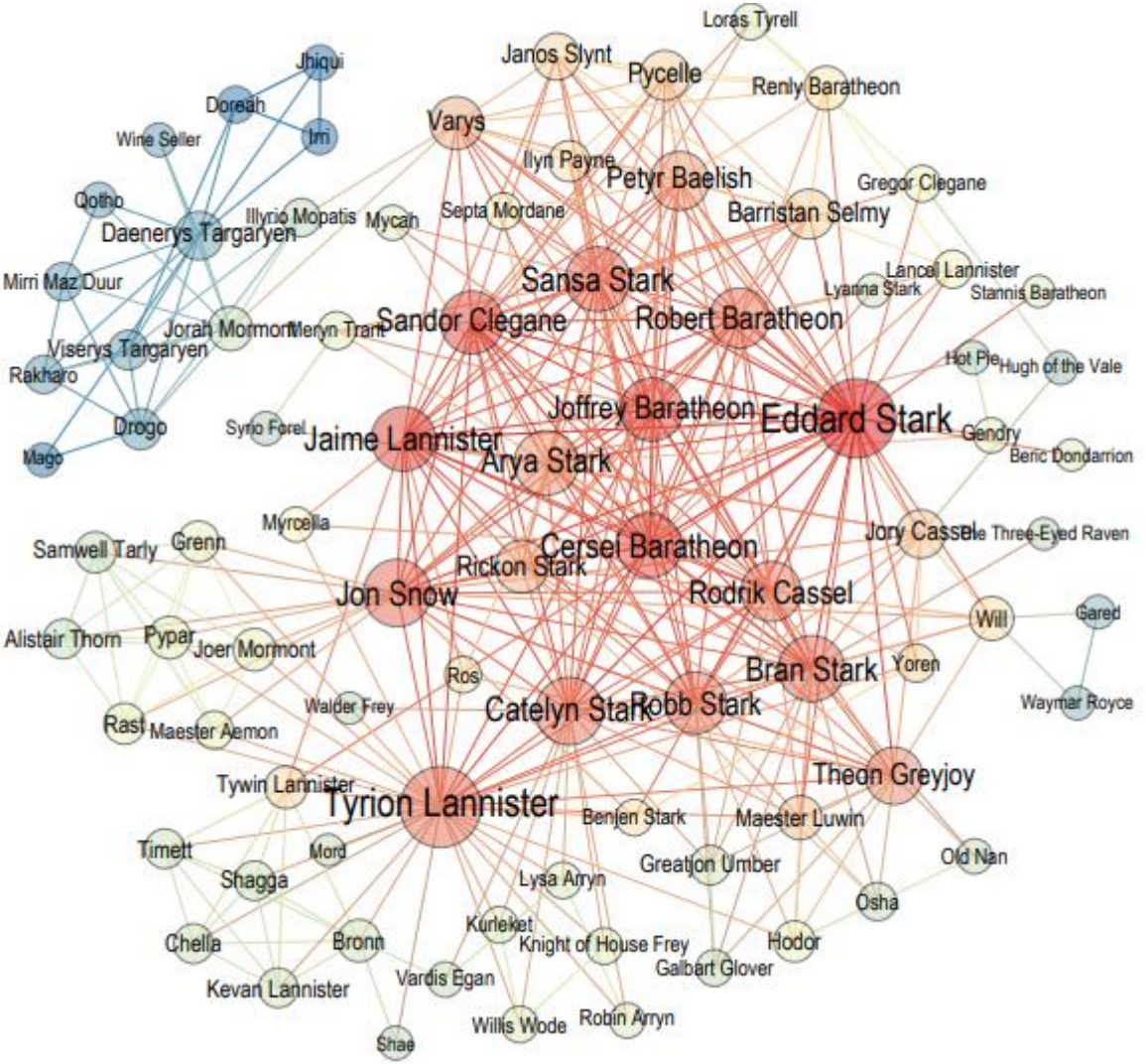
Degree Centrality



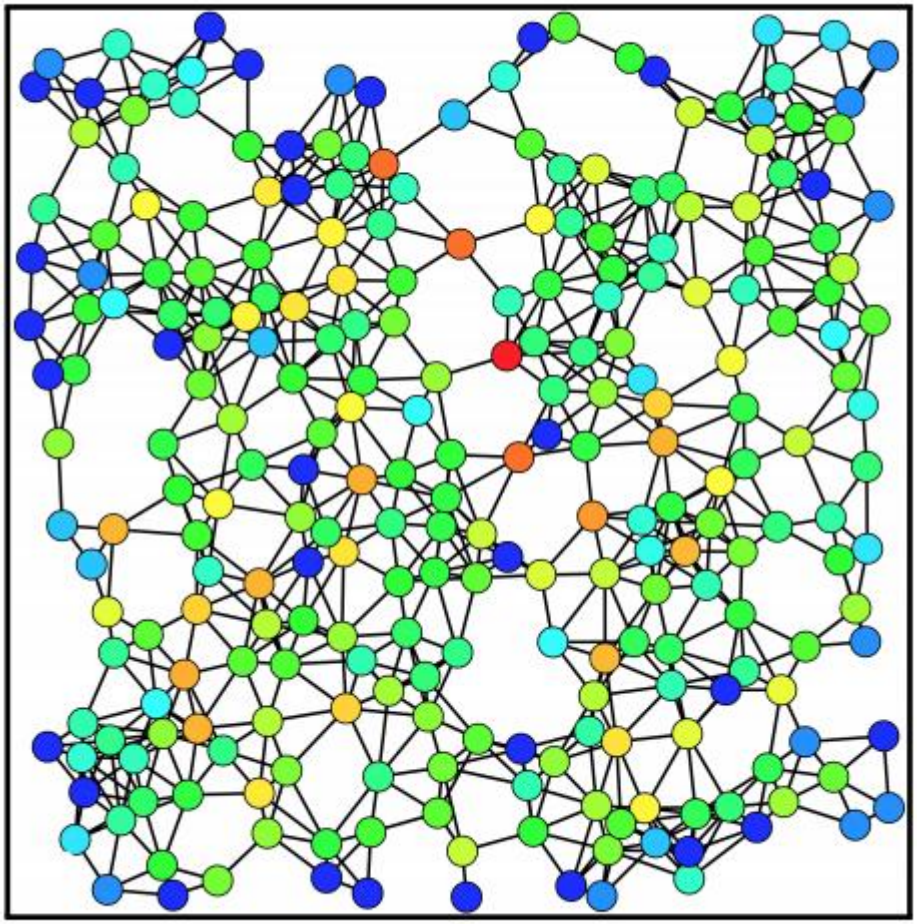


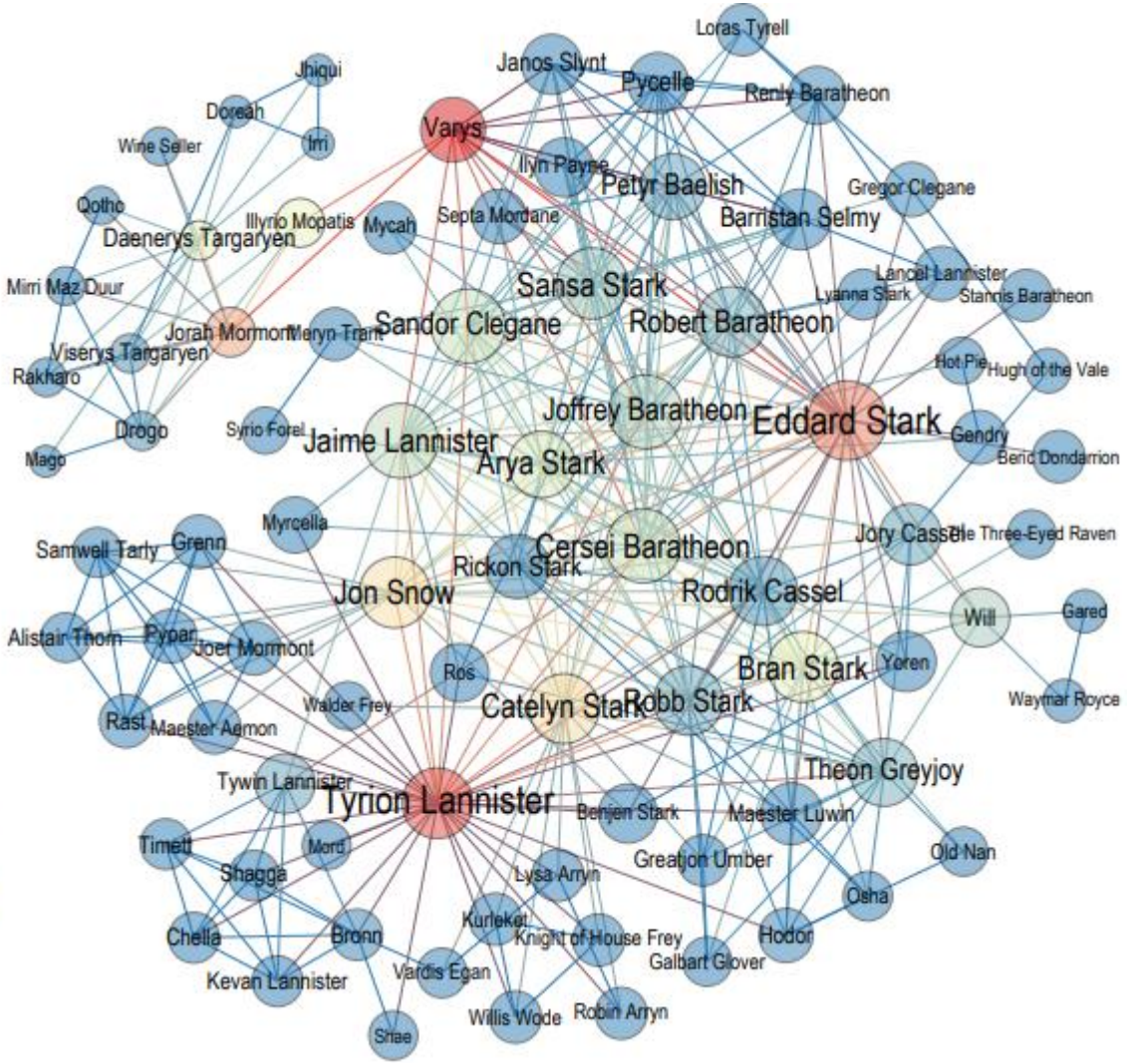
Closeness Centrality





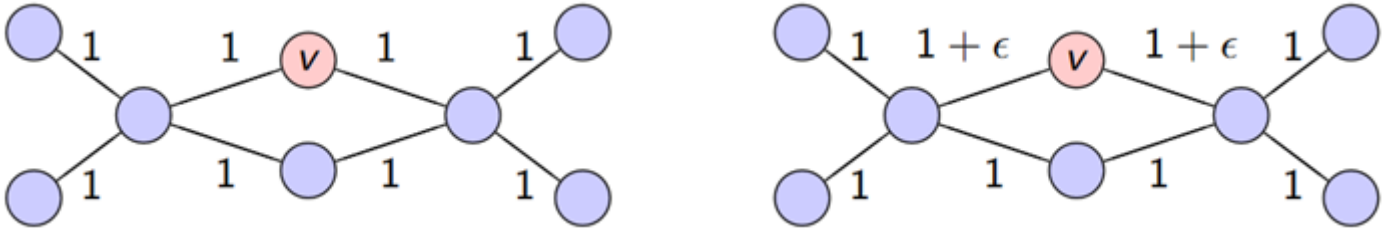
Betweeness centrality



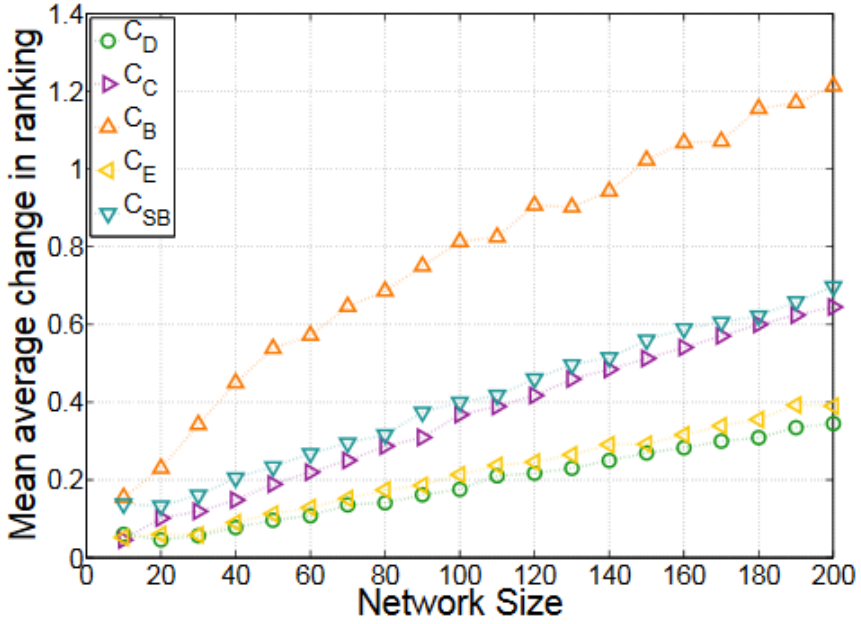


Устойчивость понятий

betweenness centrality неустойчива



есть устойчивые модификации…



Сравниваются ранки в исходном и чуть подпорченном графе (веса рёбер умножаются)

D = degree

C = closeness

B = betweenness

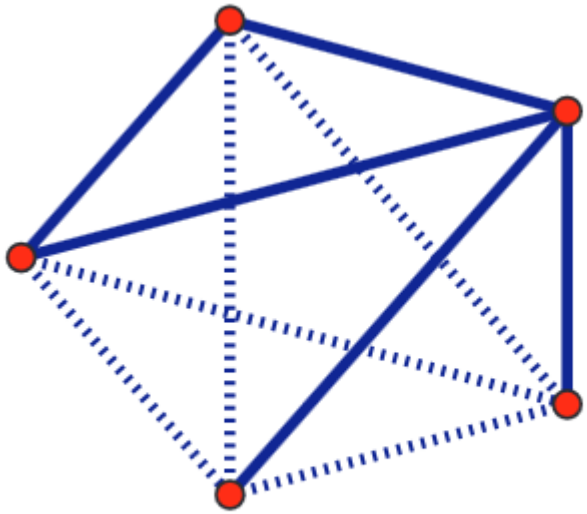
SB = stable betweenneess

S. Segarra and A. Ribeiro «Stability and continuity of centrality measures in weighted graphs» // IEEE Trans. Signal Process, 2015

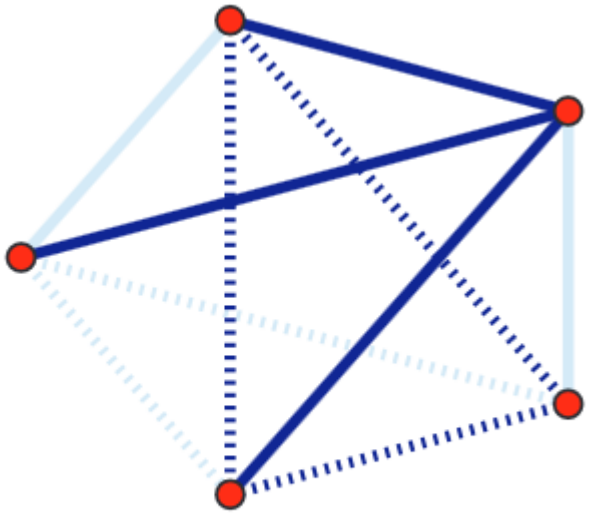
Прогнозирование появления ребра в динамическом графе (Link Prediction Problem)

Дан слепок графа соцсети.
Какие рёбра появятся в ближайшем будущем?

Чаще: для конкретных пар вершин «вероятность стать ребром»



Original graph



Link prediction

Liben-Nowell et. al. «The link-prediction problem for social networks» // J of American society for info science and technology. 2007

Прогнозирование появления ребра в динамическом графе (Link Prediction Problem)

Приложения:

социальные сети,

сотовые операторы,

мобильные операторы и т.д.

Как решать?

Для каждой пары (i, j) выпишем потенциально хорошие признаки

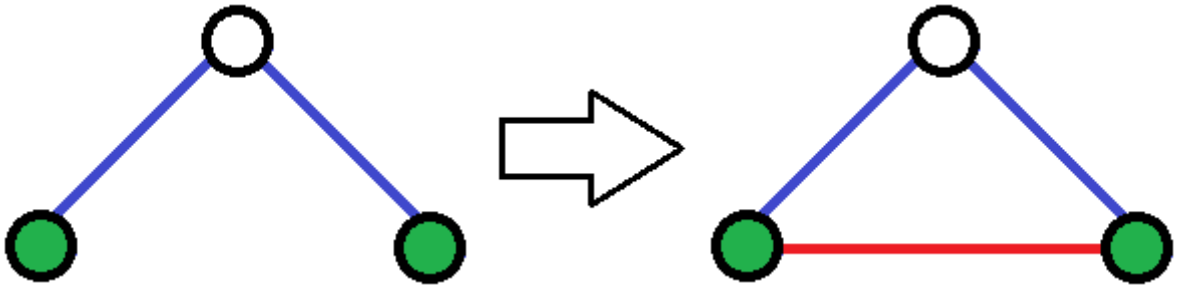
меры схожести вершин

– формирование признакового пространства

признак №0) расстояние на графе (graph distance)

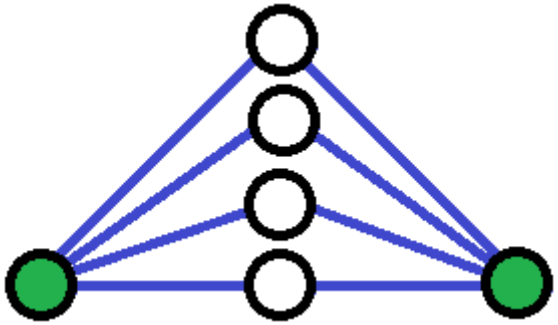
признак №1 – число соседей (common neighbors)

Принцип
«друг моего друга»



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

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

В его чём недостатки?

признак №2 – коэффициент предпочтительности

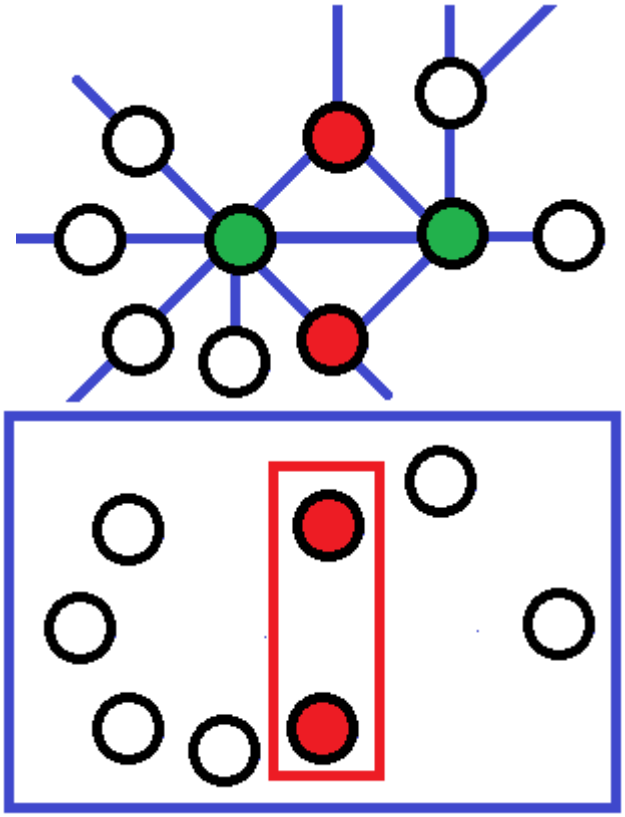
$|\Gamma(x)|\cdot|\Gamma(y)|$ **– коэффициент предпочтительности (preferential attachment)**

Чем более общительны, тем скорее подружатся

признак №3 – коэффициент Жаккара

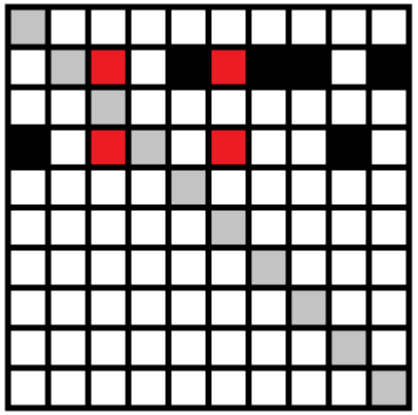
Или наоборот: чем больше процент общих друзей

$$\frac{|\Gamma(x)\cap\Gamma(y)|}{|\Gamma(x)\cup\Gamma(y)|}$$ – коэффициент Жаккара (Jaccard's coefficient)



обычные признаки
для сравнения множеств

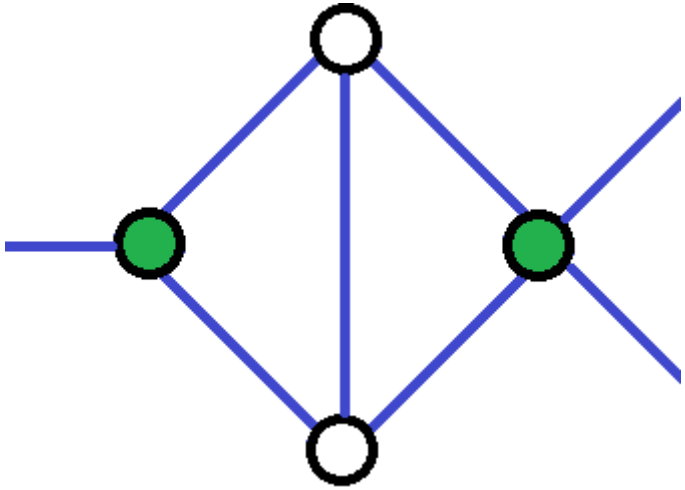
просто сравнение строк матрицы смежности

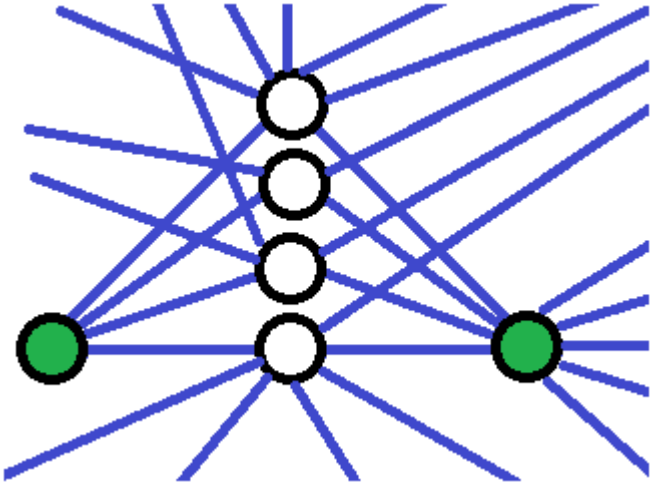


Полезно: разный подход к описанию смысла (множества, строки)

признак №4 – коэффициента Адамик/Адара

не все друзья одинаковые!

