电子科技大学
University of Electronic Science and Technology of China

Intro to Link Prediction

Xiaolin Yang

Data Mining Lab, Big Data Research Center, UESTC
School of Computer Science and Engineering
Email: xiaolinyn@gmail.com

2017.11.22

Outline

01 Definition and Motivation

Definition and Motivation

数据挖掘实验室
Data Mining Lab

Definition

A fundamental problem that attempts to ***estimate the likelihood of the existence of a link between two nodes, based on observed links and the attributes of nodes****.*

Specifically, consider an undirected network $G(V, E)$, where V is the set of nodes and E is the set of links. Denote by U, the universal set containing all $|V|(|V|-1)/2$ possible links. Then, the set of nonexistent links is $U - E$. We assume that there are ***some missing links (or the links that will appear in the future)*** *in the set $U - E$, and the task of link prediction is to find out these links.*

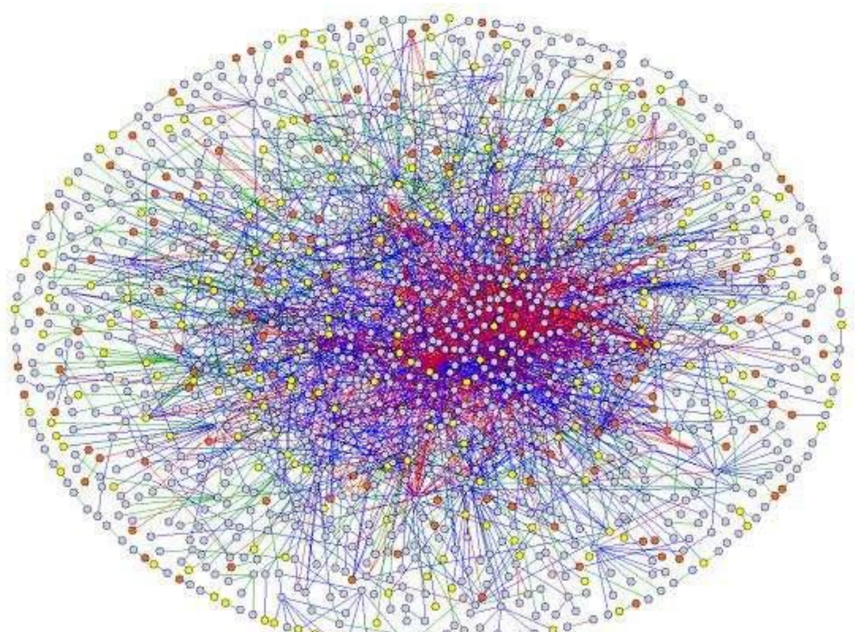
Definition and Motivation

数据挖掘实验室
Data Mining Lab

Motivation

Link prediction can be used to:

1. extract missing information;
2. identify spurious(假的，伪造的) interactions;
3. evaluate network evolving mechanisms;
4. and so on.



Definition and Motivation

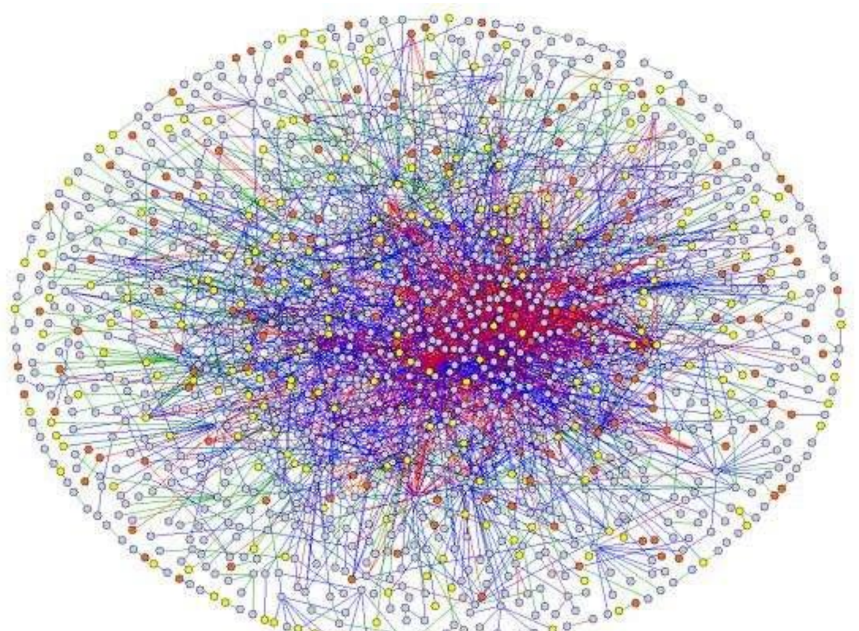
Motivation

Link prediction can be used to:

