Data Mining algorithms

2017-2018 spring

03.23.2018

1. Association rules
2. Recommender systems
3. Complex networks (PageRank, Hits, Generative models)

Apriori

Association rules:

Example:

I={bread, dyper, milk}

Disjoint sets:
X={bread,dyper}
Y={milk}

conf(X -> Y): s(S,Y)/s(X) = 2/3

Basket	Content
1	bread, milk
2	bread, dyper, beer, egg
3	milk, dyper, beer, coke
4	bread, milk, dyper, beer
5	bread, milk, dyper, coke

lift(Y |X) :P(X,Y) / P(X) P(Y) = P(Y|X)/ P(Y) =0.667/0.8 = 0.833

Apriori

Association rules:

Example:

I={bread, dyper, milk}

Disjoint sets:

X={bread,dyper}

Y={milk}

conf(X -> Y): s(S,Y)/s(X) = 2/3

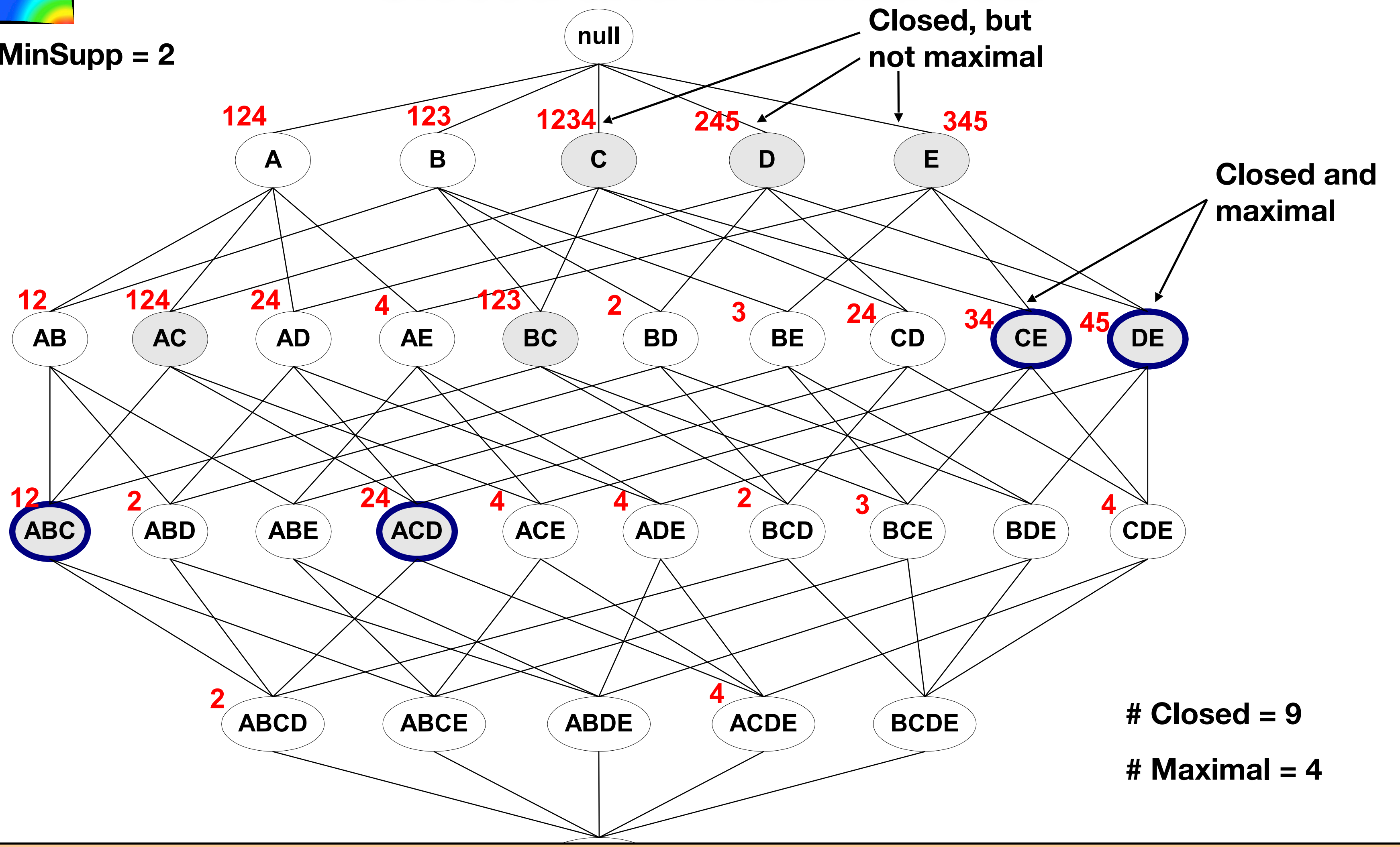
Basket	Content
1	bread, milk
2	bread, dyper, beer, egg
3	milk, dyper, beer, coke
4	bread, milk, dyper, beer
5	bread, milk, dyper, coke

lift(Y |X) :P(X,Y) / P(X) P(Y) = P(Y|X)/ P(Y) =0.667/0.8 = 0.833

negatively correlated

Closed and maximal sets

MinSupp = 2



Closed = 9

Maximal = 4

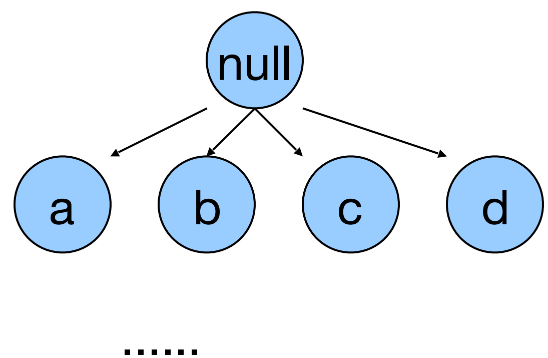
Closed: none of its subsets has the same support
Maximal: none of its subsets is frequent

Closed and maximal sets

Transaction	
ID1	{a,b,d}
ID2	{b,c,d}
ID3	{a,c}
ID4	{a,c,d}

#1-itemsets: 4
MinSupp: 2

APRIORI:
How efficient if
a) we have a fast cpu/io, but small memory
b) slow cpu/io, but a lot of memory?



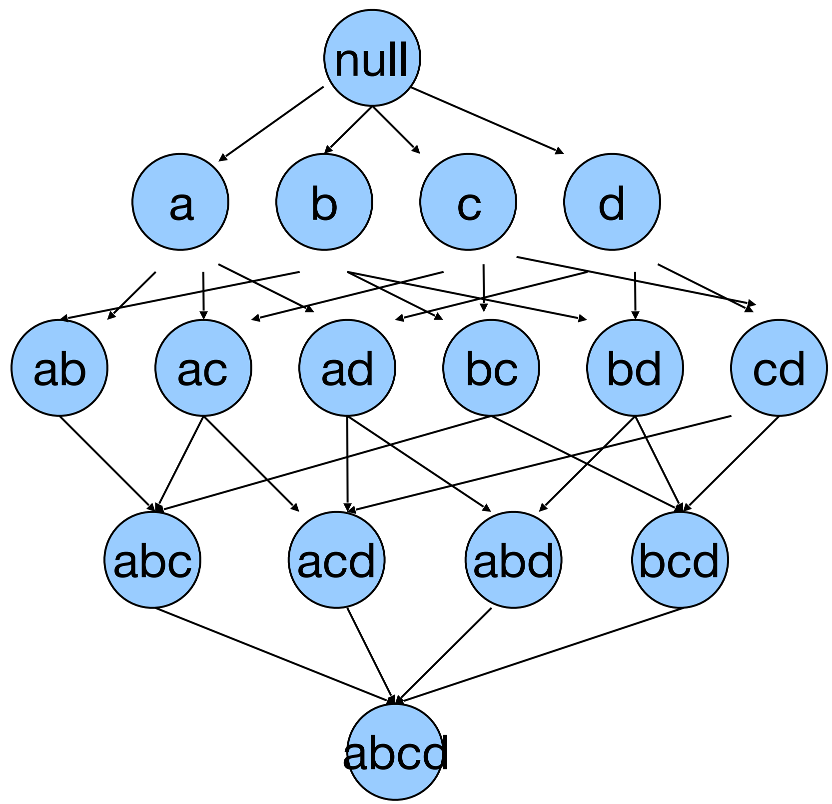
Largest itemset?
Frequent itemsets?
Closed and maximal itemsets?

Closed: none of its subsets has the same support
Maximal: none of its subsets is frequent

Closed and maximal sets

Transaction	
ID1	{a,b,d}
ID2	{b,c,d}
ID3	{a,c}
ID4	{a,c,d}

#1-itemsets: 4
MinSupp: 2



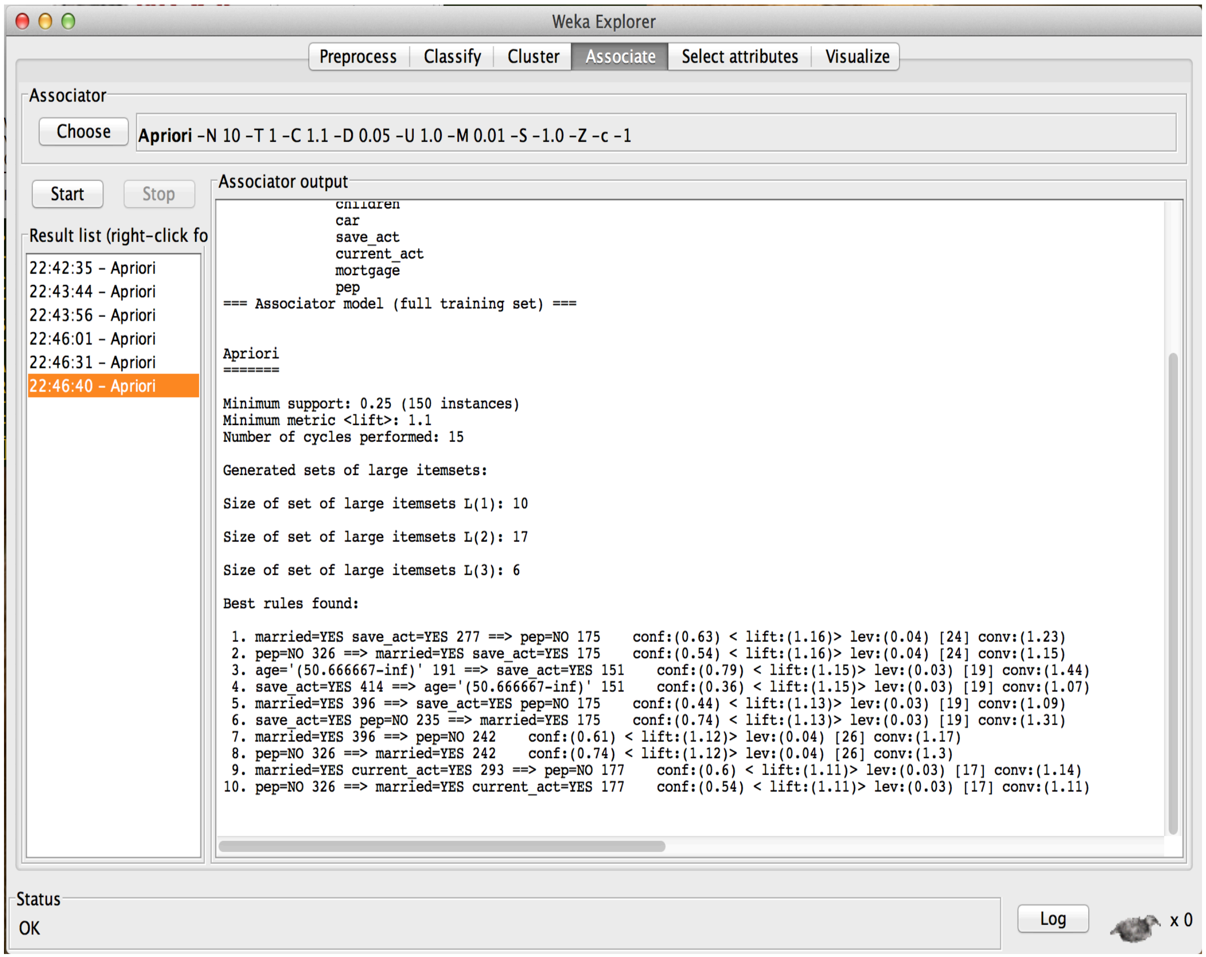
Frequent itemsets:
{a},{b},{c},{d},{a,c},{a,d},{b,d},{c,d}

Closed: {a},{c},{d},{a,c},{a,d},{b,d},{c,d}
Maximal: {a,c},{a,d},{b,d},{c,d}

Closed: none of its subsets has the same support
Maximal: none of its subsets is frequent

Weka

bank_data.arff



$$conf\,(Y|X) = \frac{P(X,Y)}{P(X)}$$

$$lift\,(Y|X) = \frac{P(X,Y)}{P(Y)P(X)} = \frac{P(Y|X)}{P(Y)}$$

$$lev\,(Y|X) = P(X,Y) - P(Y)P(X)$$

$$conv\,(Y|X) = \frac{1 - supp\,(Y)}{1 - conf\,(\mathrm{X} \text{->} \mathrm{Y})}$$

Recommender systems

Given: user-item pairs (implicit), user-item ratings (explicit)

Goal: recommend items to users (or?)