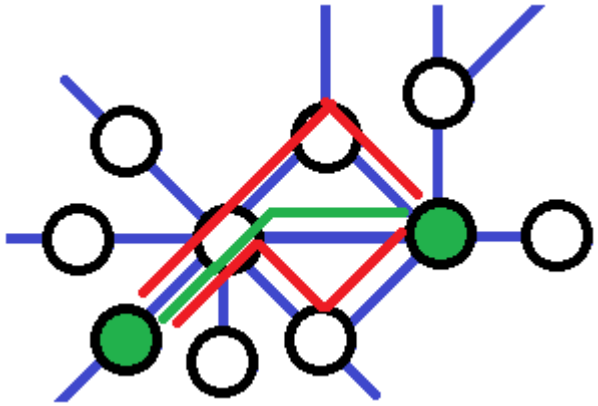




$$\sum_{z\in\Gamma(x)\cap\Gamma(y)}\frac{1}{\log|\Gamma(z)|}$$ **– коэффициента Адамик/Адара (Adamic/Adar)**

признак №5 – Katz

Учитывать целые цепочки друзей-друзей



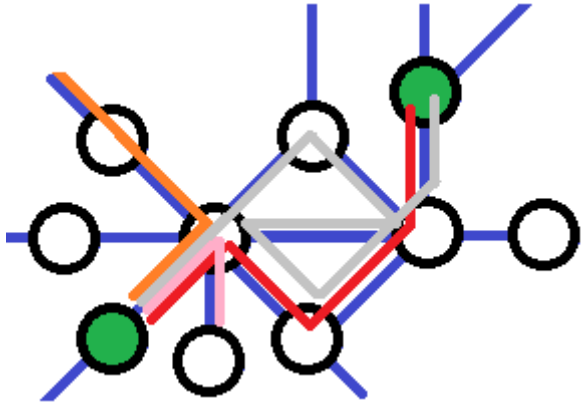
$$\sum_{l=1}^{\infty} \beta^l \, \mathrm{path}_l(x,y)$$ **– признак Katz**

равен xy-му элементу матрицы

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

Признаки №6 – 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

+ PageRank

Признаки №7 – на основе рекуррентных вычислений

SimRank

Вершины похожи,
если похожи их друзья

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

Разные итерации пересчёта можно сделать признаками!

Признаки №8 – вероятностные методы

Пусть вершина i порождается с вероятностью $\mathrm{P}(i)$

По ней порождается латентный класс с вероятностью $\mathrm{P}(z \mid i)$

По нему порождается ребро с вероятностью $\mathrm{P}(j \mid z)$

Вероятность появления ребра =

$$\mathrm{P}(i)\mathrm{P}(z \mid i)\mathrm{P}(j \mid z)$$

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

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

Алгоритм PageRank (подробнее про случайные блуждания)

Две эквивалентные интерпретации «что такое важные страницы в интернете»

I) Случайные блуждания

Если ходить по ссылкам в Интернете,
то важная страница – на которую чаще попадаешь

II) Перетекание рейтинга

«Важные»:
1. На них ссылаются (есть входящие ссылки)
2. На них ссылаются важные страницы

Алгоритм PageRank

Если страница j с важностью w_j имеет $d_{\text{out}}(j)$ выходных ссылок, каждая ссылка «передаёт» важность

$$\frac{w_j}{d_{\text{out}}(j)}$$

Важность страницы = сумма всех входных ссылок

$$w_j = \sum_{(i,j)\in E} \frac{w_i}{\deg_{\text{out}}(i)}$$

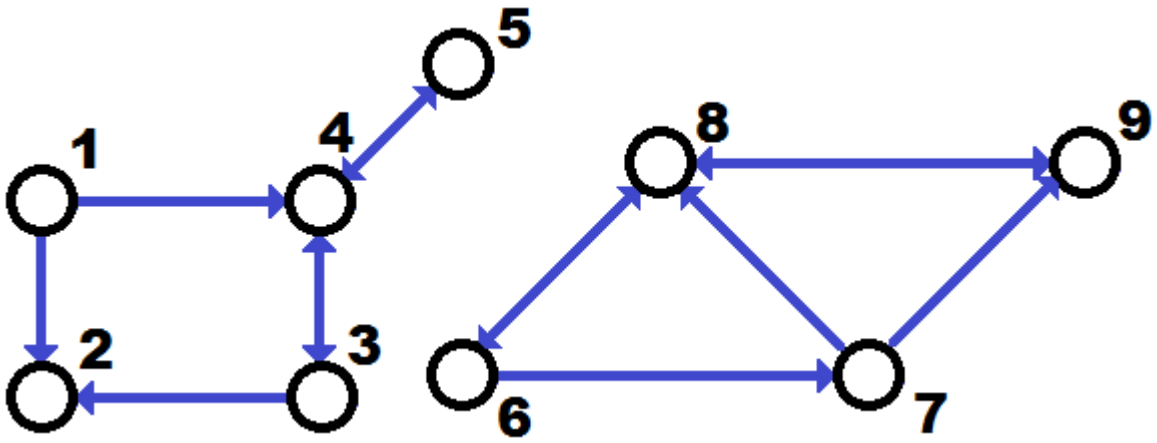
Если пронормировать матрицу смежности

$$N = D_{\text{out}}^{-1} A$$

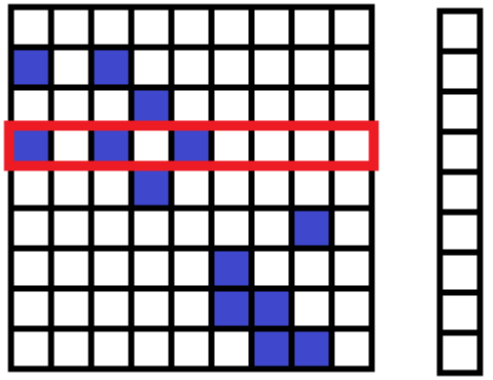
тогда вектор важности рекурсивно записывается как

$$w = N^{\text{T}} w$$

Алгоритм PageRank



тут транспонированная матрица



$$w_4 = \frac{w_1}{2} + \frac{w_3}{2} + \frac{w_5}{1}$$

Внимание на построение матрицы смежности!

Алгоритм PageRank

Решаем задачу на собственные значения

$$N^{\mathrm{T}} w = \lambda w$$

Наибольшее с.з. = 1

Берём его собственный вектор!

Итерационный метод

$$w^{(t)} = N^{\mathrm{T}} w^{(t-1)}$$

это и находит

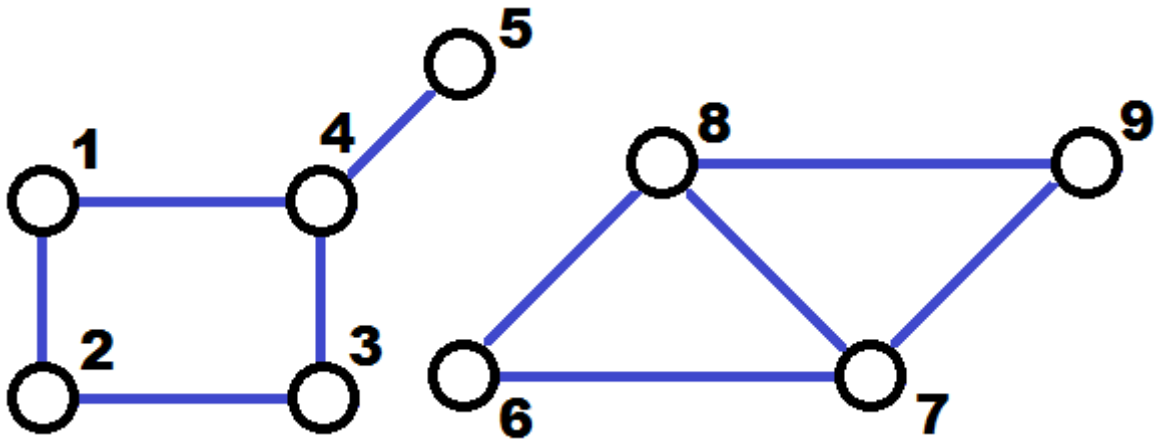
Алгоритм PageRank

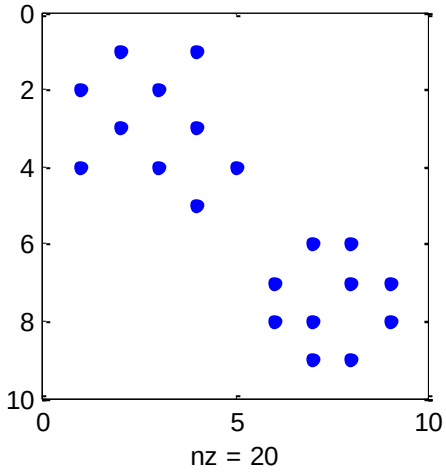
Можно по-разному формализовать

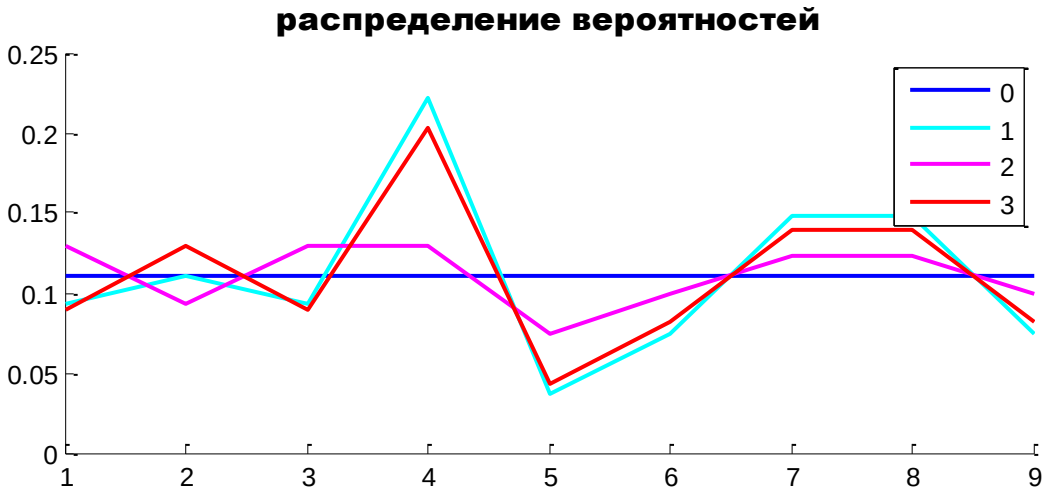
Если матрицу отнормировать, то сумма рангов в сети – константа

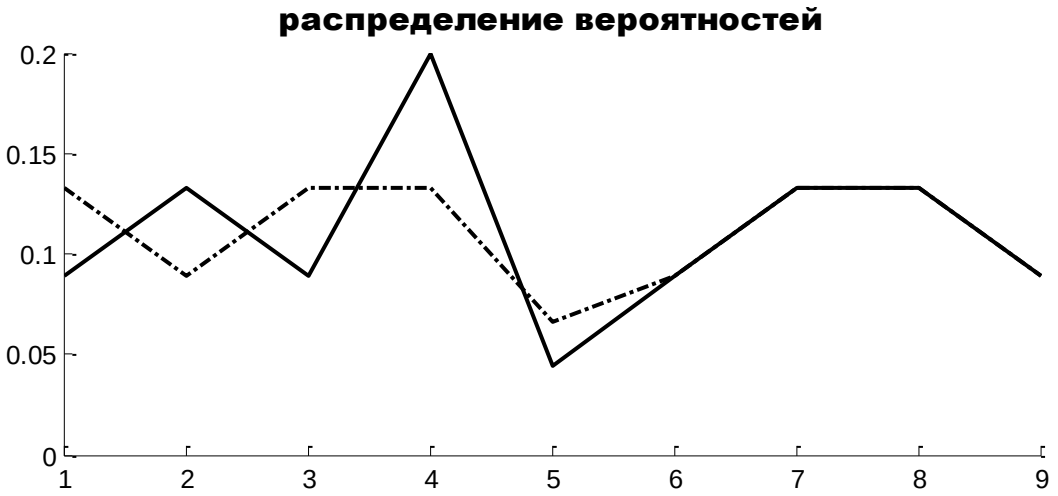
+) рейтинг не появляется, он постоянен в сообществе

Проблема: на практике не всегда получается





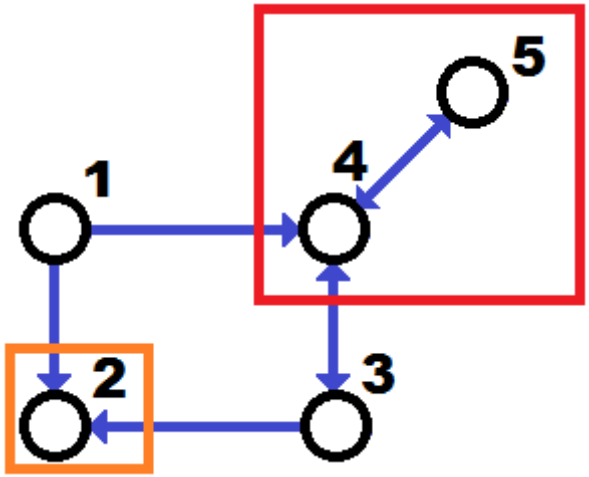




Почему?

Два типа проблем

1. Циклы (Spider traps) **2. Мёртвые вершины (Dead ends)**



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

Решение проблем

Брин, Пейдж:

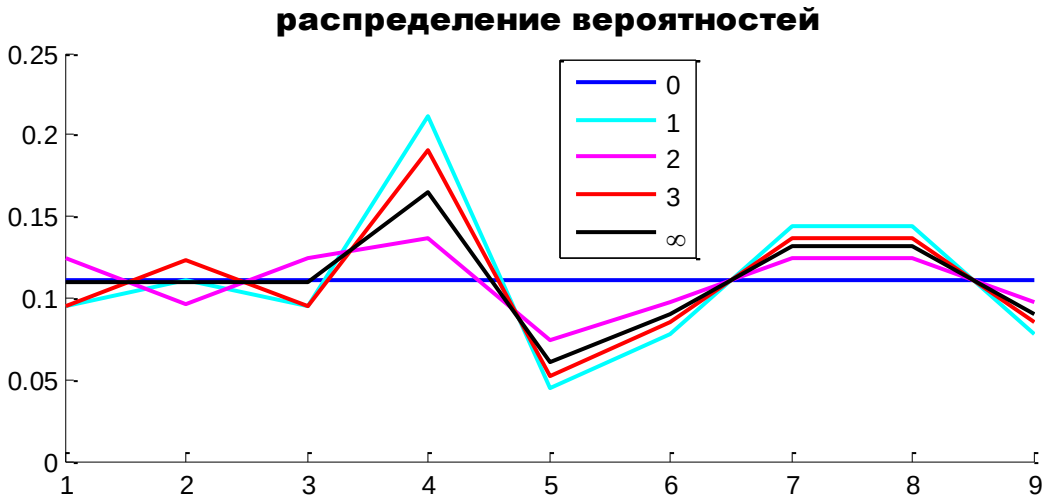
$$w_j = \beta \sum_{(i,j)\in E} \frac{w_i}{\deg_{\text{out}}(i)} + (1-\beta)\frac{1}{n}$$

$$M = \beta \cdot N + \frac{(1-\beta)}{n}\tilde{1}\cdot\tilde{1}^{\text{T}}$$

Обычно 100 итераций

Larry Page and Sergey Brin, The PageRank citation ranking: Bringing order to the web, Technical Report, Stanford Infolabs, 1999.