1. *extract missing information;*
2. *identify spurious(假的，伪造的) interactions;*
3. ***evaluate network evolving mechanisms;***
4. *and so on.*

To test the algorithm's accuracy, the observed links, E, is randomly divided into two parts: the training set, E^T, is treated as known information, while the probe set (i.e., validation subset), E^P, is used for testing and no information in this set is allowed to be used for prediction.

Use K-fold cross-validation.



Definition and Motivation

evaluate network evolving mechanisms:

Two standard metrics:

1. *AUC: the probability that a randomly chosen missing link (i.e., a link in E^P) is given a higher score than a randomly chosen nonexistent link (i.e., a link in U-E). If among **n** independent comparisons, there are n' times the missing link having a higher score and n'' times they have the same score, the AUC value is*

$$AUC = \frac{n' + 0.5n''}{n}$$

2. *Precision: if we take the top-L links as the predicted ones, among which L_r links are right (i.e., there are L_r links in the probe set E^P), then the precision equals L_r/L*

Definition and Motivation

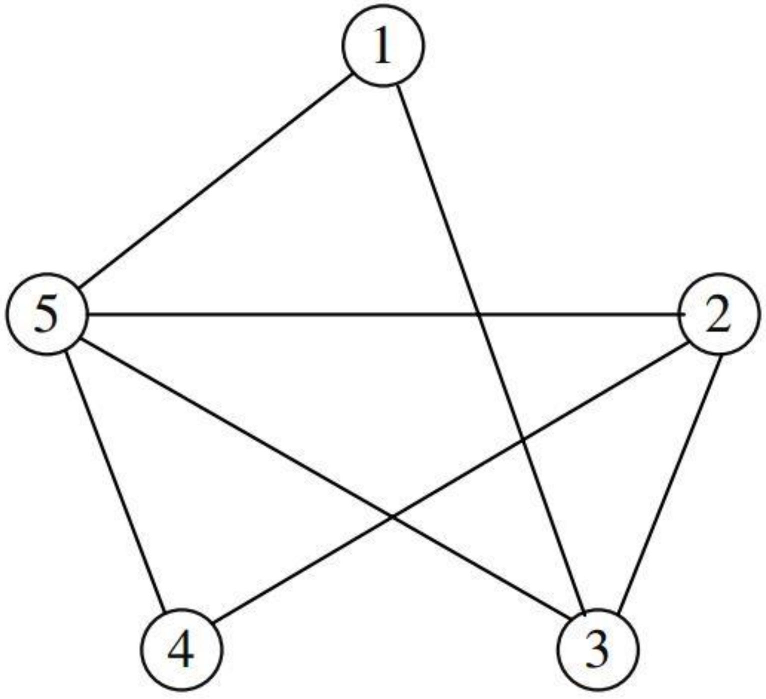
数据挖掘实验室
Data Mining Lab

evaluate network evolving mechanisms:

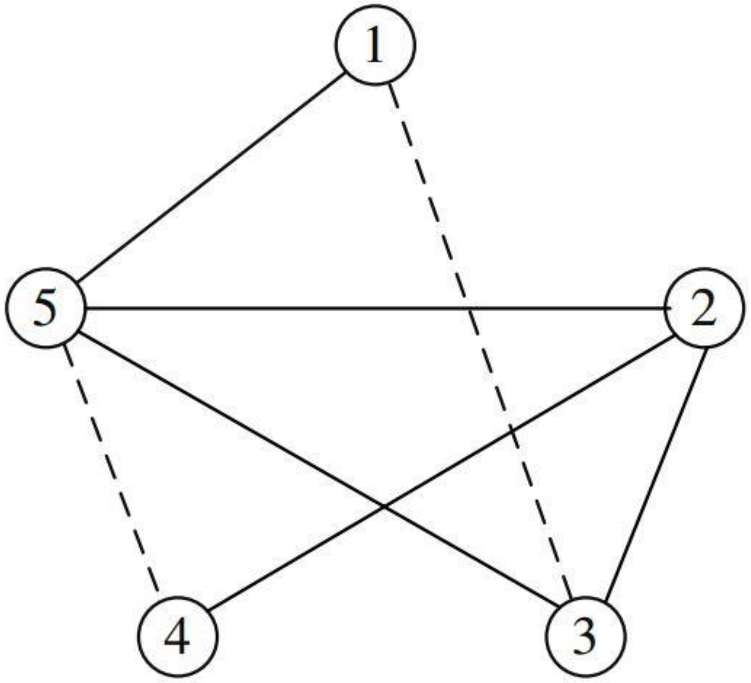
An example:

If an algorithm assigns scores of all non-observed links as

$s_{12} = 0.4, s_{13} = 0.5, s_{14} = 0.6, s_{34} = 0.5, s_{45} = 0.6$



Whole graph



Training graph

Definition and Motivation

数据挖掘实验室
Data Mining Lab

evaluate network evolving mechanisms:

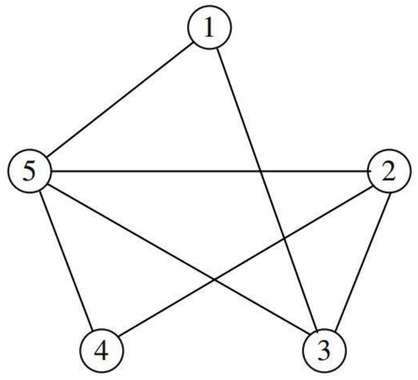
An example:

If an algorithm assigns scores of all non-observed links as

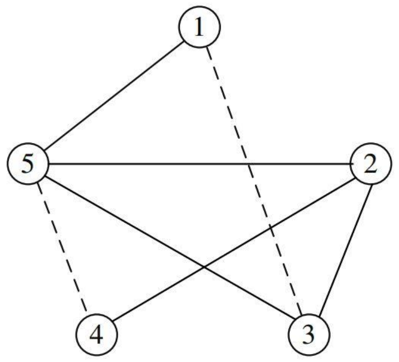
$s_{12} = 0.4, s_{13} = 0.5, s_{14} = 0.6, s_{34} = 0.5, s_{45} = 0.6$

1. *AUC: There are six pairs in total,* $s_{13} > s_{12}, s_{13} < s_{14}, s_{13} = s_{34}, s_{45} > s_{12}, s_{45} = s_{14}, s_{45} > s_{34}$. *Hence, the AUC value equals* $(3\times1+2\times0.5)/6\approx0.67$.

2. *Precision: if L=2, the predicted links are (1, 4) and (4, 5). Clearly, the former is wrong while the latter is right, and thus the precision equals 0.5 .*



Whole graph



Training graph

02 *Methods or Models*

Methods or Models

Four mainstream approaches

1. *Similarity-based algorithms*
   - *Local similarity indices*
   - *Global similarity indices*
   - *Quasi-local indices*
2. *Maximum likelihood methods*
   - *Hierarchical structure model*
   - *Stochastic block model*
3. *Matrix and tensor factorizations*
   - *Matrix factorizations*
   - *Tensor factorizations*
4. *Probabilistic models*

Methods or Models

1. ***Similarity-based algorithms***

each pair of nodes, x and y, is assigned a score s_{xy}, which is directly defined as the similarity/proximity between x and y. All non-observed links are ranked according to their scores, and ***the links connecting more similar nodes are supposed to be of higher existence likelihoods****.*

structural similarity

vs.