Example
user → movie/song/theatre/restaurant etc.

Basic questions:

- How to measure the performance?
- Outliers: too much variance, too many ratings (e.g. 17k)
- The distribution changes rapidly
  Terminátor 2 in 1991 or in 2017
- Special data:
  - sparse, very sparse (99% of the ratings are missing)
  - the missing values are also valuable
  - large graph

Recommender systems

Content based filtering (CB):
Some meta is given about the items or/and the users

Collaborative Filtering (CF):
Previous ratings

Nearest-Neighbour (NN):
some similarity measure

Latent factor:
matrix factorization (SVD,SGD)

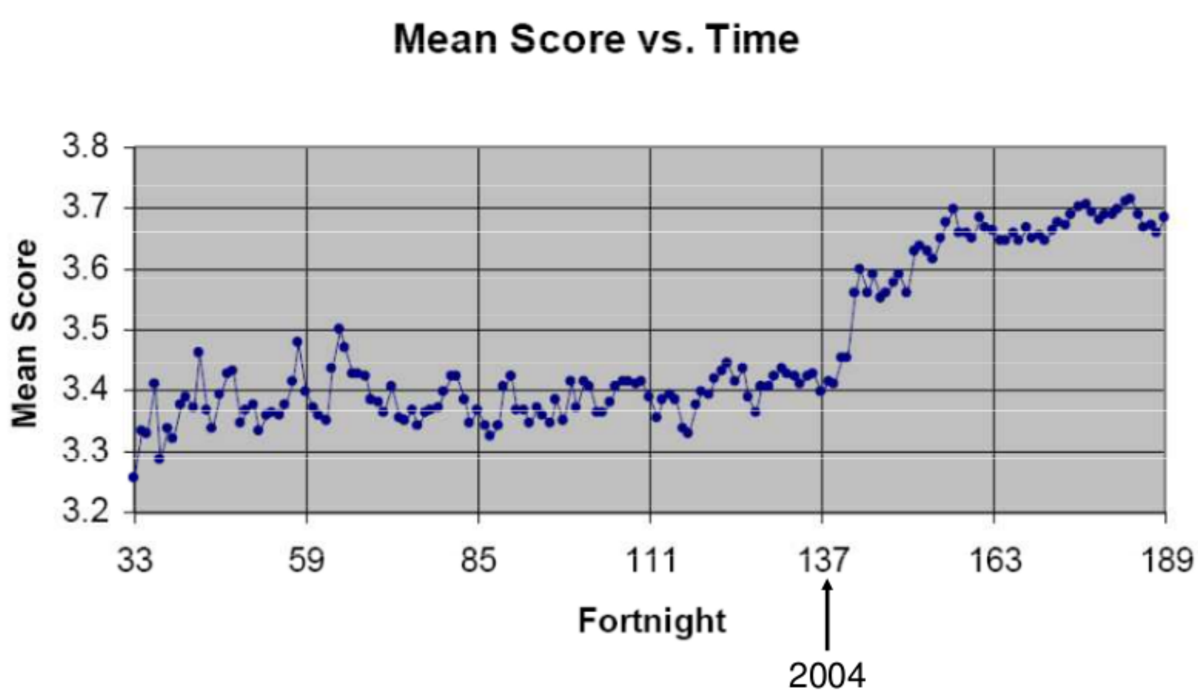
Artificial Neural Networks:
Restricted Boltzmann Machines (by NN)

Recommender systems

Time: where we can utilize time?

1. The model was built on previous interval -> the distribution is not the same
2. Even the preferences change in time
3. Rapid: news recommendation?

Netflix: what happened in 2004?



Example: MovieLens

User/entity:

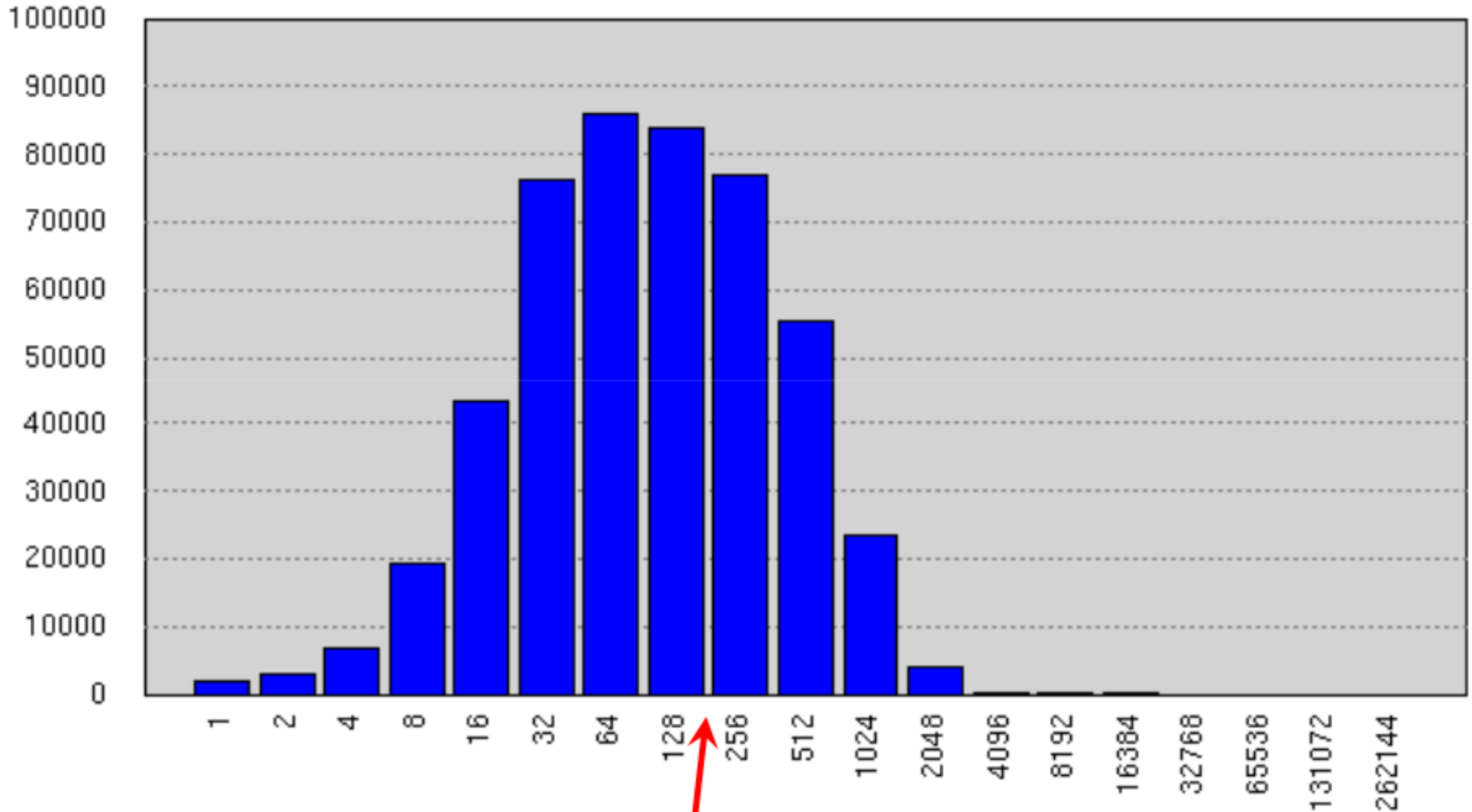
- Not representative: <2% of the ratings are known
- Zero variance?
- Less trustable
- Content:
  - Age
  - Gender
  - Birth place
  - Live in elsewhere
  - Marital status
  - Childern
  - Education
  - job
  - Etc.

Are they important?

Netflix

User ID	# Ratings	Mean Rating
305344	17,651	1.90
387418	17,432	1.81
2439493	16,560	1.22
1664010	15,811	4.26
2118461	14,829	4.08
1461435	9,820	1.37
1639792	9,764	1.33
1314869	9,739	2.95

Netflix



Avg #ratings/user: 208

Example: MovieLens

Item:

- Not representative: same reason
- Std. var. is also a problem
- Content (more trustable?):
    - genre
    - director/director of cinematography etc.
    - TV/movie etc.
    - actors
    - year
    - original language/origin

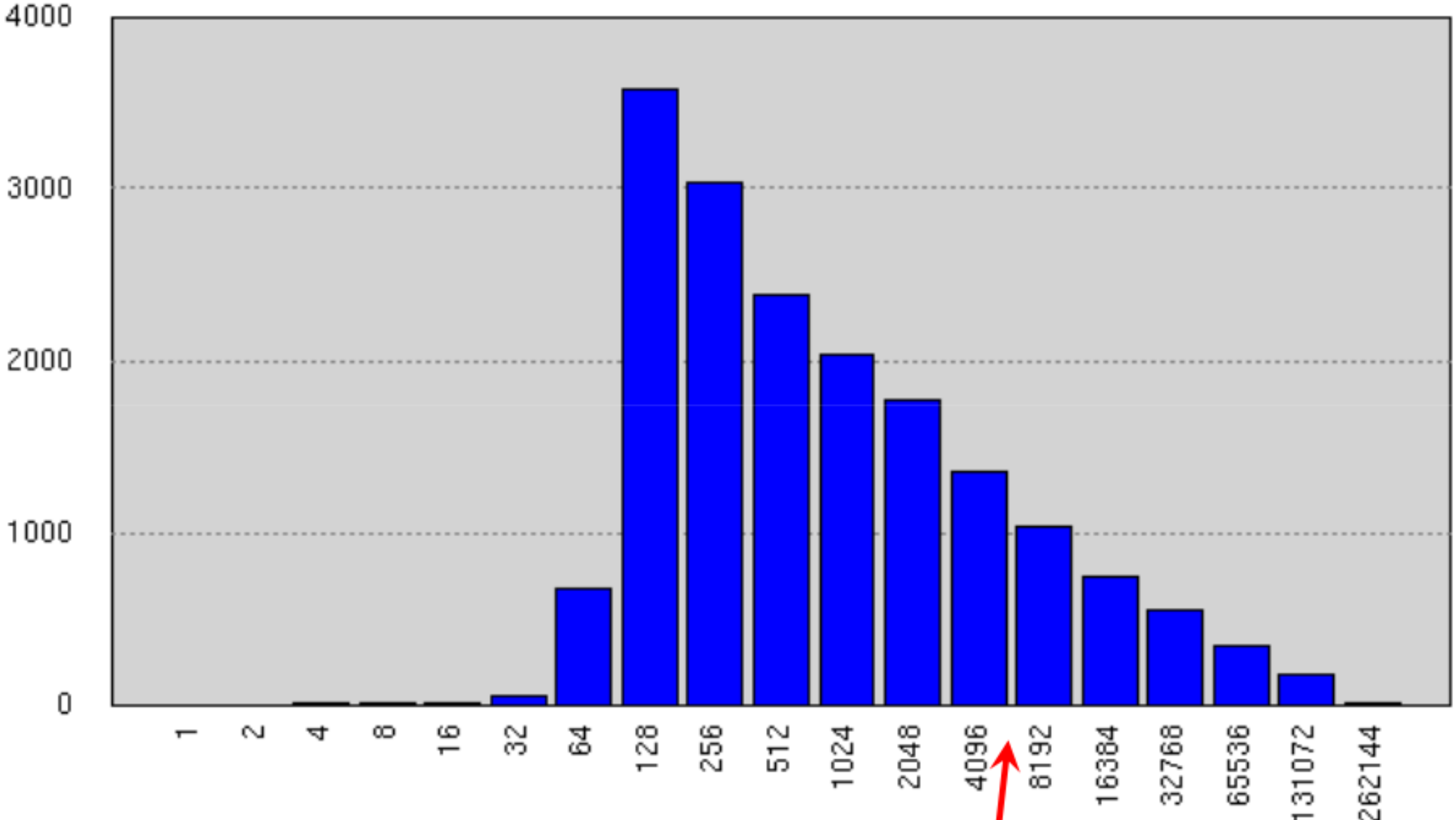
Netflix

Most Loved Movies	Avg rating	Count
The Shawshank Redemption	4.593	137812
Lord of the Rings :The Return of the King	4.545	133597
The Green Mile	4.306	180883
Lord of the Rings :The Two Towers	4.460	150676
Finding Nemo	4.415	139050
Raiders of the Lost Ark	4.504	117456

Most Rated Movies
Miss Congeniality
Independence Day
The Patriot
The Day After Tomorrow
Pretty Woman
Pirates of the Caribbean

Highest Variance
The Royal Tenenbaums
Lost In Translation
Pearl Harbor
Miss Congeniality
Napolean Dynamite
Fahrenheit 9/11

Netflix



Avg #ratings/movie: 5627

Collaborative Filtering

User-item graph:

- Bipartite graph: each node is either a user or an item
- There are only edges between users and items
- Is it directed? (need to be?)
- Presumption: less than 1% is known

Ratings matrix: R (in case of explicit)

- Sparse
- discrete (implicit: 0/1/(-1), explicit: 1-5 ...)