В результате



Практические аспекты

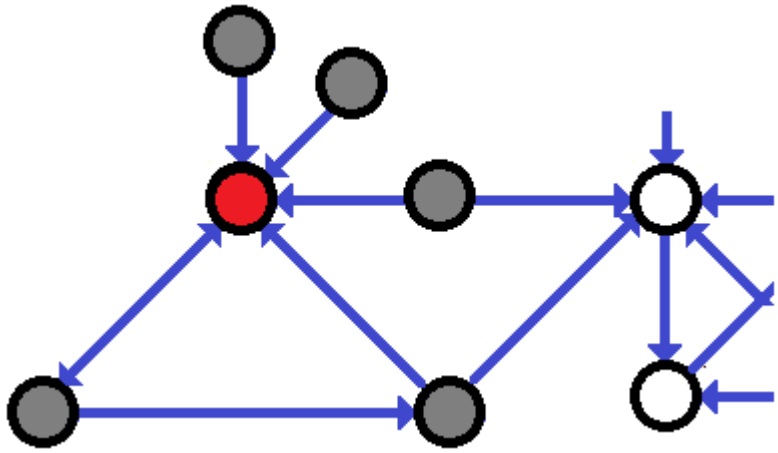
Переход не в произвольную вершину, а

- **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,

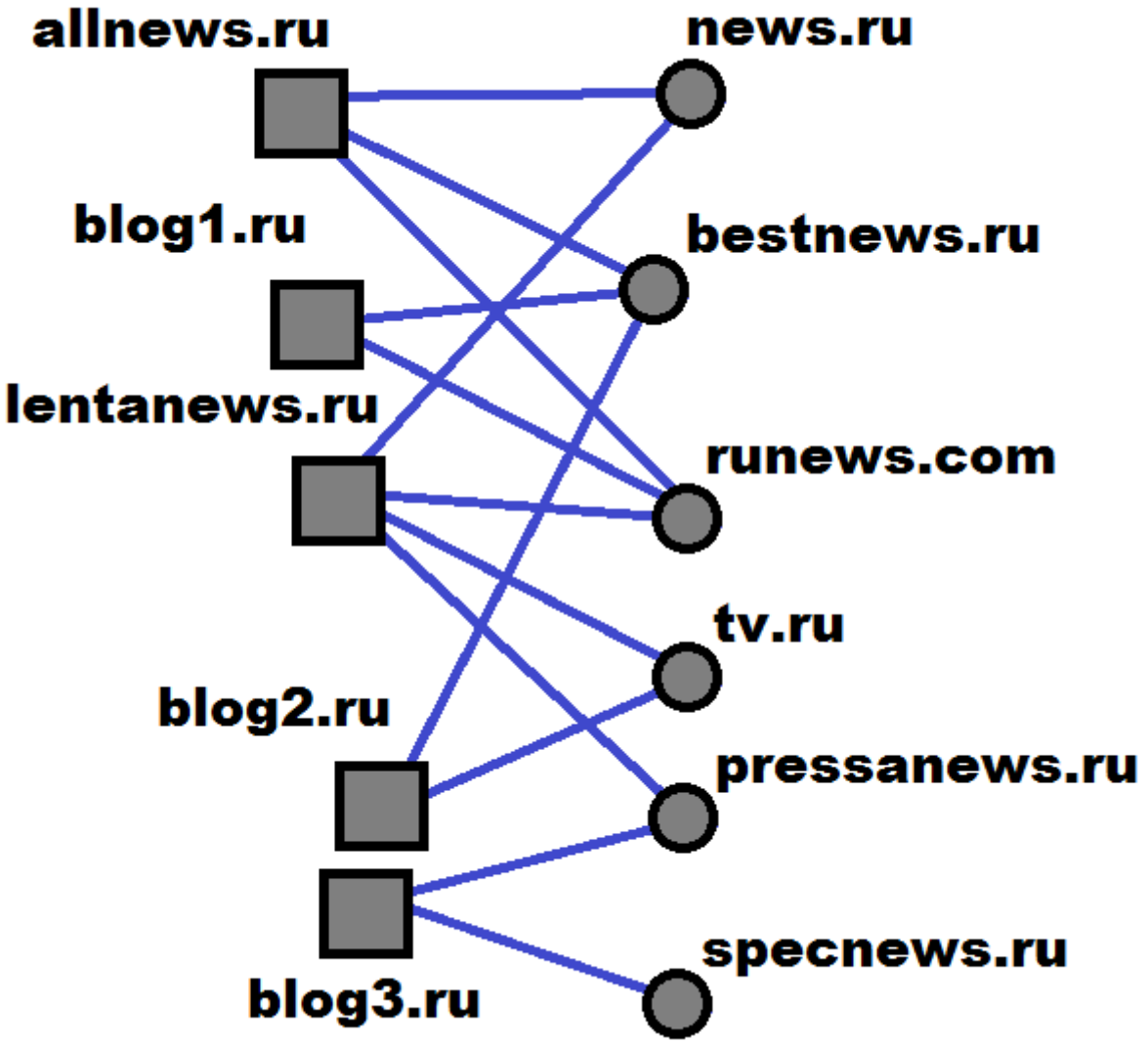
– спам

Для формирования доверенной зоны можно использовать эксперта

Ещё итерационные алгоритмы

поиск ценных источников информации

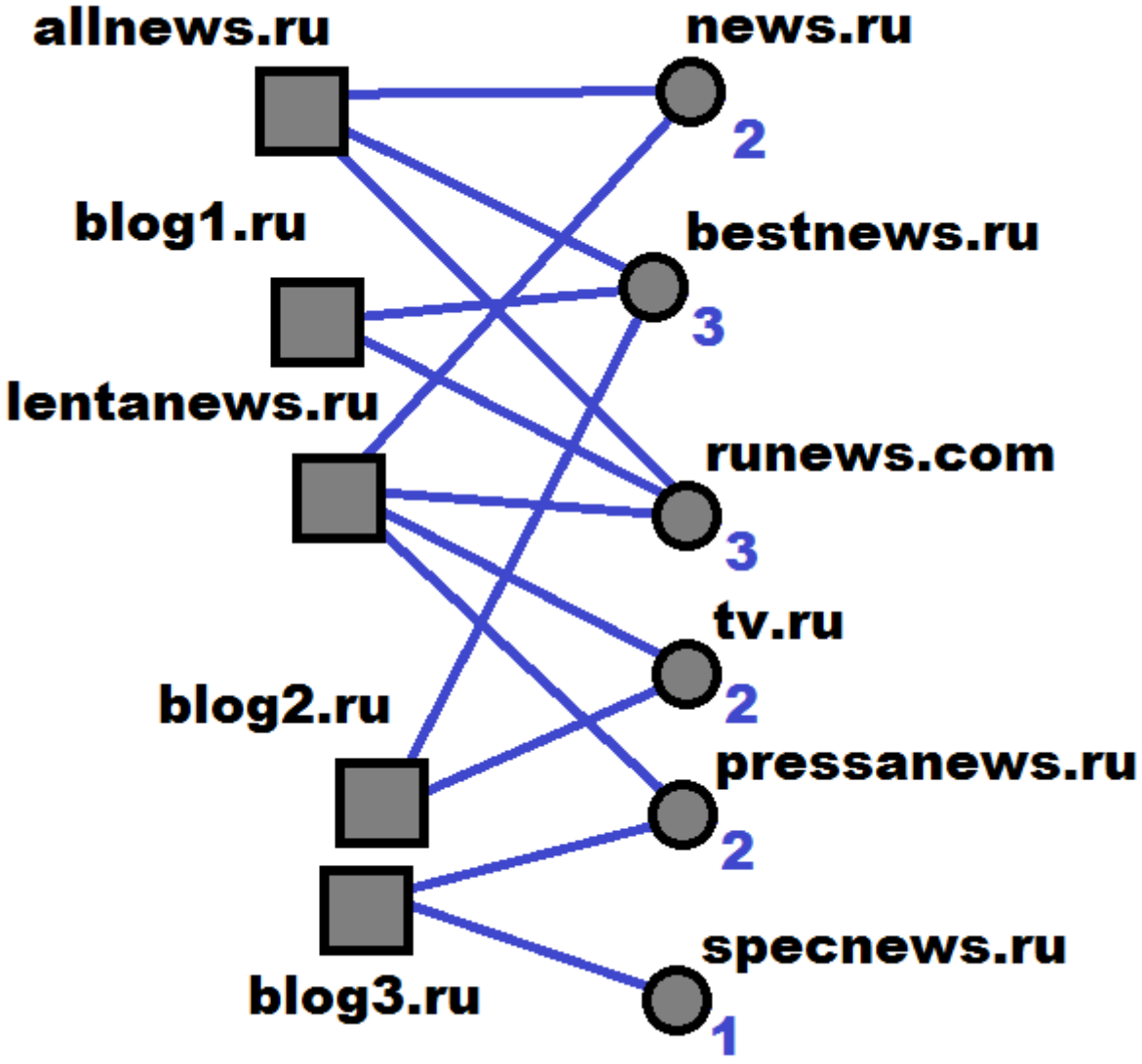
Агрегаторы



Новостные сайты

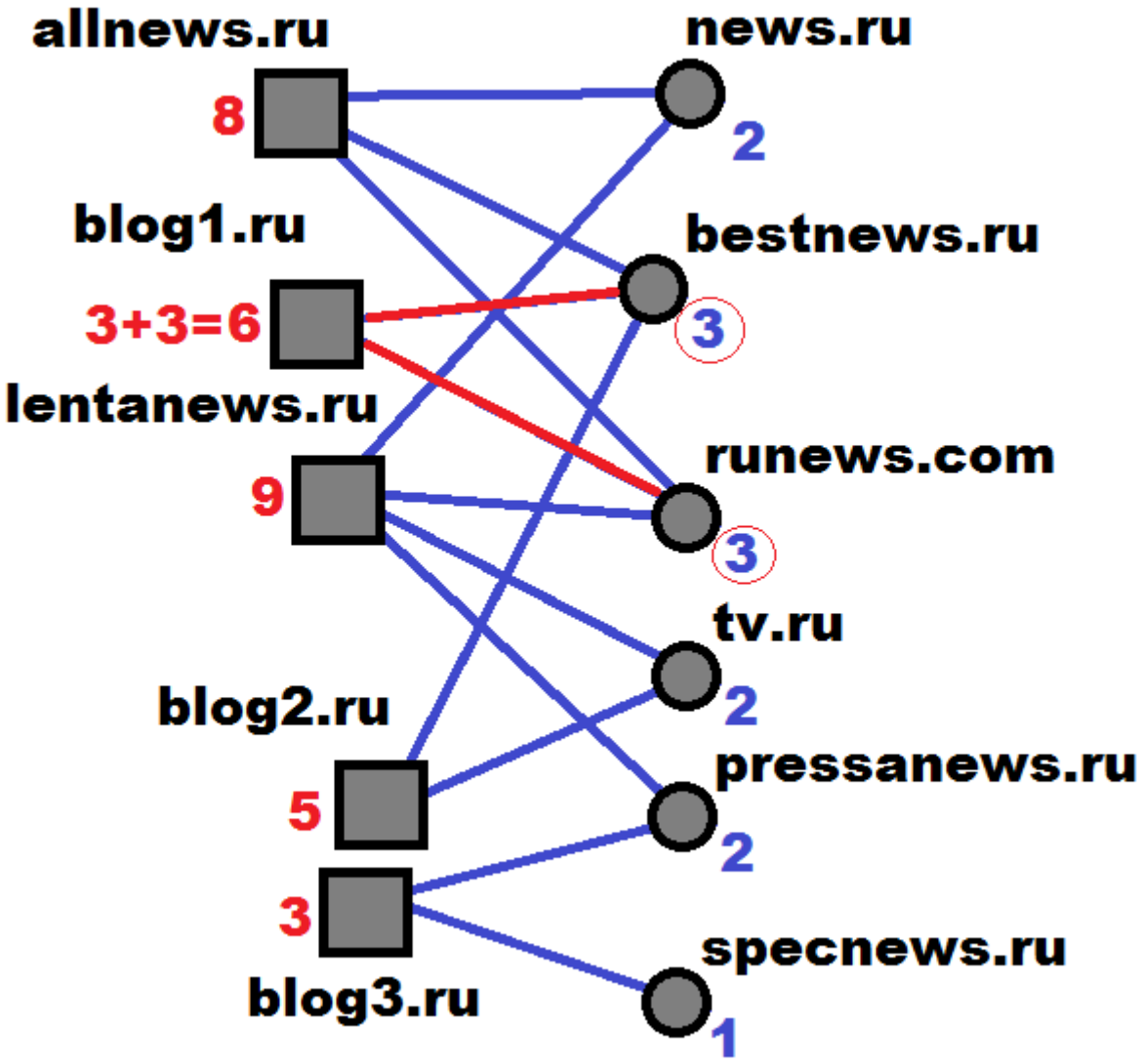
Ещё итерационные алгоритмы

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?

http://liacs.leidenuniv.nl/~takesfw/SNACS/lecture4.pdf

HITS=«Hyperlink Induced Topic Search» (алгоритм Кленберга)

Пусть в графе вершины $V = H \cup A$**:** $H = \{h_i\}, A = \{a_j\}$**,**

рёбра $E \subseteq H \times A$**,**

1. Инициализация:

$$w(h_i) = \frac{1}{|H|}, w(a_j) = \frac{1}{|A|}$$

2. Повторять

$$w(a_j) = \sum_{(i,j)\in E} w(h_i), \; w(h_i) = \sum_{(i,j)\in E} w(a_j)$$

$$w(a_j) = \frac{w(a_j)}{\sum_t w(a_t)}, \; w(h_i) = \frac{w(h_i)}{\sum_t w(h_t)}$$

до сходимости

$$\sum_t w(h_t) < \varepsilon, \; \sum_t w(a_t) < \varepsilon$$

HITS

$$\begin{cases} a = M^{\mathrm{T}} h \\ h = Ma = MM^{\mathrm{T}} h \end{cases}$$

$$\begin{cases} a^{(t)} = M^{\mathrm{T}} h^{(t-1)} \\ h^{(t)} = MM^{\mathrm{T}} h^{(t-1)} = (MM^{\mathrm{T}})^t h^{(0)} \end{cases}$$

Иногда используют другие нормировки

Недостатки:

– Строгое разграничение: хаб / ресурс

– Надо нормировать, в отличие от PageRank

Kleinberg, Jon «Hubs, Authorities, and Communities» Cornell University. 1999.

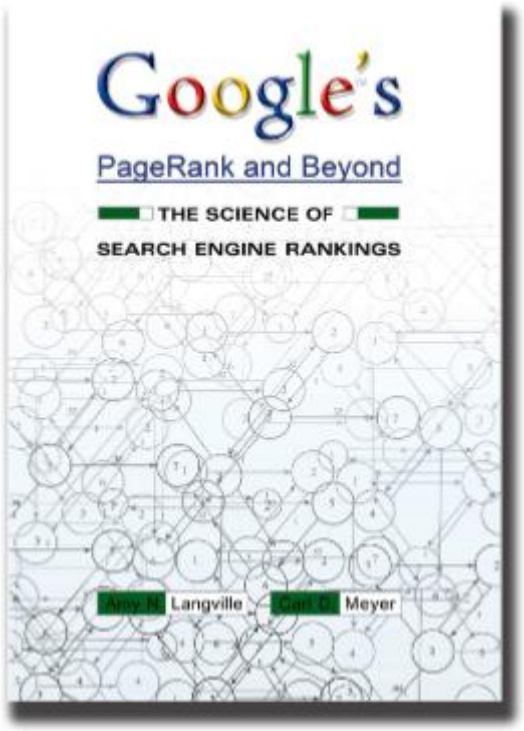
Case: где ещё применяется

«Impact Factor» научных журналов
– среднее число цитирований статей,
опубликованных в этом журнале за последние 2 года

«New Lung Cancer Study Takes Page from Google's Playbook»
http://www.scripps.edu/news/press/2013/20130325lung_cancer.html

Что почитать

«Google's PageRank and beyond»

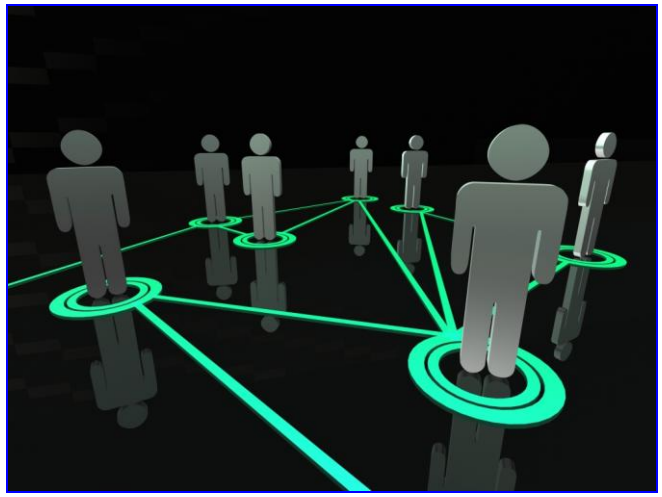


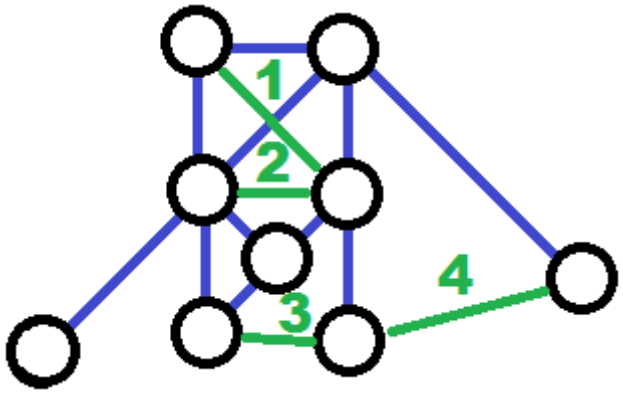
http://geza.kzoo.edu/~erdi/patent/langvillebook.pdf

case: Прогнозирование появления ребра в динамическом графе (Link Prediction Problem)

Международное соревнование
«IJCNN Social Network Challenge»

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



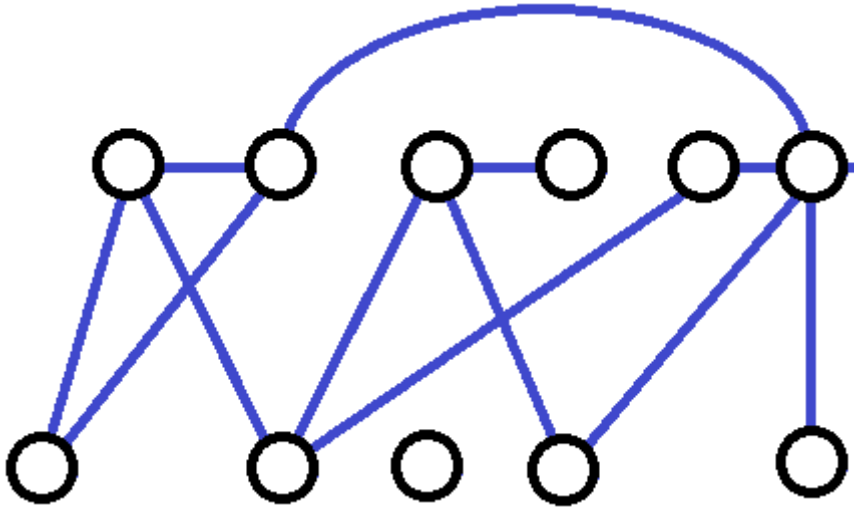


Дан граф,
Список потенциальных рёбер

Необходимо ранжировать список по вероятности появления

Соревнование «IJCNN Social Network Challenge»

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



вершин = 1’100’000

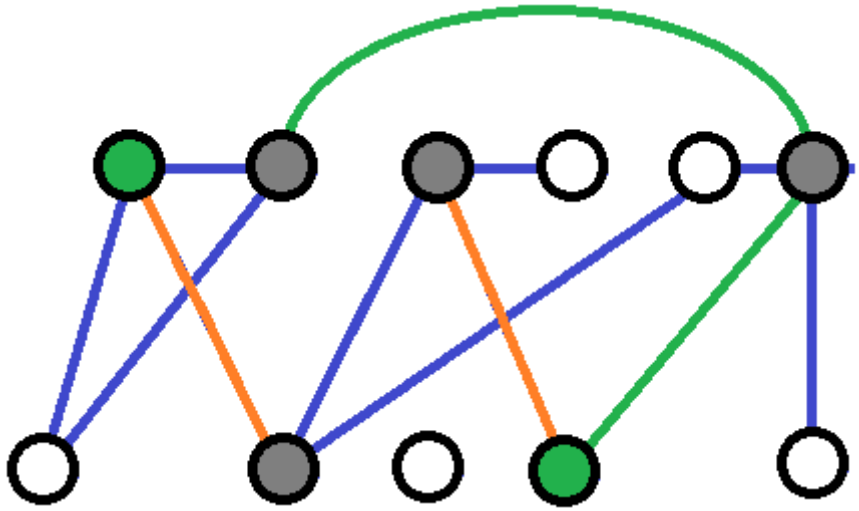
рёбер = 7’200’000

Сеть Flickr

Тест = 4480+4480
потенциальных рёбер

Как решать?

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,

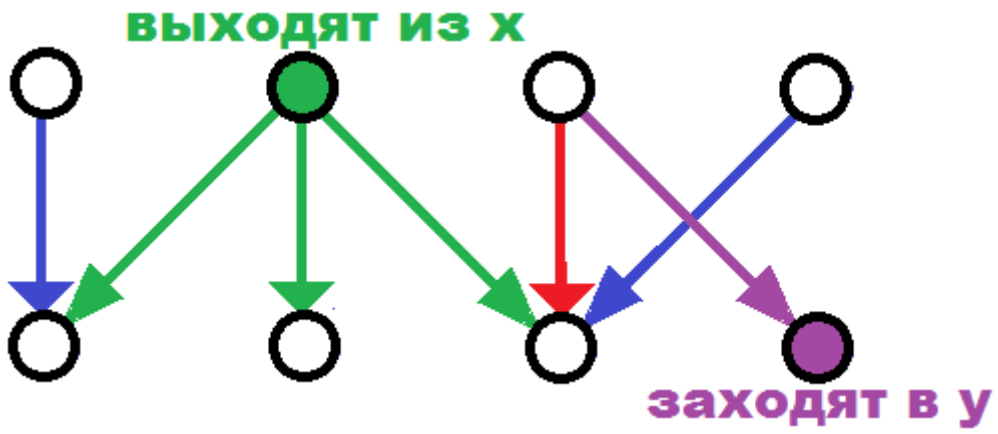
то они могут скоро стать рёбрами… а этому мы и должны научиться

Первый подход

друг друга

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

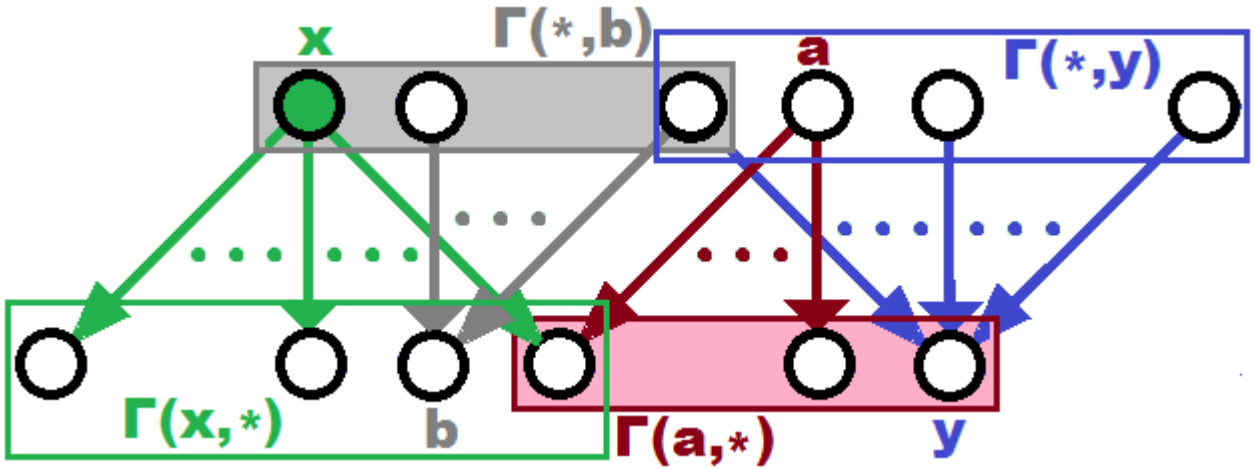
$$\Gamma(x,*)=\{y\in V\,|\,(x,y)\in E\}$$



Улучшение качества при таком признаке

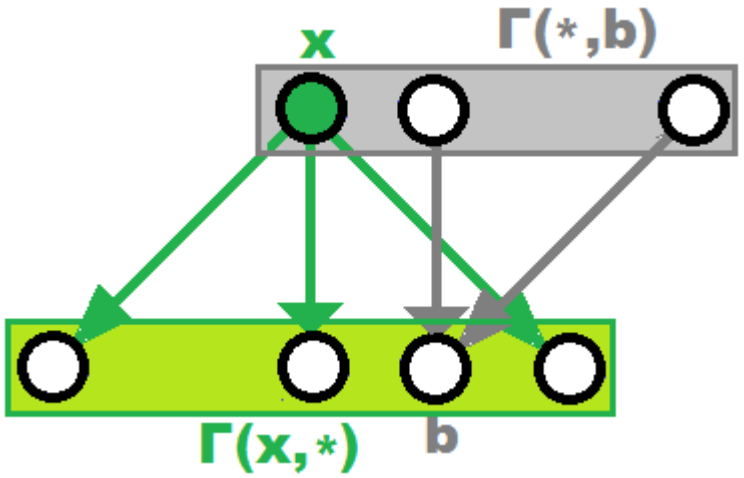
$$\frac{\sum\limits_{\substack{a\in\Gamma(*,y)\\ b\in\Gamma(x,*)}} \frac{|\Gamma(a,*)\cap\Gamma(x,*)|\cdot|\Gamma(*,b)\cap\Gamma(*,y)|}{\sqrt{|\Gamma(a,*)|\cdot|\Gamma(*,b)|}}}{|\Gamma(x,*)|\cdot|\Gamma(*,y)|+1}$$

