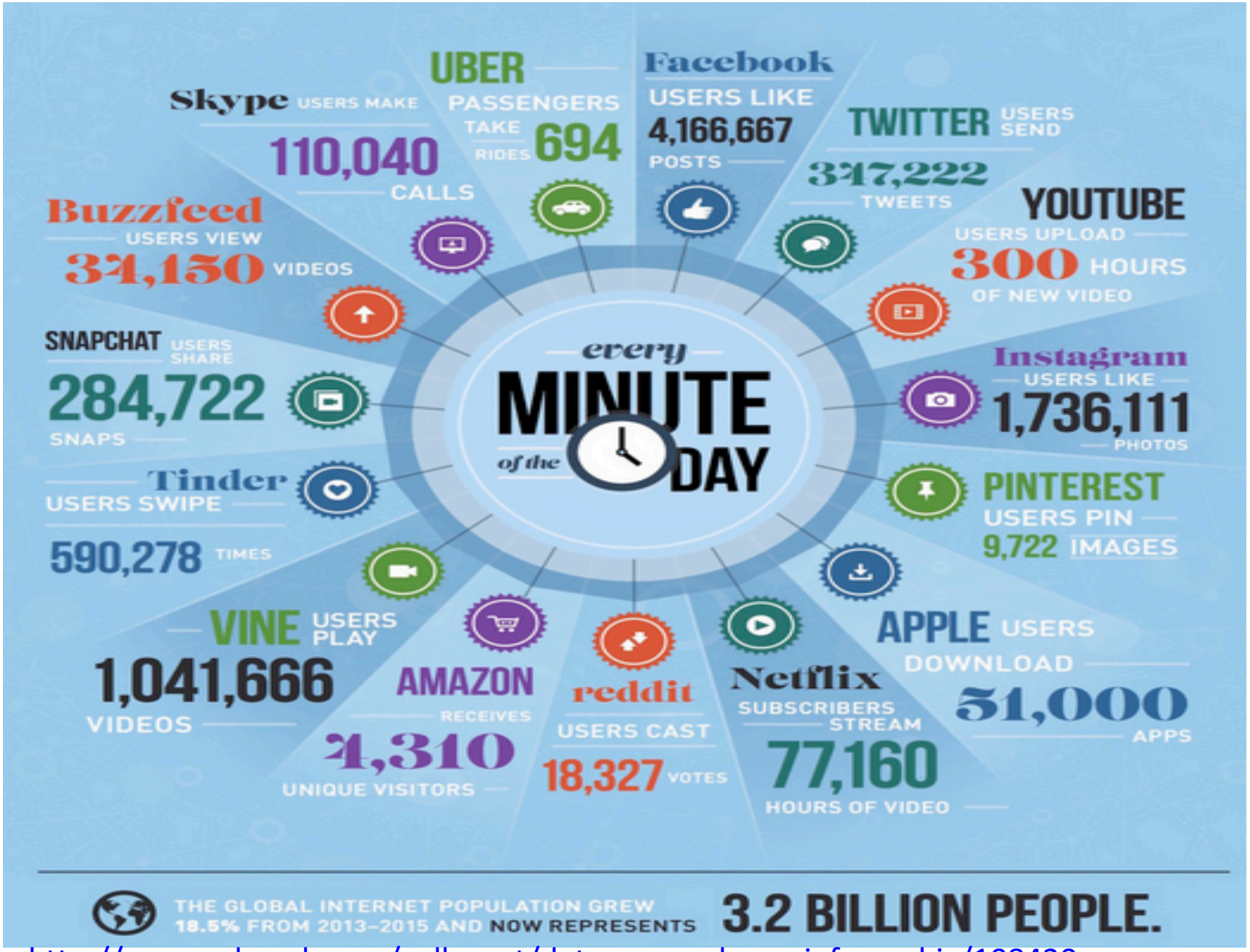
Web Mining

Michalis Vazirgiannis

November, 2019



http://www.adweek.com/galleycat/data-never-sleeps-infographic/108420

Topics

1. **Recommendation/Personalization**
2. **Pagerank**
3. **Influence maximization**

Web personalization and recommendations

- ~25% of Internet users reading online reviews prior to paying for an offline service,
  - 80% claimed reviews had significant influence on their purchasing habits.
- Users pay a mark-up of 20% to 100% for services/products with excellent peer ratings on review sites.
- Humans are notoriously bad at choosing between too many choices,
  - rely on external recommendations and reviews to narrow the set of possible choices.

Personalization

- Personalized reviews tend to dominate
- Netflix: personalized video-recommendation system based on ratings and reviews by its customers.
- In 2006, offered a $1,000,000 prize to the first developer of a video-recommendation algorithm that could beat its existing algorithm

Recommender Data Model

- Set $U=\{u_1, \ldots, u_n\}$ of users
- Set $I=\{i_1, \ldots, i_m\}$ of items (e.g. products)
- Elements from U and I can be described by a vector respectively
  - $(a_1, \ldots, a_s)$ → attributes of user profile
  - $(b_1, \ldots, b_t)$ → description of items (meta data, features, …)
- Goal of recommendation process: recommend new items for an active user u
- Overview of process
  - User modeling (explicit or implicit, e.g. user rates items)
  - Personalization, generate list of recommended items

User-Item Ranking

- Recommendation often based on ratings of an item ij by a user u_k:
- Rating $r_{j,k}$: I → [0,1] ∪ ø
- Other range of values possible, e.g. {*, **, ***, ****, *****}
- ø := no rating for Item (or "0")
- Example user-item matrix of ratings

	V for Vendetta	La Vita e Bella	Lion King	Wall-e
Alice	4	3	2	4
Bob	∅	4	5	5
Cindy	2	2	4	∅
David	3	∅	5	2

Types of Recommender Systems

- Collaborative filtering (CF)
- Content-based filtering (CB)
  - Individual recommender algorithms
  - Also utility- or knowledge-based approaches

- Case-based recommendation
- Hybrid recommender systems
  - Combination of several other recommenders
- Additional important variants
  - Context-aware and multi-dimensional recommenders
  - Decentralized recommender systems
  - Recommending for groups

Issues of Recommender Systems

- Cold start and latency problems
- Sparseness of user-item matrix
- Diversity of recommendations
- Scalability
- Privacy and trust
- Robustness
- Utilization of domain knowledge
- Changing user interests (dynamics)
- Evaluation of recommender systems

Cold Start Problems

- “New user” and “new item” problem
- Systems cannot recommend items to new users with no profile or no interaction history
- Same for new items
  - Also “latency problem”: items need some time until they can be recommended
- Chicken-and-egg problem
  - Users will not use system without good recommendations
  - No incentive to rate items etc.
  - System cannot generate good recommendations
- Possible solutions
  - include explicit user profiling methods to start interaction

Data Sparseness

- Common situation
  - Lots of users and items
  - But only few ratings
  - Sparseness of user-item matrix
  - Recommender algorithms will not work very well
- In addition, new items are continuously added
  - Users should also rate these items
  - Number of ratings has to keep up with new users and items
- Possible solution
  - Include the automatic generation of ratings
  - Implicit user profiling, use of transaction history of users, e.g. click on a video constitutes a positive rating

Diversity of Recommendations

- Focus usually on generating recommendations as "good" as possible
  - But also important: new, unexpected items
  - Do not recommend items that are already known
  - Do not recommend items that are too similar to already known items
    - E.g. user likes "Lord of the Rings 1" → user possibly also likes "Lord of the Rings 2", but is this really a useful recommendation?
- Possible solutions
  - Use content-based approaches to easier integrate new items in recommendation process
  - Use collaborative filtering to allow "cross-domain" recommendations

Privacy and Trust

- Collecting and interpreting personal data, e.g. ratings
  - For example, bought items or visited product Web pages on Amazon
  - Control for users?
    - Bought product may have been gift for other person
  - Privacy problem!
- Tradeoff with recommender quality
  - The more information about the user the system is able to collect, the higher the recommendation quality is in general
- Also trust, how can user trust the quality of a recommended item?
- Possible solutions include
  - Consider social relationships ("social recommender", "Web of Trust")
  - Let user control their profile information
  - Explanations of recommendations
    - Why was an item recommended?

Robustness

- Quality of (collaborative) recommenders depends on quality of ratings
  - Manipulation by users possible
    - E.g. by automatic registration of a large number of “users” and ratings
  - Also called “shilling”, “profile injection”
  - Attacks in principle
    - “push”: Aim is to push item(s) by inserting a large number of good ratings
    - “nuke”: Same with negative ratings
- Possible solutions include
  - Make registration for service harder, e.g. request and check personal information
  - Detect attacks and remove corresponding users and ratings
  - Adjust algorithms, some algorithms have proven to be more robust

Utilization of Domain Knowledge

- Systems often regard items in isolation
  - No relationships between items
  - No domain knowledge
- Example: searching for (books or other products on) “baseball”
  - Too many hits → restriction to “baseball technique”, or “baseball player”, for example
    - Based on user model and domain ontology
  - Too few hits → broading to “sport”, for example
- Some approaches in current research literature utilize Semantic Web technologies
  - Build and maintain item ontologies
  - Also for users
    - E.g. „GUMO“ (General User Modeling Ontology)

Changing User Interests (Dynamics)

- User model is often relatively static
- But dynamic evolution over user interests
  - Changes over time, older ratings may not be valid any more
- Also the context of recommendations
  - Example: Mobile restaurant guide
    - Restaurant may be too far away from current position (location)
    - Restaurant may be closed today (time)
  - A good rating for a restaurant after a dinner on a weekend may not be relevant for recommending a restaurant for a quick lunch on a workday
- Solutions in research literature include
  - E.g. explicit distinction between short- and long-term interests
  - Context-aware recommender systems

Evaluation of Recommender Systems

- Goal of personalization is to improve the interaction of users with the system
  - May be subjective, hard to evaluate
- General method for recommender systems
  - Let users rate recommended items and compare actual user ratings with predicted rating
  - Most important metrics
    - “precision”: probability rate that users did like recommended items
    - “recall”: probability rate that preferred items by users are recommended
  - In addition user studies
    - User evaluate system in questionnaire etc.

Collaborative Filtering (CF)

- Basic idea: System recommends items which were preferred by similar users in the past
  - Based on ratings
    - Expressed preferences of the active user
    - And also other users → Collaborative approach
  - Works on user-item matrix
    - Memory-based or model-based
    - No item meta data etc.!
- Assumption: Similar taste in the past implies similar taste in future
- CF is formalization of “word of mouth“ among buddies

General Process

1. Users rate items
2. Find set S of users which have rated similar to the active user u in the past (→ neighborhood)
   - Similarity calculation
   - Select the k nearest users to the active user
3. Generate candidate items for recommendation
   - Items which were rated in neighborhood of u,
   - but were not rated by u yet
4. Predict rating of u for candidate items
   - Select and display n best items

Example (I)

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

Source: http://www.dfki.de/~jameson/ijcai03-tutorial/

Example (II)

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

Example (III)

	Hoop Dreams	Star Wars	Pretty Woman	Titanic	Blimp	Rocky XV
Joe	D	A	B	D	?	?
John	A	F	D		F	
Susan	A	A	A	A	A	A
Pat	D	A		C		
Jean	A	C	A	C		A
Ben	F	A				F
Nathan	D		A		A	

Required Metrics

- Metric for user-user similarity
  - Mean-squared difference
  - Cosine
  - Pearson/Spearman correlation
- Select set S of most similar users (to active user u)
  - Similarity threshold
  - Aggregate neighborhood
  - Center-based
- Metric to predict the rating of u for an item i

User-User Similarity

- Item set I
- Users U,V with u[i] denoting rating of item i by user u
  - the rating vector of user u is denoted by $\vec{u}$
  - the vector norm is denoted by $|\vec{u}|$
  - n is the number of items rated by both U and V

- Mean squared difference:
  - Small values show similar users $sim_1(U,V) = \frac{(\vec{u} - \vec{v})^2}{n}$

- Cosine similarity:
  - Large values show similar users $sim_2(U,V) = \frac{\vec{u} * \vec{v}}{|\vec{u}| * |\vec{v}|}$

Pearson/Spearman Correlation

- Average rating is taken into account
  - The vector of average ratings is denoted by $\overrightarrow{u}$
- Not suitable for unary ratings
  - Unary: Item is marked (or not)
    - e.g. “Product was purchased“
  - Binary: good/bad, +/- etc.
  - Scalar: Numerical rating (e.g. 1-5) etc.
  - Consider only items which were rated by both users
- Values near 1 show similar users

$$sim_3(U,V) = \frac{(\overrightarrow{u} - \overrightarrow{\overrightarrow{u}}) * (\overrightarrow{v} - \overrightarrow{\overrightarrow{v}})}{|(\overrightarrow{u} - \overrightarrow{\overrightarrow{u}})| * |(\overrightarrow{v} - \overrightarrow{\overrightarrow{v}})|}$$

Required Metrics

- Metric for user-user similarity
  - Mean-squared difference
  - Cosine
  - Pearson/Spearman correlation
- Select set S of most similar users (to active user u)
  - Similarity threshold
  - Aggregate neighborhood
  - Center-based
- Metric to predict the rating of u for an item i

Neighborhood of Similar Users

- Goal: Determine set S of users which are most similar to the active user u
- Center-based
  - S contains k most similar users
    - Problem: maybe some of the users are not really that similar, if k was chosen too large, deviators possible
- Similarity threshold
  - S contains all users with a similarity bigger than a threshold **t**
    - Problem: maybe too few users in S
- Aggregate neighborhood
  - Follow similarity threshold method first
  - If S is too small (less than k users)
    - Determine "centroid" of set S and add users which are most similar to centroid (→ less deviators than center-based method)

Required Metrics

- Metric for user-user similarity
  - Mean-squared difference
  - Cosine
  - Pearson/Spearman correlation
- Select set S of most similar users (to active user u)
  - Similarity threshold
  - Aggregate neighborhood
  - Center-based
- Metric to predict the rating of u for an item i

CF Recommender (I)

- Given
  - Set S with most similar users to u
  - s[i] rating of a user (from S) from an item i
- Goal: Predict the rating of u for i
- Easiest option: Arithmetic mean

$$r_1(U, i) = \frac{1}{|S|} \sum_{s \in S} s[i]$$

- Problems
  - Similarity of u with members of S is not taken into account
    - Solution: Weighting based on similarity

CF Recommender (II)

- Different users utilize rating scale differently
  - Solution: Consider deviation from average rating (for user)

$$r_3(U,i) = \overline{u} + \frac{1}{\sum_{s \in S} sim(U,s)} \sum_{s \in S} sim(U,s) * (s[i] - \overline{s})$$

- Note
  - Many variations of algorithms in research literature
    - For various application domains, with different properties

Advantages Collaborative Filtering

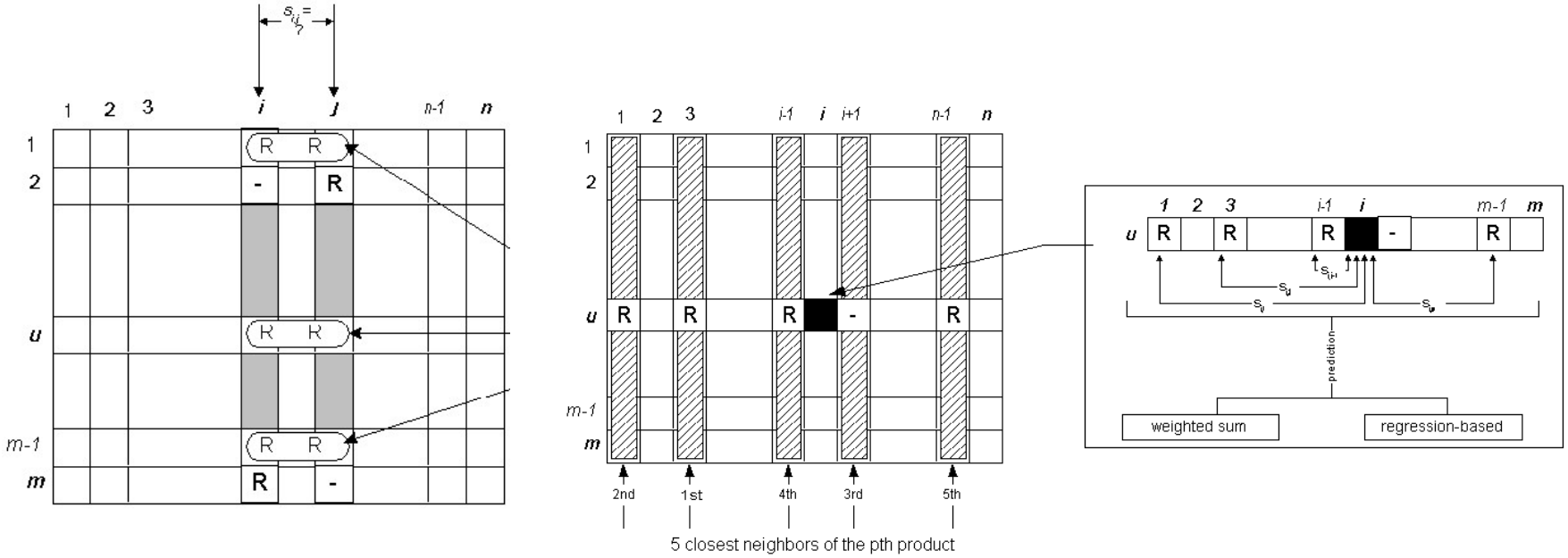
- Works well in practice
- Quality of recommendations improves with density of ratings
- Only ratings as input data required
  - In particular, no information (meta data, description) about items needed
- CF is able to generate cross-domain ("cross genre") recommendations → high diversity
  - Because item categories etc. are not considered
  - Has proven useful in practice
- Implicit user feedback often adequate (CTR)
  - Unary ratings, e.g. rating = "Click on product Web page"