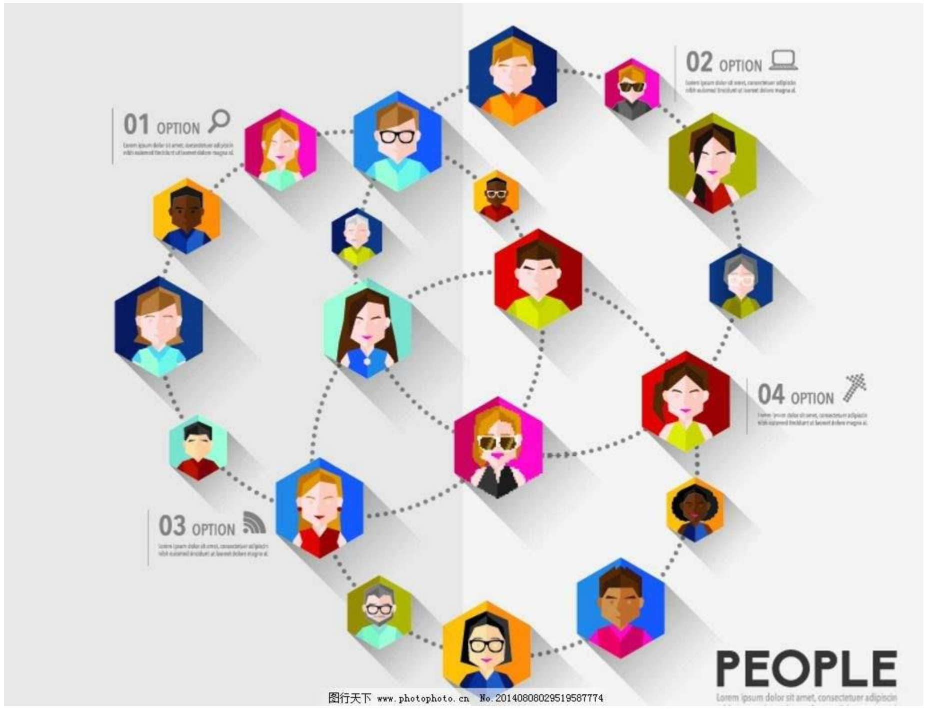
social similarity, textual similarity…

Methods or Models

数据挖掘实验室
Data Mining Lab

1. Similarity-based algorithms

The fundamental starting point for most measures of structural similarity is ***the assumption that the edges in a network themselves indicate a similarity between the vertices they connect*** *(assortativity同配性).*



Methods or Models

数据挖掘实验室

Data Mining Lab

1. Similarity-based algorithms

1.1 Local similarity indices

Common Neighbors (CN): $s_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)|$

Salton Index: $$s_{xy}^{Salton} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x \times k_y}}$$

Jaccard Index: $$s_{xy}^{Jaccard} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

$\Gamma(x)$ *: the set of neighbors of x*

k_x *: the degree of node x*

Methods or Models

数据挖掘实验室
Data Mining Lab

1. Similarity-based algorithms

1.1 Local similarity indices

Sørensen Index: $$s_{xy}^{Sørensen} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k_x + k_y}$$

Hub Promoted Index (HPI): $$s_{xy}^{HPI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x, k_y\}}$$

Hub Depressed Index (HDI): $$s_{xy}^{HDI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{k_x, k_y\}}$$

$\Gamma(x)$: *the set of neighbors of x*

k_x : *the degree of node x*

Methods or Models

1. Similarity-based algorithms

1.1 Local similarity indices

Leicht–Holme–Newman Index(LHN1):

$$s_{xy}^{LHN1} = \frac{|\Gamma(x) \cap \Gamma(y)|}{k_x \times k_y}$$

Preferential Attachment Index(PA):

$$s_{xy}^{PA} = k_x \times k_y$$

Adamic–Adar Index(AA):

$$s_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}$$

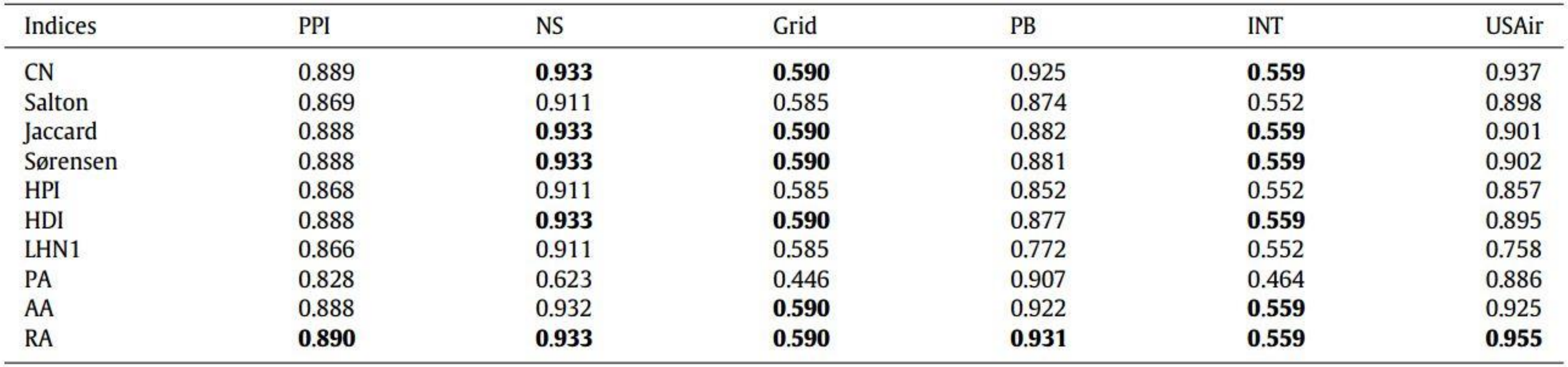
Resource Allocation Index (RA):