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

I = 87.5

I + II = 90.7

III = 90.7

I + II + III = 92.6

PR = 93.0

I + II + III + PR = 95.0

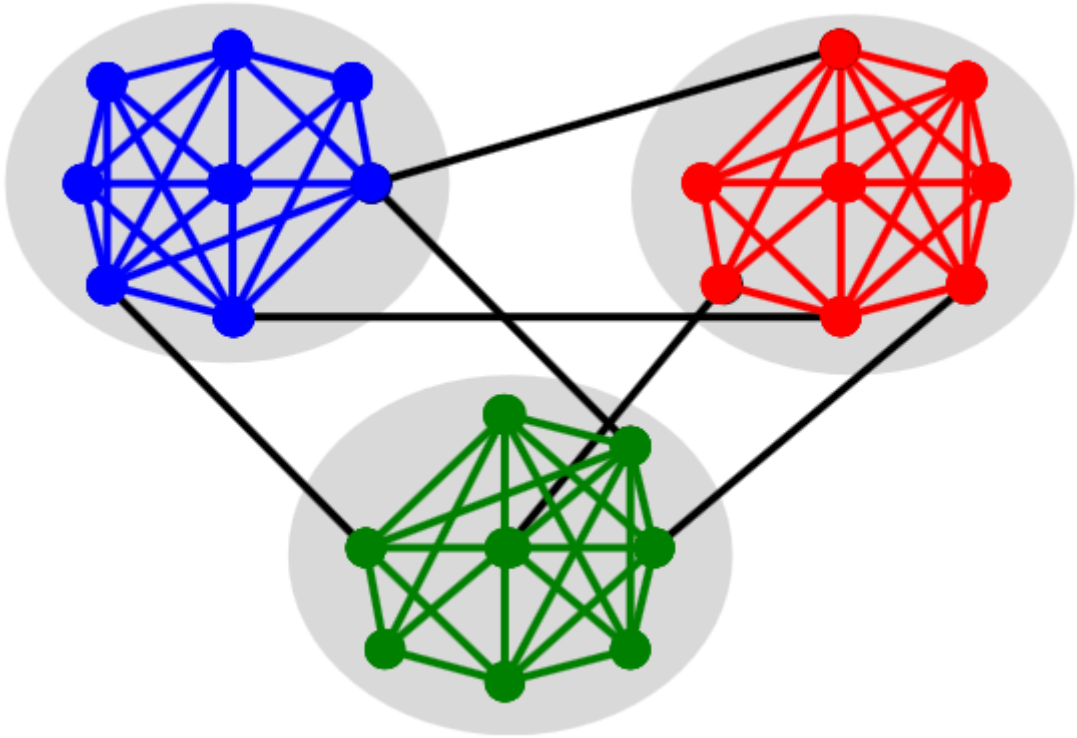
Сообщество в графе

нет чёткого определения

рёбер внутри сообщества много,

рёбер соединяющих сообщество с остальными вершинами мало

малый радиус сообщества



Какие бывают определения сообщества

1. Чёткие

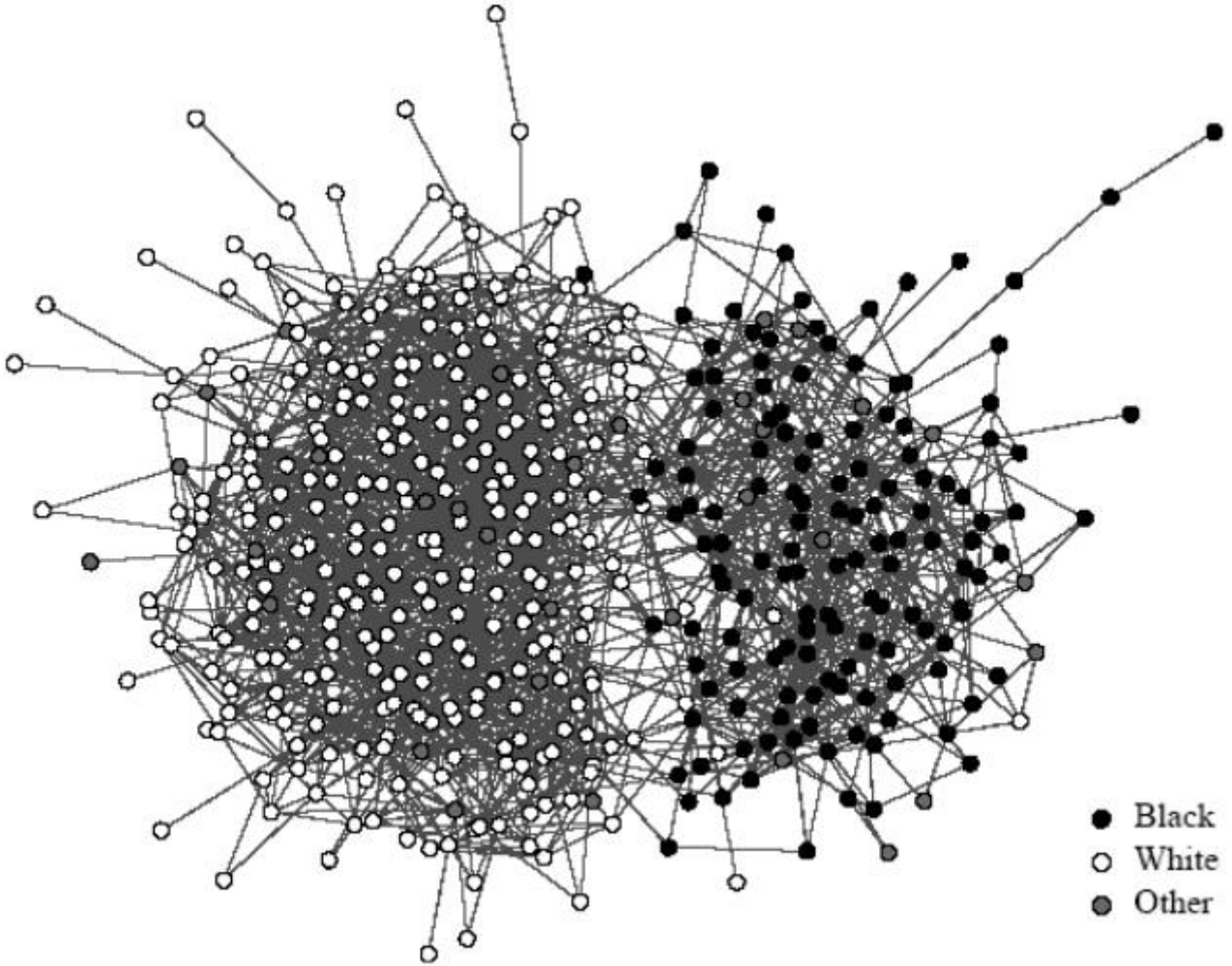
2. Нечёткие (не определения, см. выше)

3. Алгоритмические

(то что получается в результате действия алгоритма)