Collaborative Filtering: NN

K-Nearest Neighbour (Bell-Koren):

Hypothesis: “like-minded” entities are having similar ratings

Three main parts:

1) normalization
2) Identification of neighbours
3) Weight determination (similarity)

Collaborative Filtering: NN

Given n users and m items.

Recommendation: estimate $R=\{r_{ui}\}$

u : user

i : item

Hypothesis:

$$r_{ui} = \frac{\sum_{v \in N(u,i)} s_{uv} r_{vi}}{\sum_{v \in N(u,i)} s_{uv}}$$

where s_{uv} is the similarity of users u and v

Collaborative Filtering: NN

Same but based on the items:

$$r_{ui} = \frac{\sum_{j \in N(i,u)} s_{ij} r_{uj}}{\sum_{j \in N(i,u)} s_{ij}}$$

Which one to choose?

Notes (disadvantages):

- similarity measure? (Pearson, cosine, Jaccard etc.)
- overweight correlated movies:
  LOTR^3,Terminator^3, Harry Potter 1-8 etc.
- a five years old rating is treated as a recent one
  (is it a problem?)

Collaborative Filtering: NN

Normalization

Let be the ratings:

$$r_{ui} = \theta_u x_{ui} + err$$

where x_{ui} is the connectedness of u and i (e.g. if the user only saw small number of movies, it is high) and

$$\theta_u = \frac{\sum r_{ui} x_{ui}}{\sum x_{ui}^2}$$

Collaborative Filtering: NN

Normalization:

- time factor: the normalization is root time from the first occurrence of the user or item

- outlier: smaller weight, measured by difference from the mean, number of ratings etc.

5-10% increase over a simple CF NN model.

Collaborative Filtering: NN

Weight determination:

By default it is uniform.

Assumption: linear combination of previous ratings of the user approximate well the actual rating:

$$r_{ui} = \sum \omega_{ij} r_{uj}$$

Loss function:

$$min\left(r_{ui} - \sum \omega_{ij} r_{uj}\right)^2$$

Collaborative Filtering

Notes on NN:

- small models (VC-theorem)
- interpretable:
  list of similar items (e.g. Amazon)
- not so complicated to implement
- could be slow

Latent models: matrix factorization

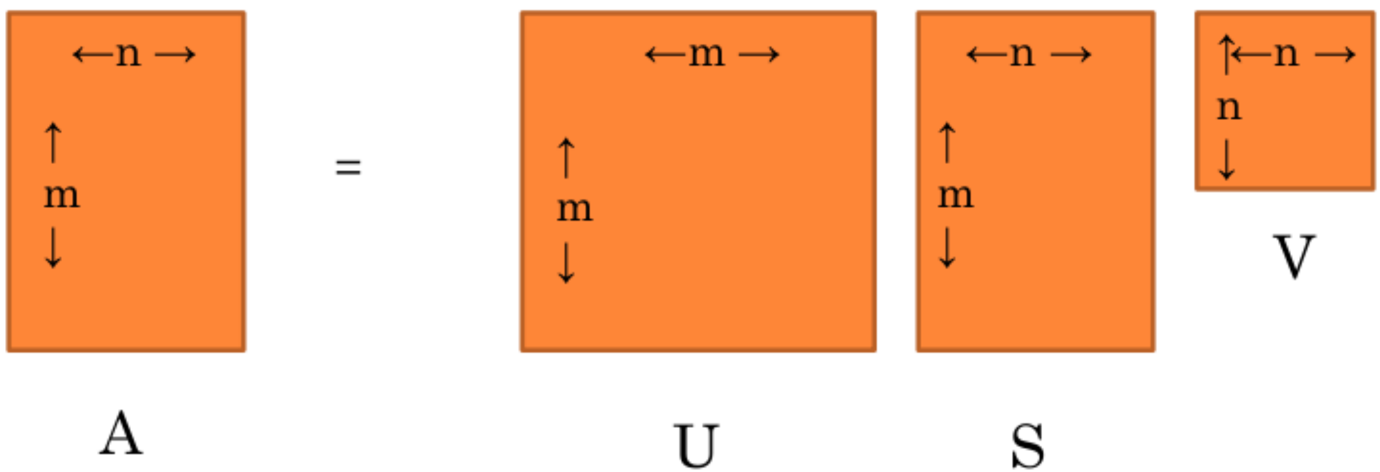
Alternating Least Squares
Stochastic Gradient Descent
Singular Value Decomposition

Let be R a matrix with $m \times n$ and $k<m$, $k<n$, then we approximate $R^k = PQ^T$ where $P = P_{m \times k}$ és $Q = Q_{n \times k}$:

$$min \| R - R^k \|_{Frobenius}$$

Singular Value Decomposition

$$SVD(A) = U \times S \times V^T$$



Singular Value Decomposition

SVD is not suitable:

- complexity
- missing values?
- regularization?

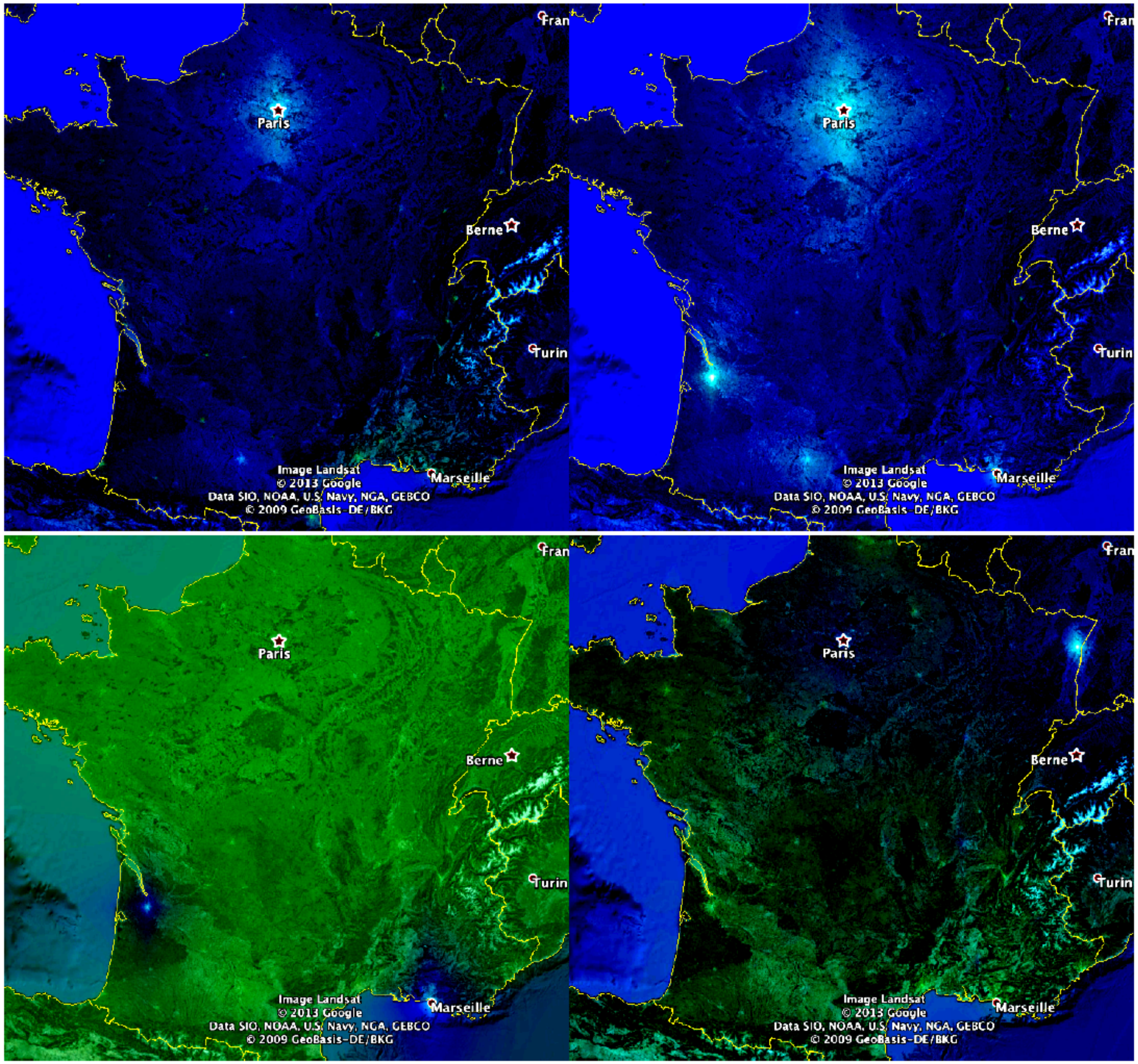
Presumed:
- S diagonal and rank is r $S_1 > S_2 > S_3 ... > S_r$
- U and V orthogonal
- Frobenius norm:

$$\|R\|_{Frobenius} = \sqrt{\sum_{ij} |r_{ij}|^2}$$

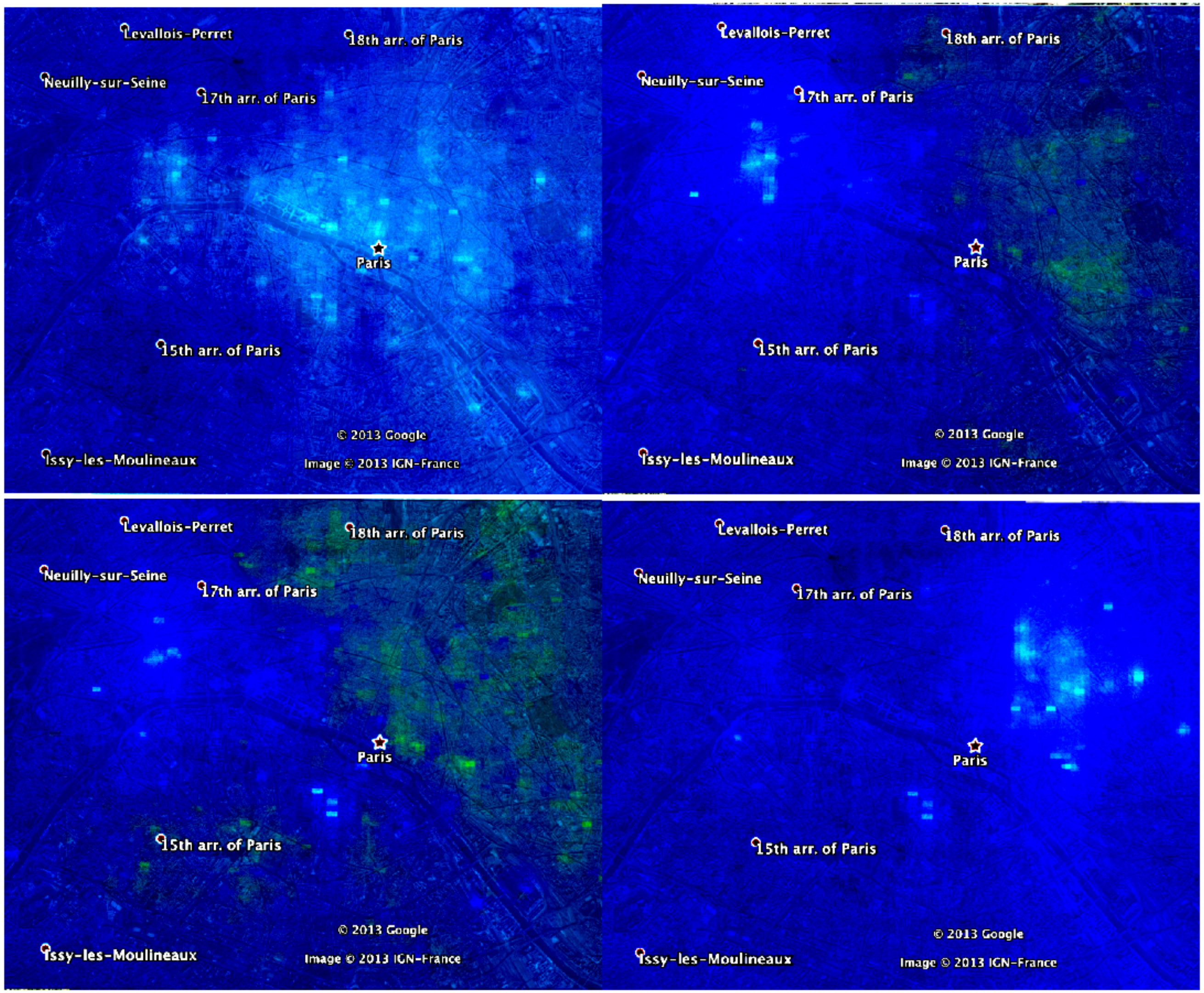
Low (k) rank approximation
Predicted values (let be the number of factors k, k << n és k << m):