Disadvantages Collaborative Filtering

- New user and new item problem
  - Serious issue in practice
- Often sparseness in user-item matrix
  - Algorithms generate worse results with too few ratings
- “Grey sheep” problem
  - Does not work very well for users with “extraordinary” taste
    - Because similar users are not available
  - Also “black sheep”, users that intentionally make incorrect ratings
    - CF is prone to manipulation
    - Trust and robustness are issues

Item-to-Item Collaborative filtering (Amazon)

- Item representation through a N-dimensional vector.
  - Each dimension corresponds to a user's action on this item.
- Rather than matching the user to similar customers, build a similar-items table by finding that customers tend to purchase together.
- Recommend items with high-ranking based on similarity



Matrix completion

User	Jurassic Parc	Cast Away	Titanic	Amelie
Joan	4	5		2
Maria	1	2	5	
Jenny	5	3	1	
Salem	2			4

$$\mathbf{R} = \begin{pmatrix} 4 & 5 & & 2 \\ 1 & 2 & 5 & \\ 5 & 3 & 1 & \\ 2 & & & 4 \end{pmatrix}$$

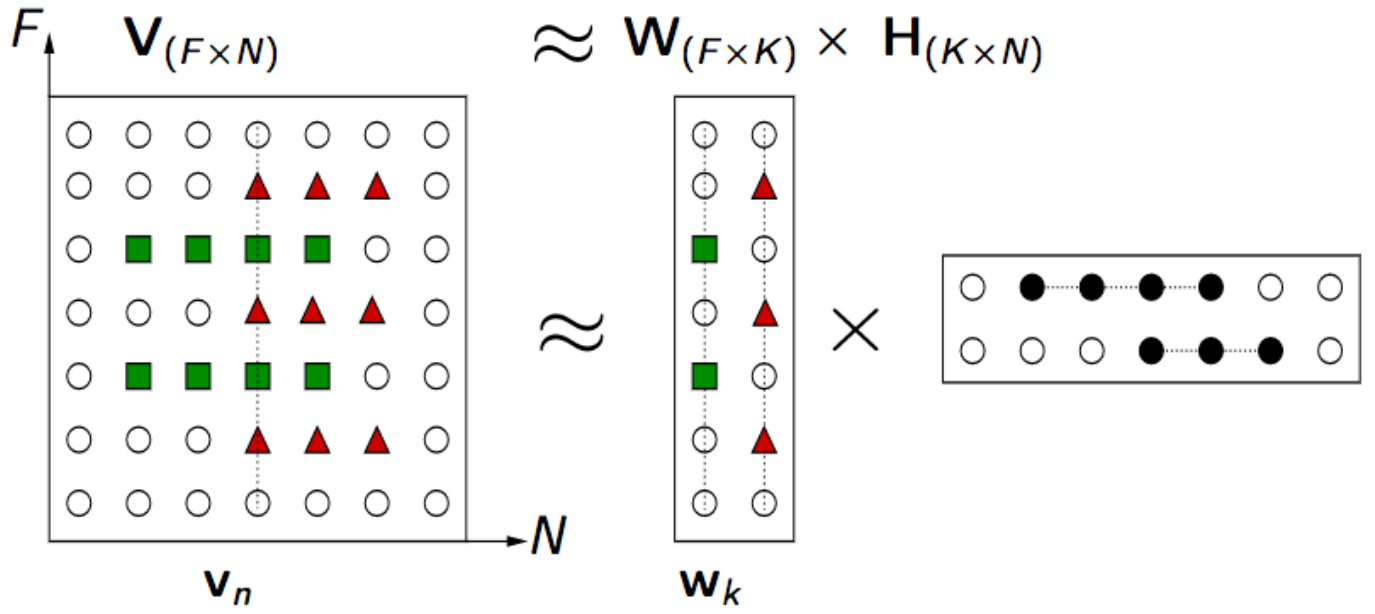
Low Rank approximation

- Consider the rating matrix R with elements R_{ui} If we have a set of users U and a set of items I,
- then R is a $|U| \times |I|$ matrix
- We can approximate this by a low-rank approximation:
- $R \approx AB$
- where A is a $|U| \times K$ matrix and B is a $K \times |I|$ matrix
- K is the rank of AB (number of non-zero eigenvalues)

Low Rank approximation

- ratings R_{ui} for $(u,i) \in D$ where $|D| \ll |U| \times |I|$ (number of ratings is typically much less than the total possible number of ratings)
- In low rank approximation we are trying to fit to $|D|$ ratings of the matrix with $(|U| + |I|) \times K$ elements
- K controls the complexity of the model
- Large K will fit the data better but at the cost of possibly over-fitting

Explaining data by factorization



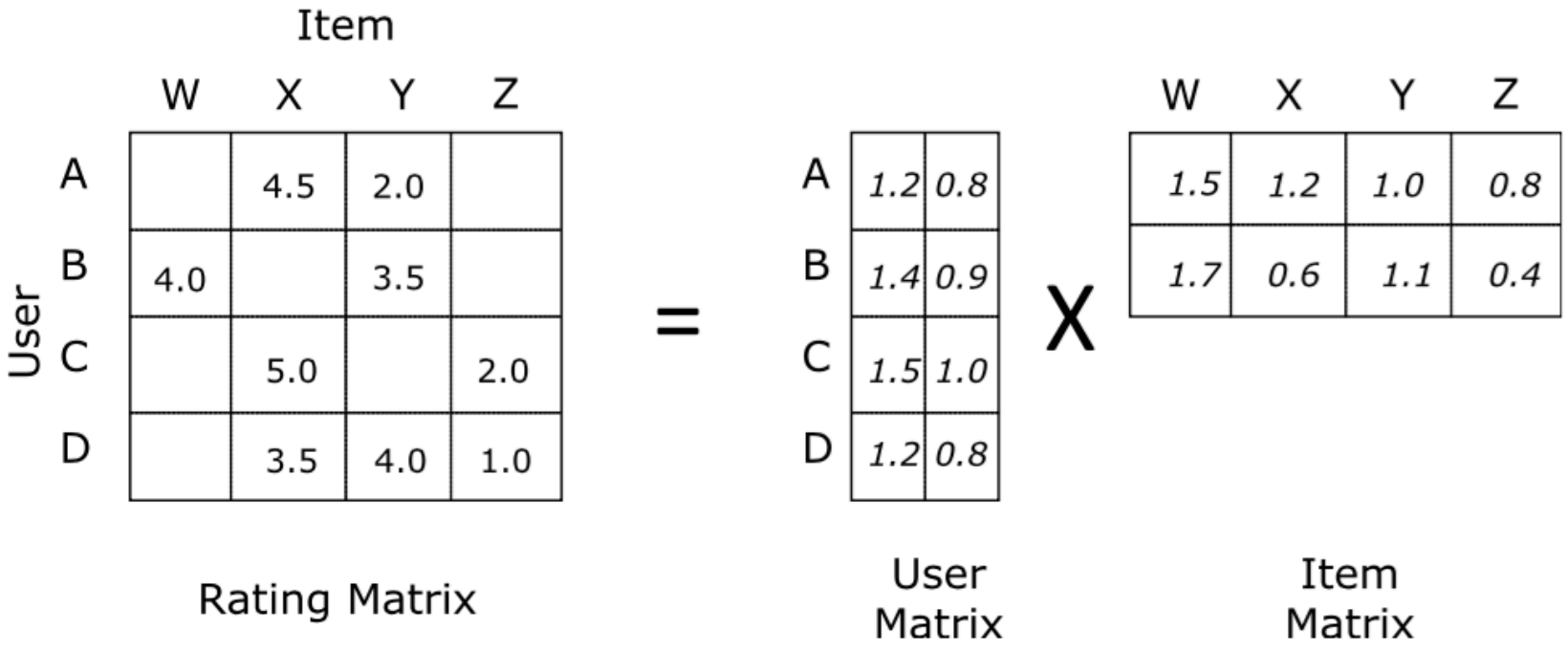
data matrix

"explanatory variables"
"basis", "dictionary",
"patterns", "topics"

"regressors",
"activation coefficients",
"expansion coefficients"

Illustration by C. Févotte

Low Rank Approximation Example



Least square method

- We can seek a least-squares solution for **A** and **B**

$$(\mathbf{A}, \mathbf{B}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{(u,i)\in D} \left(R_{ui} - \sum_{l=1}^{K} A_{ul} B_{li}\right)^2$$

- Using low-rank approximation provides very effective model
- If we are unlucky **A** and **B** may not be uniquely definedthen it is difficult to find a solution

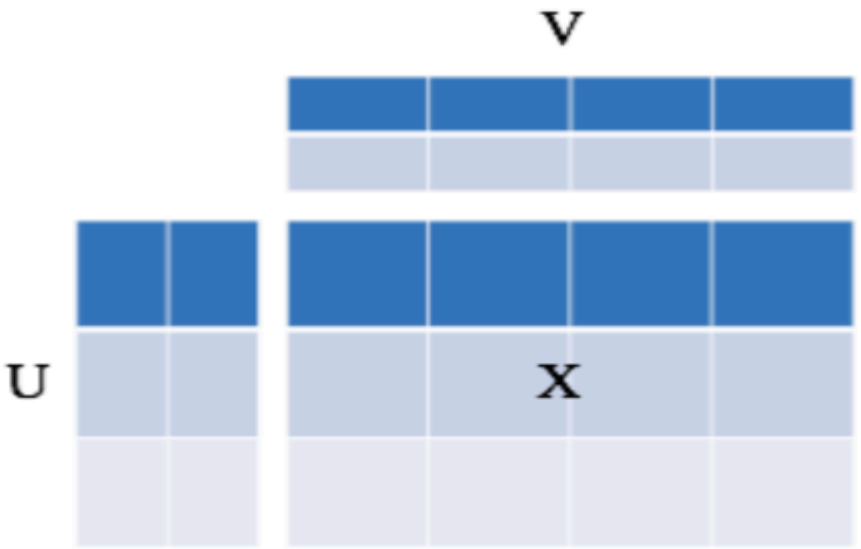
Non Negative Matrix factorization (NMF)

- - Data is often nonnegative by nature
  - pixel intensities; occurrence counts; food or energy consumption; user scores; stock market values;
- - Interpretability of the results, optimal processing of nonnegative data may call for processing under Non-negativity constraints

- - Applying SVD results in factorized matrices with positive and negative elements may contradict the physical meaning of the result.
- - *Nonnegative matrix factorization (NMF)*
- find the reduced rank *nonnegative factors* to approximate a given nonnegative data matrix.

NMF

- $X \simeq UV^T$
  - $U = [u_{fk}], w_{fk} >= 0$
  - $V = [v_{kn}], h_{kn} >= 0$
  - $k << f, n$



NMF

- Assume X (mxn) data matrix and $r << m, n$
- NMF aims to find non negative matrices

$$U \in R^{mxr}, V \in R^{rxn} : X \approx UV^T$$

- To find U, V, optimization problem:

$$\min_{(U,V)} \left\| X - UV^T \right\|_2$$

- Alternative error function:

$$\min_{U,V} f(U,V) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(X_{ij} \log \frac{X_{ij}}{(UV^\top)_{ij}} - X_{ij} + \left(UV^\top\right)_{ij} \right)$$
$$\text{s.t. } U_{ia} \geq 0, V_{jb} \geq 0, \forall i, a, b, j$$

Alternating Least squares

- 1. Suppose we know U, *with* V unknown.
- for each j we could minimize $||X_{.j} - UV_{.j}^T||_2$
  - find $V_{.j}$ that minimizes with $X_{.j}$ and U known.
  - Frobenius norm: sum of squares,
  - minimization is a least-squares problem, i.e. linear regression
  - "predicting" $X.j$ from W.

$$V_{.j} = (U^T U)^{-1} U^T X_{.j}$$

  - repeat for all columns $V_{.j}$

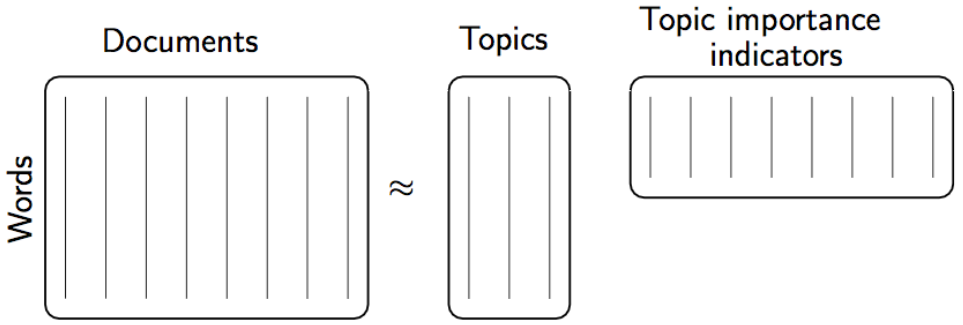
- 2. assume V, *with* U unknown: $X^T = VU^T$
  - Interchange roles of U, V in the above optimization
  - Compute a row of U, repeat for all rows

Alternating Least squares

- Putting all this together
- first choose initial guesses, random numbers, for U and V
- alternate:
  - Compute U assuming V known
  - Compute V based on that new U
  - …
- - may generate some negative values: simply truncate to 0

NMF issues, applications

- Uniqueness and Convergence
- U_{mxr} , r (rank) choice: via SVD…
- Applications
  - Topic detection
  - Source separation (music , speech)
  - Clustering
  - Recommendations



References

- D. Billsus and M. J. Pazzani, "Learning collaborative information filters", In Proceedings of the Fifteenth International Conference on Machine Learning, pages 46-54, 1998
- "A Comparative Study of Collaborative Filtering Algorithms", Joonseok Lee, Mingxuan Sun, Guy Lebanon, http://arxiv.org/pdf/1205.3193.pdf
- A. Paterek. Improving regularized singular value decomposition for collaborative filtering, Statistics, 2007:2{5, 2007.
- J. Leskovec, A. Rajaraman and J. Ullman. Mining of Massive Datasets. Cambridge University Press, 2014.
- J. Breese, D. Heckerman and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998.

References

- D. Billsus and M. J. Pazzani, "Learning collaborative information filters", In Proceedings of the Fifteenth International Conference on Machine Learning, pages 46{54, 1998
- "A Comparative Study of Collaborative Filtering Algorithms", Joonseok Lee, Mingxuan Sun, Guy Lebanon, http://arxiv.org/pdf/1205.3193.pdf
- A. Paterek. Improving regularized singular value decomposition for collaborative filtering, Statistics, 2007:2{5, 2007.
- J. Leskovec, A. Rajaraman and J. Ullman. Mining of Massive Datasets. Cambridge University Press, 2014.
- J. Breese, D. Heckerman and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998.

Topics

1. **Recommendation/Personalization**
2. **Pagerank**
3. **Influence mazimization**

Data are connected!