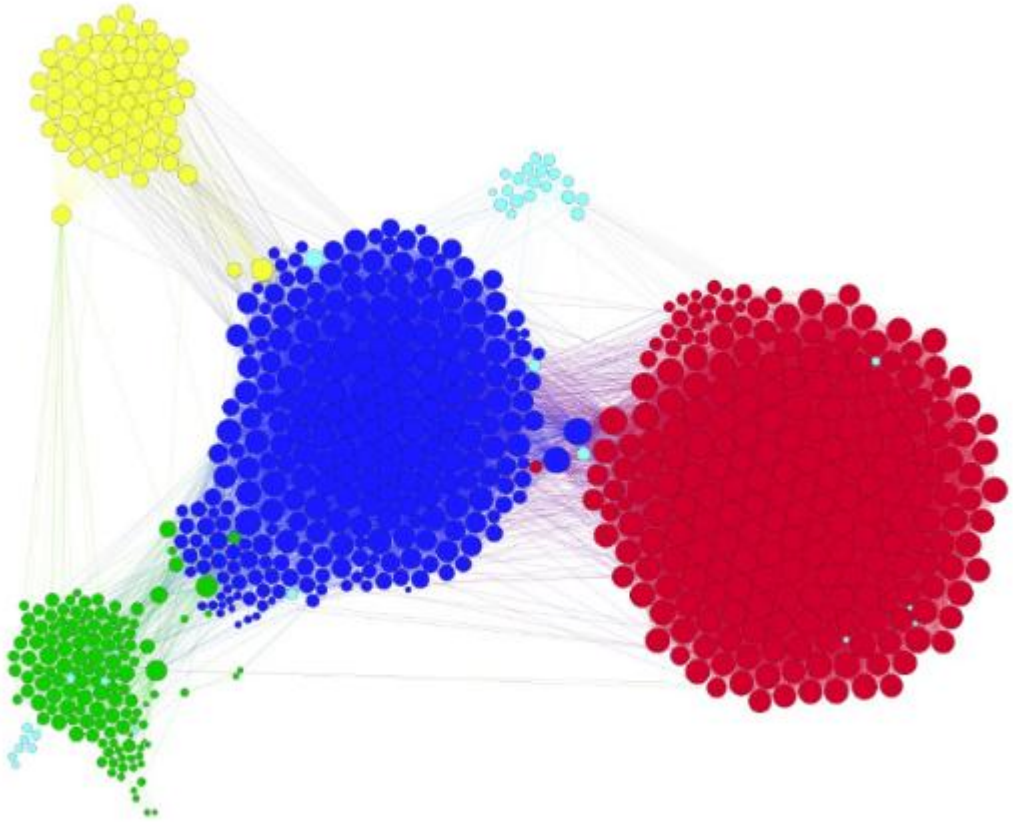
Примеры сообществ

Сеть социальных отношений в high-school

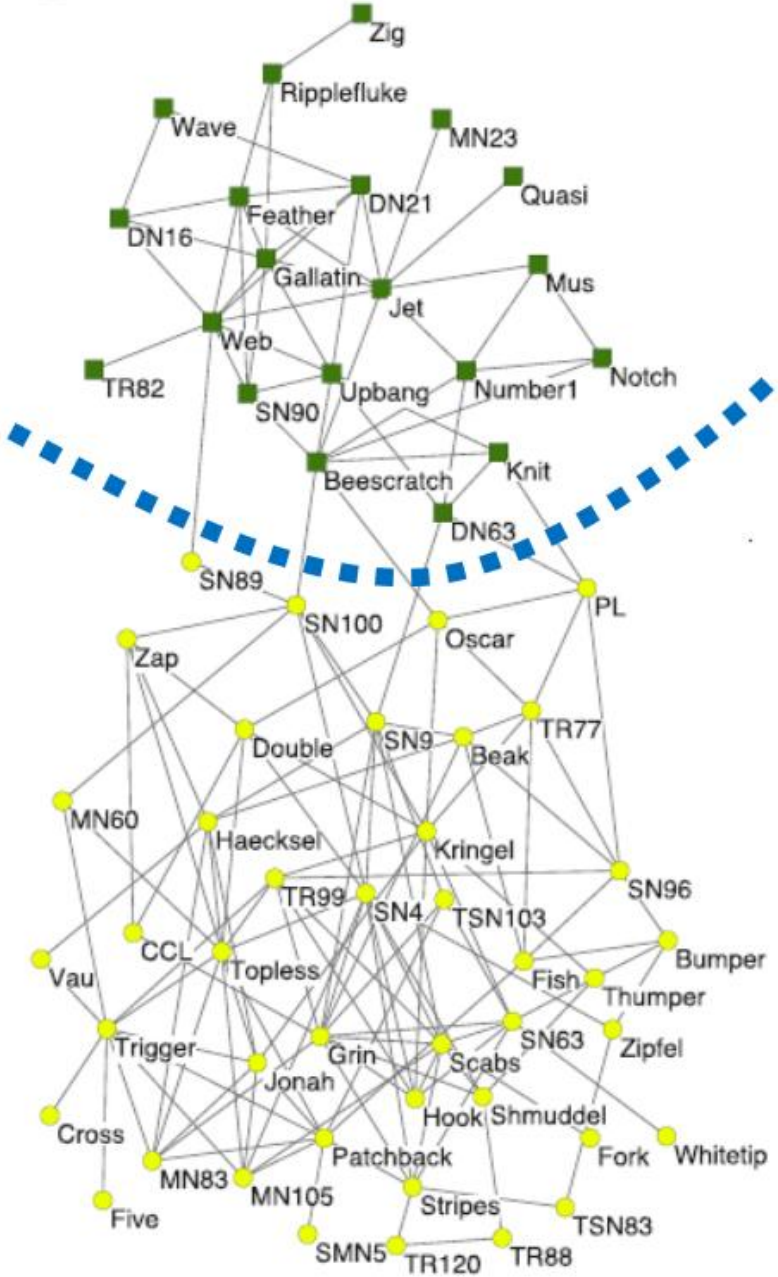


Примеры сообществ

Эго-сеть фейсбука

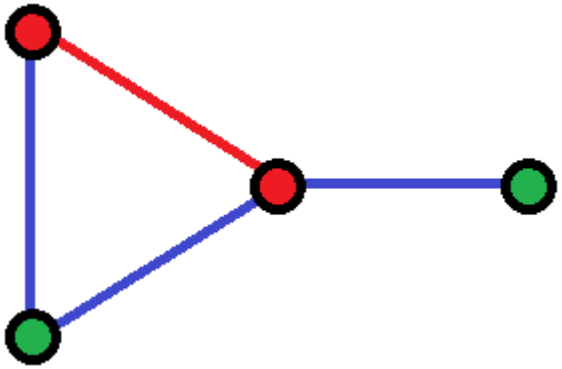


Примеры сообществ

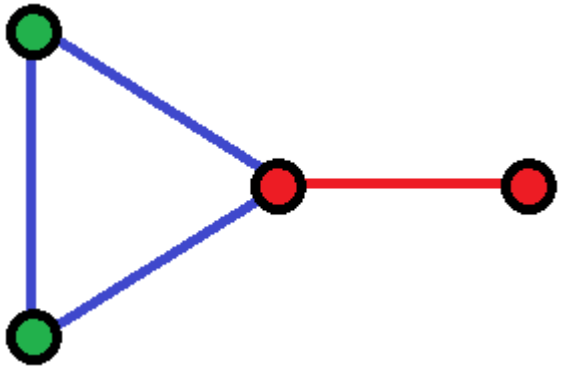


Сообщество в графе

Идеальный кандидат – клика

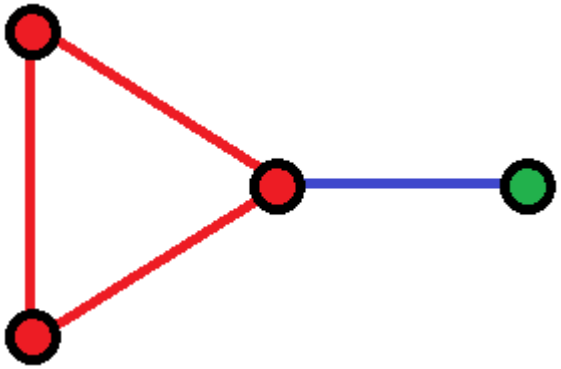


Клика



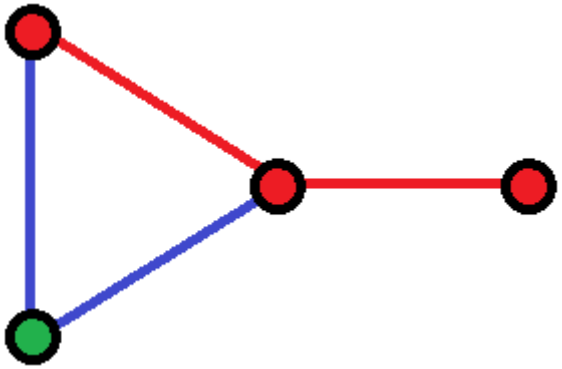
Максимальная клика

Не может быть расширена



Наибольшая клика

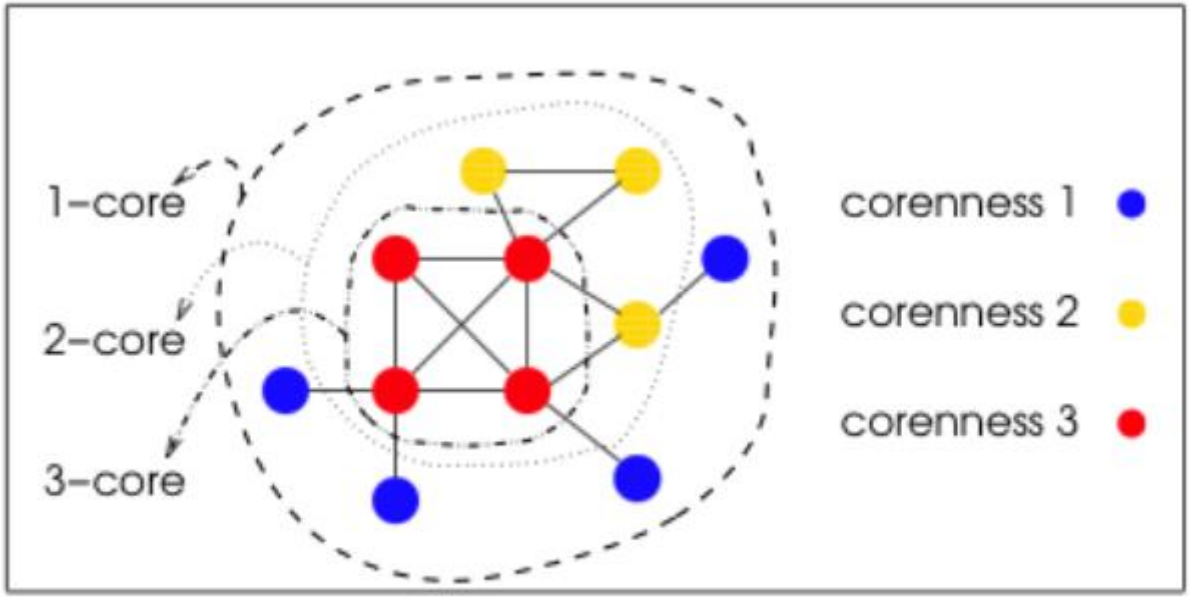
Клика наибольшего размера



Не клика

Но вычислительные сложности…

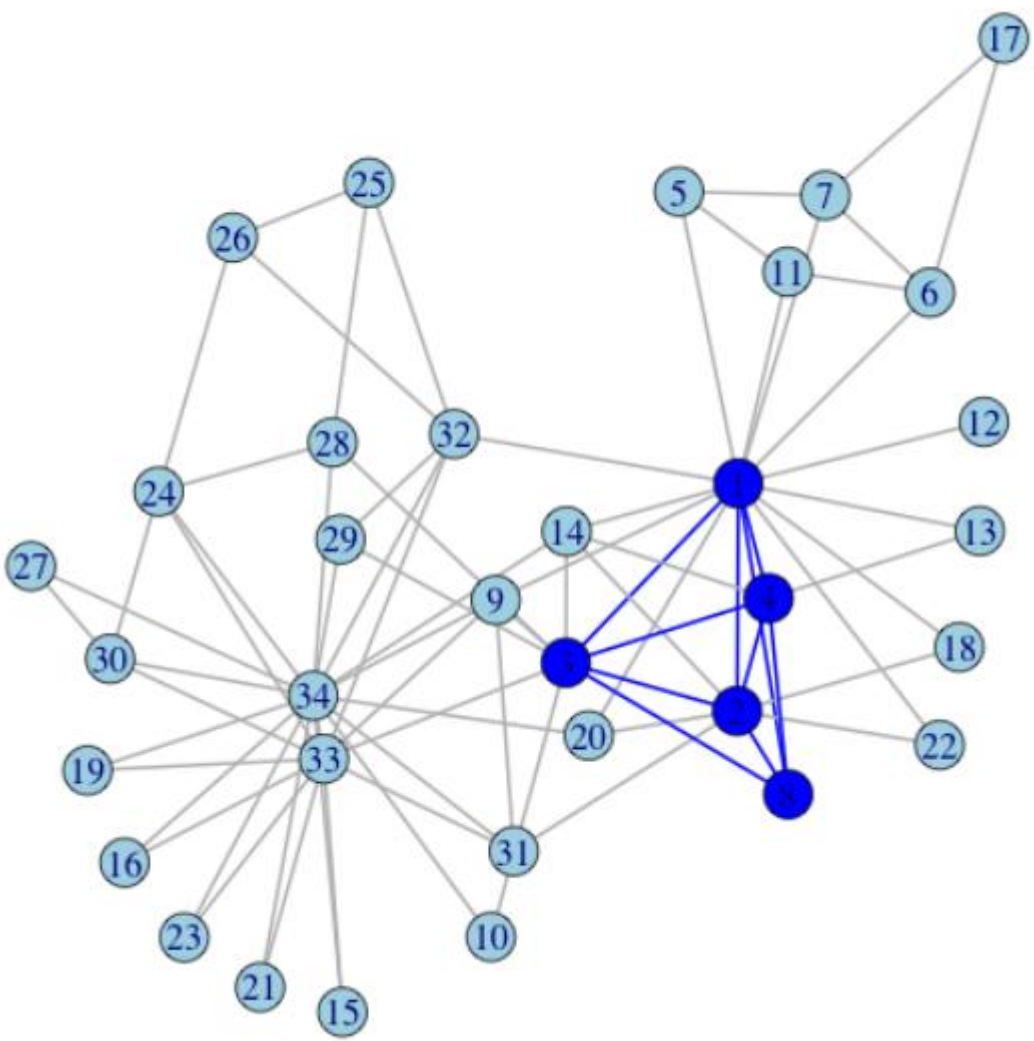
k-ядра

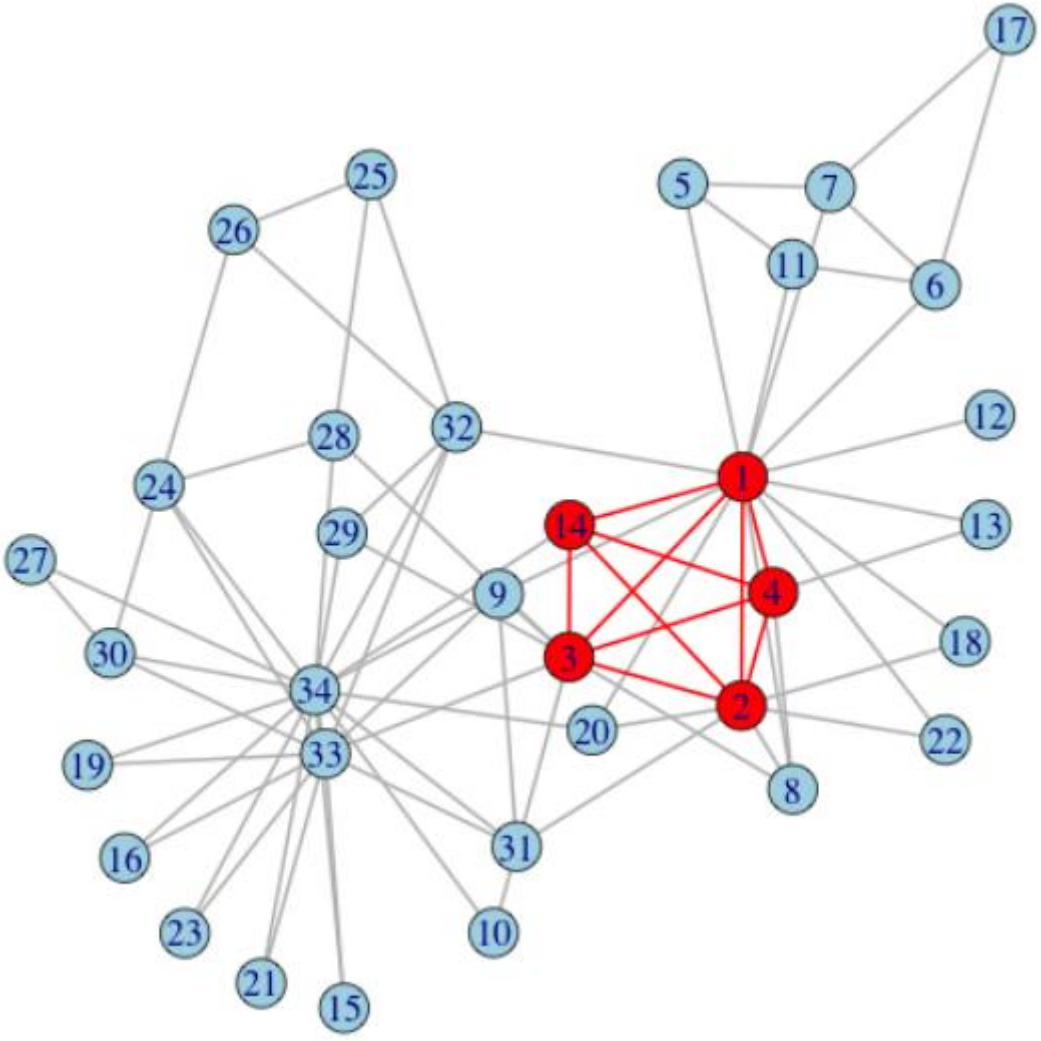


Alvarez-Hamelin et.al., 2005

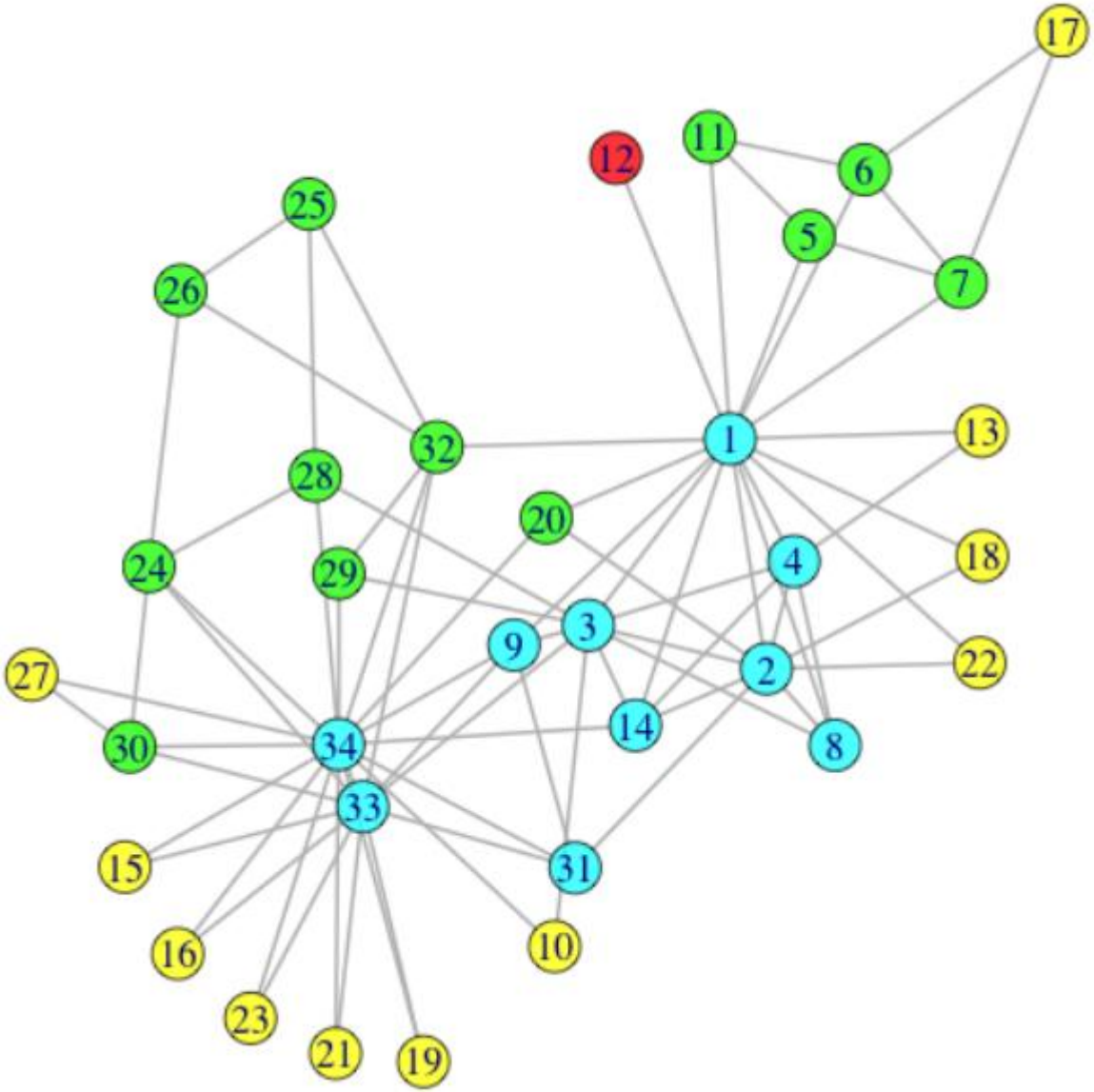
k-ядро = степень каждой вершины >= k

Наибольшие клики (Карате клуб)





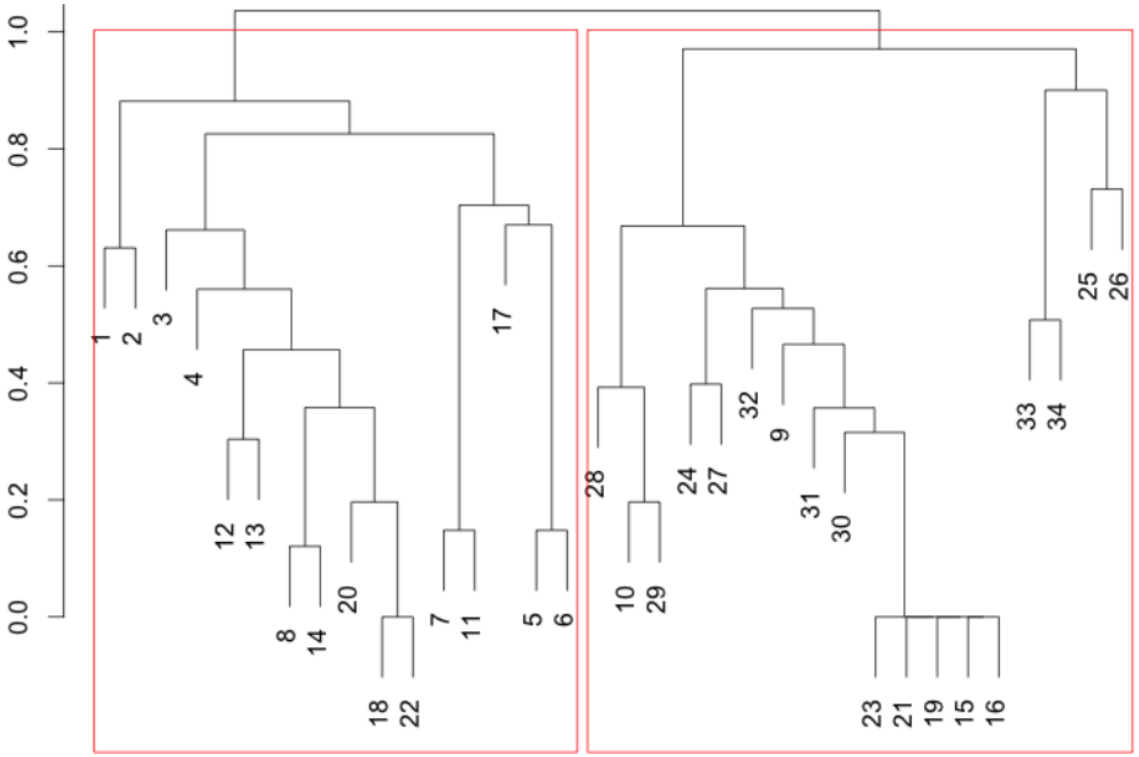
Ядра (Карате клуб)

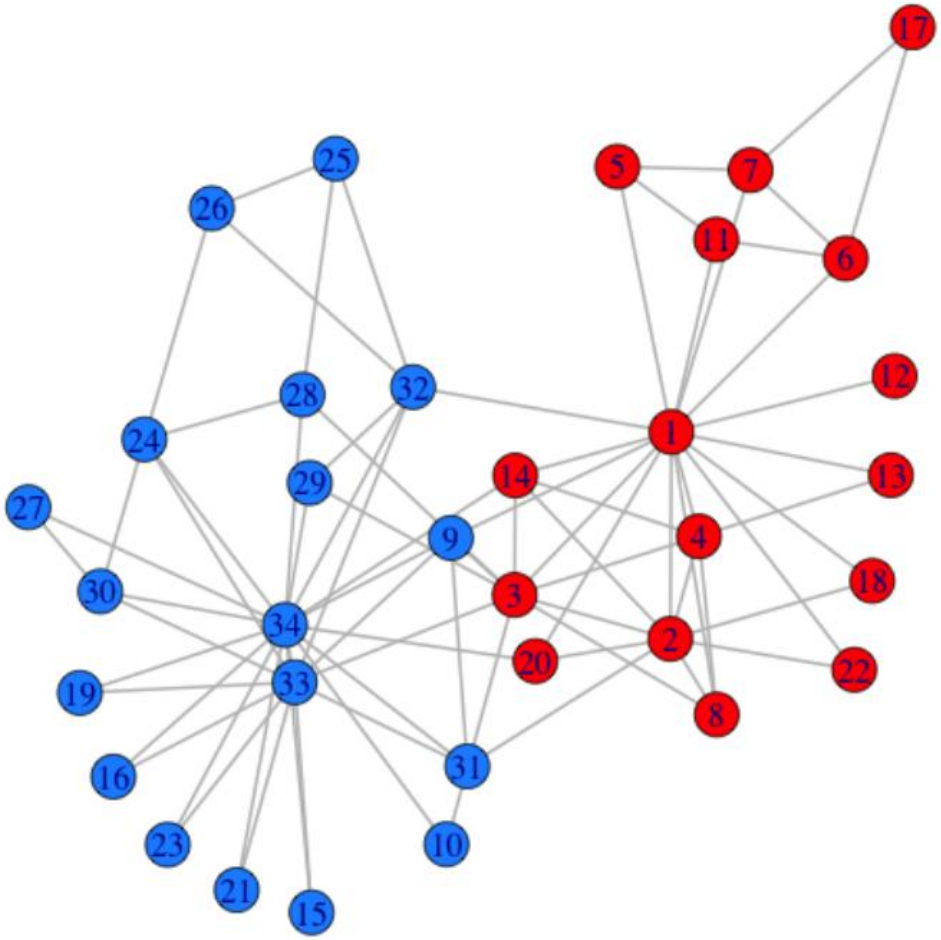


Выделение сообществ

1й способ

Обычная кластеризация с мерой схожести вершин





Выделение сообществ

1й способ - недостатки

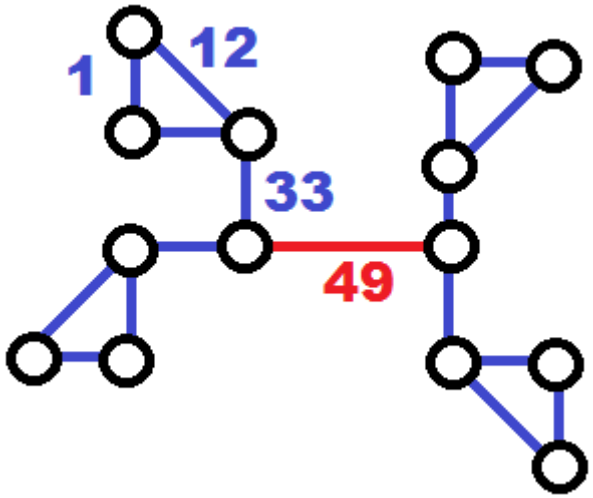
Формально не пытаемся выполнить условия «сообщности»:
много рёбер внутри сообщества
слабые связи между сообществами

Выделение сообществ

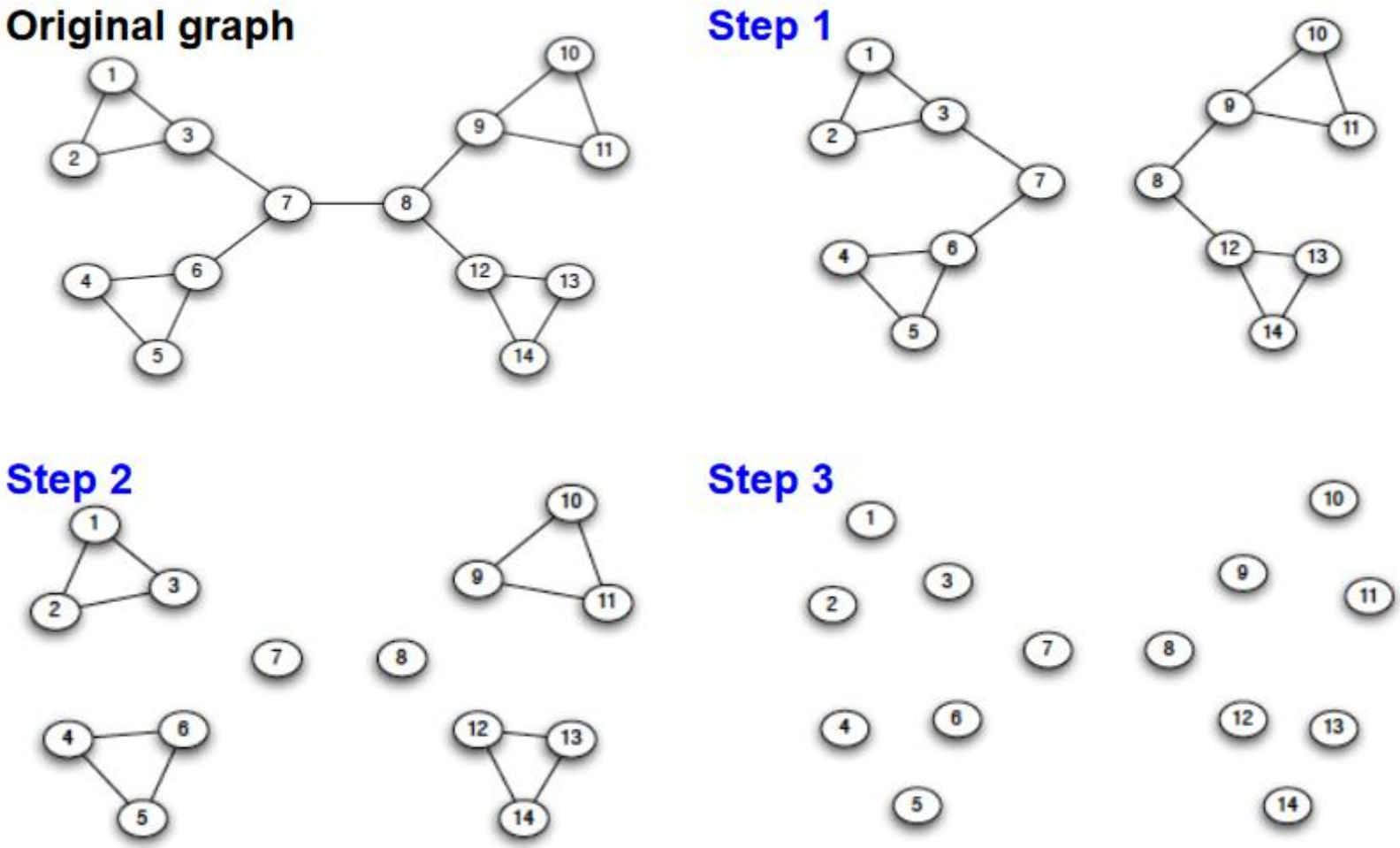
2-й способ – Edge betweenness (Girvan-Newmann's method)

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)

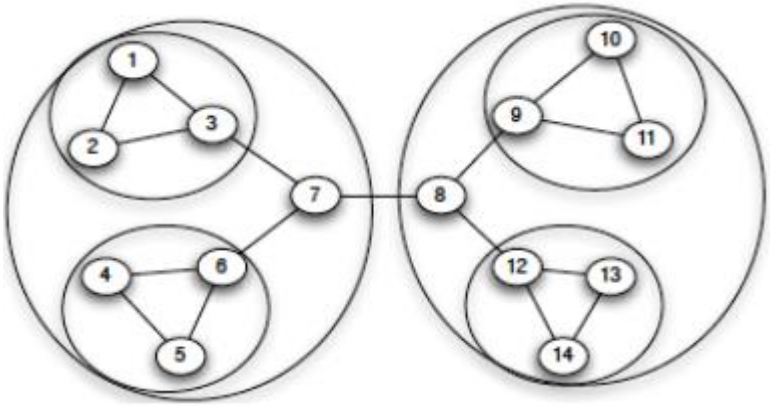


На каком этапе останавливаться (в иерархическом делении)

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**

Обоснование модулярности

Уже был приём…

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

Если просуммировать – вектор степеней

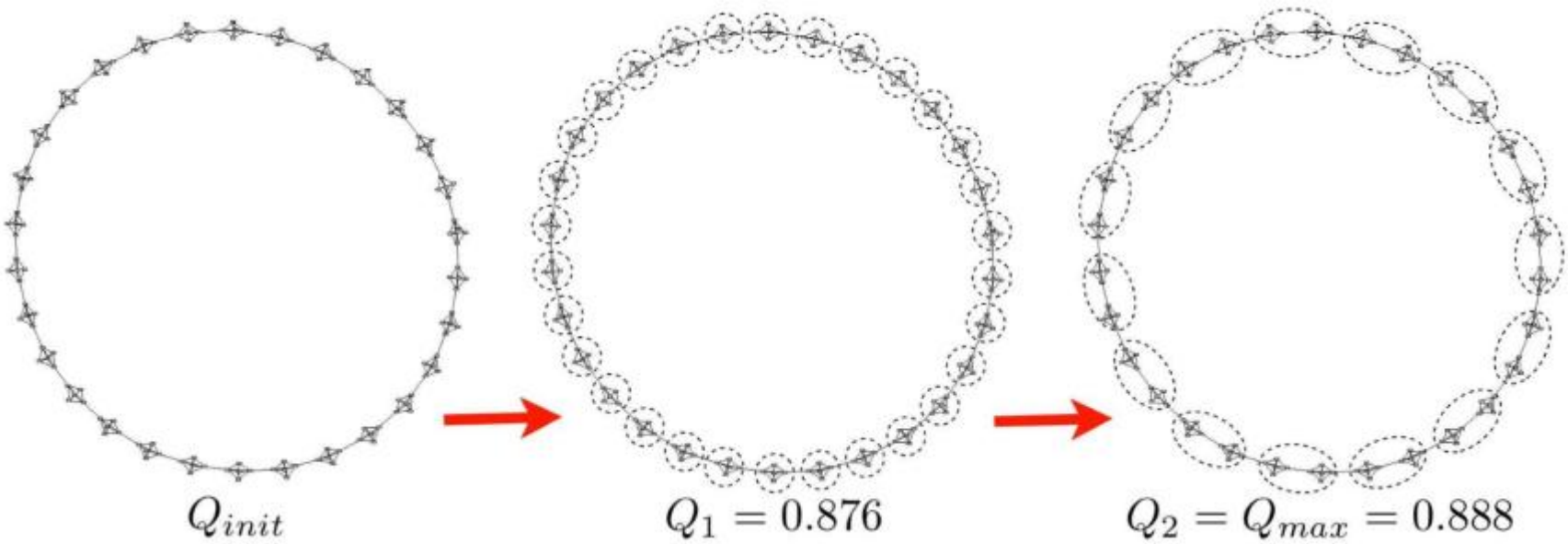
$$\mathrm{sum}(A, \mathrm{axis} = \mathrm{any}) = d = (d_1, \ldots, d_n)^{\mathrm{T}}$$

Хотим «случайную матрицу» вероятностей с такими же суммами:

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

+ нормализация, чтобы была на отрезке [–1, +1]

Иногда модулярность подводит...

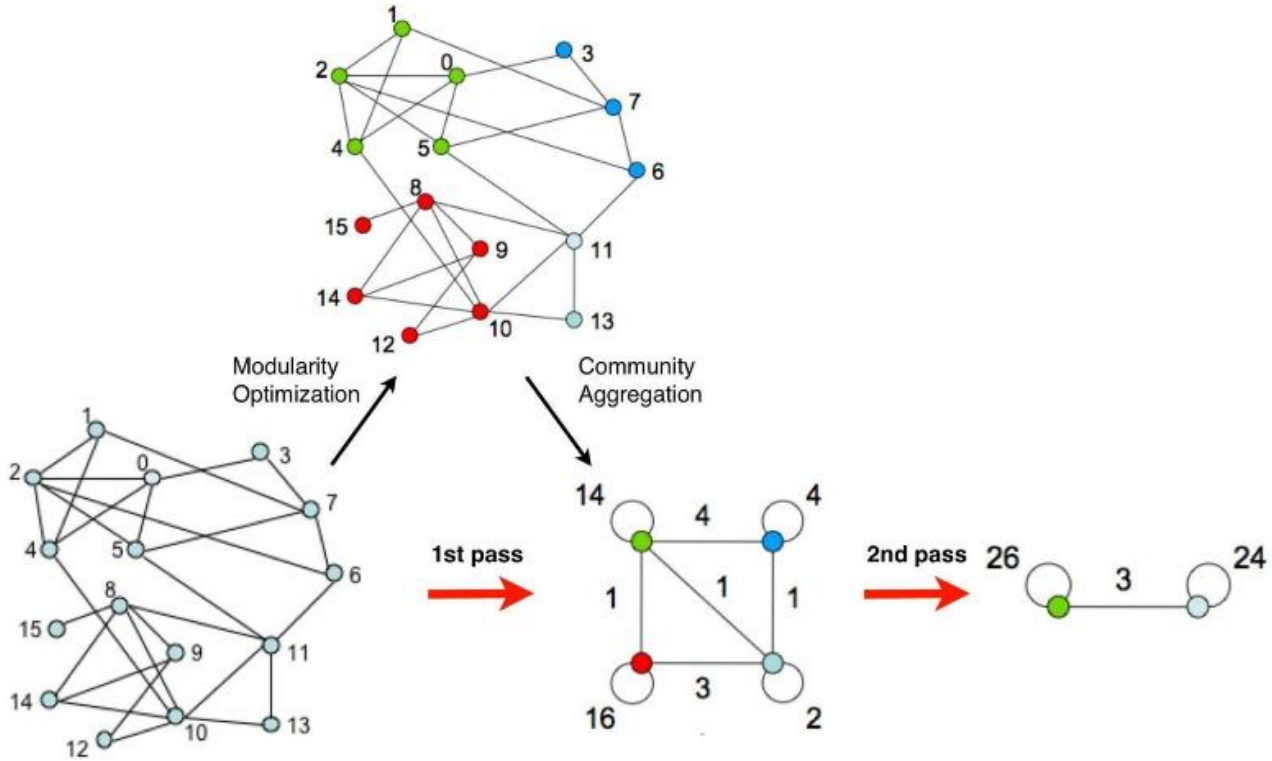


Источник?