$$r_{ui} = \bar{r}_u + \left(U_k \sqrt{S_k}^T (u)\right)\left(\sqrt{S_k}^T V_k(i)\right)$$

Example: first 4 factors of restaurants in France



Example: first 4 factors of restaurants in Paris



Collaborative Filtering SGD

Stochastic Gradient Descent

In comparison to SVD:
- optimization over the known ratings
- low complexity -> fast
- but in case of implicit: negative samples are needed (how?)

Predicted ratings: $$r'_{ui} = p_u q_i = \sum_{k=1}^{K} p_{uk} q_{ik}$$

$$err = \sum_{v(u,i) \in G(E,V)} (r_{ui} - r'_{ui})^2 = \sum_{v(u,i) \in G(E,V)} \left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik}\right)^2$$

Gradient:

$$\frac{\partial\, err}{\partial\, p_{uk}} = -2\left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik}\right) q_{ik} \qquad \frac{\partial\, err}{\partial\, q_{ik}} = -2\left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik}\right) p_{uk}$$

Collaborative Filtering SGD

Stochastic Gradient Descent

Regularization (prevents overfitting):

$$err = \sum_{v(u,i) \in G(E,V)} \left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik}\right)^2 + \alpha \sum_u \|p_u\|^2 + \beta \sum_i \|q_i\|^2$$

Social regularization
(there is a known social graph):

$$err = \sum \left(r_{ui} - \sum_{k=1}^{K} p_{uk} q_{ik}\right)^2 + \alpha \sum_u \|p_u\|^2 + \beta \sum_i \|q_i\|^2 + \gamma \sum_u \|p_u - \sum_{u' \in N(u)} w(u,u') p_{u'}\|^2$$

Hypothesis: we have similar taste to our friends (do we?)

Implicit vs. Explicit recommendation?

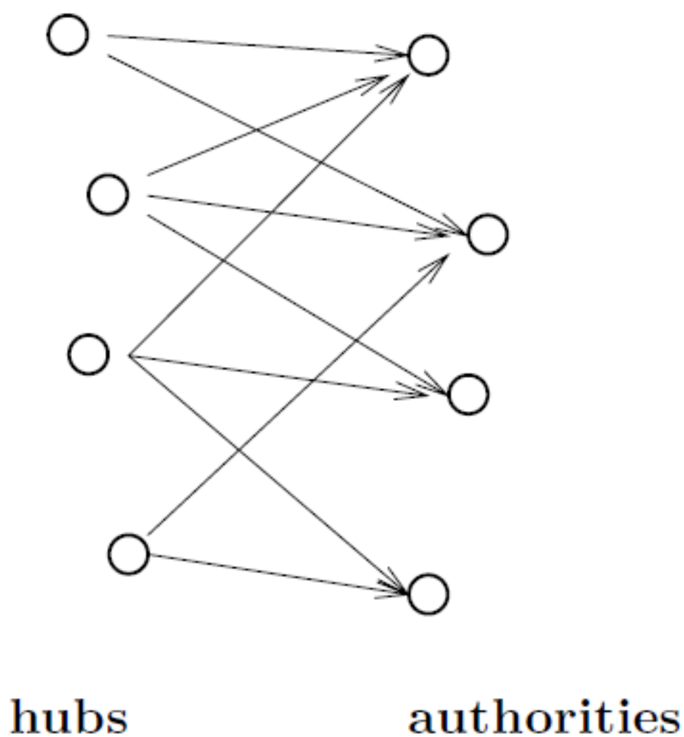
Unsolved problems:

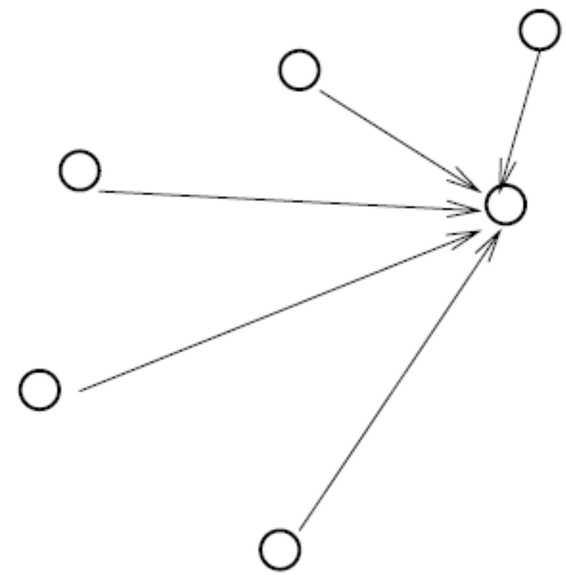
MF models
“Cold start”?
Group recommendation
Evaluation? RMSE vs. nDCG
Rapid changes in ditribution
Online recommendation
Distributed MF
Session based models (Item2Item (Koenigstein, Koren 2013))
Search engine vs. recommender

Content similarity (DNN?)

Web graph: HITS intro

Hyperlink-Induced Topic Search (HITS)





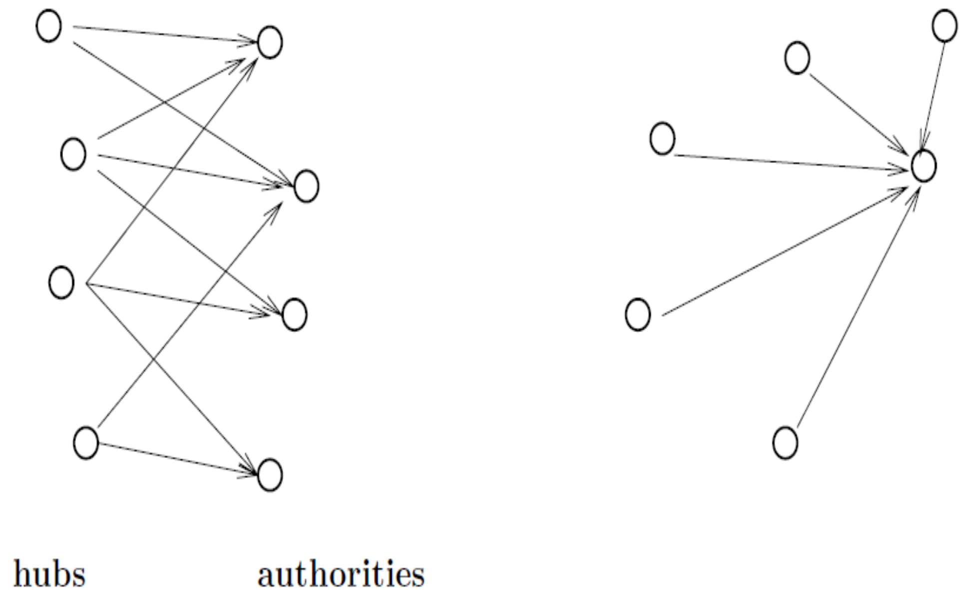
Web graph: HITS

Hyperlink-Induced Topic Search (HITS) Kleinberg ’98

1. Hubs: links to authorities, act as stations
2. Authorities: relevant web sites,

where the good hubs are linking

Goal: identify the hubs/authorities
by two positive scores for a specific query!
(x: authority, y: hubness)



$$I(G, x, y): x^p \leftarrow \sum_{q:(q,p)\in E} y^q$$

$$O(G, x, y): y^p \leftarrow \sum_{q:(p,q)\in E} x^q$$

$$\sum_p (x^p)^2 = 1 \qquad \sum_p (y^p)^2 = 1$$

Web graph: HITS

HITS(G,k,q)

G: a subgraph (focus graph), the set of nodes connected to the hits based on the query q

k: constans

Z: n dimensional real vector (1; 1; 1; : : :; 1)

Let x_0 := z:

Let y_0 := z:

for i = 1,2 .. k

$O(G, x_{i-1}; y_{i-1}) \rightarrow x'$

$I(G, x_{i-1}; y_{i-1}) \rightarrow y'$

Normalize x' -> x_i

Normalize y' -> y_i

Theorem: $(x_1, x_2, x_3, \ldots)$ and $(y_1, y_2, y_3, \ldots.)$ are converging

HITS

Proof: A is the adjacent matrix on the focus graph

$$x^{(k+1)} = y^{(k)} A$$
$$y^{(k+1)} = x^{(k+1)} A^T$$

expand:

$$x^{(k+1)} = x^{(1)} (A^TA)^k = x^{(1)} U W U^T$$

$$y^{(k+1)} = y^{(1)} (AA^T)^k = y^{(1)} V W V^T$$

where W is diagonal.

Theorem: the normalized $x^{(k)}$ and $y^{(k)}$ series are converging
(start with x=y=(1,1,…,1))

HITS

Or equivalently:

$(AA^T)^j y^{(1)}/||(AA^T)^j y^{(1)}||$ and
$(A^TA)^j x^{(1)}/||(A^TA)^j x^{(1)}||$ converging

Lemma: if an nxn matrix positive-semidefinite and symmetric (AA^T and A^TA), and its eigenvalues are $\lambda_1>\lambda_2\geq\lambda_3...\geq\lambda_k\geq 0$ ($k<n$) then for all n dimensional v vector can be expressed as a linear combination of the eigenvectors $\Sigma_{i=1..k}\, \alpha_i\omega^{(i)}$ where for all i $||\omega^{(i)}||=1$ and $\omega^{(i)\,T}\omega^{(j)} = 0$ if $i\neq j$.

Since $\omega^{(i)}$ is the i-th eigenvector: $M\, \omega^{(i)} = \lambda_i\omega^{(i)}$

HITS

And since $AA^T = M$ positive semidefinite and symmetric:

$$\frac{M^j v}{||M^j v||} = \frac{\sum_{i=1}^{k} \alpha_i M^j w^{(i)}}{||\sum_{i=1}^{k} \alpha_i M^j w^{(i)}||} = \frac{\sum_{i=1}^{k} \alpha_i \lambda_i^j w^{(i)}}{\sqrt{\sum_{i=1}^{k} (\alpha_i \lambda_i^j)^2}}$$

$$\frac{\alpha_1 \lambda_1^j w^{(1)} + \sum_{i=2}^{k} \alpha_i \lambda_i^j w^{(i)}}{\sqrt{(\alpha_1 \lambda_1^j)^2 + \sum_{i=1}^{k} (\alpha_i \lambda_i^j)^2}} \cdot \frac{\frac{1}{\lambda_1^j}}{\frac{1}{\lambda_1^j}} = \frac{\alpha_1 w^{(1)} + \sum_{i=2}^{k} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^j w^{(i)}}{\sqrt{\alpha_1^2 + \sum_{i=1}^{k} (\alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^j)^2}} \longrightarrow w^{(1)}$$

src.:Bodon

Random surfer

Hypothesis: random walk over the edges in WWW

If uniform:

$$Pr(i \mid j) = 1/d(j)$$

where the d(j) is the out degree.