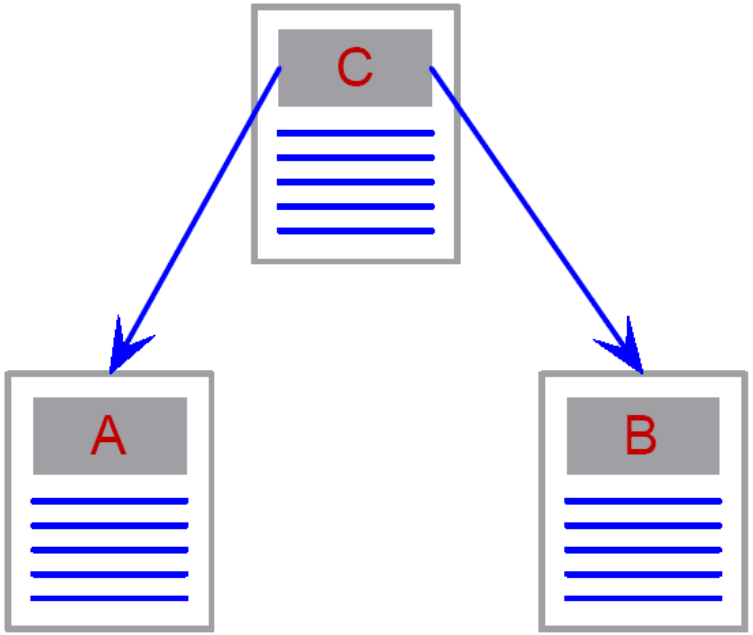
- A.boss = B, B.friends = {C,D,F}
- Social networks, and the web is not just a collection of documents – they form a graph structure
- A link from page *A* to page *B* may indicate:
  - *A* is related to *B*, or
  - *A* is recommending, citing or endorsing *B*
- Links are either
  - referential – *click here and get back home*, or
  - Informational – *click here to get more detail*

Citation Analysis

- The **impact factor** of a journal = *A/B*
  - *A* is the number of current year citations to articles appearing in the journal during previous two years.
  - *B* is the number of articles published in the journal during previous two years.

Journal Title	**Impact Factor (2002)**
J. Mach. Learn. Res.	3.818
IEEE T. Pattern Anal.	2.923
Mach. Learn.	1.944
IEEE Intell. Syst.	1.905
Artif. Intell.	1.703

Co-Citation



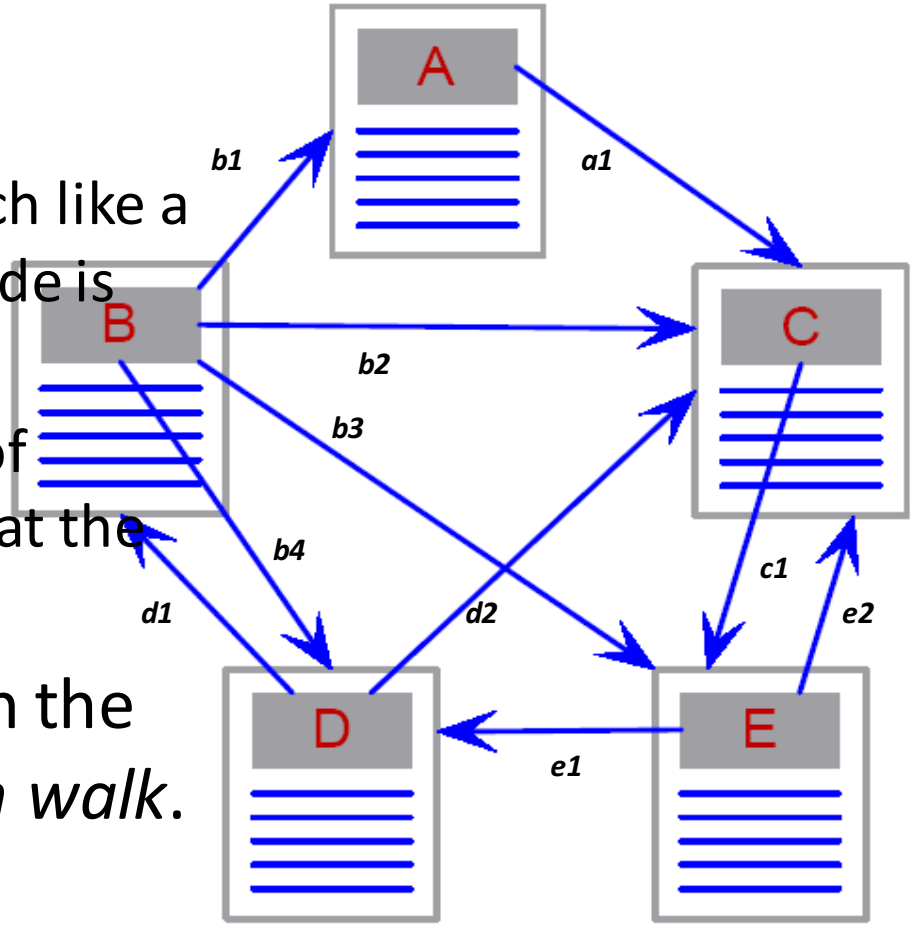
- *A* and *B* are co-cited by *C*, implying that
  - they are related or associated.
- The strength of co-citation between *A* and *B* is the number of times they are co-cited.

What is a Markov Chain?

A Markov chain has two components:

- A network structure much like a web site, where each node is called a state.
- A transition probability of traversing a link given that the chain is in a state.

A sequence of steps through the chain is called a *random walk*.



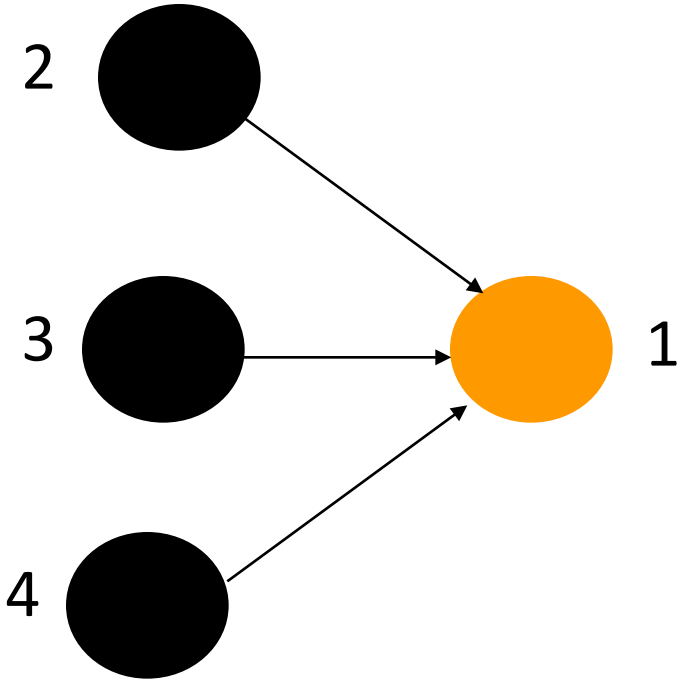
HITS - Kleinberg's Algorithm

- HITS – Hypertext Induced Topic Selection
- For each vertex v Є V in a subgraph of interest:

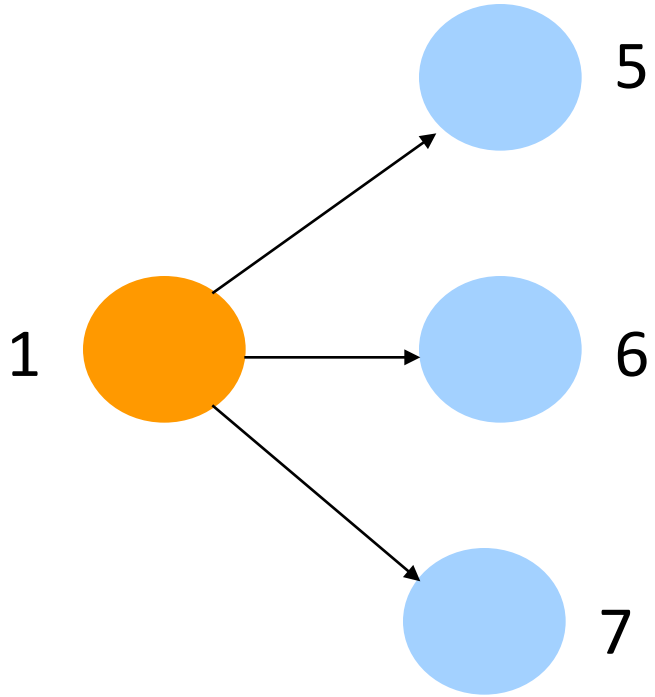
  a(v) - the authority of v
  h(v) - the hubness of v

- A site is very authoritative if it receives many citations. Citation from important sites weight more than citations from less-important sites
- Hubness shows the importance of a site. A good hub is a site that links to many authoritative sites

Authority and Hubness



$a(1) = h(2) + h(3) + h(4)$



$h(1) = a(5) + a(6) + a(7)$

Authority and Hubness Convergence

- Recursive dependency:

$$\alpha(v) = \sum_{w \in pa(v)} h(w)$$

$$h(v) = \sum_{w \in ch(v)} \alpha(w)$$

- Using Linear Algebra, we can prove:

$$\alpha(v), h(v) \quad \text{converge}$$

HITS Example

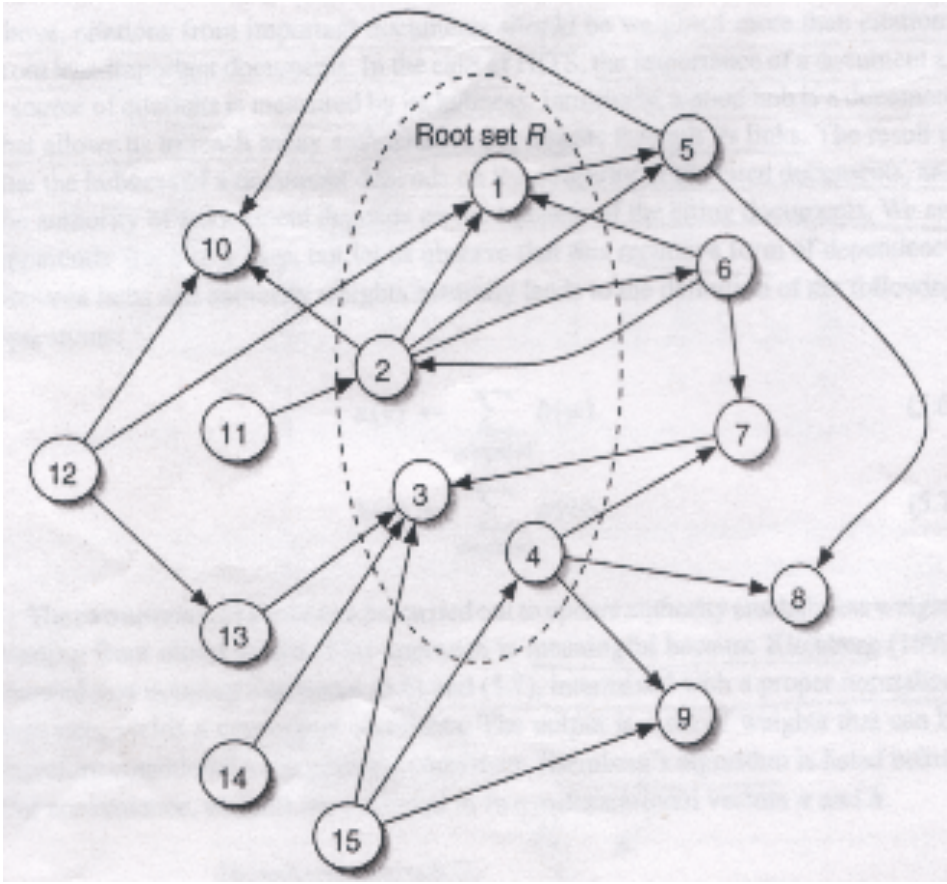
Find a base subgraph:

- Start with a root set R {1, 2, 3, 4}

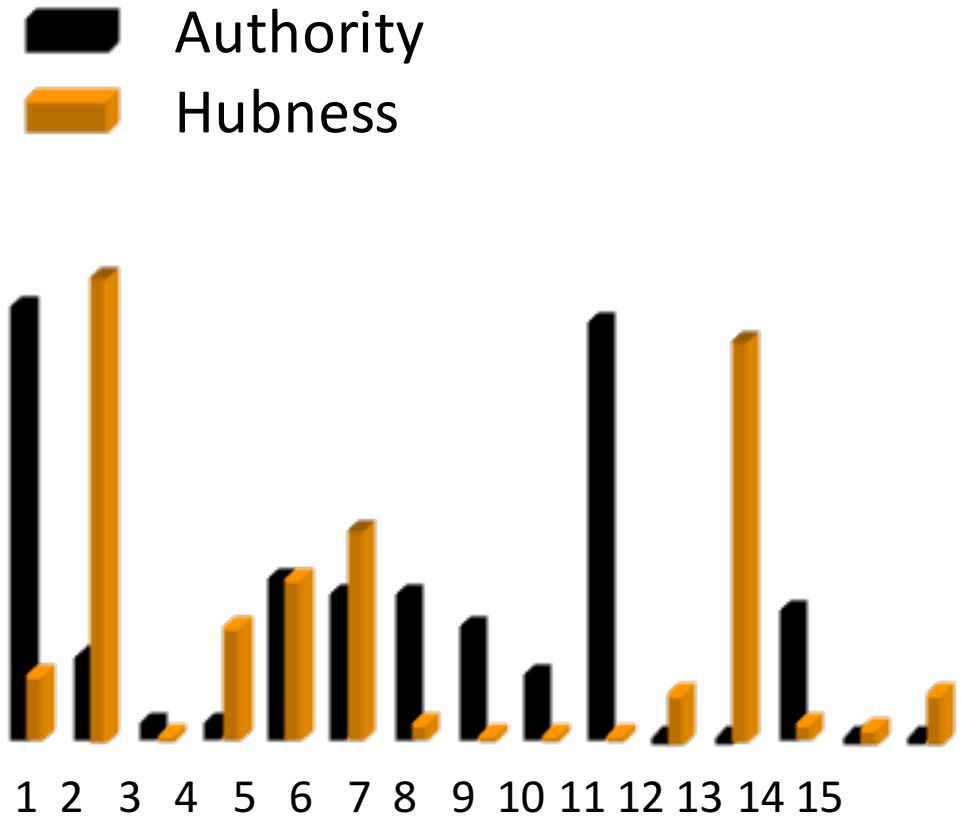
- {1, 2, 3, 4} - nodes relevant to the topic

- Expand the root set R to include all the children and a fixed number of parents of nodes in R