Fast community unfolding [Multilevel]

1. Каждая вершина приписывается в своё сообщество
2. Пока возможно:
   a. Для каждой вершины – изменение модулярности при перемещении её в сообщество (каждого) соседа
   b. Максимальное изменение реализуем
3. Пока увеличивается модулярность:

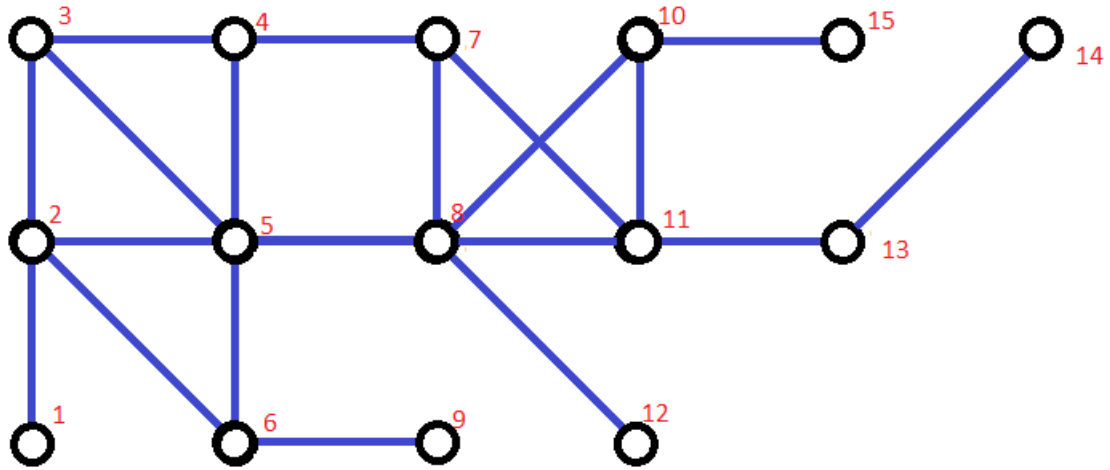
вершины сообществ превращаем в мета-вершины



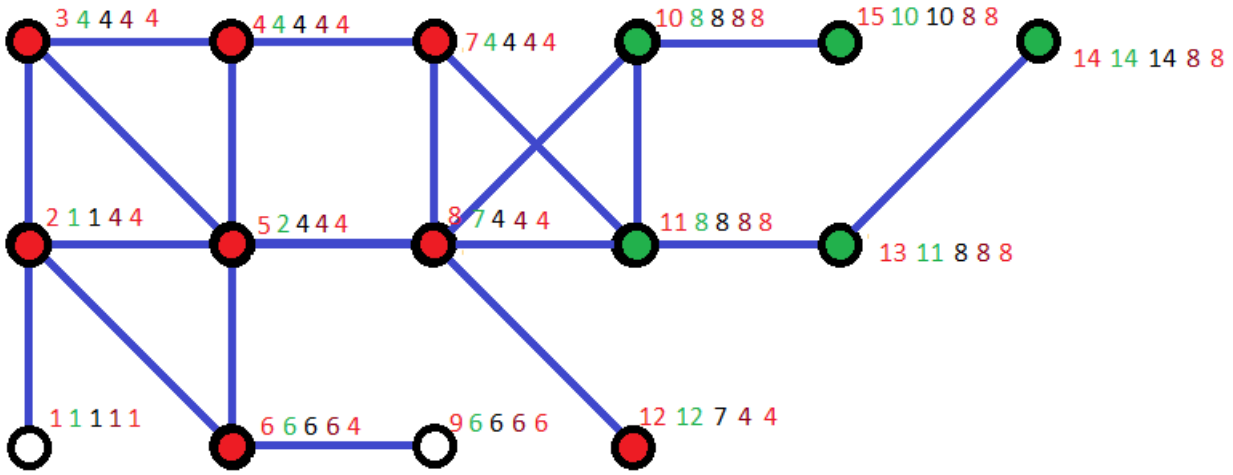
4-й способ: Label Propagation

1. Случайно приписать метки вершинам
2. Цикл по вершинам (в случайном порядке)

a. Метка вершины заменяется на самую частую метку соседей



Label Propagation



5й способ: Walktrap

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

Как измеряется близость сообществ

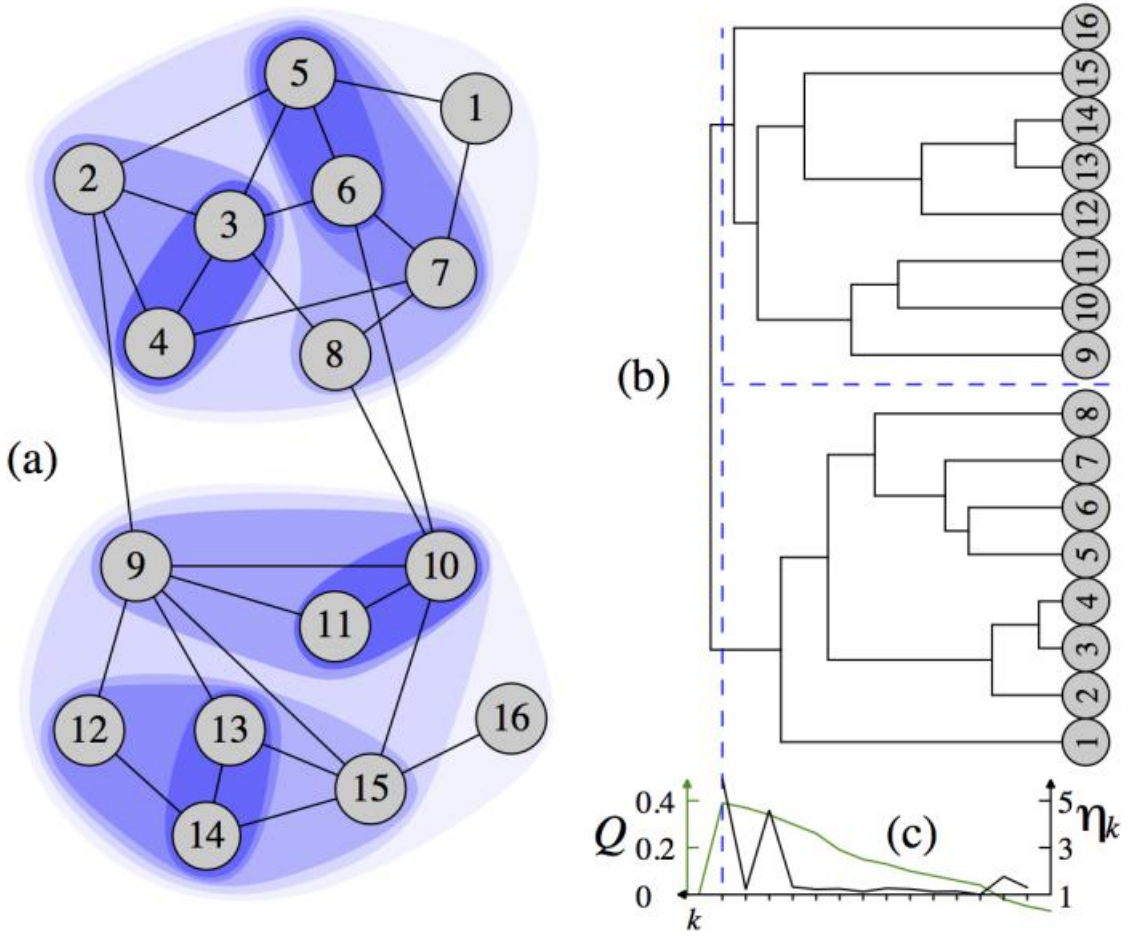
$$r_{A,B}(t)=\sqrt{\sum_{i=1}^{n}\frac{(P_{A,i}^{t}-P_{B,i}^{t})^{2}}{\deg(i)}}=\| D^{-0.5}P_{A}^{t}-D^{-0.5}P_{B}^{t}\|,$$

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

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

(можно вычислить приближённо – случайными блужданиями)

Walktrap



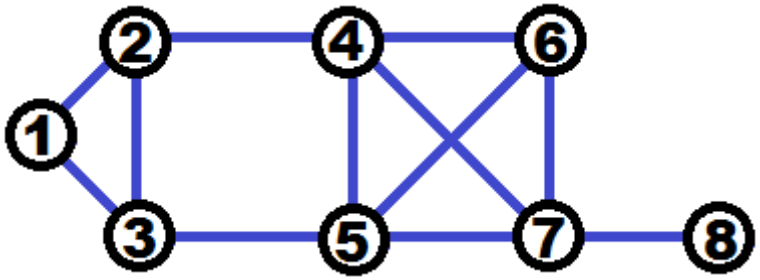
Другая идея выделения сообществ

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	

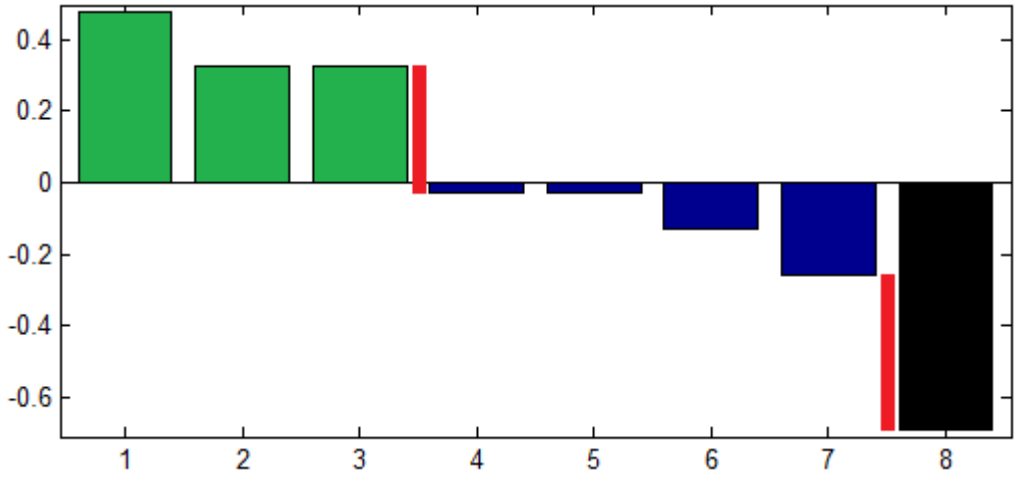


Матрица Лапласа

	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!

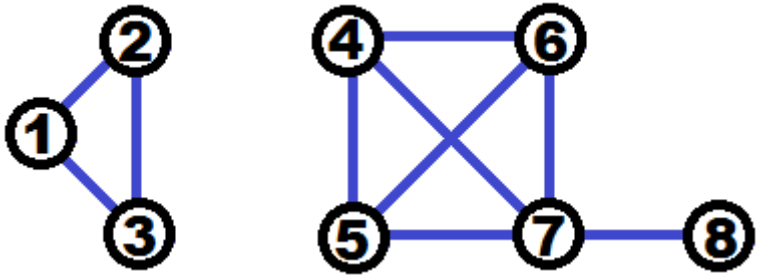
Потом – теоретическое обоснование

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
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]

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

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

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

Но это сложная переборная задача, поэтому вместо

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

Решают вещественную задачу с ограничениями

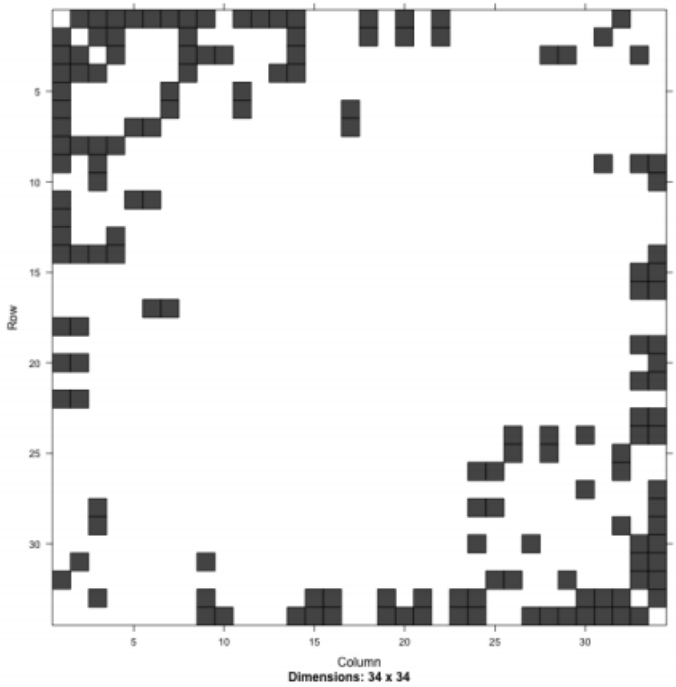
$$\tilde{1}^{\mathrm{T}} x = 0,\ \| x \| = 1.$$

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

Потом $(\mathrm{sgn}(x_1), \ldots, \mathrm{sgn}(x_n))$**.**

Совмещение идей

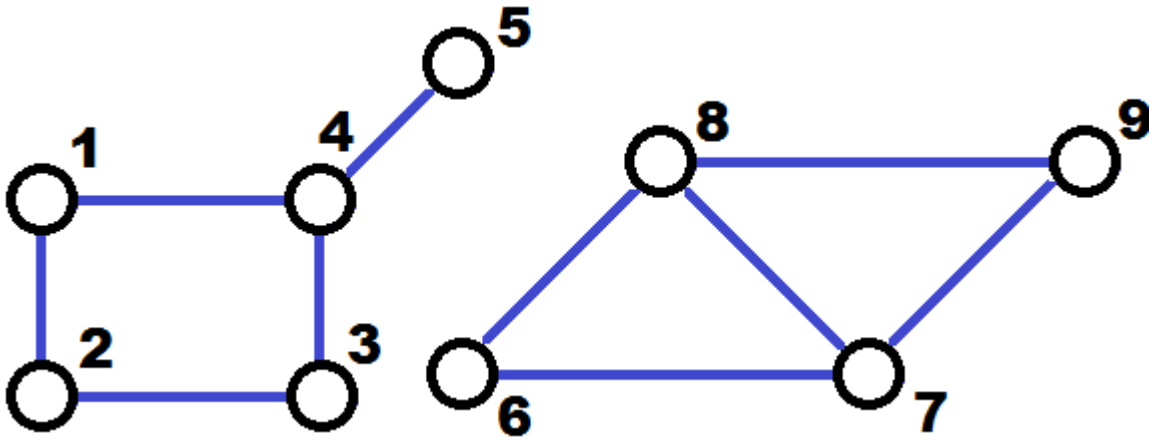
1. Найти второй собственный вектор
2. По его значениям упорядочить вершины



change

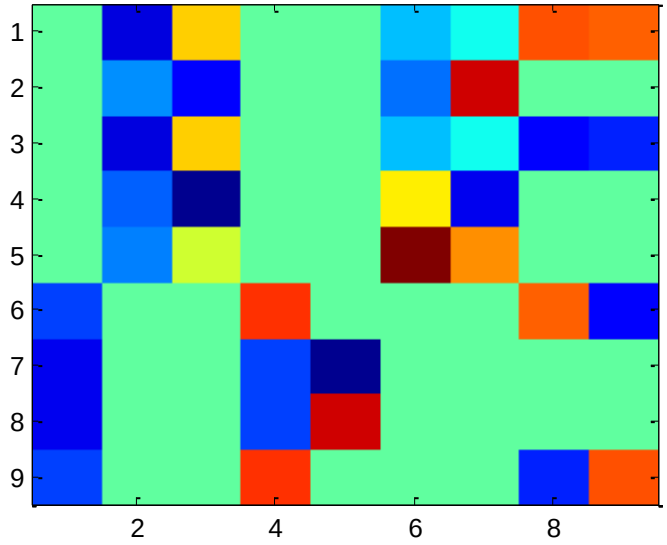
3. Как именно делить решаем по отдельному функционалу (ex: модулярность), надо перебрать всего n-1 деление.

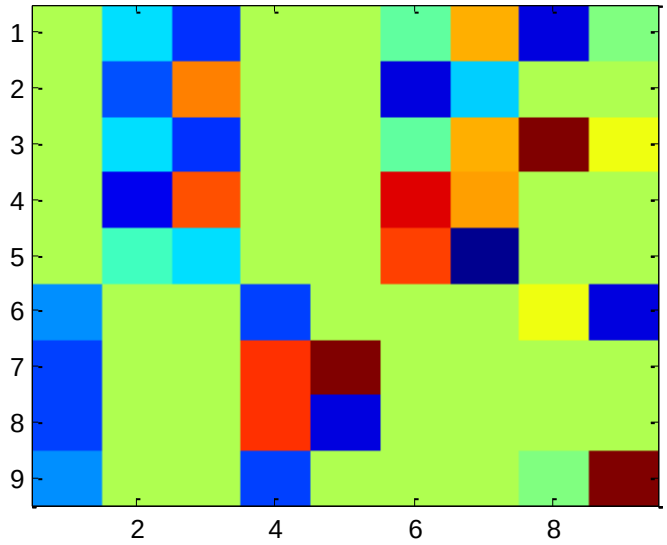
SVD над матрицей смежности



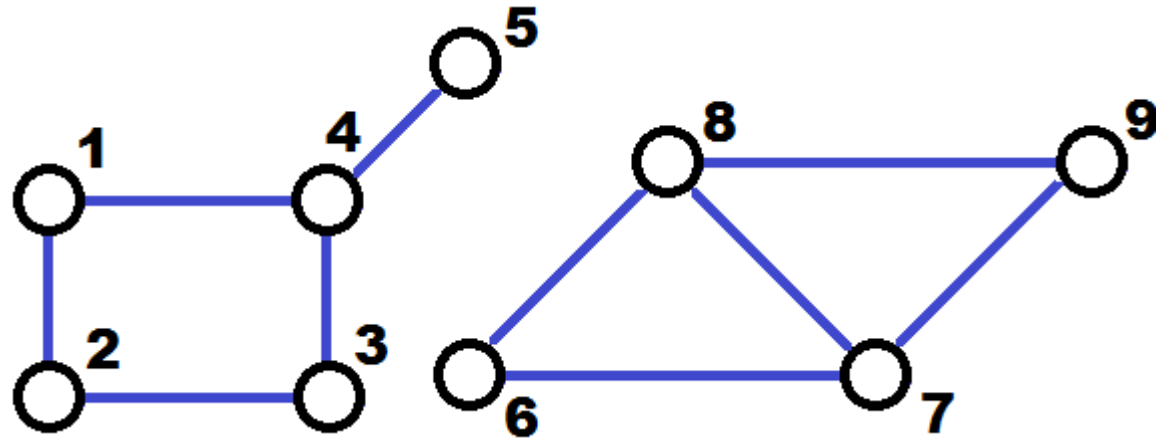

```
S = sparse([1 1 2 2 3 3 4 4 6 6 7 7 8 8 8 9 9 5 4 7], ...
           [2 4 1 3 2 4 1 5 7 8 8 9 6 7 9 8 7 4 3 6], 1)
[U L V] = svds(S,9);
disp(U)
disp(V)
disp(diag(L)')
```

```
    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
```