Let us define the adjacency matrix of our graph A.

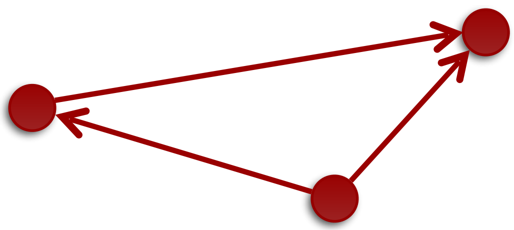
Replace the values with probabilities: M

Sum (norm) of rows and columns?

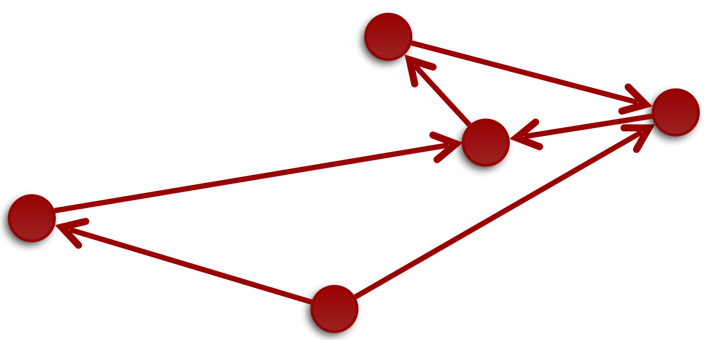
Random surfer

When is it not one?

dead end:

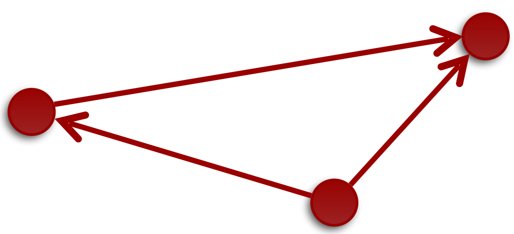


spiderweb:

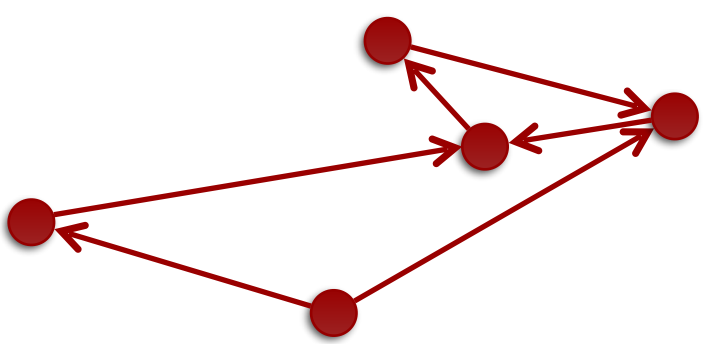


Random surfer

Dead end:



Spiderweb:



Ergodic:

- strongly connected
- aperiodic

If a Markov chain is ergodic:
(Perron-Frobenius):

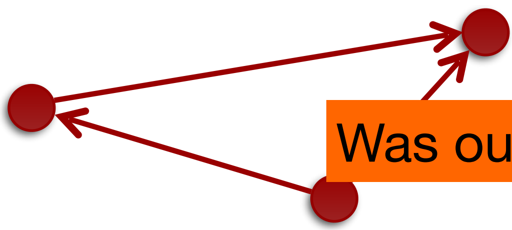
There exists a stationary state:

$$\pi^T M = \pi^T$$

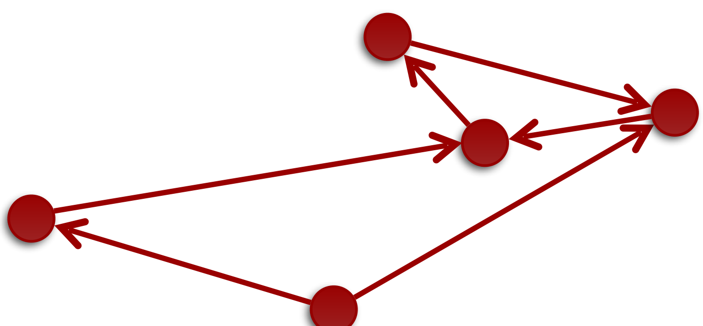
And:
The biggest eigenvalue is 1!

Random surfer

Dead end:



Spiderweb:



Ergodic:

- strongly connected
- aperiodic

Was our original graph ergodic?

(Perron-Frobenius):

Exist a stationary state:

$$\pi^T M = \pi^T$$

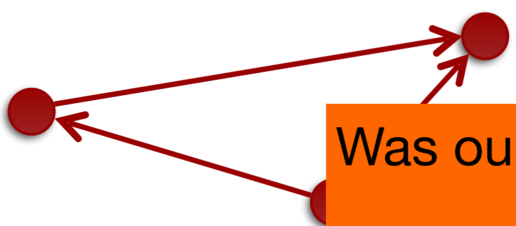
And:
The biggest eigenvalue is 1!

Random surfer

Ergodic:

- strongly connected
- aperiodic

Dead end:

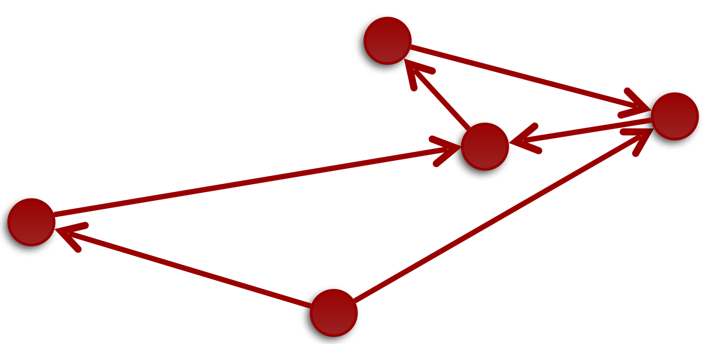


Was our original graph ergodic?

Teleportation?

c:

Spiderweb:



Exist a stationary state:

$$\pi^T M = \pi^T$$

And:
The biggest eigenvalue is 1!

PageRank

Larry Page , Sergey Brin , Rajeev Motwani and Terry Winograd (WWW'98).

Hyperlink graph -> www

Traditional text query based search engines:

- hit: set of relevant documents to query
  (sites containing the query terms)
- ranking? : tf-idf, bm25 etc. -> based on the content!

When it is not efficient or even result false hits at the top?

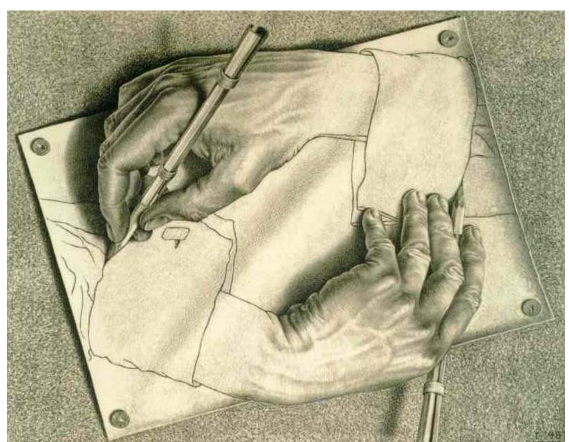
The idea is similar to HITS: the relevant pages are the pages cited by relevant pages

PageRank

The PageRank of a document (page) A is PR(A):

$$PR(A) = \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

where I_A is the set source pages of incoming edges to A and L(B) is the outdegree of node B (PR(B) is the PageRank of B in the last iteration)

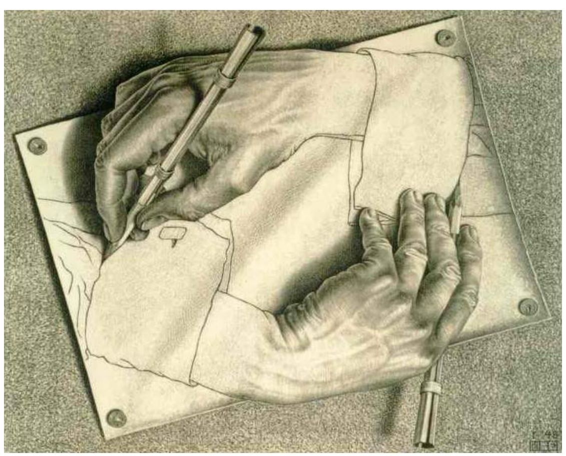


PageRank

The PageRank of a document (page) A is PR(A):

$$PR(A) = \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

where I_A is the set source pages of incoming edges to A and L(B) is the outdegree of node B (PR(B) is the PageRank of B in the last iteration)



What is the connection between the random surfer model and PageRank?

PageRank

"Random surfer" model: activity decrease through surfing → teleportation

$$PR(A) = 1 - d + d \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

or

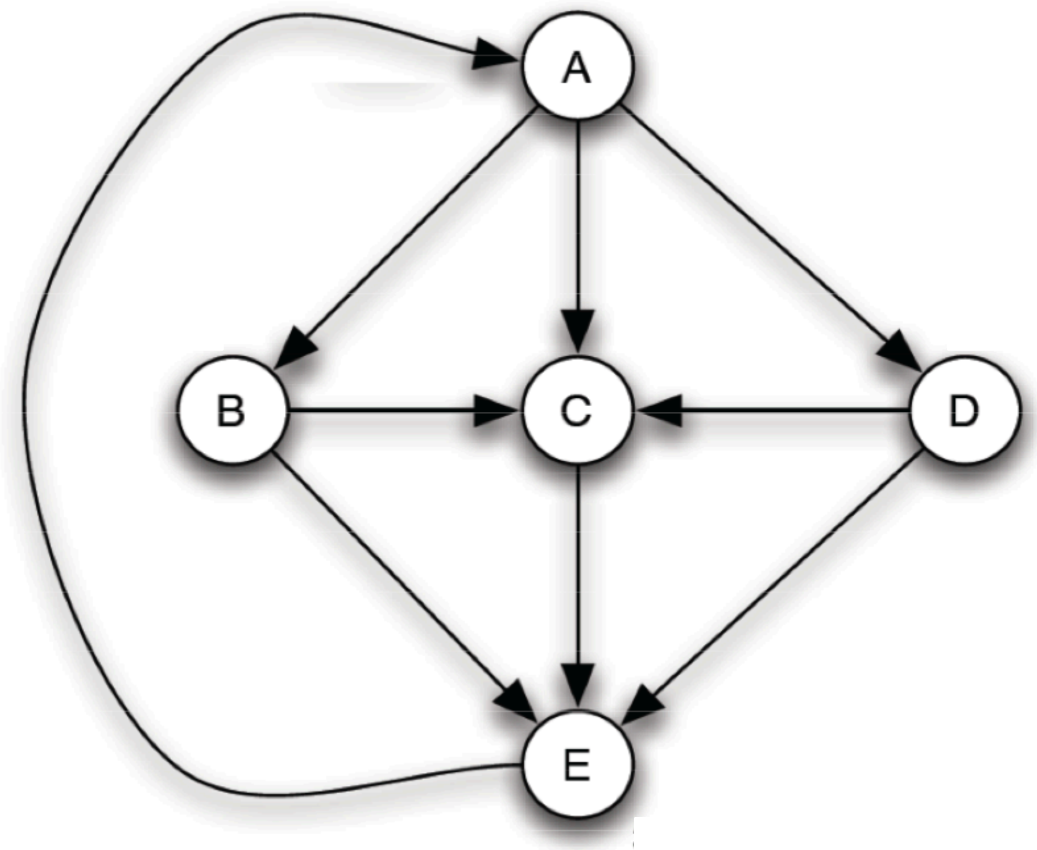
$$PR(A) = \frac{1-d}{N} + d \sum_{B \in I_A} \frac{PR(B)}{L(B)}$$

where N is the number of nodes and "d" is the damping factor.