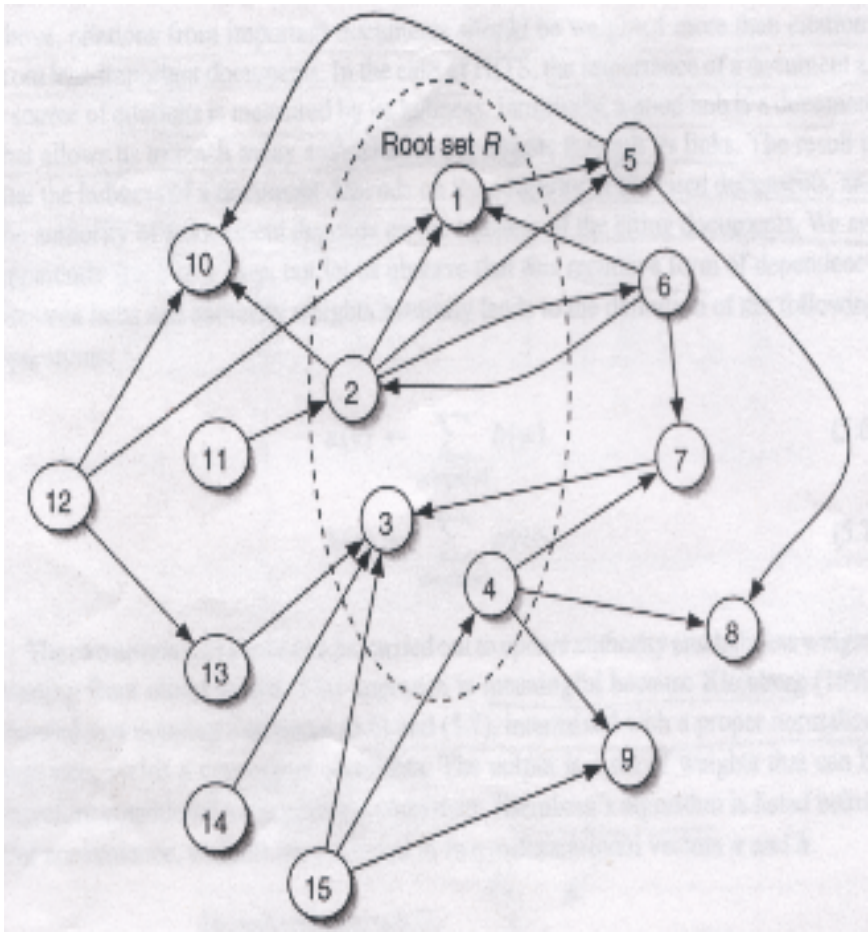
→ A new set S (base subgraph) →



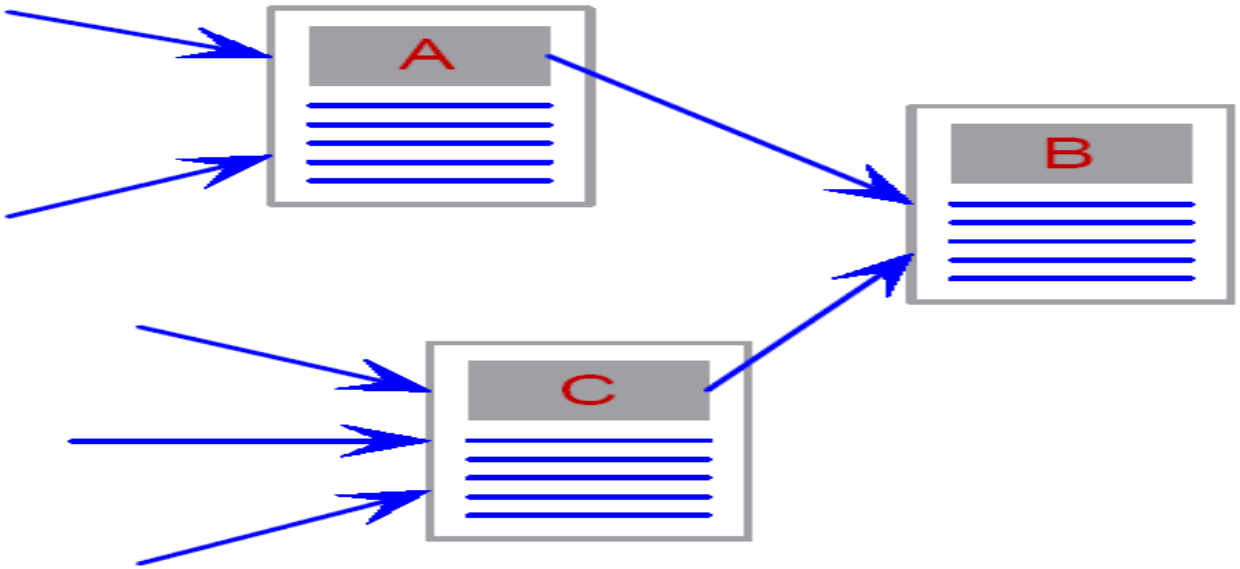
HITS Example Results



Authority and hubness weights



PageRank - Motivation

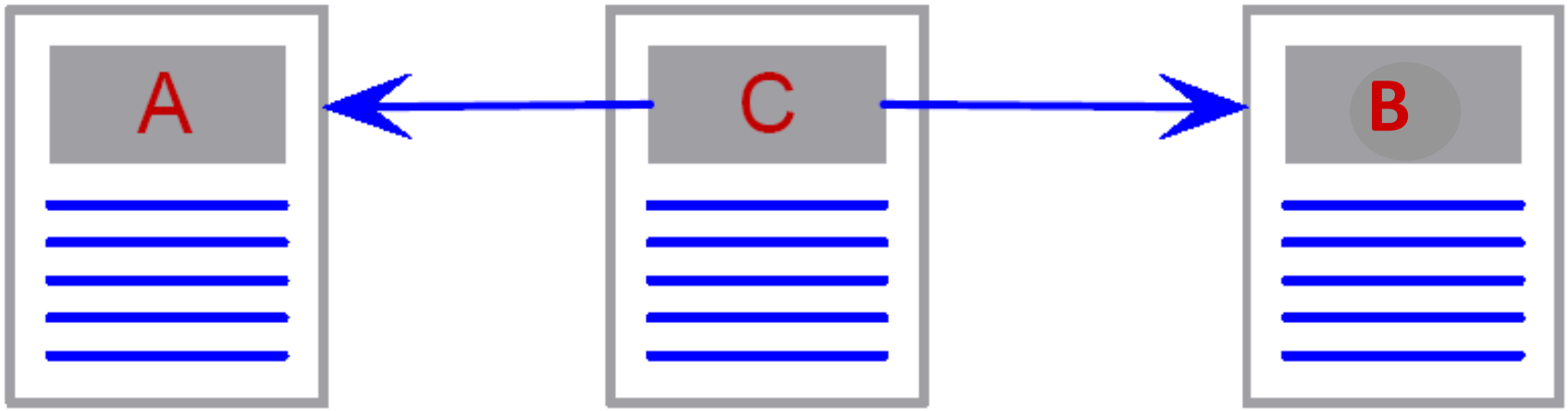


- A link from page *A* to page *B* is a **vote** of the author of *A* for *B,* or a **recommendation** of the page.
- The number incoming links to a page is a measure of importance and authority of the page.
- Also take into account the quality of recommendation, so a page is more important if the sources of its incoming links are important.

The Random Surfer

- Assume the web is a Markov chain.
- Surfers randomly click on links, where the probability of an outlink from page A is *1/m*, where *m* is the number of outlinks from *A.*
- The surfer occasionally gets *bored* and is *teleported* to another web page, say *B,* where *B* is equally likely to be any page.
- Using the theory of Markov chains it can be shown that if the surfer follows links for long enough, *the PageRank of a web page is the probability that the surfer will visit that page*.

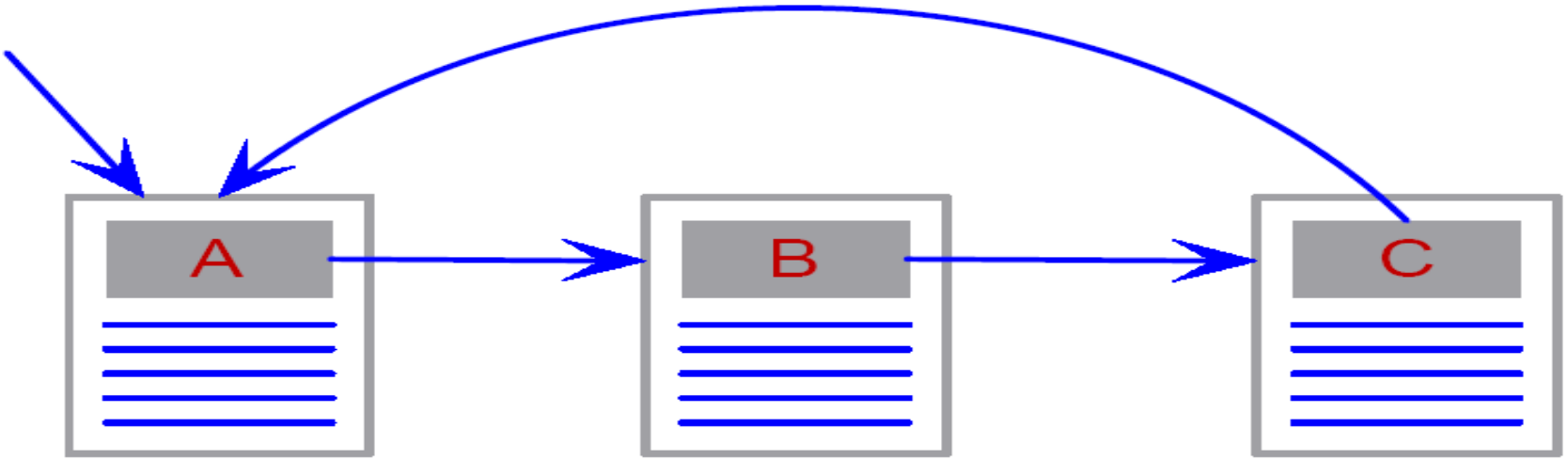
Dangling Pages



- Problem: *A* and *B* have no outlinks.

Solution: Assume *A* and *B* have links to all web pages with equal probability.

Rank Sink



- Problem: Pages in a loop accumulate rank but do not distribute it.
- Solution: Teleportation, i.e. with a certain probability the surfer can jump to any other web page to get out of the loop.

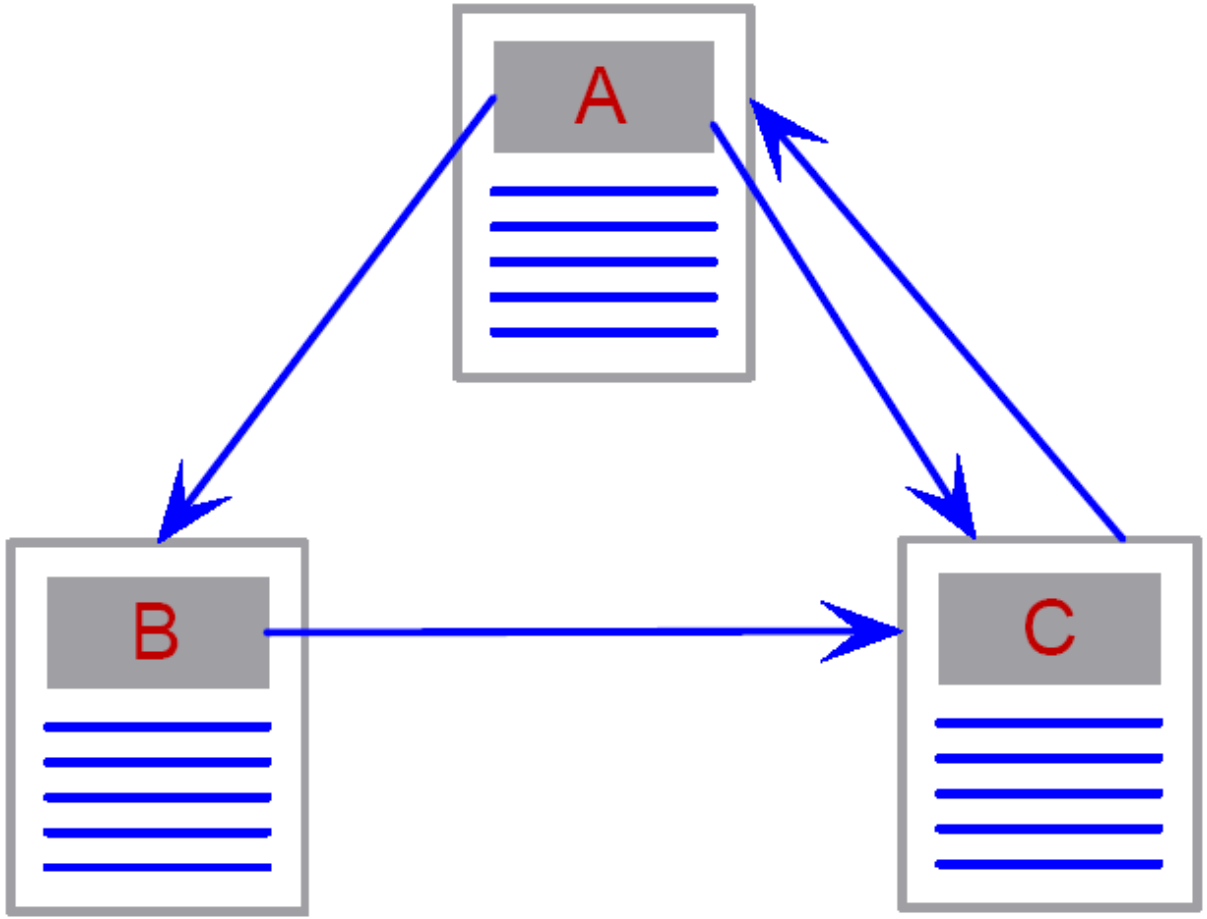
PageRank (*PR*) – Definition

$$p(P) = \frac{d}{N}(1-d)(\frac{p(P_1}{O(P_1)} + \frac{p(P_2}{O(P_2)} + \ldots + \frac{p(P_n}{O(P_n)})$$

- P is a web page
- Pi are the web pages that have a link *to P*
- $O(Pi)$ is the number of outlinks from *Pi*
- d is the teleportation probability
- N is the size of the web

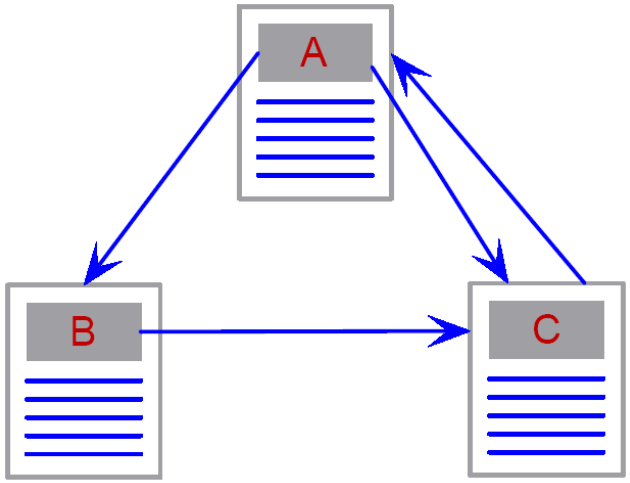
- Difference to HITS
  - HITS takes Hubness & Authority weights
  - The page rank is proportional to its parents' rank, but inversely proportional to its parents' outdegree

Example Web Graph



Iteratively Computing PageRank

- d is normally set to 0.15
- Set initial *PR* values to 1/3
- *Solve the following equations iteratively:*



$$PR(A) = 0.15/3 + 0.85PR(C)$$

$$PR(B) = 0.15/3 + 0.85(PR(A)/2)$$

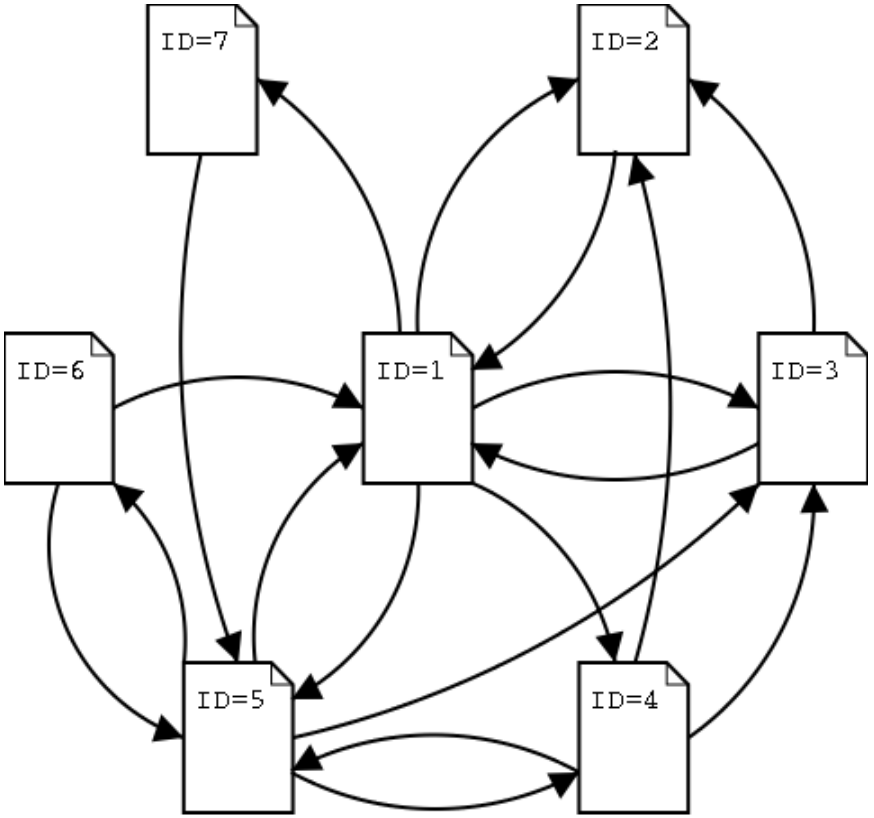
$$PR(C) = 0.15/3 + 0.85(PR(A)/2 + PR(B))$$

Example Computation of PR

	PR(A)	PR(B)	PR(C)	ERROR
1	0,333333333	0,333333333	0,333333333	
2	0,333333333	0,191666667	0,475	0,04014
3	0,45375	0,191666667	0,354583333	0,029
4	0,351395833	0,24284375	0,405760417	0,01571
5	0,394896354	0,199343229	0,405760417	0,00378
6	0,394896354	0,217830951	0,387272695	0,00068
7	0,379181791	0,217830951	0,402987258	0,00049
8	0,39253917	0,211152261	0,396308569	0,00027
9	0,386862284	0,216829147	0,396308569	6,4E-05
10	0,386862284	0,214416471	0,398721246	1,2E-05
11	0,388913059	0,214416471	0,396670471	8,4E-06
12	0,3871699	0,21528805	0,39754205	4,6E-06
13	0,387910742	0,214547208	0,39754205	1,1E-06
14	0,387910742	0,214862066	0,397227192	2E-07
15	0,387643113	0,214862066	0,397494821	1,4E-07
16	0,387870598	0,214748323	0,397381079	7,8E-08
17	0,387773917	0,214845004	**0,397381079**	1,9E-08

- Error converges fast, ~10 repetitions
- Page C is the top ranked one

Matrix Notation



Page ID	OutLinks
1	2,3,4,5,7
2	1
3	1,2
4	2.3.5
5	1,3,4,6
6	1.5
7	5

Adjacency Matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

* http://www.kusatro.kyoto-u.com

A Adjacency Matrix: Lets decomponse A into its Eigenvector 'v' and Eigenvalue 'lambda'
=> A.v = lambda.v
Eigenvector v is an (n * 1) matrix
v is the right Eigenvector

If u is a left EV to A ALORS u.A = ketta.u
=> ATransp.uTransp = ketta.uTransp
=> 'u' the left eigenvector of A is the same as the transpose of a right eigenvector of ATranspose, with the same eigenvalue.