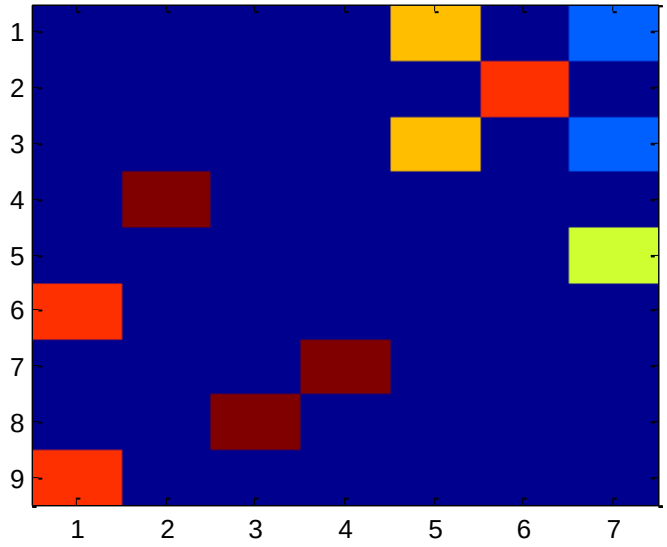


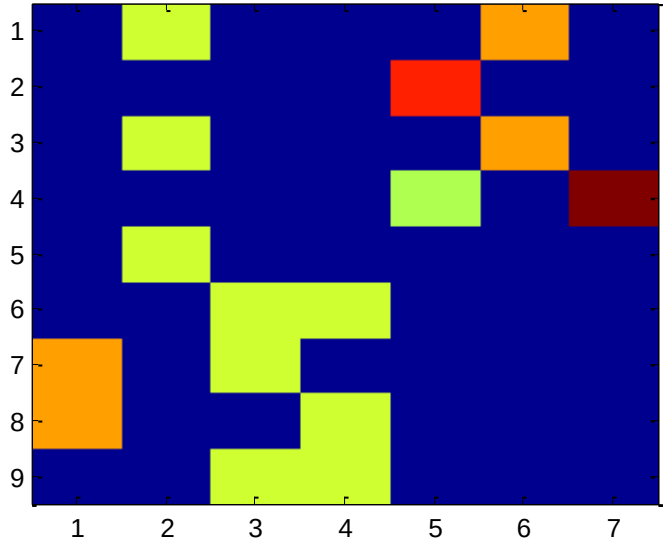
Неотрицательные матричные разложения




```
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**

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

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

Вычислить $k = \deg(A)$**,**

$$B = A - \frac{1}{2m}kk^{\mathrm{T}}\text{,}$$

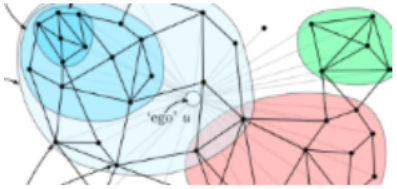
Найти max с.в. $Bv = \lambda v$

$$\mathrm{sgn}(v)$$

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

Enter/Merge by

Tue 6 May 2014 Tue 28 Oct 2014 (27 days to go)

Dashboard

Home
- Data
- Make a submission

Information
- Description
- Evaluation
- Rules
- FAQ
- Timeline

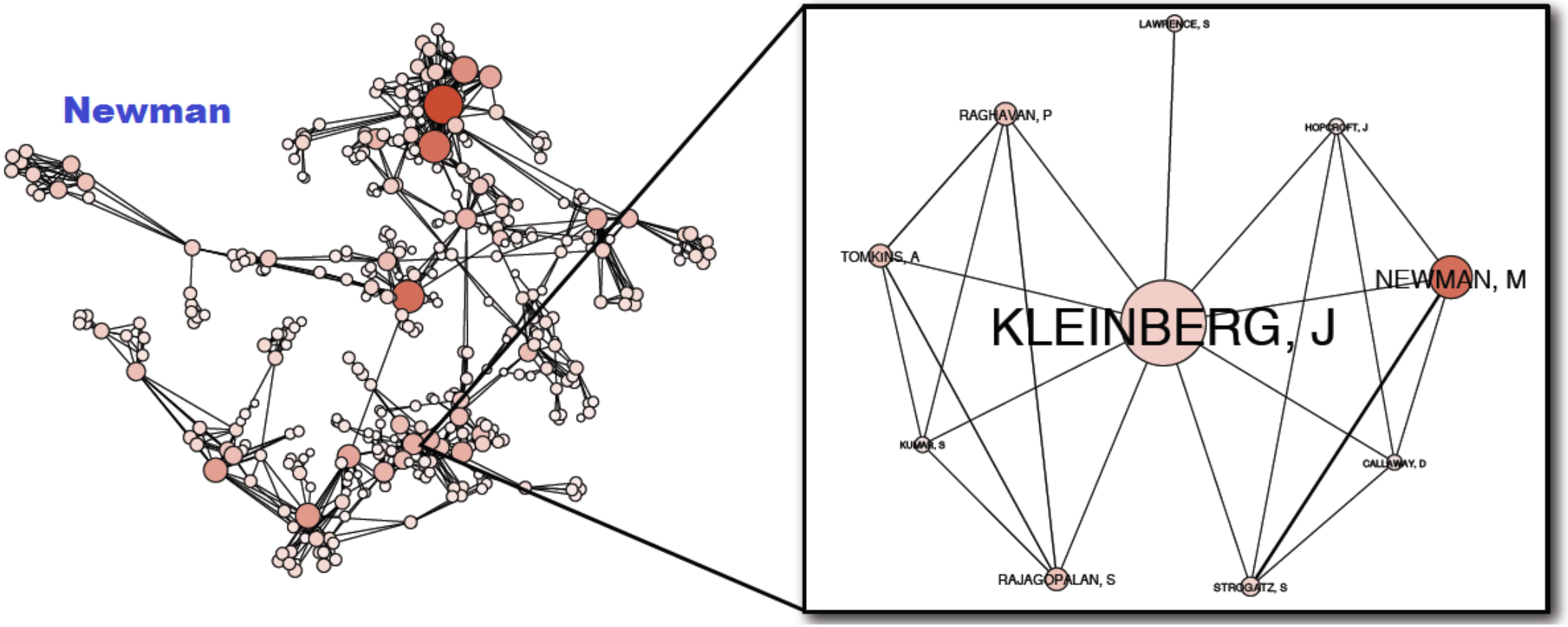
Forum

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.

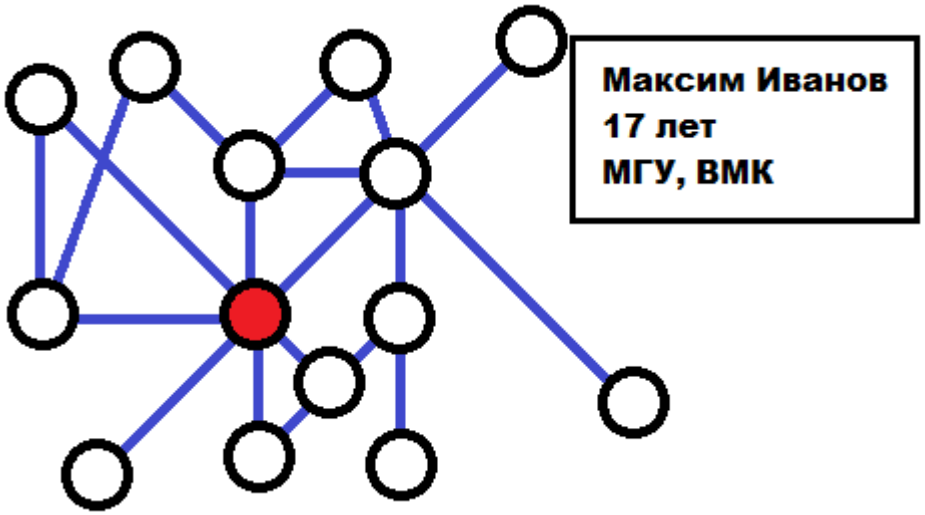
Эго-подграфы



окрестность порядка 1

(не обязательно связный граф – без порождающей вершины)

Задача определения кругов



Здесь: соцсеть =

граф + признаки вершин

Круг – подмножество друзей

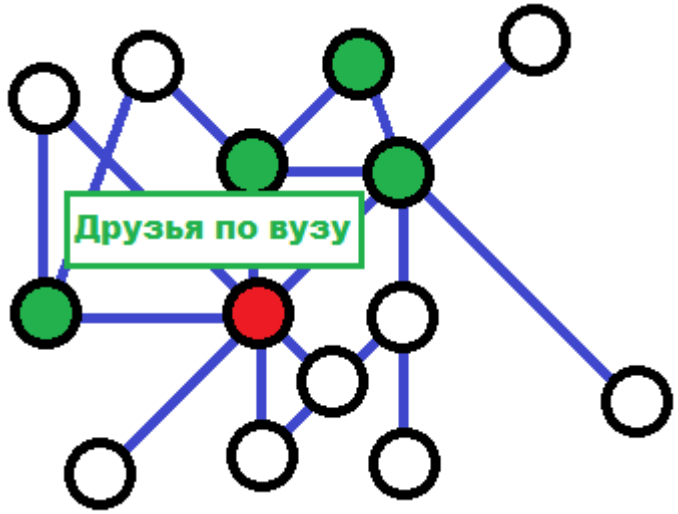
Определяет пользователь

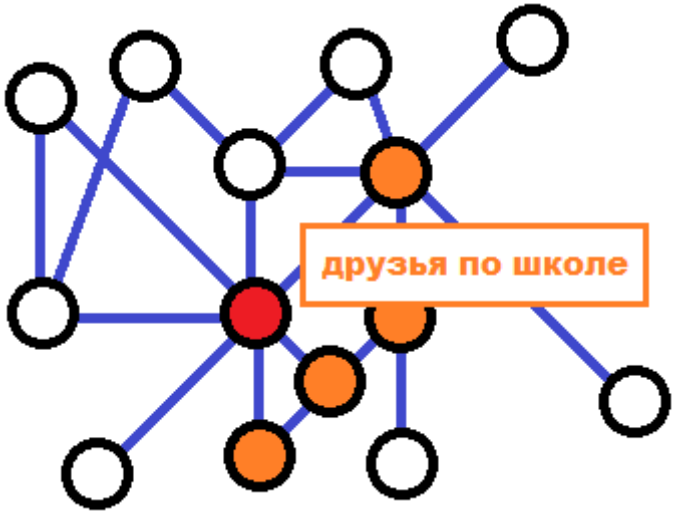
Себя в круг не включает

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

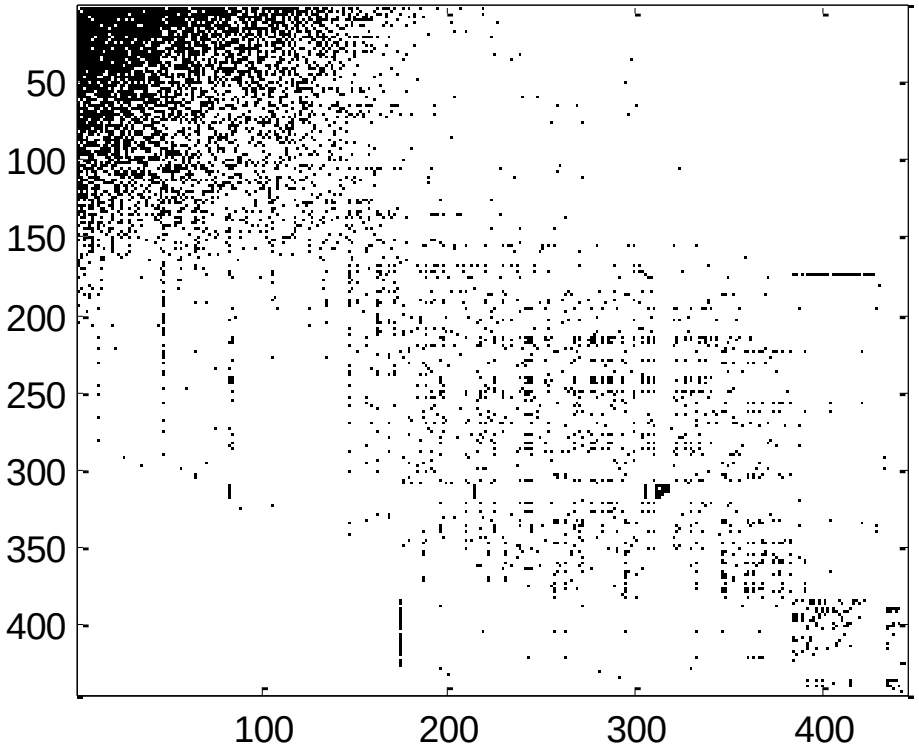

```
[U L V] = svds(M*M' + alpha*X*X', k_svd);
```

К сожалению, нет хорошего эффекта…

Вопрос: какую матрицу раскладывать,

смежности, Лапласа, с нормировками…

Оправдание алгоритма



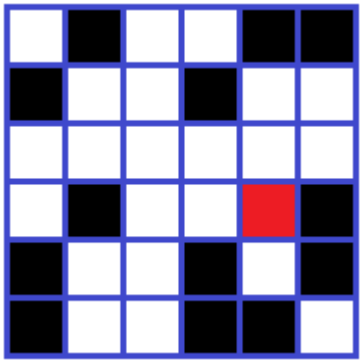
Матрица смежности (упорядоченность вершин по первой компоненте) действительно, есть факторизация

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

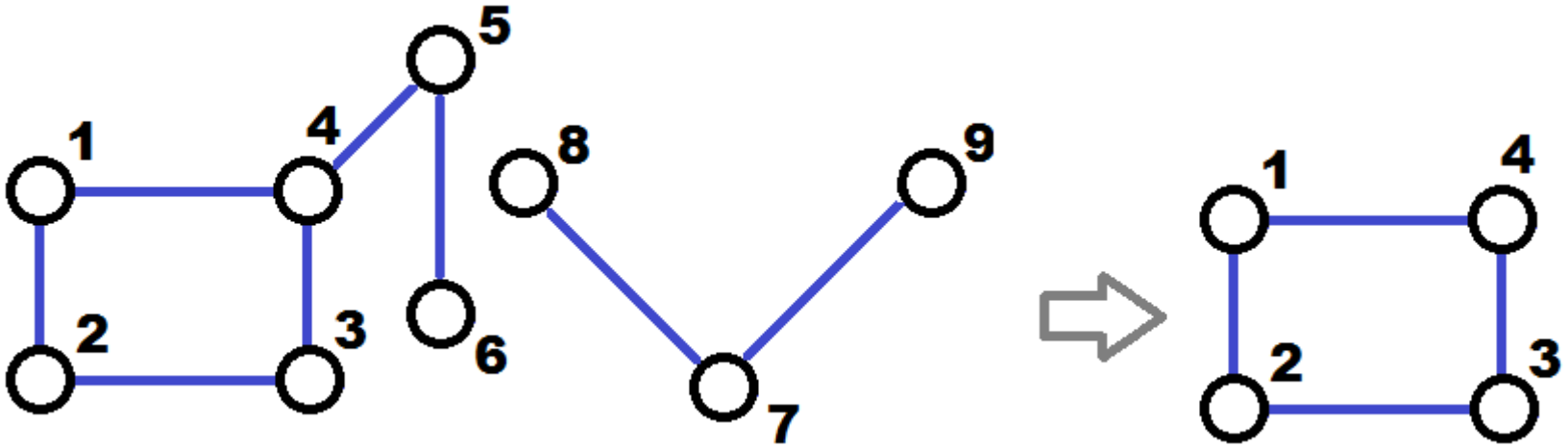
Этапы алгоритма

1. Получение матрицы смежности (симметризация)

не все матрицы были симметричными



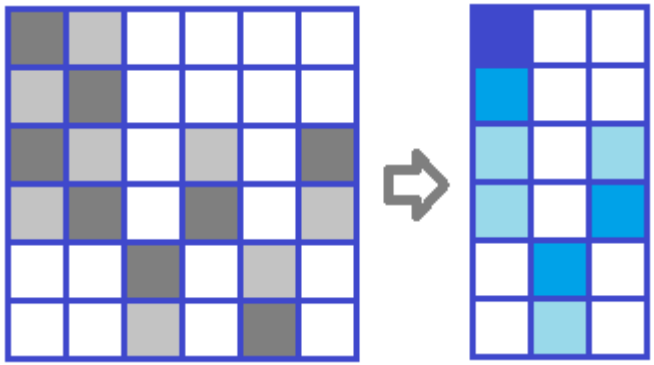
2. Удаление висячих вершин



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

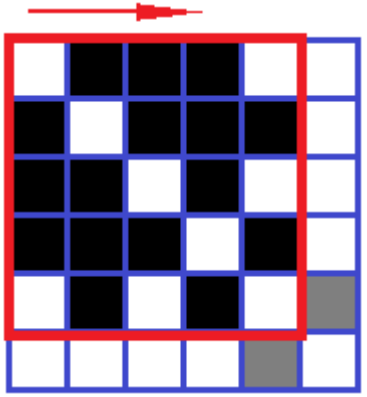
4. Добавление круга

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);
```