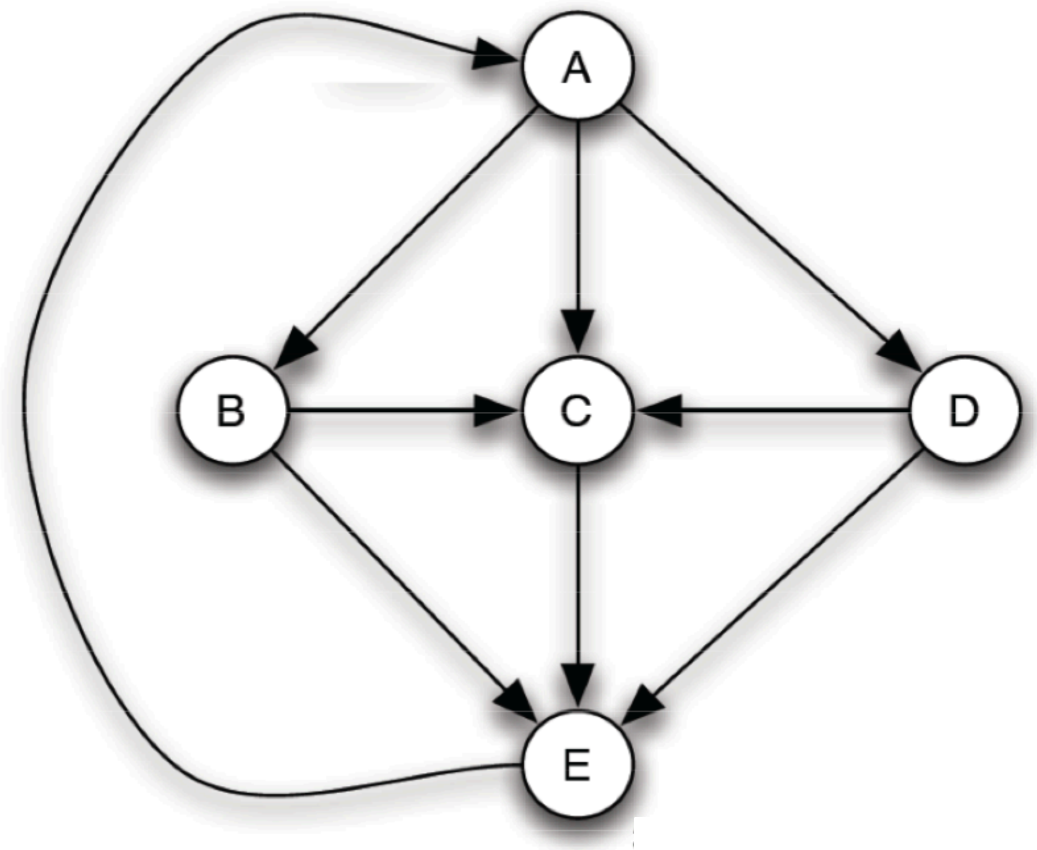
But: link farms…

10-15% of the pages are link farms -> Internet archives vs. webSpam

Example: Page Rank



Example: Page Rank



The PR will not change ☺️

HITS vs. PageRank

	HITS	PageRank
Graph	Unique for each query	fix
Measures	Hubness and authority values	PR values

Different motivation:

The PR’s origial goal is to measure the centrality over the full graph, while the HITS is a good ranking algorithm for relevant hits.

But the HITS is much more demanding -> PR is more common

Before again something completely different:
Other centrality measures worth to check: Betweenness, Kendall tau, Katz, degree etc.

In python: NetworkX package

Generative network models

Spread analysis, network structures based on distributions

e.g. infection or social networks influence

Complex networks

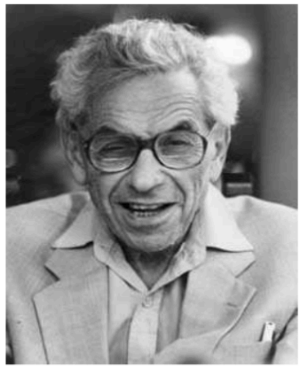
We will go only into the simulation of “realistic” networks

What we know about typical real world networks?

What can we measure?

Random graph and Erdős-Rényi

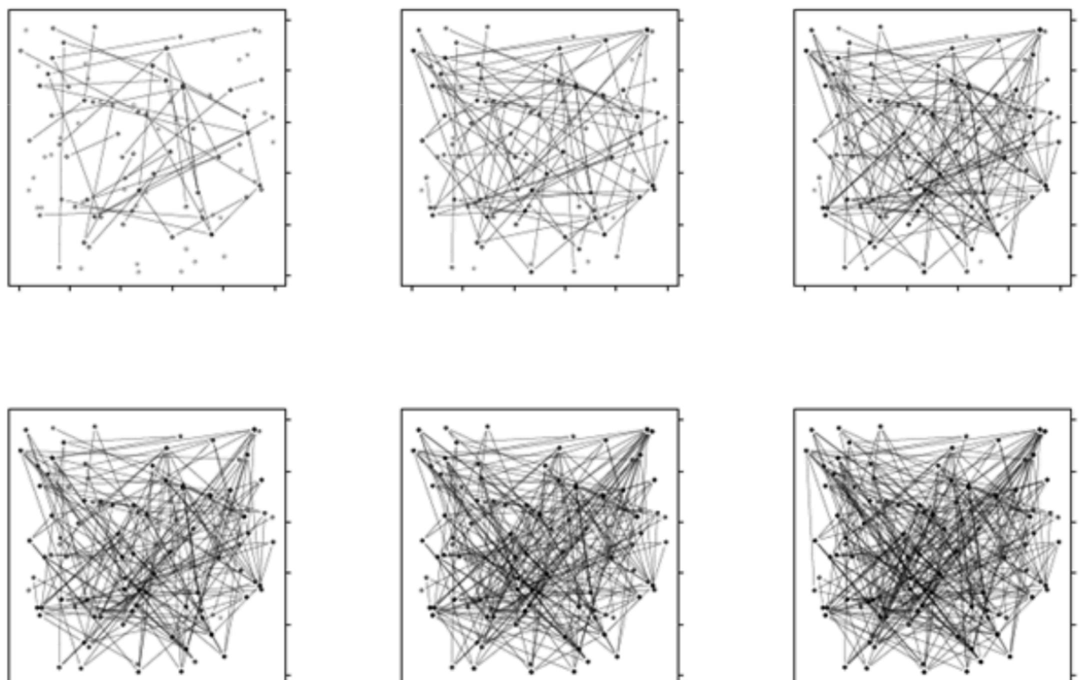
src: A. Benczúr



Erdös

Rényi

Erdös–Rényi



Random graph and Erdős-Rényi

G(n,p): with n nodes the probability of an existing edge is p
(independence!)

We know a lot about it (ER 1959, Bollobás et al 2002):

Expected number of edges:

$$|E| = \binom{n}{2} p = pn(n-1)/2$$

Average degree:

$$z = \frac{2|E|}{n} = \frac{2\binom{n}{2}p}{n} = (n-1)p$$

Random graph or Erdős-Rényi

The degree distribution is binomial:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

But if n is large, the limit distribution is Poisson
(z is the average degree)

$$P(k) = \frac{z^k e^{-z}}{k!}$$

Question 1: how the degree distribution look alike in natural networks?

Random graph or Erdős-Rényi

Connectedness: if … , then with high probabilty

$np < 1$: the largest connected component is $O(\log(n))$

$np = 1$: the largest connected component is $O(n^{(2/3)})$

$np > 1$ constant: $O(n)$ the largest connected component is and the second largest is at best $O(\log(n))$

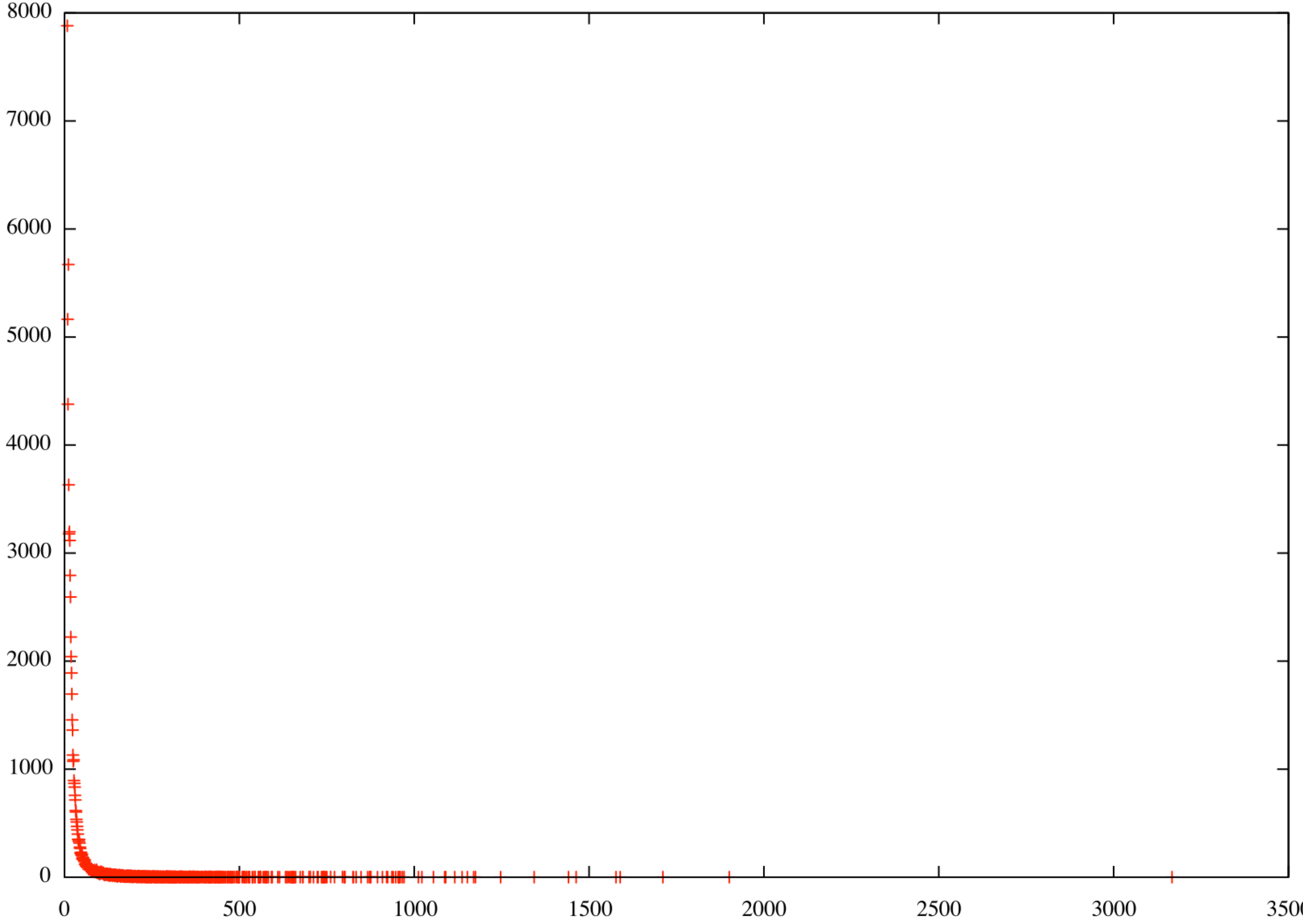
$np < (1-e) \log(n)$: there is at least one isolated node