* https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors + https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix

After decomposing the Adjacency Matrix, the normalized Eigenvector is the PageRank

Matrix Notation

PageRank: eigenvector of **A** relative to max eigenvalue

$$\mathbf{A} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T$$

L: diagonal matrix of eigenvalues $\{\lambda_1, \ldots \lambda_n\}$

U: regular matrix that consists of eigenvectors

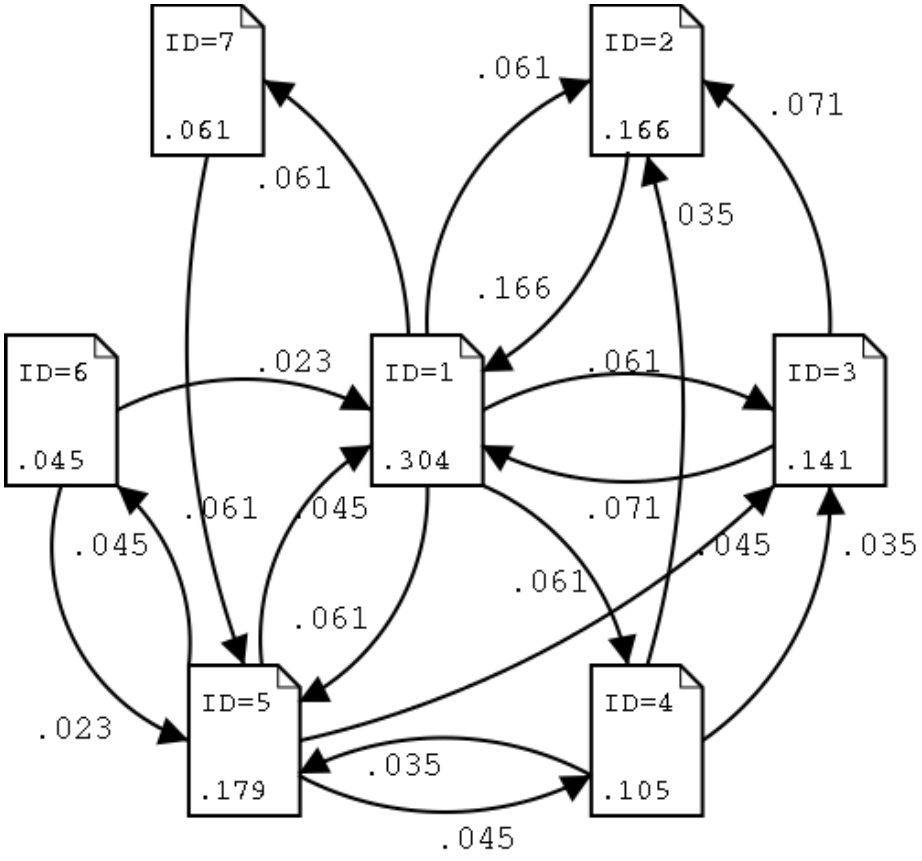
$$\begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \ddots \\ 0 & & \lambda_n \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_n \end{pmatrix}$$

Approximation: Power method: $\mathbf{P^{i+1} = A\,P^i}$

PageRank $\mathbf{r}_1$ = $\begin{pmatrix} 0.69946 \\ 0.38286 \\ 0.32396 \\ 0.24297 \\ 0.41231 \\ 0.10308 \\ 0.13989 \end{pmatrix}$ → normalized $\begin{pmatrix} 0.303514 \\ 0.166134 \\ 0.140575 \\ 0.105431 \\ 0.178914 \\ 0.044728 \\ 0.060703 \end{pmatrix}$

Matrix Notation



PR	ID	OutLink	InLink
0.304	1	2,3,4,5,7	2,3,5,6
0.179	5	1,3,4,6	1,4,6,7
0.166	2	1	1,3,4
0.141	3	1,2	1,4,5
0.105	4	2,3,5	1,5
0.061	7	5	1
0.045	6	1,5	5

- Confirm the result

  # of inlinks from high ranked page

  hard to explain about 5&2, 6&7

- Interesting Topic

  How do you create your homepage highly ranked?

“Rank Sink” Problem

- In general, many Web pages have no inlinks/outlinks
- It results in dangling edges in the graph

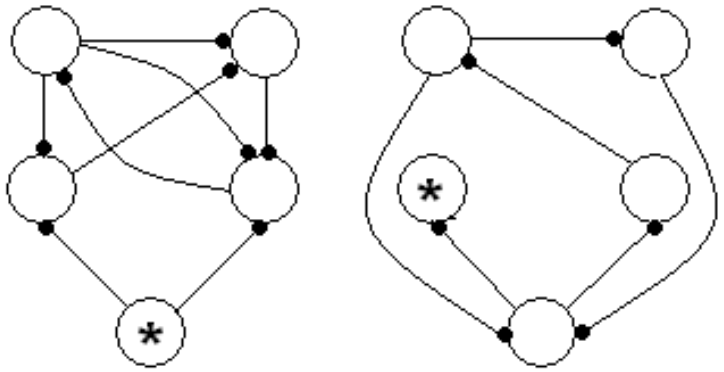
E.g.

no parent → rank 0

M^T converges to a matrix whose last column is all zero

no children → no solution

M^T converges to zero matrix



Modification

- Surfer will restart browsing by picking a new Web page at random

$$\mathbf{M} = (\mathbf{B} + \mathbf{E}), \quad e_{vw} = \begin{cases} 0 & \text{if } |ch[v]| > 0 \\ \frac{1}{n} & \text{otherwise} \end{cases}$$

E : escape matrix

M : stochastic matrix

- Still problem?
  - It is not guaranteed that **M** is primitive
  - If **M** is stochastic and primitive, PageRank converges to corresponding stationary distribution of **M**

PageRank Algorithm

PAGERANK(M, n, ϵ)

1 $\mathbf{1} \leftarrow [1, \ldots, 1] \in \mathbb{R}^n$
2 $z \leftarrow \frac{1}{n}\mathbf{1}$
3 $x_0 \leftarrow z$
4 $t \leftarrow 0$
5 **repeat**
6 $\quad t \leftarrow t + 1$
7 $\quad x_t \leftarrow M^{\mathrm{T}} x_{t-1}$
8 $\quad d_t \leftarrow \|x_{t-1}\|_1 - \|x_t\|_1$
9 $\quad x_t \leftarrow x_1 + d_t z$
10 $\quad \delta \leftarrow \|x_{t-1} - x_t\|_1$
11 **until** $\delta < \epsilon$
12 **return** x_t

* Page et al, 1998

The Largest Matrix Computation in the World

- Computing PageRank can be done via matrix multiplication, where the matrix has several billion rows and columns.
- The matrix is sparse as average number of outlinks is between 7 and 8.
- Setting d = 0.15 or below requires at most 100 iterations to convergence.
- Researchers still trying to speed-up the computation.

Ranking function web search

- Web search engines take into account 100's of features to rank documents assuming a query
- Two important features are
  - The *PageRank* value of the page containing the *query* terms
  - The *relevance* of the term to the specific page
- Given a term *t* the score of a document *d* is computed as:

$$score_t(di) = w_1(\text{relevance}(t, d_i)) + w_2 pr(d_i)$$

- Where relevance: tf-idf, BM25 etc.
- In a specific case we used:

$$score_t(d) = (\text{tf/idf}(t, d) \cdot \text{title}(t, d))^{1.5} \cdot pr(d)$$

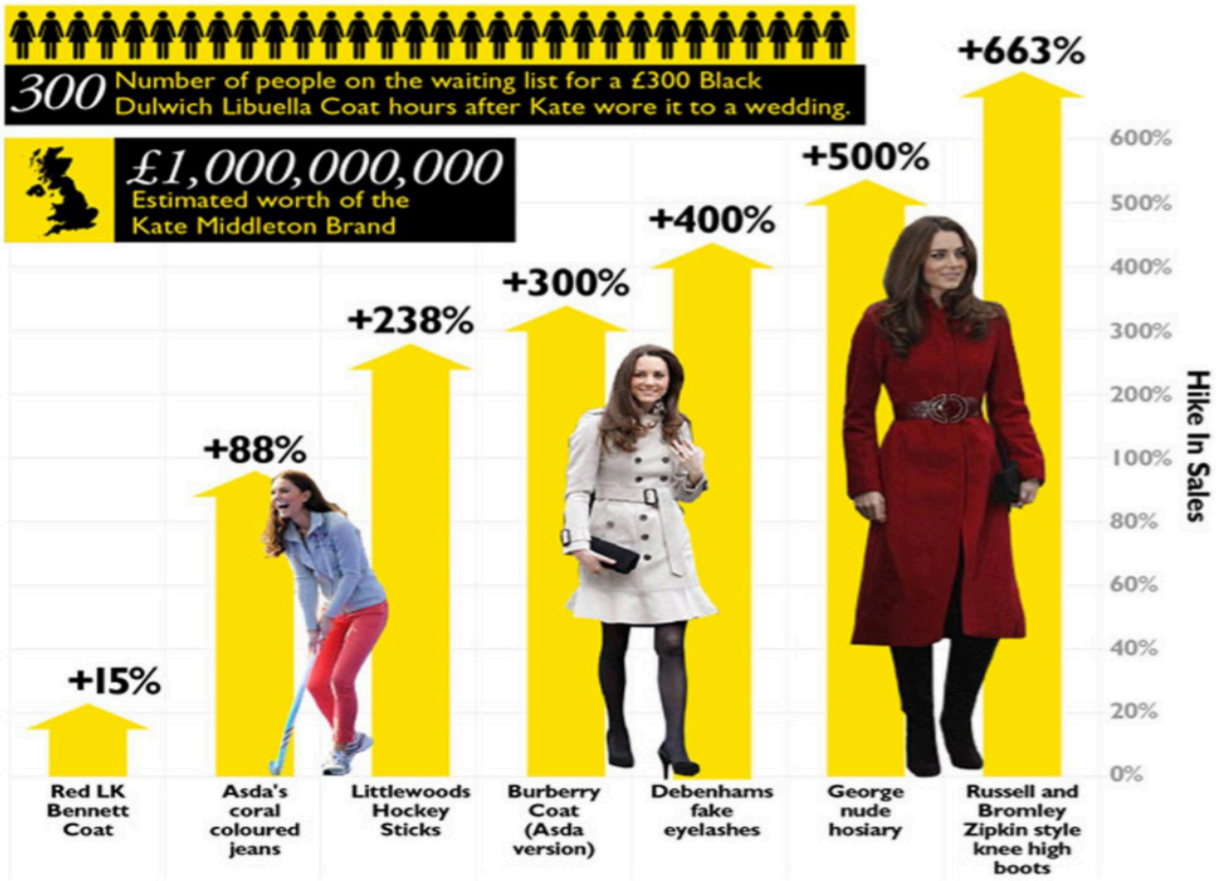
References

- “Graph databases”, Ian Robinson Jim Weber and Emily Eifren, O’Reilly
- Amy Nicole Langville, Carl Dean Meyer: Survey: Deeper Inside PageRank. Internet Mathematics 1(3): 335-380 (2003)
- “PageRank Computation and the Structure of the Web: Experiments and Algorithms”, Arvind Arasu, Jasmine Novak, Andrew Tomkins & John Tomlin

Topics

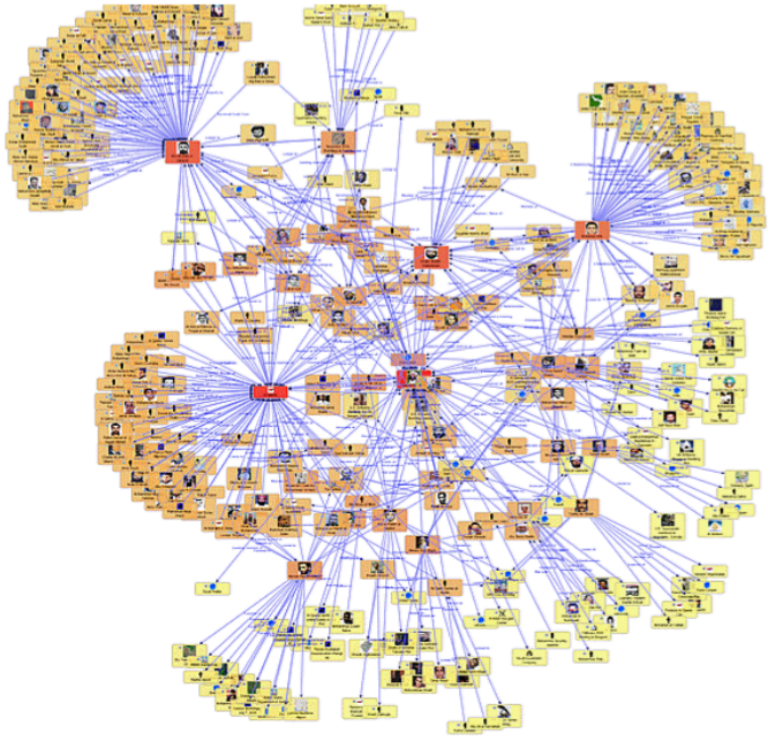
1. **Recommendation/Personalization**
2. **Pagerank**
3. **Influence mazimization**

Kate Middleton Effect



Online Social Influence Analysis

- Big Datasets
  - Real time, Dynamic



- New Services
  - Social Influence → $

Outline

- The basic ideas on Influence Marketing
- **Competing approaches**
- Our approach on Influence Marketing
- The next level of Influence Marketing

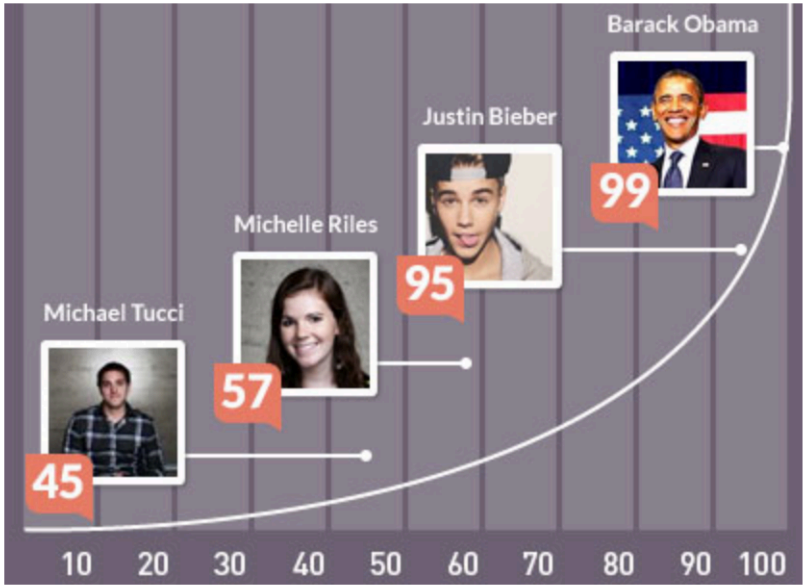
Competing approaches
Influence Marketing world

- **Many start ups are positioned for Influence Marketing:**

  Hypr, Julius, Sysomos, Scrunch, Brandnew, Reech, Tapinfluence,..