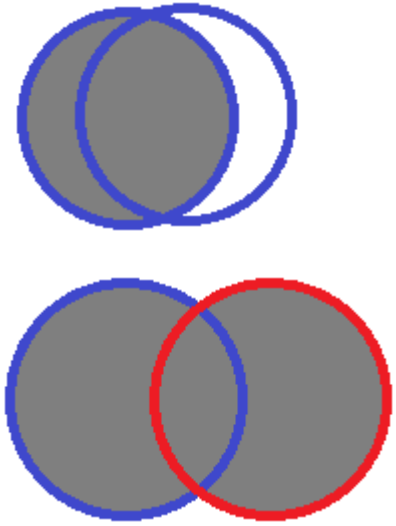
Этапы алгоритма

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

5. Окончательное добавление

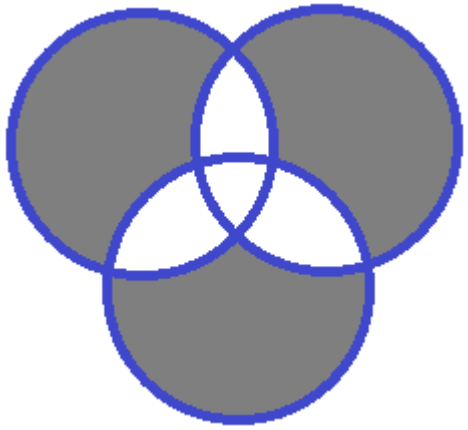
Если большое пересечение с уже существующим – не добавлять

```
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

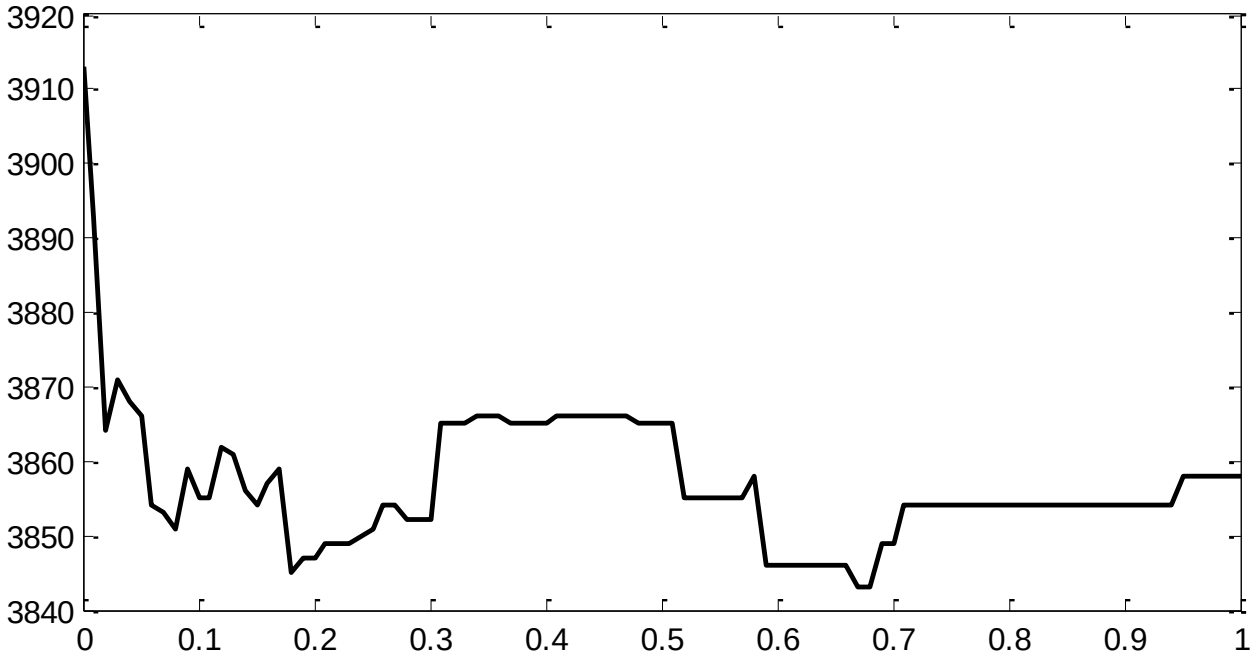
6. Удаление пересечений



следует из функционала качества

1) Настройка параметров

Типичная картинка



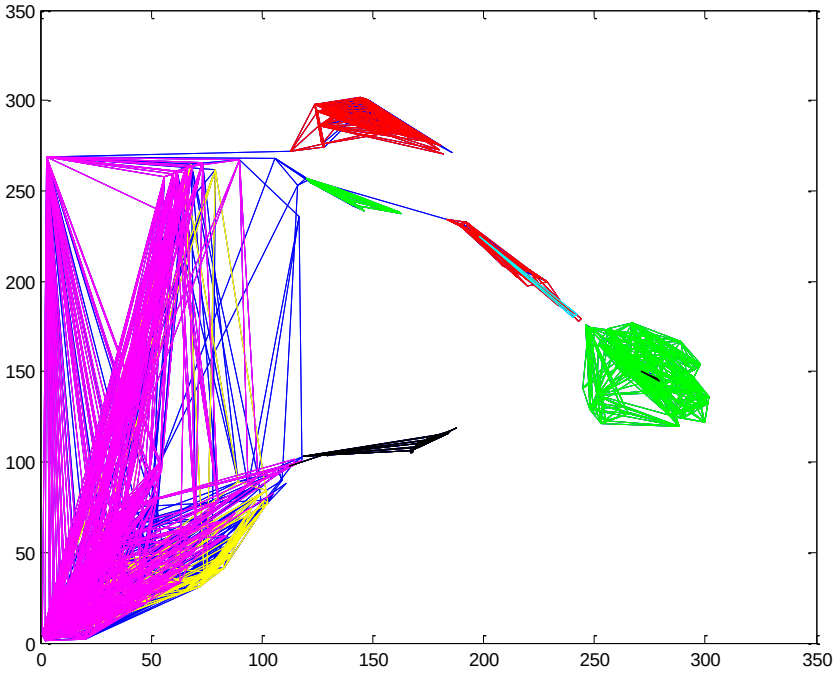
порог в добавлении кругов.

Уже по картинке видно:

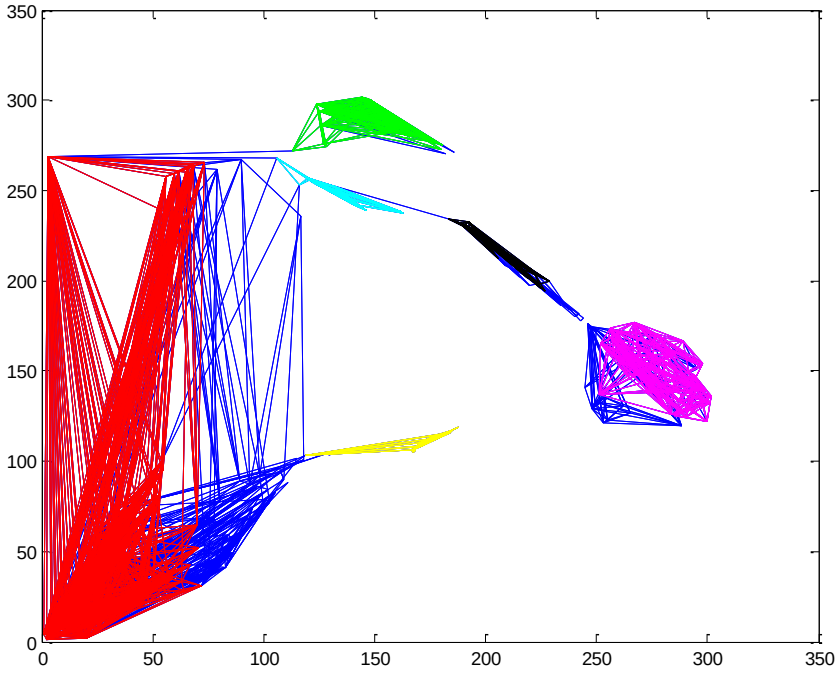
Мало статистики!!!

Работа алгоритма

Визуализация по 1й и 2й SVD-компоненте



правильный ответ

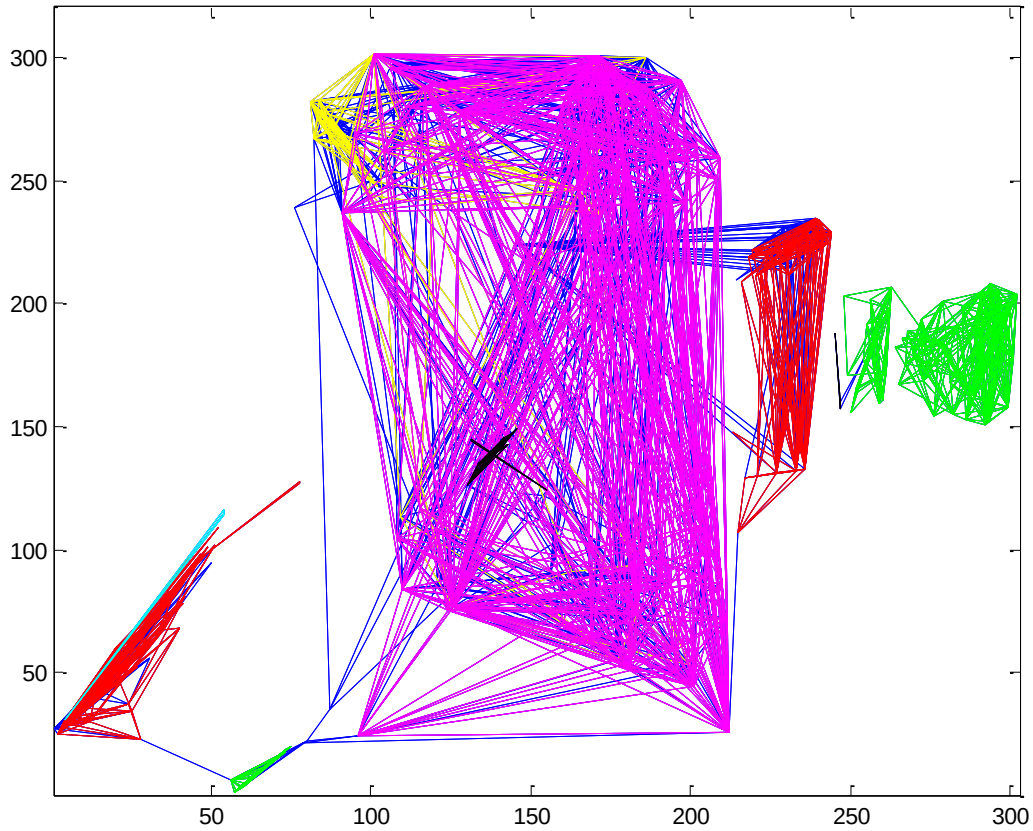


ответ алгоритма

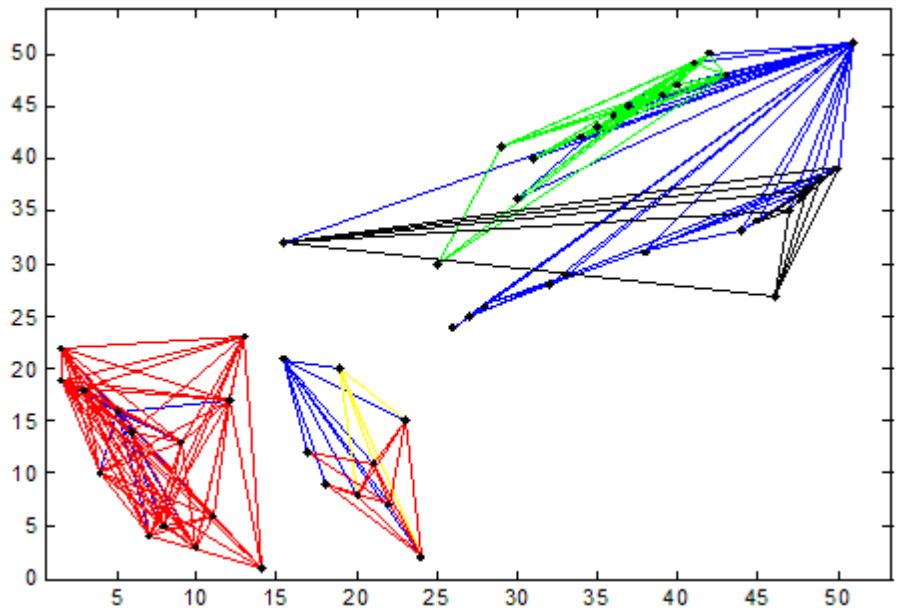
Хитрость: координаты – не значения компонент, а tiedrank…

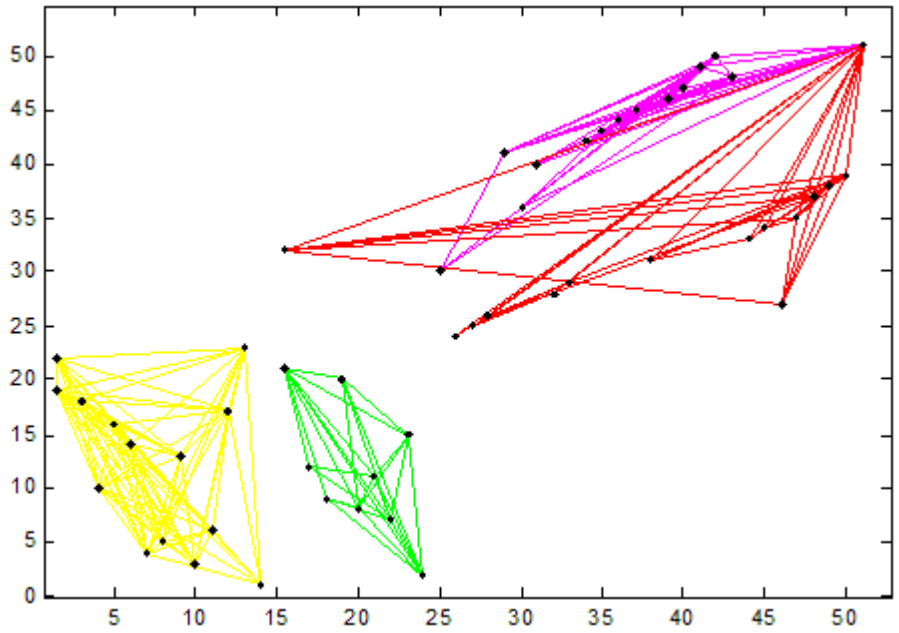
Работа алгоритма

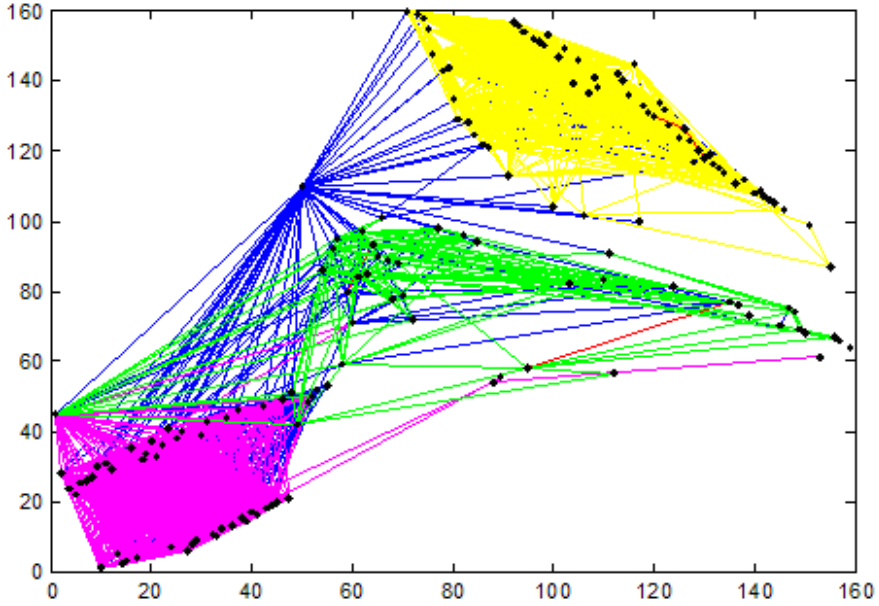
Визуализация по 3й и 4й SVD-компоненте

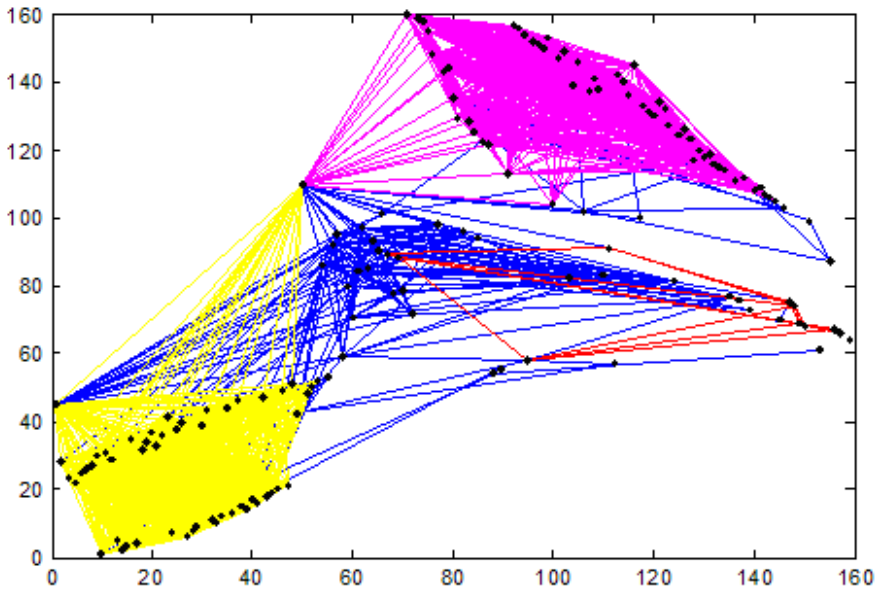


Работа алгоритма



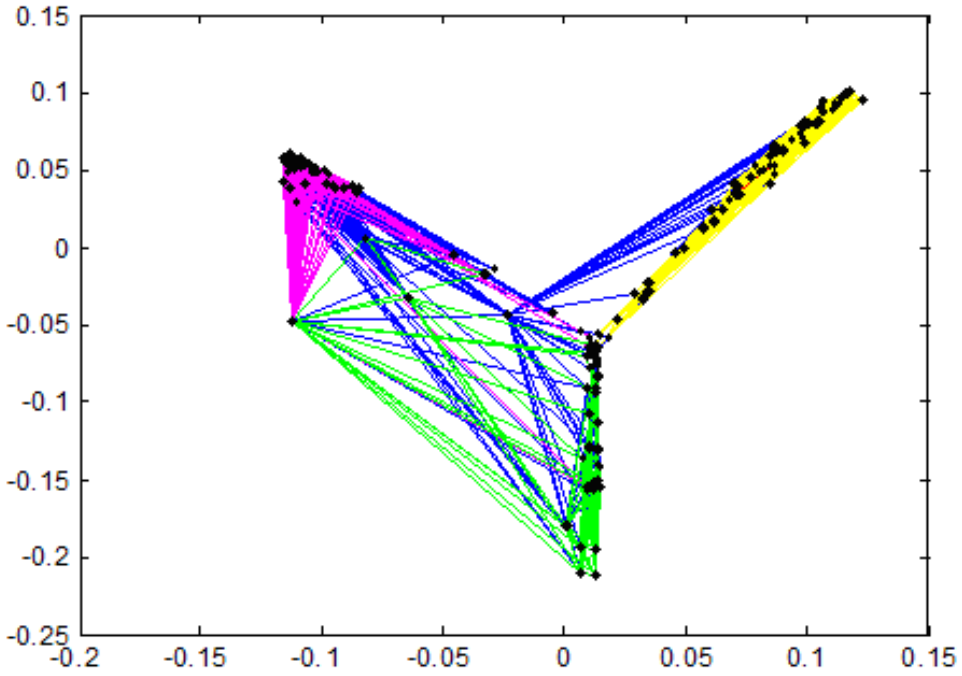






MDS

Можно проецировать граф на плоскость с сохранением расстояний



Но получается не очень информативно

Что можно было сделать ещё…

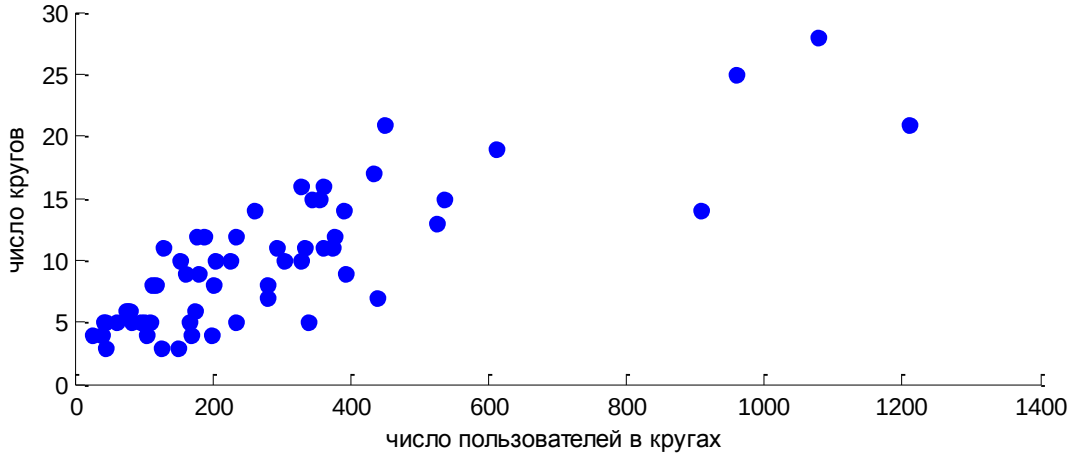
1) кластеризация в пространстве первых компонент SVD

(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…



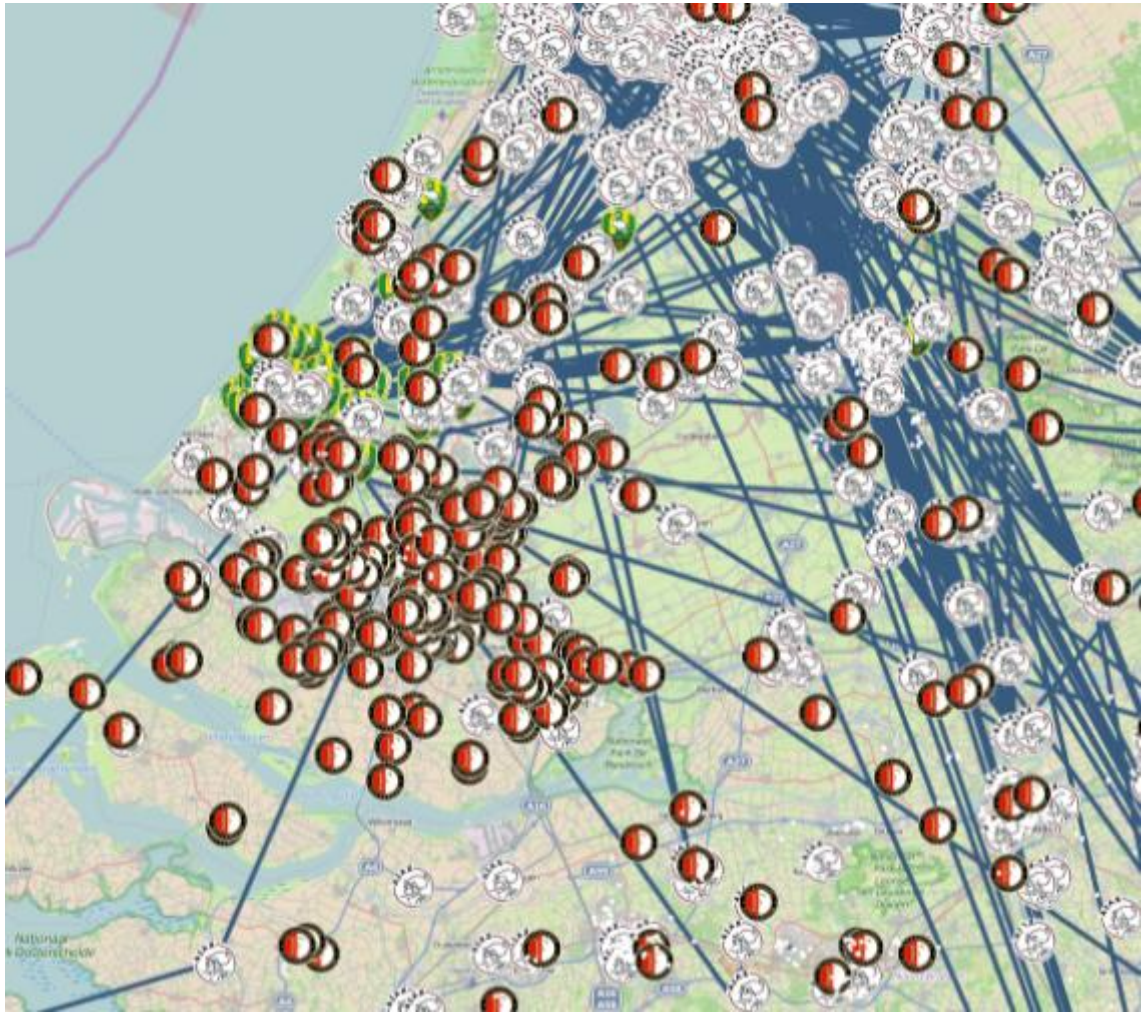
но, как правило, это не работает!

4) объединение ответов кластеризаторов

(собственно, уже делал через SVD – хорошая тема)

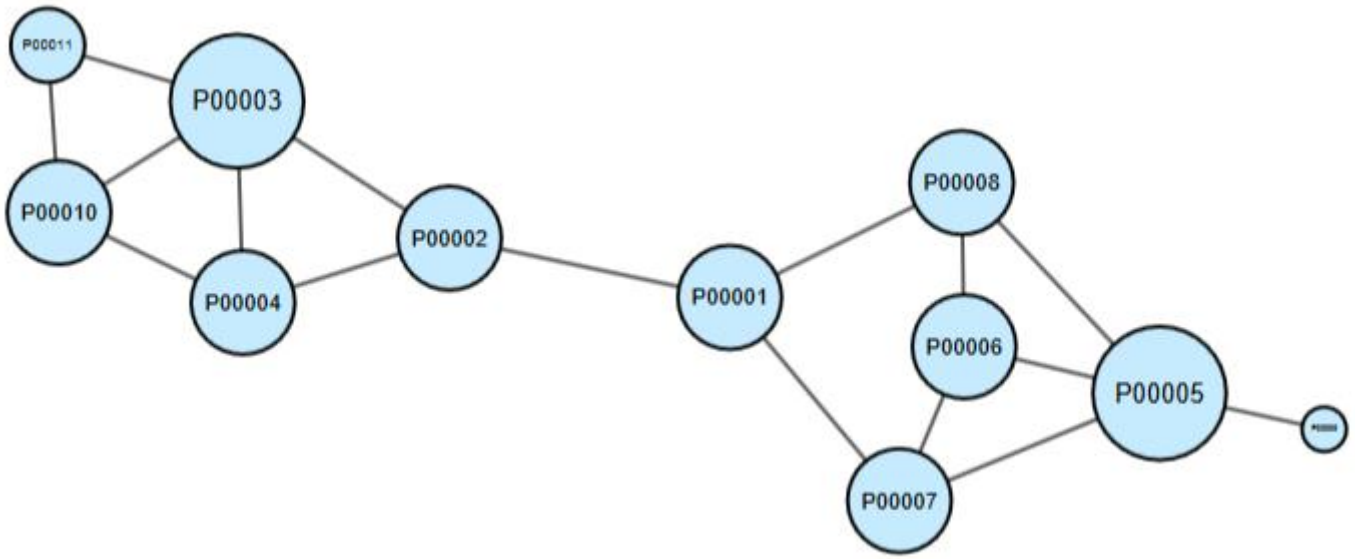
Case: анализ фанатских сообществ

Проект Dutch National Police



Статистика преступлений футбольных фанатов

Case: анализ фанатских сообществ



Строим граф фанатов

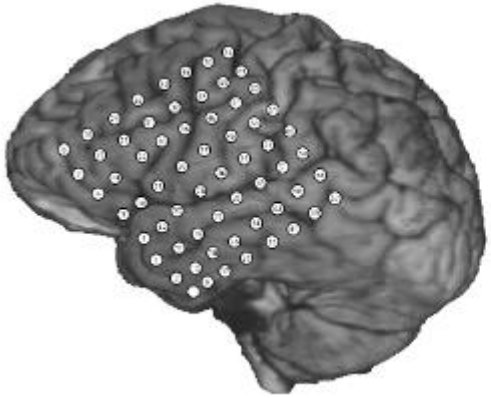
находим самые важные вершины

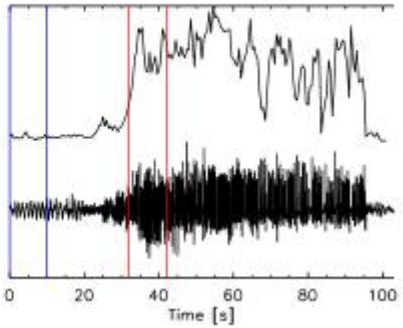
http://liacs.leidenuniv.nl/~takesfw/SNACS/lecture3.pdf

Case: детектирование эпилепсии

Приступы ~ ненормальная нейронная активность

Electrocorticogram (ECoG)



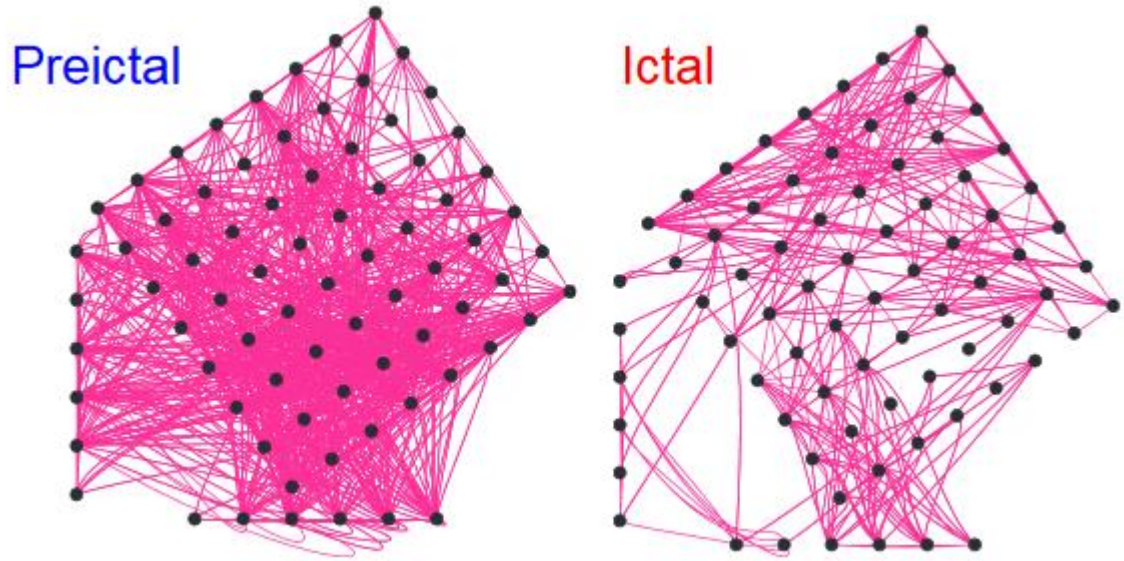


M. A. Kramer et al, «Emergent network topology at seizure onset in humans» // Epilepsy Res.,vol. 79, pp. 173-186, 2008

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.)

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

Очень хороший

Hadi Amiri «Social Media Computing - CMSC 498J»

http://legacydirs.umiacs.umd.edu/~hadi/cmsc498j/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