$np > (1+e) \log(n)$: connected

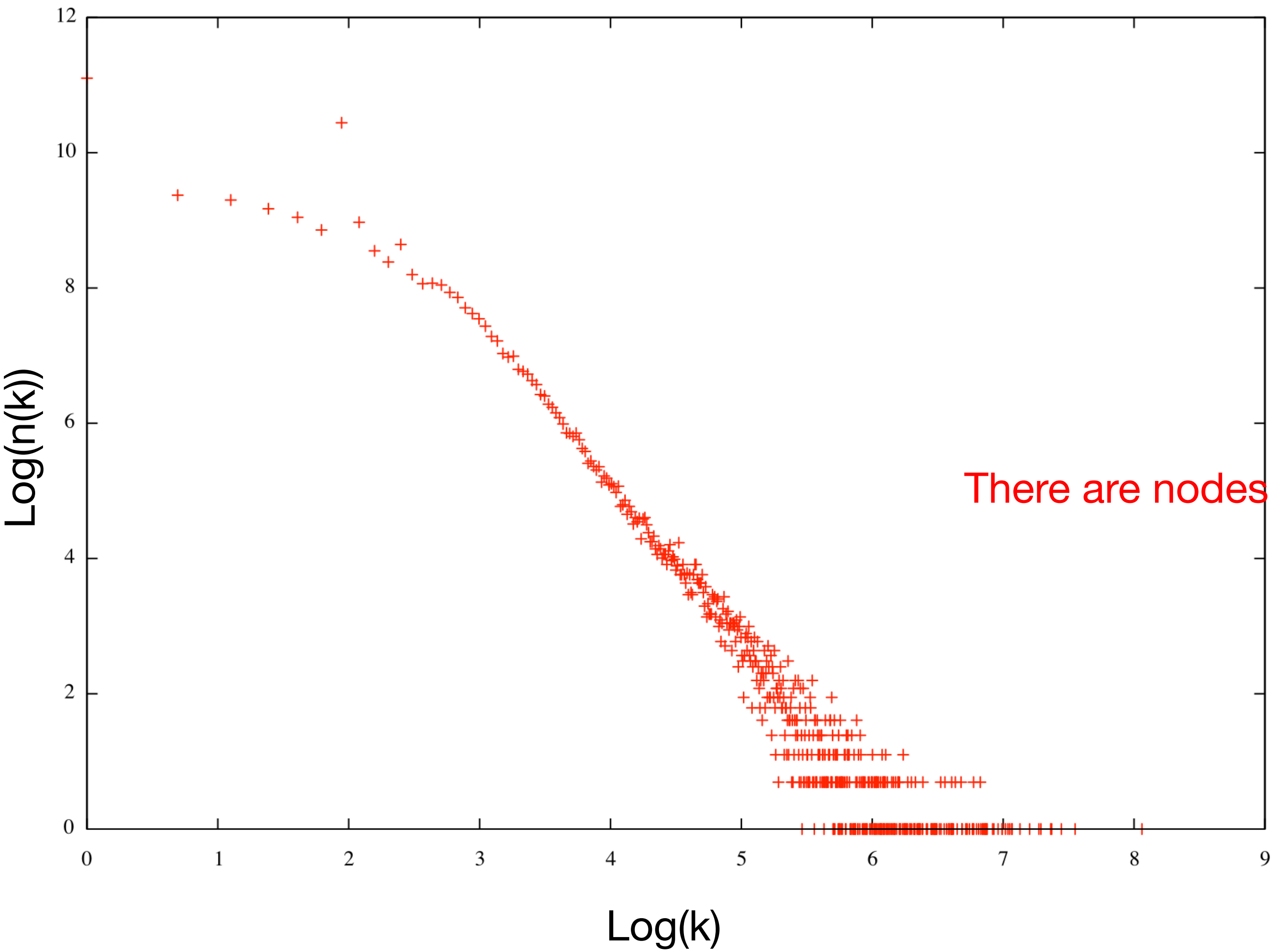
Diameter:

$$\log(n)/\log(p(n-1))$$

Wiki graph (z=11.1667)

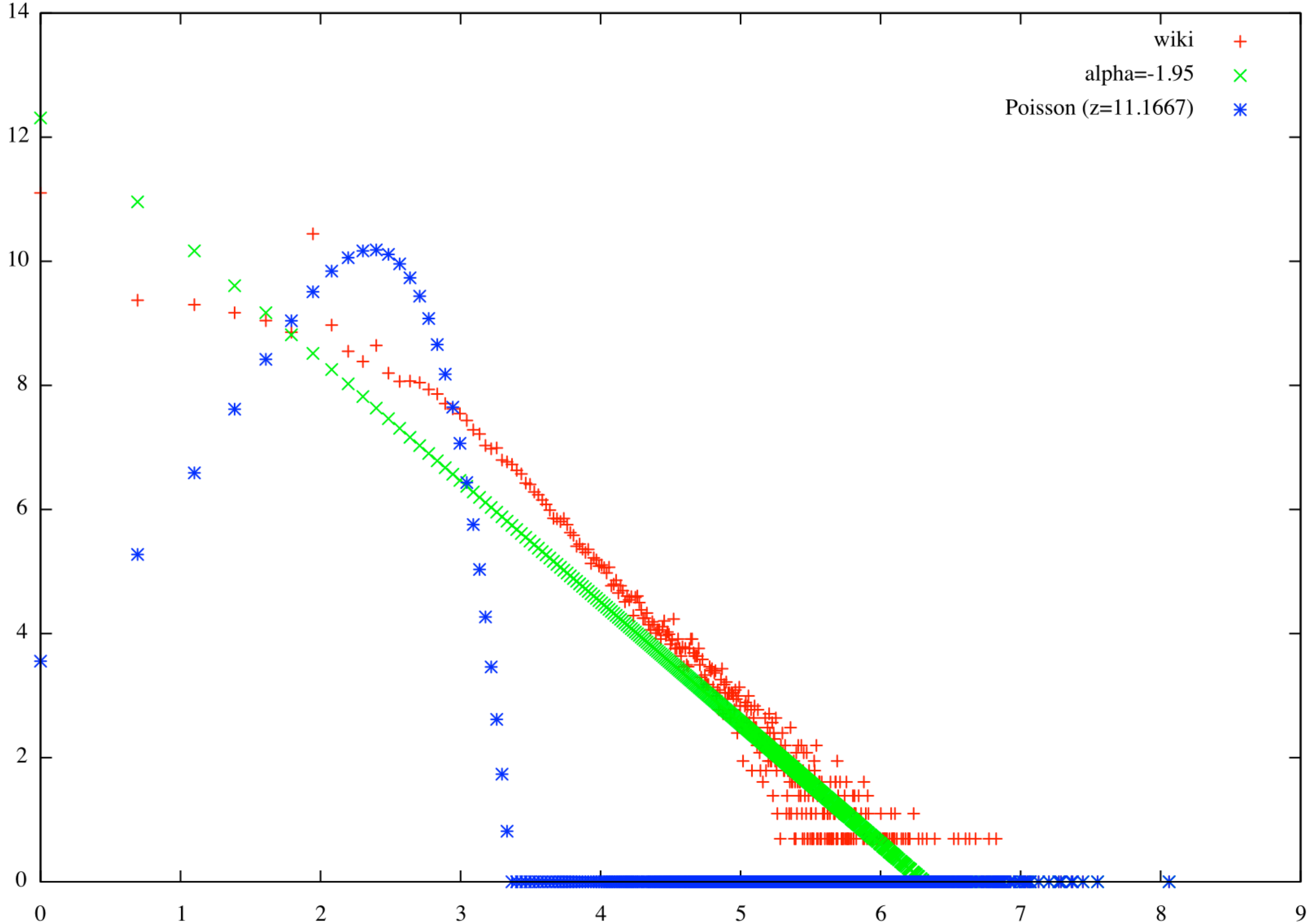


So maybe on log-log scale

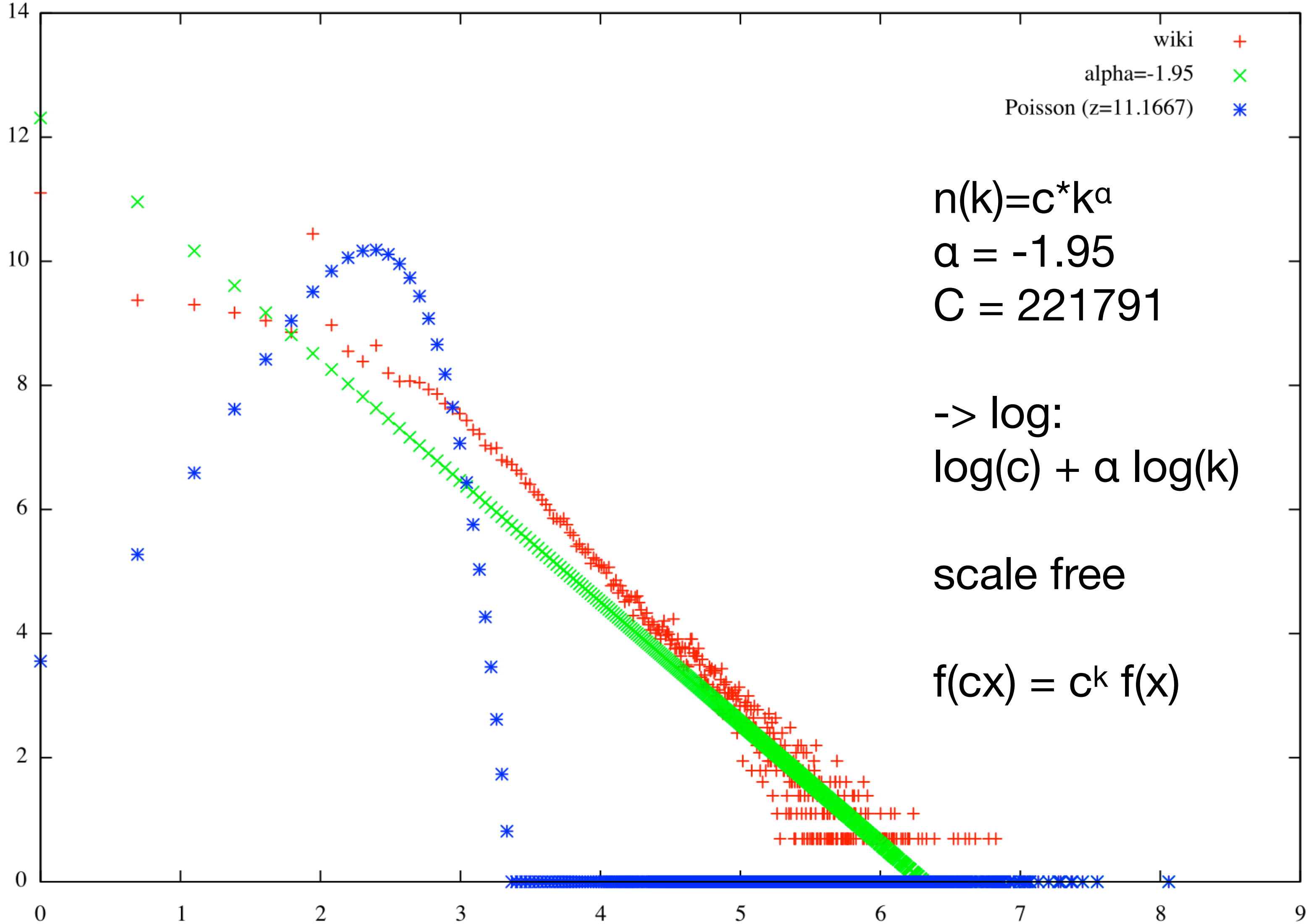


There are nodes with large degree!
(heavy tail)

Wiki graph is power law (or is it?)!



Wiki graph is power law (or is it?)!



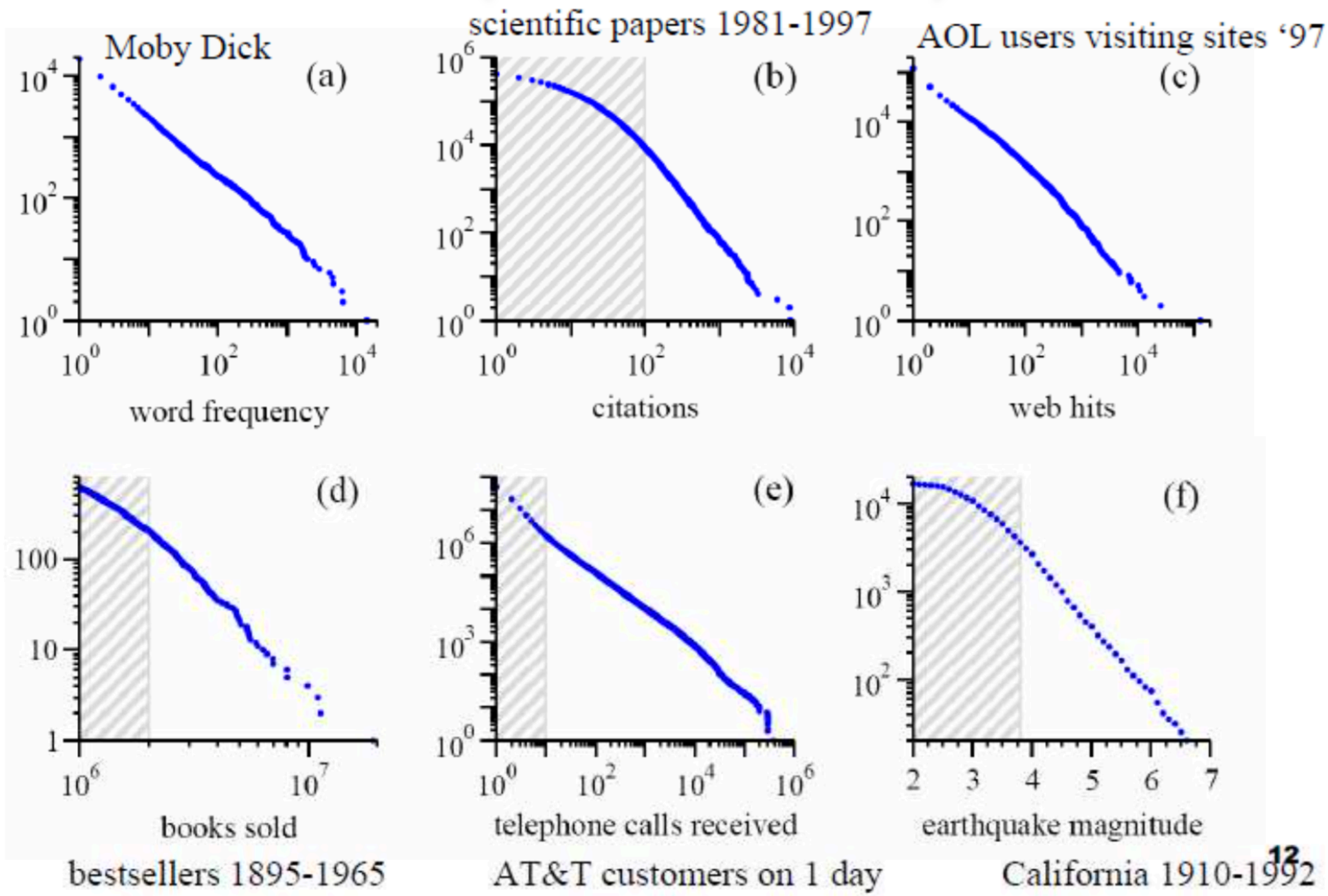
$n(k)=c*k^{\alpha}$
$\alpha = -1.95$
$C = 221791$