- **The process is threefold**
  1. **Influence measure (#followers – Klout)**
  2. **Selection of top influencers**
  3. **Measurement of campaign success**

Influence measure
Klout score

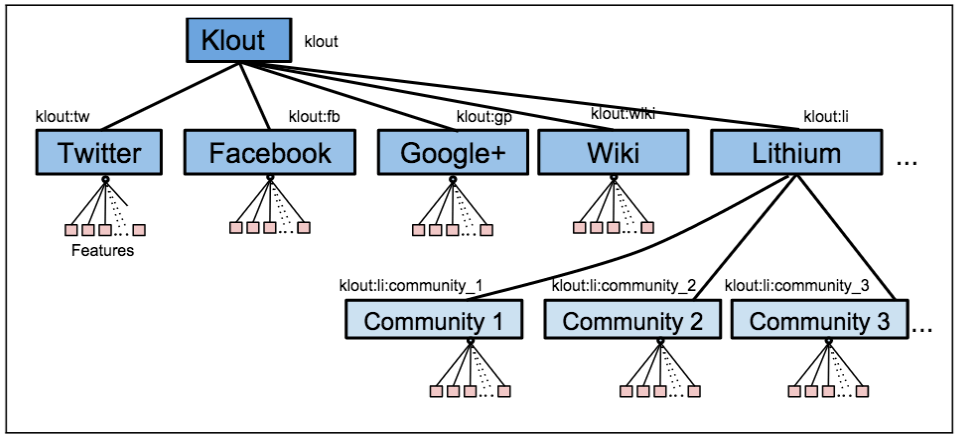


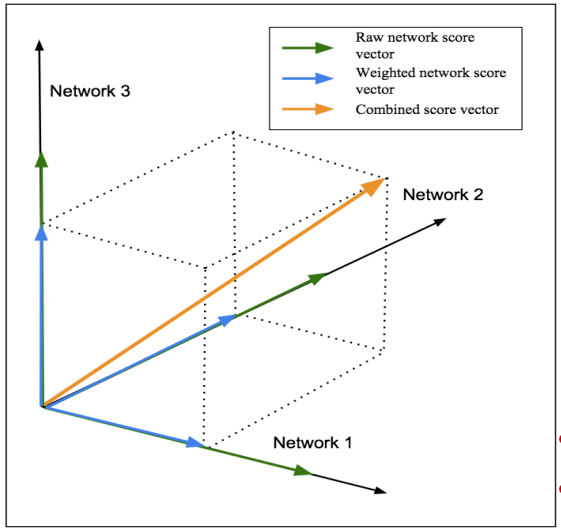
HOW IT'S CALCULATED

We measure multiple pieces of data from several social networks, and also real world data from places like Bing and Wikipedia. Then we apply them to our Klout Score algorithm, and then show the resulting number on your profile. The higher your Klout Score, the tougher it becomes to increase. More information about measuring influence can be found in the published paper here.

https://klout.com/corp/score

Influence measure
Klout – score computation





- Heuristics
- Simple Aggregation

Network Scalability:

- user's spans multiple social and professional networks.

Interaction Graph:

- influence measurement based solely on structural graph properties (degree and centrality heuristics) do not perform well- essential to consider information dynamics in the network.
- interactions may indicate a greater degree of influence than others.

Temporal factors:

- long-lasting vs. dynamic/changing influence.
- klout time window: 90 days to consider

Offline factors:

- signals on social networks are only a partial representation of a user's overall influence,
- crucial to incorporate proxy sources that signify a user's real world influence (i.e. Wikipedia, news articles).

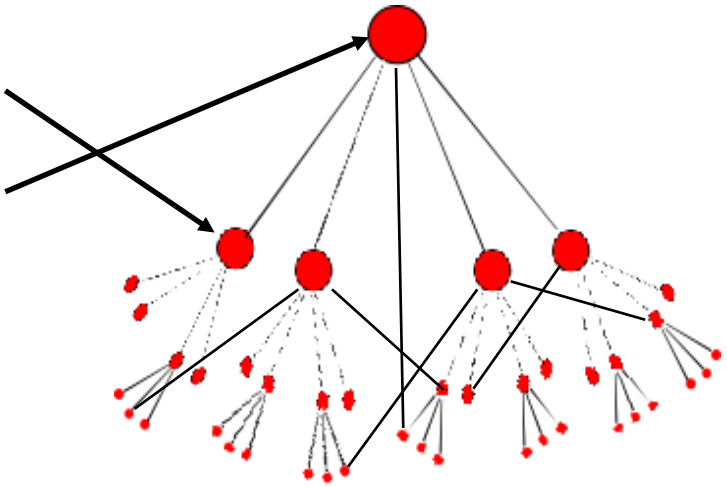
Significant resources needed:

- 45 bn interactions each pipeline run,
- 0.5 bn new interactions/day.
- daily footprint: 196.14CPU days, 18.46TB reads, 9.53TB writes

Influence measure
Klout – Graph based features

Table I: Types of Long-lasting features

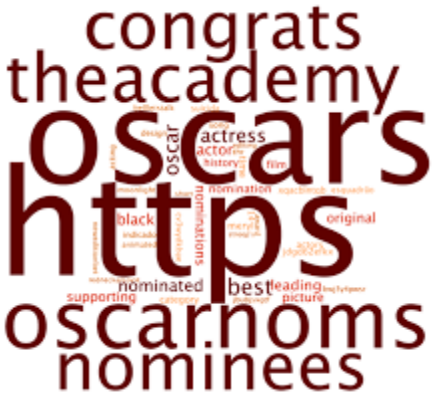
Feature Type	Features	Networks
Node Degree	Followers, Friends, Fans, Subscribers, In-links	TW, FB, IG, GP, WK, YT
Graph Properties	PageRank, Inlink to Outlink ratio	WK
Categories	Job Title, Education Level, Endorsements, Recommendations, Awards, Community Badges	LI, LT



https://arxiv.org/pdf/1510.08487.pdf

Selection of top influencers

- Profiling of influencers:
  - keywords/affinity to topics
  - buzzgraph **[Sysomos]**
- Profiling of influencers' audiences:
  - Keywords/affinity to topics
  - Demographics/location
  - IBM Watson **[Tapinfluence]**

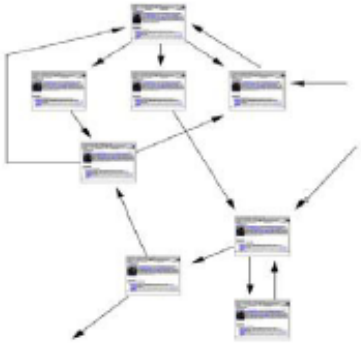


Measure of campaign success

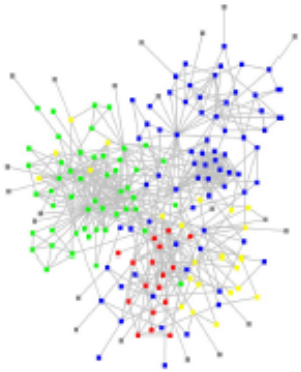
- Amount of content generated: volume of message reproduction (i.e. retweet) by users
- Engagement: number of times people interact with the brand on each social media platform
- Number of clicks: measures how many clicks the brand's links receive
- Conversion Rate:
  - Conversion lead: number of leads divided by total traffic
  - Conversion sale: ROI=number of sales divided by total traffic

Graphs are ubiquitous!

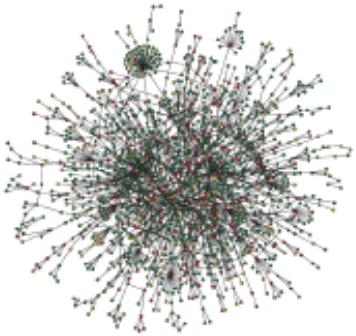
(a) Internet



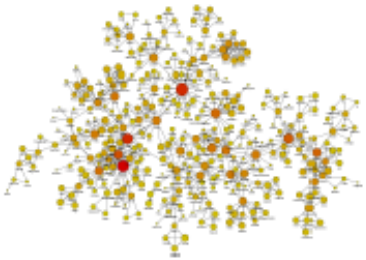
(b) World Wide Web



(c) Email network



(d) Protein interactions

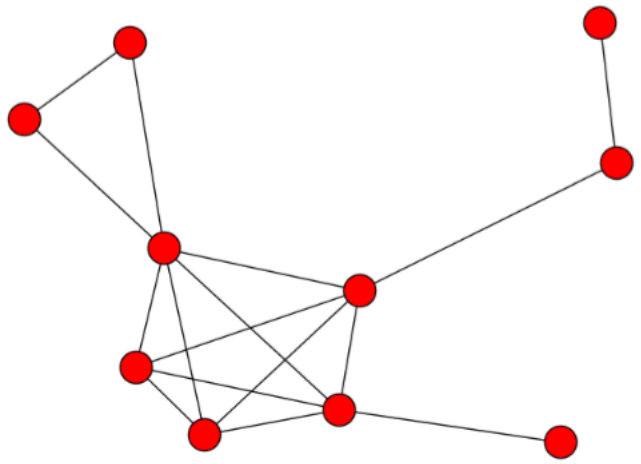


(e) Collaboration network

(f) Citation network

Dense Subgraph Discovery

- Given a graph or a network (social network, biological network, term network,...), the problem of dense subgraph discovery is to find a subgraph that is **dense** and has a **large number of nodes**.

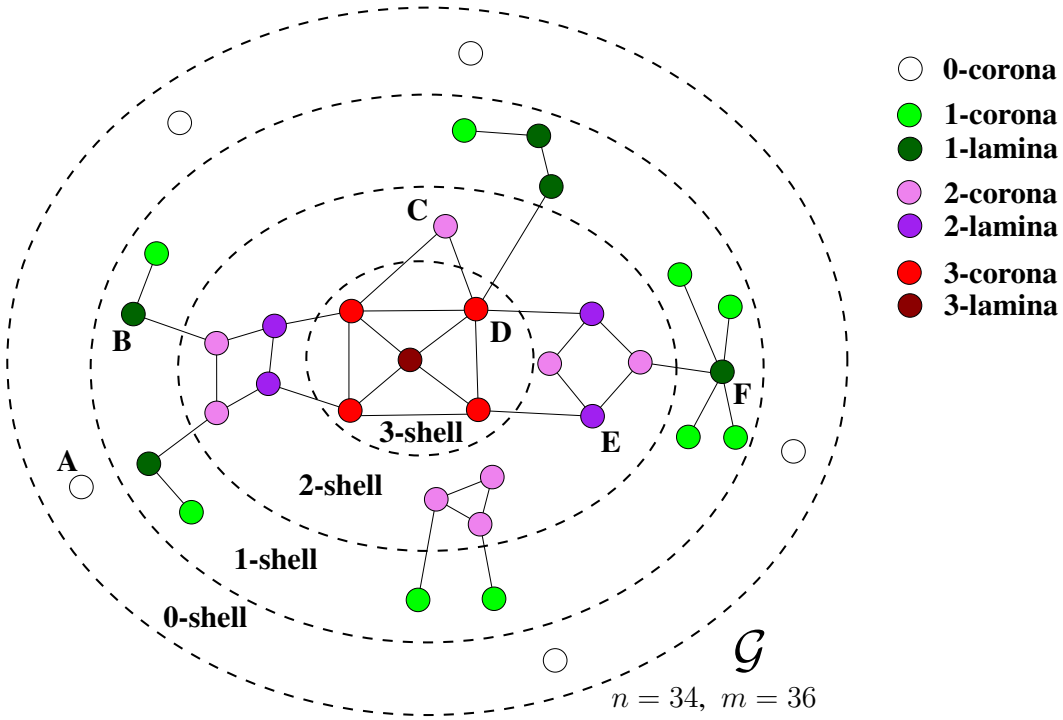


Numerous applications:

- bioinformatics [FNBB06]
- spam detection [GKT05]
- event detection [AKS+14]
- fraud detection [BXG+13]
- community detection [CS12]
- graph visualization [AHDBV05]

Graph Degeneracy

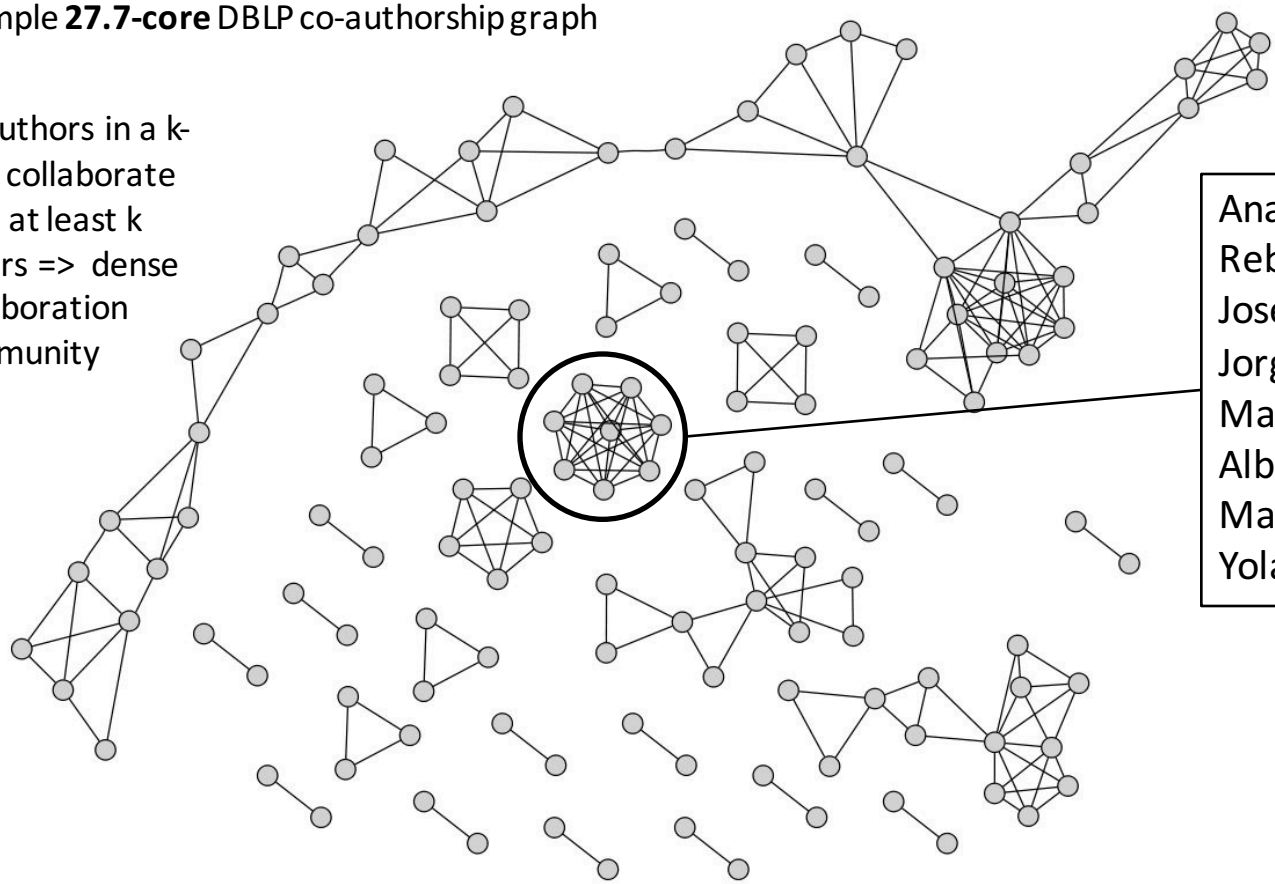
- **k-core**: maximal size subgraph each node has at least k neighbors in the subgraph - k : rank of such a core).
- The maximum k for which a graph contains a k-core is known as its degeneracy.
- k-core is an efficient approximation of the densest subgraph



Collaboration identification in co-authorship graphs

Example **27.7-core** DBLP co-authorship graph

All authors in a k-core collaborate with at least k others => dense collaboration community

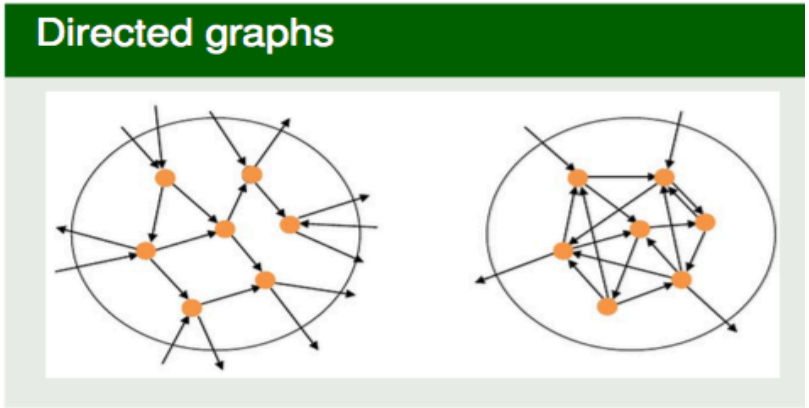


Ana Fernandez Vilas
Rebeca P. Diaz Redondo
Jose J. Pazos Arias
Jorge Garcia Duque
Martin Lopez Nores
Alberto Gil-Solla
Manuel Ramos Cabrer
Yolanda Blanco-Fernandez

http://www.graphdegeneracy.org/

Degeneracy in directed graphs

- Directed graphs
  - **Wikipedia**
  - **DBLP** - Citation graph



- Is there a degeneracy notion for **directed graphs**?
- We extend the ***k***-core concept in directed graphs by applying a limit on in/out edges respectively
- This provides a two dimensional range where cores degenerate
- Trade off between in/out edges can give us a more specific view of the cohesiveness and the **social** behavior