$$s_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}$$

1. Similarity-based algorithms

1.1 Local similarity indices

Accuracy (measured by the AUC value):



Indices	PPI	NS	Grid	PB	INT	USAir
CN	0.889	**0.933**	**0.590**	0.925	**0.559**	0.937
Salton	0.869	0.911	0.585	0.874	0.552	0.898
Jaccard	0.888	**0.933**	**0.590**	0.882	**0.559**	0.901
Sørensen	0.888	**0.933**	**0.590**	0.881	**0.559**	0.902
HPI	0.868	0.911	0.585	0.852	0.552	0.857
HDI	0.888	**0.933**	**0.590**	0.877	**0.559**	0.895
LHN1	0.866	0.911	0.585	0.772	0.552	0.758
PA	0.828	0.623	0.446	0.907	0.464	0.886
AA	0.888	0.932	**0.590**	0.922	**0.559**	0.925
RA	**0.890**	**0.933**	**0.590**	**0.931**	**0.559**	**0.955**

PPI: a protein–protein interaction network

NS: a coauthorship network of scientists who are themselves publishing on the topic of network

Grid: an electrical power grid of the western

PB: a network of the US political blogs

INT: a routerlevel Internet collected by Rocketfuel Project

USAir: a network of the US air transportation system

Methods or Models

数据挖掘实验室

Data Mining Lab

1. *Similarity-based algorithms*

1.2 Global similarity indices

Katz Index: $s_{xy}^{Katz} = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \cdots$

Leicht–Holme–Newman Index (LHN2): $S = 2M\lambda_1 D^{-1} \left(I - \frac{\phi A}{\lambda_1} \right)^{-1} D^{-1}$

$D_{xy} = \delta_{xy} k_x$ (δ_{xy}*is the Kronecker function:* δ_{xy} *= 1 if x = y, otherwise equals 0)*

P *: the transition matrix with* $P_{xy} = 1/k_x$

Methods or Models

数据挖掘实验室
Data Mining Lab

1. *Similarity-based algorithms*

1.2 *Global similarity indices*

$$\underline{S_{ij}} = \phi \sum_{v} A_{iv} \underline{S_{vj}} + \psi \delta_{ij} \Rightarrow S = \phi A S + \psi I \Rightarrow S = \psi (I - \phi A)^{-1}$$

set $\psi = 1$:

$$S = (I - \phi A)^{-1} = I + \phi A + \phi^2 A^2 + \cdots$$

But vertices with very high degree, for instance, will almost certainly have one or several paths of length two connecting them, even if connections between vertices are just made at random. So simple counts of number of paths are not enough to establish similarity. We ***need to know when a pair of vertices has more paths of a given length between than we would expect by chance****.*

Methods or Models

数据挖掘实验室
Data Mining Lab

1. Similarity-based algorithms

1.2 Global similarity indices

The expected value of $(A^l)_{xy}$, *namely* $E[(A^l)_{xy}]$, *equals* $(k_x k_y/2M)\lambda_1^{l-1}$, *where* λ_1 *is the largest eigenvalue of A and M is the total number of edges in the network (derived from the configuration model). Replace* $(A^l)_{xy}$ *in with* $(A^l)_{xy}$ */E* $(A^l)_{xy}]$:

$$s_{xy}^{LHN2} = \delta_{xy} + \frac{2M}{k_x k_y}\sum_{l=0}^{\infty}\phi^l\lambda^{1-l}\left(A^l\right)_{xy} = \left[1 - \frac{2M\lambda_1}{k_x k_y}\right]\delta_{xy} + \frac{2M\lambda_1}{k_x k_y}\left[\left(I - \frac{\phi}{\lambda_1}A\right)^{-1}\right]_{xy}$$