-> log:
$\log(c) + \alpha \log(k)$

scale free

$f(cx) = c^k f(x)$

Examples for Power law distros (src: Daniel Bilar)



Discovered multiple times

(src: Benczúr)

Pareto (1897): 80-20 rule

Yule (1925): evolution

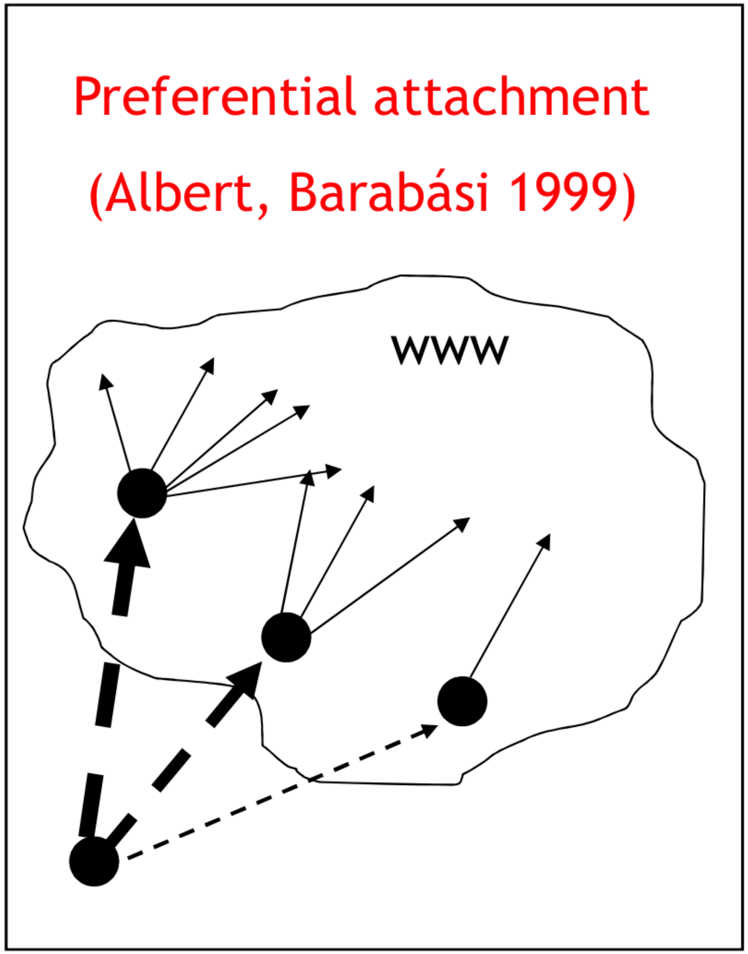
Zipf (1949): distribution of terms

Simon (1955): Zipf cont.

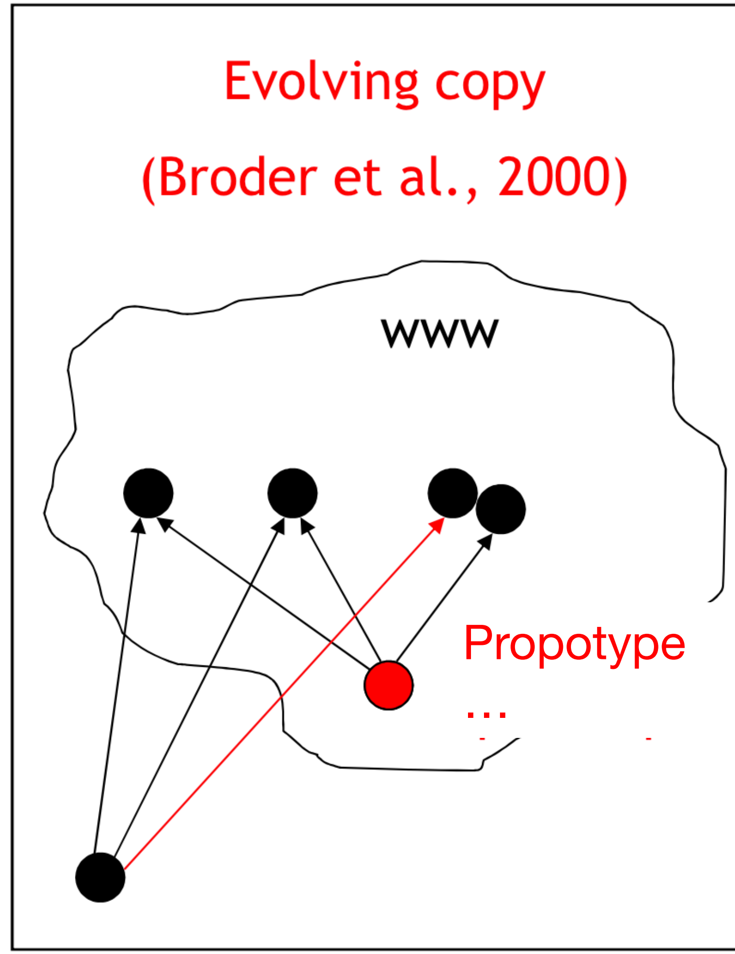
Price (1976): citation graph!

and Barabási-Albert model (1999): WWW graph is PL

We happy?



Parameter: a new node will connect with m edges



Parameter: the quality of copying

Barabási-Albert model

1. In each iteration the model add a new node

2. According the existing degree distribution it connects the new node with m nodes

The distribution of the degree of a node which was added at the i-th Iteration after t iterations

$$P(k_i(t) = k) = m^2 t \left(\frac{1}{k^2} - \frac{1}{(k-1)^2} \right) \sim k^{-3}$$

1. It needs to grow (ER is not growing!)
2. The grow should follow BA

If either of them is not true -> it will not follow PL

Are we done?

We have a PL!

But the natural networks have other properties:

Are we done?

We have a PL!

But the natural networks have other properties:

Small world model (Milgram in 1967 and Watts-Strogatz)

1. Low average distance

    We can reach through small number of edges every node (on average)

    In natural nets: log(N) is a good first approx.

Note:
Friendship paradoxon (Scott L. Feld 1991):
On average we have less friends than our friends

Are we done?

We have a PL!

But the natural networks have other properties:

Small world model (Milgram in 1967 and Watts-Strogatz)

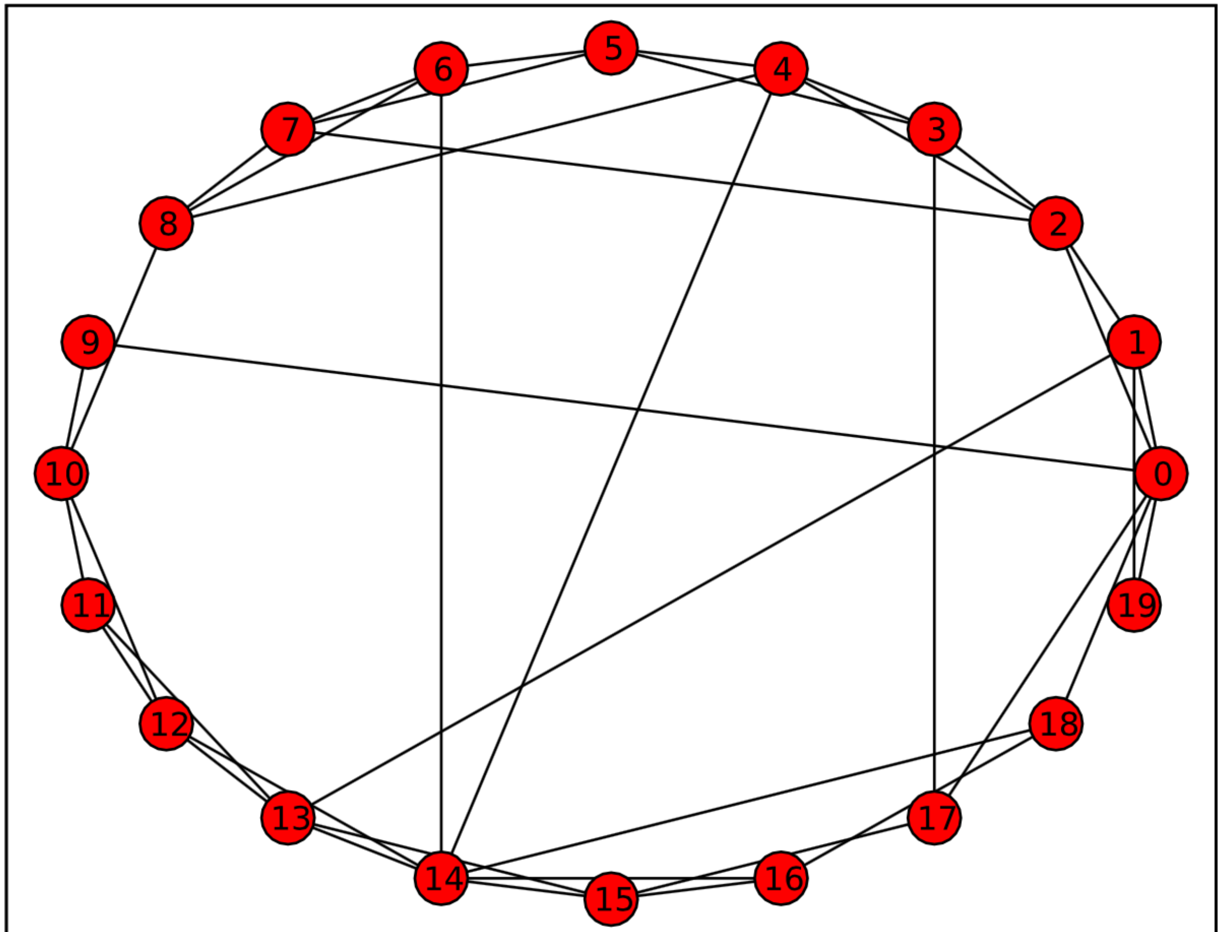
1. Low average distance

2. Strongly clustered:
   The neighbours of the nodes are ”strongly” connected (cliques?)

   per node (let be the our degree k_i):

$$C_i = \frac{|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}$$

Watts-Strogatz model (1998)



r regular ring
(each node has r neighbours)

Random remove an edge and another similar to ER

It will result a small world Network ☺️

But: not PL ☹

Summary

	Degree distr.	Clustering coeff.	Average dist.
"Real" nets.	PL	strongly	small
Erdős-Rényi	Poisson	low	small
Barabási-Albert	PL	low	small
Watts-Strogatz	Poisson	strongly	small
Broder et al.	PL	strongly	small

Src: Benczúr