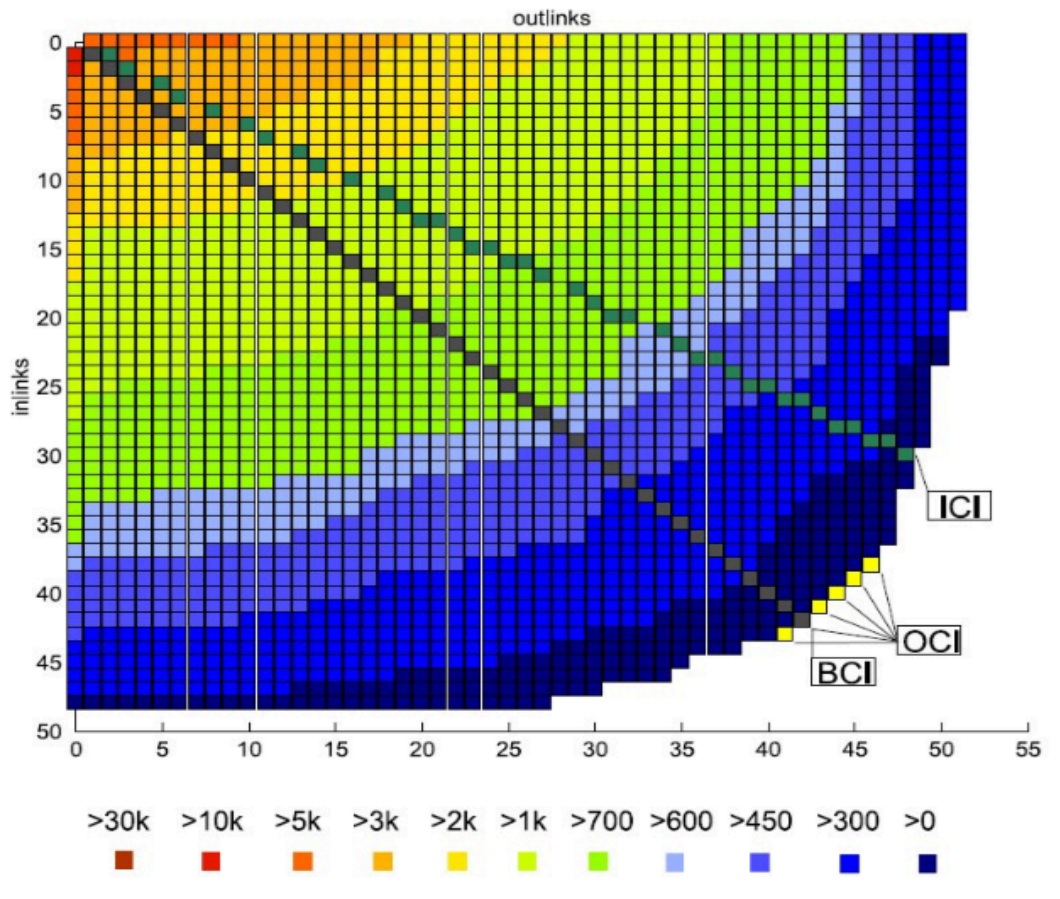
[Giatsidis et al., ICDM '11; KAIS '13]

D-core: definitions

- Given a directed graph D. We define:
  - $\delta^{in}(D) = \min\{x \big| deg_D^{in}(x) \big| x \in V(D)\}$
  - $\delta^{out}(D) = \min\{x | deg_D^{out}(x) | x \in V(D)\}$
  - A (k,l)-D-*core* of D is a maximal sub-digraph F of D: $\delta^{out}(F) > k \quad \text{and} \quad \delta^{\text{in}}(\text{F}) > \text{l}$
- This creates a two dimensional range where cores degenerate

D-core Matrix of DBLP citations

D-core of DBLP

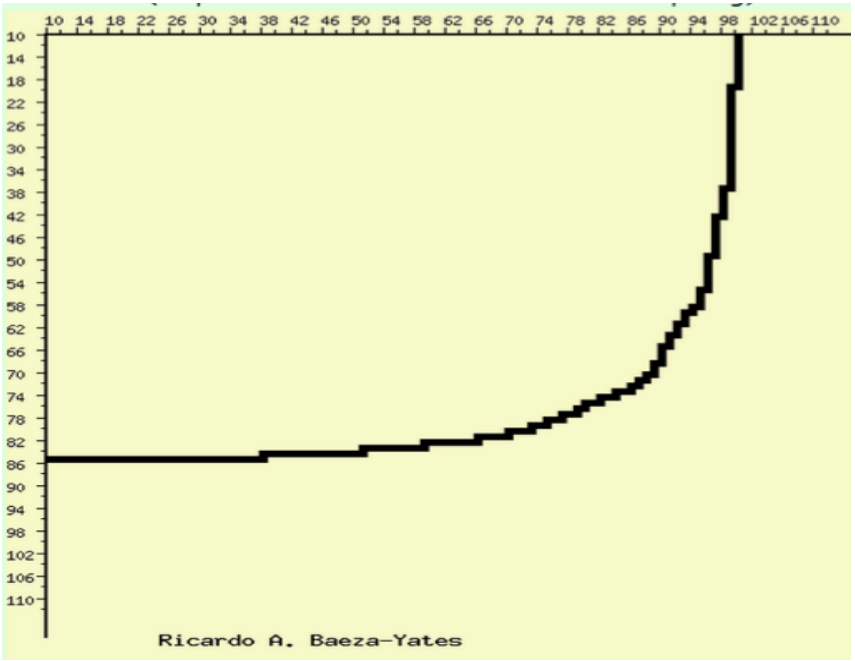


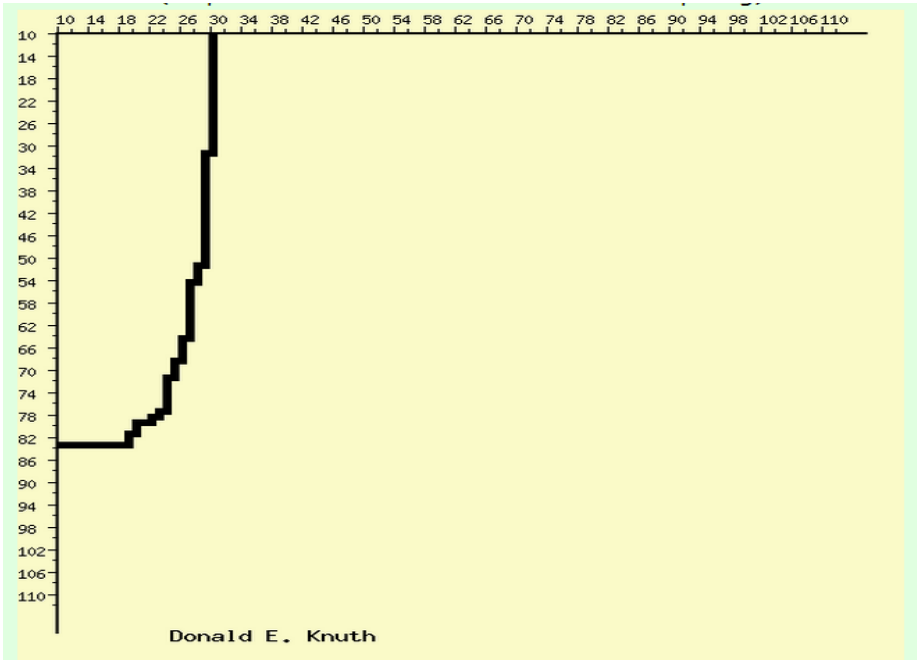
DBLP: the main core authors

- JosÃ© A. Blakeley
- Hector Garcia-Molina
- Abraham Silberschatz
- Umeshwar Dayal
- Eric N. Hanson
- Jennifer Widom
- Klaus R. Dittrich
- Nathan Goodman
- Won Kim
- Alfons Kemper
- Guido Moerkotte
- Clement T. Yu
- M. Tamer Ã–zsu
- Amit P. Sheth
- Ming-Chien Shan
- Richard T. Snodgrass
- David Maier
- Michael J. Carey
- David J. DeWitt
- Joel E. Richardson
- Eugene J. Shekita
- Waqar Hasan
- Marie-Anne Neimat
- Darrell Woelk
- Roger King
- Stanley B. Zdonik
- Lawrence A. Rowe
- Michael Stonebraker
- Serge Abiteboul
- Richard Hull
- Victor Vianu
- Jeffrey D. Ullman
- Michael Kifer
- Philip A. Bernstein
- Vassos Hadzilacos
- Elisa Bertino
- Stefano Ceri
- Georges Gardarin
- Patrick Valduriez
- Ramez Elmasri
- Richard R. Muntz
- David B. Lomet
- Betty Salzberg
- Shamkant B. Navathe
- Arie Segev
- Gio Wiederhold
- Witold Litwin
- Theo HÃ¤rder
- FranÃ§ois Bancilhon
- Raghu Ramakrishnan
- Michael J. Franklin
- Yannis E. Ioannidis
- Henry F. Korth
- S. Sudarshan
- Patrick E. O'Neil
- Dennis Shasha
- Shamim A. Naqvi
- Shalom Tsur
- Christos H. Papadimitriou
- Georg Lausen
- Gerhard Weikum
- Kotagiri Ramamohanarao
- Maurizio Lenzerini
- Domenico SaccÃ
- Giuseppe Pelagatti
- Paris C. Kanellakis
- Jeffrey Scott Vitter
- Letizia Tanca
- Sophie Cluet
- Timos K. Sellis
- Alberto O. Mendelzon
- Dennis McLeod
- Calton Pu
- C. Mohan
- Malcolm P. Atkinson
- Doron Rotem
- Michel E. Adiba
- Kyuseok Shim
- Goetz Graefe
- Jiawei Han
- Edward Sciore
- Rakesh Agrawal
- Carlo Zaniolo
- V. S. Subrahmanian
- Claude Delobel
- Christophe LÃ©cluse
- Michel Scholl
- Peter C. Lockemann
- Peter M. Schwarz
- Laura M. Haas
- Arnon Rosenthal
- Erich J. Neuhold
- Hans-JÃ¶rg Schek
- Dirk Van Gucht
- Hamid Pirahesh
- Marc H. Scholl
- Peter M. G. Apers
- Allen Van Gelder
- Tomasz Imielinski
- Yehoshua Sagiv
- Narain H. Gehani
- H. V. Jagadish
- Eric Simon
- Peter Buneman
- Dan Suciu
- Christos Faloutsos
- Donald D. Chamberlin
- Setrag Khoshafian
- Toby J. Teorey
- Randy H. Katz
- Miron Livny
- Philip S. Yu
- Stanley Y. W. Su
- Henk M. Blanken
- Peter Pistor
- Matthias Jarke
- Moshe Y. Vardi
- Daniel BarbarÃ¡
- Uwe Deppisch
- H.-Bernhard Paul
- Don S. Batory
- Marco A. Casanova
- JÃ¼rgen Koch
- Joachim W. Schmidt
- Guy M. Lohman
- Bruce G. Lindsay
- Paul F. Wilms
- Z. Meral Ã–zsoyoglu
- Gultekin Ã–zsoyoglu
- Kyu-Young Whang
- Shahram Ghandeharizadeh
- Tova Milo
- Alon Y. Levy
- Georg Gottlob
- Johann Christoph Freytag
- Klaus KÃ¼spert
- Louiqa Raschid
- John Mylopoulos
- Alexander Borgida
- Anand Rajaraman
- Joseph M. Hellerstein
- Masaru Kitsuregawa
- Sumit Ganguly
- Rudolf Bayer
- Raymond T. Ng
- Daniela Florescu
- Per-Ã…ke Larson
- Hongjun Lu
- Ravi Krishnamurthy
- Arthur M. Keller
- Catriel Beeri
- Inderpal Singh Mumick
- Oded Shmueli
- George P. Copeland
- Peter Dadam
- Susan B. Davidson
- Donald Kossmann
- Christophe de Maindreville
- Yannis Papakonstantinou
- Kenneth C. Sevcik
- Gabriel M. Kuper
- Peter J. Haas
- Jeffrey F. Naughton
- Nick Roussopoulos
- Bernhard Seeger
- Georg Walch
- R. Erbe
- Balakrishna R. Iyer
- Ashish Gupta
- Praveen Seshadri
- Walter Chang
- Surajit Chaudhuri
- Divesh Srivastava
- Kenneth A. Ross
- Arun N. Swami
- Donovan A. Schneider
- S. Seshadri
- Edward L. Wimmers
- Kenneth Salem
- Scott L. Vandenberg
- Dallan Quass
- Michael V. Mannino
- John McPherson
- Shaul Dar
- Sheldon J. Finkelstein
- Leonard D. Shapiro
- Anant Jhingran
- George Lapis

D-Core frontier for individuals

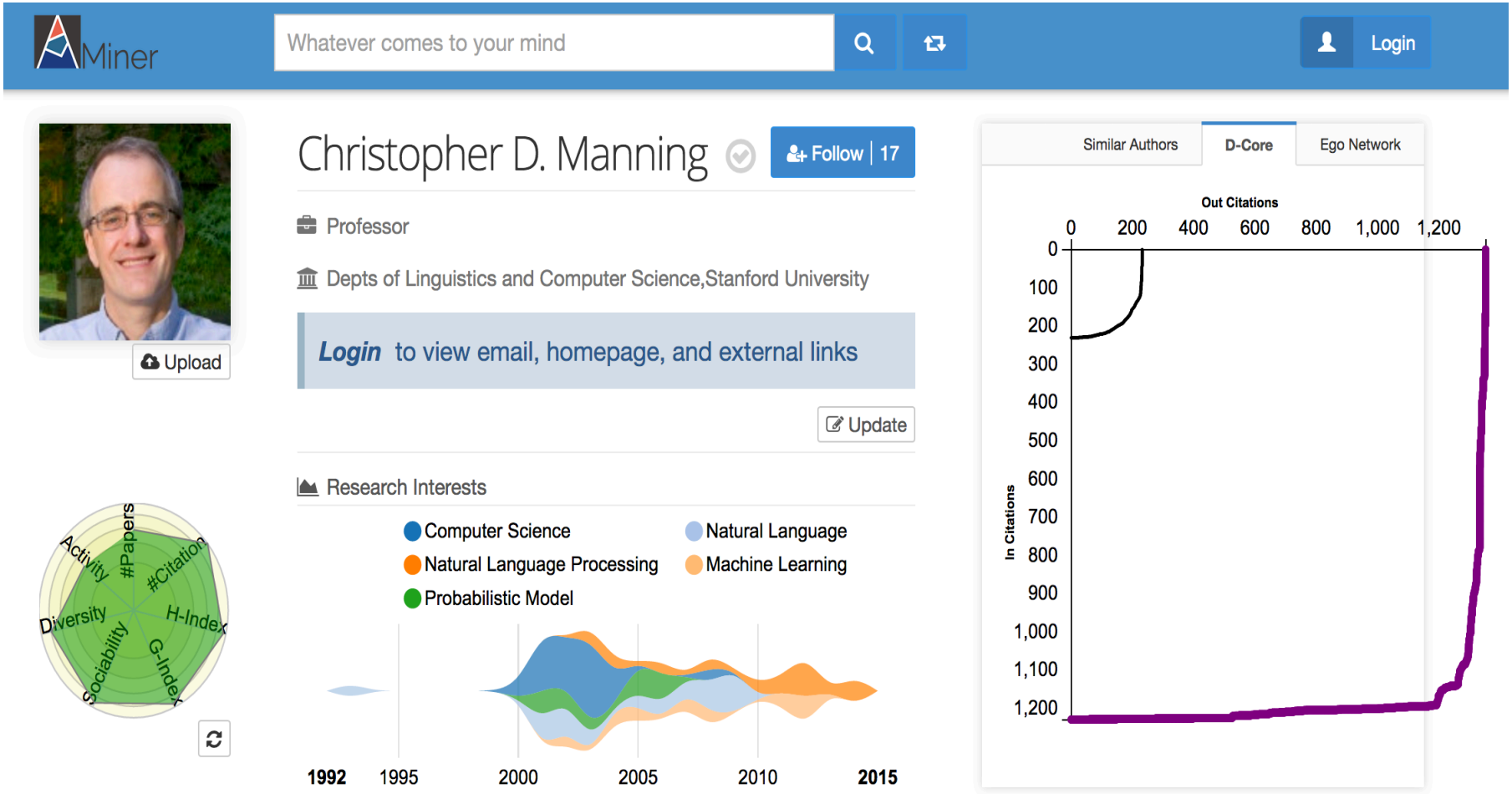
- The frontier of an individual: defined by the outmost d-cores that the individual belongs to
- Its shape defines authority vs. collaboration features
- The AUC is significant





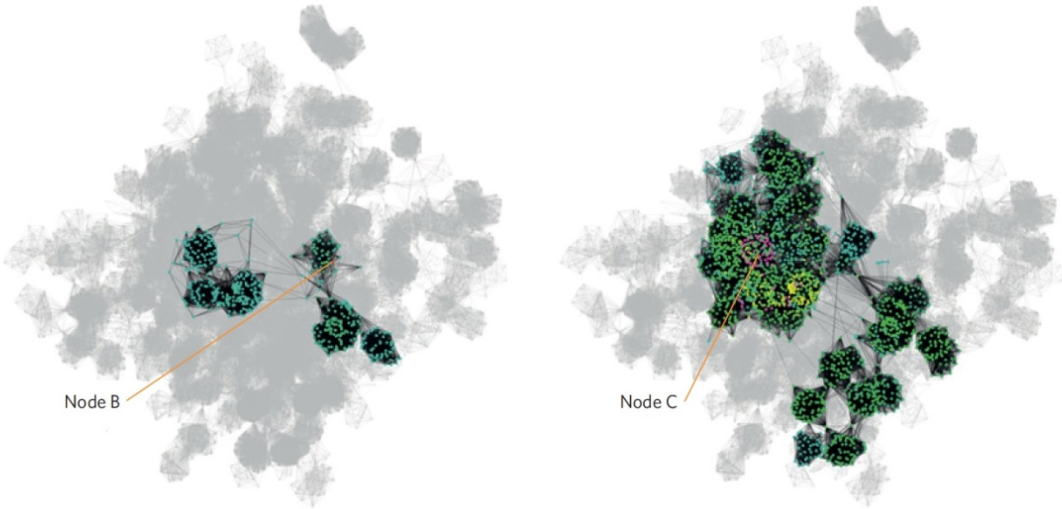
http://www.graphdegeneracy.org/

D-core adopted by aminer.org



https://aminer.org/profile/christopher-d-manning

Identifying Influential Spreaders

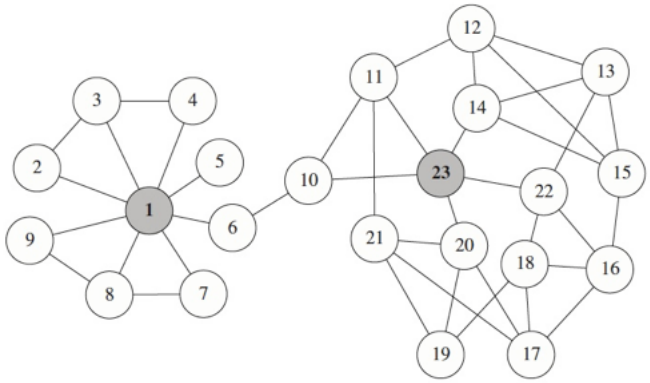


Goal: Find those nodes in the network that have a good influential power in a certain time frame

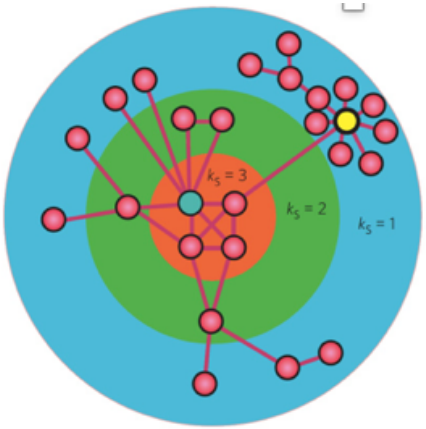
[M. Vazirgiannis et. al, Locating influential nodes in complex networks, Scientific Reports / Nature 2016]

Identifying Influential Spreaders

Degree centrality: straightforward metric to identify leaders in social networks



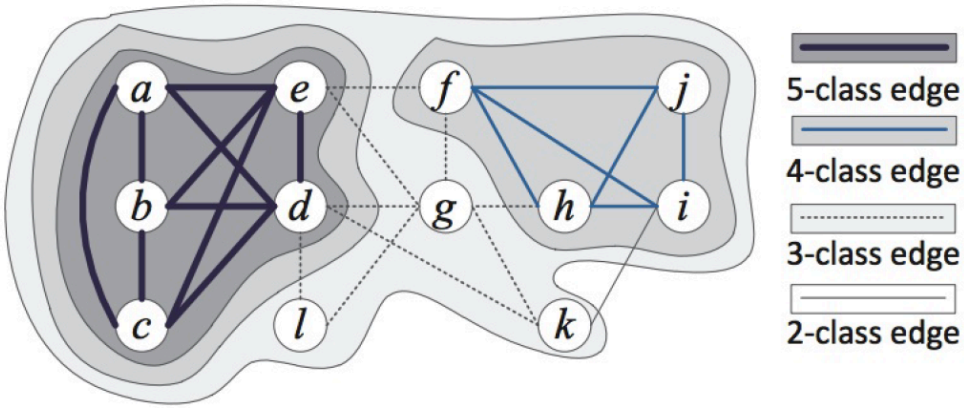
Most efficient spreaders are located within the k-core of the network



[Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., Makse, H. A. (2010). Identification of influential spreaders in complex networks. Nature Physics, 6(11), 888-893.

Identifying Influential Spreaders with K-truss

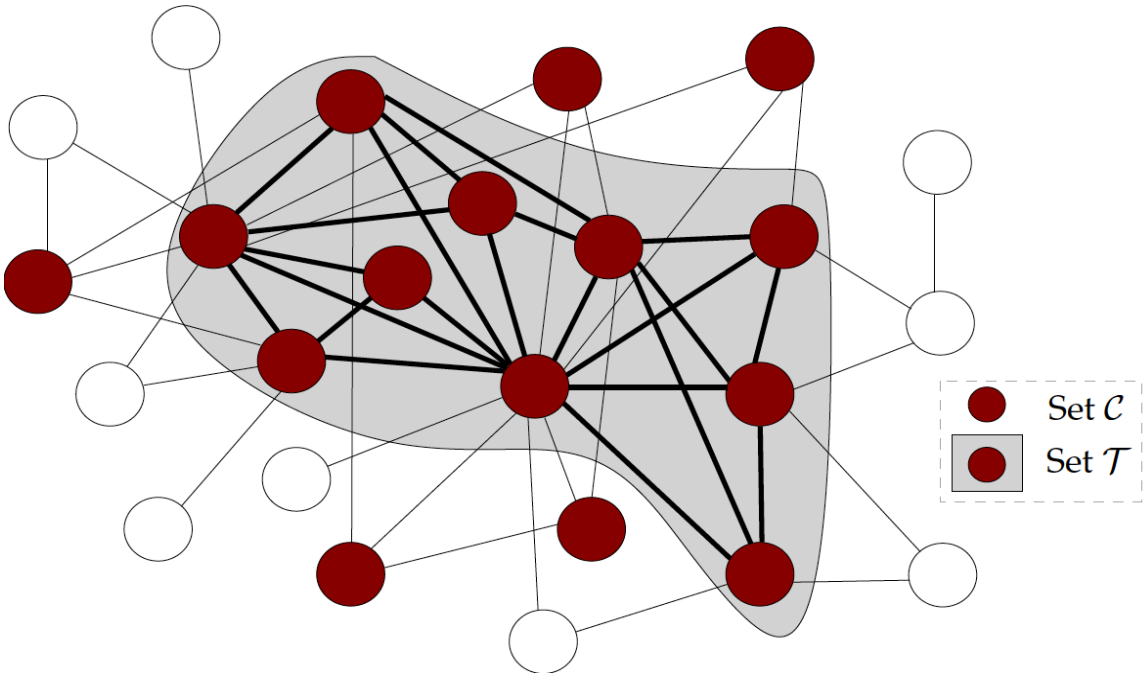
K-truss subgraph of G , the largest subgraph where all edges belong to K-2 triangles (T_K, K>2)



Complexity:
- k-core decomposition O (m) – m: # edges
- K-truss: $O(m^{1.5})$.

Identifying Influential Spreaders with K-truss

- Maximal k-core and K-truss subgraphs overlap.
- K-truss is a subgraph of k-core.



Identifying Influential Spreaders

Spreading Process Simulation:

S(t) : number of **S**usceptible nodes

I(t) : number of **I**nfected nodes

R(t) : number of **R**ecovered nodes

beta: infection rate

gamma : recovery rate

Anderson, R. M.,\& May, R. M. (1991). Infectious diseases of humans (Vol. 1). Oxford: Oxford university press.

Identifying Influential Spreaders Experiments

Network Name	Nodes	Edges	k_{max}	K_{max}	$\|\mathcal{C}\|$	$\|\mathcal{T}\|$
EMAIL-ENRON	33,696	180,811	43	22	275	45
EPINIONS	75,877	405,739	67	33	486	61
WIKI-VOTE	7,066	100,736	53	23	336	50
EMAIL-EUALL	224,832	340,795	37	20	292	62
SLASHDOT	82,168	582,533	55	36	134	96
WIKI-TALK	2,388,953	4,656,682	131	53	700	237

Jure, L. and Andrej, K. Stanford Network Analysis Project. http://snap.stanford.edu

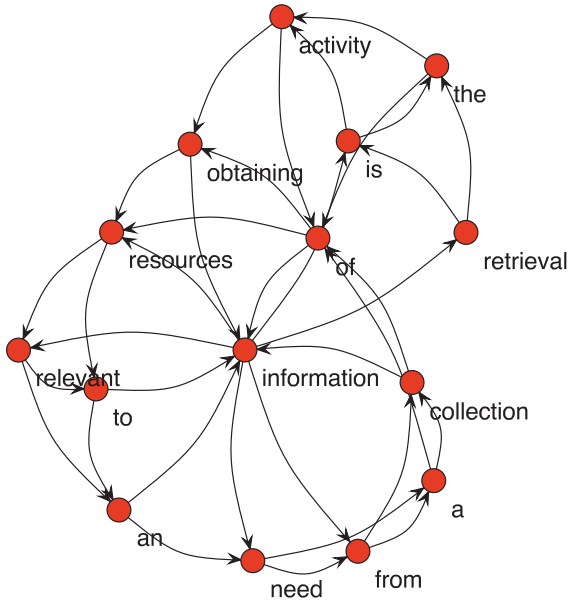
Identifying Influential Spreaders Experiments

		Time Step							
	Method	2	4	6	8	10	*Final step*	σ	*Max step*
EMAIL-ENRON	**truss**	8.44	46.66	204.08	418.77	355.84	2,596.52	136.7	33
	core	4.78	31.97	152.55	367.28	364.13	2,465.60	199.6	37
	top degree	6.89	34.13	155.48	360.89	357.08	2,471.67	354.8	36
EPINIONS	**truss**	4.17	19.70	75.04	204.14	329.08	2,567.69	227.8	37
	core	3.45	14.72	55.27	158.56	280.03	2,325.37	327.2	43
	top degree	4.22	16.03	58.84	166.23	289.49	2,414.99	331.7	47
WIKI-VOTE	**truss**	2.92	6.92	15.27	28.73	42.46	560.66	114.9	52
	core	1.92	4.78	10.65	20.66	32.40	466.01	104.5	57
	top degree	2.43	5.46	12.05	23.05	35.55	502.88	104.5	62
EMAIL-EUALL	**truss**	11.62	62.25	240.97	584.87	725.42	5,018.52	487.94	36
	core	9.85	40.82	158.72	433.81	644.76	4,579.84	498.71	38
	top degree	17.96	39.93	144.69	503.18	548.25	4,137.56	1,174.84	39
SLASHDOT	**truss**	5.36	66.21	461.35	1,390.52	1,359.99	8,207.46	368.37	32
	core	6.48	61.13	410.19	1,272.29	1,344.33	8,002.76	518.43	32
	top degree	13.95	83.29	483.95	1,426.81	1,403.80	8,489.45	59.01	32
WIKI-TALK	**truss**	64.21	3,259.05	34,543.23	9,853.84	1,186.41	93,491.81	476.22	21
	core	41.77	2,027.69	31,223.21	13,055.45	1,664.52	93,496.50	767.35	23
	top degree	88.84	2,475.01	29,694.45	13,720.15	1,937.89	93,411.18	1,166.77	24

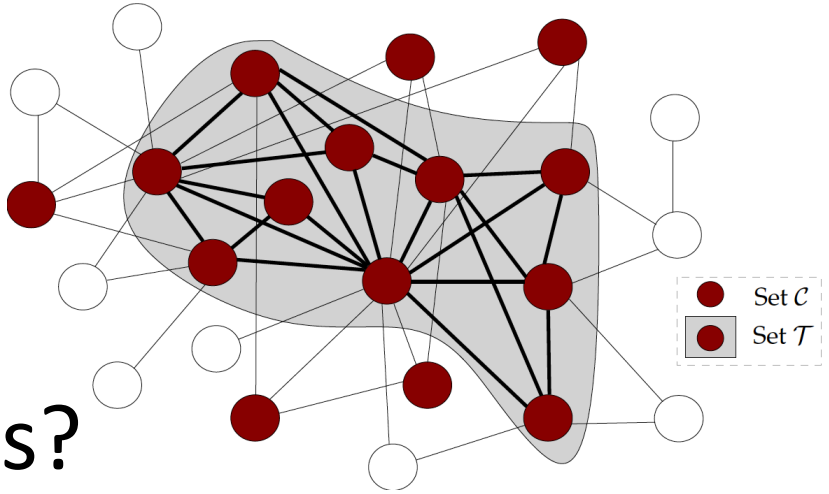
Evaluation of the spreading performance per step of the process.

F. D. Malliaros, M. G. Rossi, and M. Vazirgiannis. "Locating influential nodes in complex networks." Nature/Scientific reports 6 (2016).

THANK YOU!



Questions?



Recent Relevant publications

1. Christos Giatsidis, Dimitrios M. Thilikos, Michalis Vazirgiannis: D-cores: Measuring Collaboration of Directed Graphs Based on Degeneracy. IEEE - ICDM 2011: 201-210
2. D-cores: measuring collaboration of directed graphs based on degeneracy Christos Giatsidis, Dimitrios M. Thilikos, Michalis Vazirgiannis in Knowledge and Information Systems, Knowledge and Information Systems, May 2013, Volume 35, Issue 2, pp 311–343
3. "Locating influential nodes in complex networks", Fragkiskos D. Malliaros, Maria-Evgenia G. Rossi and Michalis Vazirgiannis, Scientific Reports/Nature journal.
4. "Vulnerability Assessment in Social Networks under Cascade-based Node Departures", F. Malliaros, Michalis Vazirgiannis) in EPL (Europhysics Letters) 110 (6), 68006
5. F. Malliaros, M. Vazirgiannis,"Clustering and Community Detection in Directed Networks: A Survey", Physics Reports Journal 533 (4), 95-142, Elsevier, 2013
6. Matching Node Embeddings for Graph Similarity, Giannis Nikolentzos, Polykarpos Meladianos and Michalis Vazirgiannis, **AAAI2017**
7. Konstantinos Skianis, François Rousseau, Michalis Vazirgiannis: Regularizing Text Categorization with Clusters of Words. EMNLP 2016: 1827-1837
8. Antoine J.-P. Tixier, Fragkiskos D. Malliaros, Michalis Vazirgiannis: A Graph Degeneracy-based Approach to Keyword Extraction. EMNLP 2016: 1860-1870
9. GoWvis: a web application for Graph-of-Words-based text visualization and summarization, AJP Tixier, K Skianis, M Vazirgiannis ACL 2016, 151 2015
10. Jonghoon Kim, François Rousseau and Michalis Vazirgiannis, Convolutional Sentence Kernel with Word Embedding for Short Text Categorization, proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP '15).
11. François Rousseau, Emmanouil Kiagias and Michalis Vazirgiannis, Text Categorization as a Graph Classification Problem, Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics and the 6th International Joint Conference on Natural Language Processing (ACL-IJCNLP '15).
12. Francois Rousseau and Michalis Vazirgiannis. K-core on Graph-of-words for Single-Document Keyword Extraction, European Conference on Information Retrieval, Vienna, Austria, 2015
13. M. Rossi, F. Malliaros, M. Vazirgiannis, Spread it Good, Spread it Fast: Identification of Influential Nodes in Social Networks. WWW (Companion Volume) 2015: 101-102
14. P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, M. Vazirgiannis, "Degeneracy-Based Real-Time Sub-Event Detection in Twitter Stream", in the proceedings of the AAAI-ICWSM 2015 conference