Since the first item is a diagonal matrix, it can be dropped and thus we arrive to a compact expression:

$$S = 2M\lambda_1 D^{-1}\left(I - \frac{\phi A}{\lambda_1}\right)^{-1} D^{-1}$$

Methods or Models

数据挖掘实验室

Data Mining Lab

1. *Similarity-based algorithms*

1.2 *Global similarity indices*

Katz Index: $s_{xy}^{Katz} = \beta A_{xy} + \beta^2 (A^2)_{xy} + \beta^3 (A^3)_{xy} + \cdots$

Leicht–Holme–Newman Index (LHN2): $S = 2M\lambda_1 D^{-1} \left(I - \frac{\phi A}{\lambda_1} \right)^{-1} D^{-1}$

Random Walk with Restart (RWR): $\overrightarrow{q_x} = cP^T \overrightarrow{q_x} + (1 - c)\overrightarrow{e_x}$

$D_{xy} = \delta_{xy} k_x$ (δ_{xy}*is the Kronecker function:* $\delta_{xy} = 1$ *if* $x = y$*, otherwise equals 0)*

P *: the transition matrix with* $P_{xy} = 1/k_x$

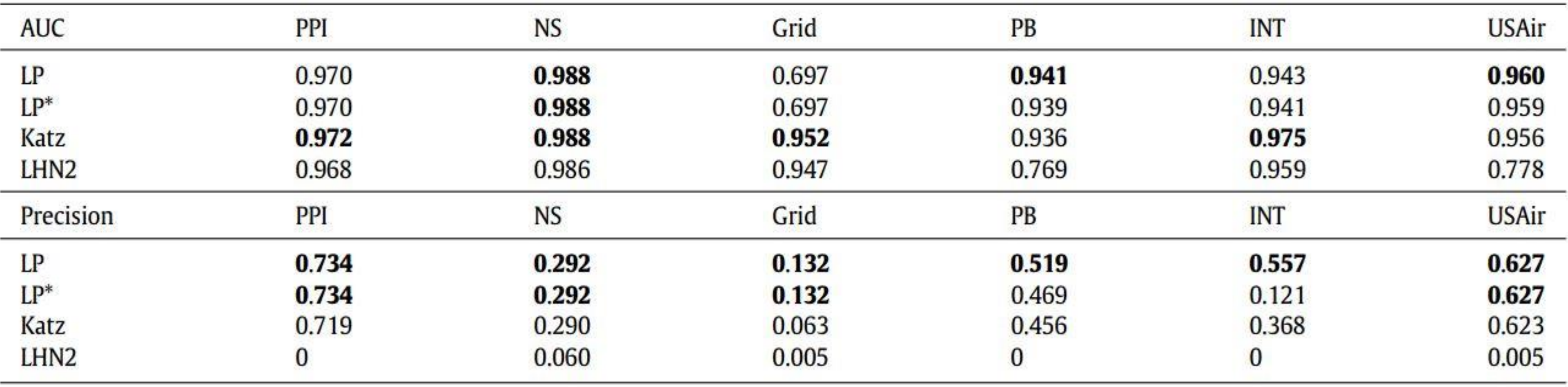
Methods or Models

数据挖掘实验室
Data Mining Lab

1. Similarity-based algorithms

1.2 Global similarity indices

Accuracy (LP is a quasi-local indice):



AUC	PPI	NS	Grid	PB	INT	USAir
LP	0.970	**0.988**	0.697	**0.941**	0.943	**0.960**
LP*	0.970	**0.988**	0.697	0.939	0.941	0.959
Katz	**0.972**	**0.988**	**0.952**	0.936	**0.975**	0.956
LHN2	0.968	0.986	0.947	0.769	0.959	0.778

Precision	PPI	NS	Grid	PB	INT	USAir
LP	**0.734**	**0.292**	**0.132**	**0.519**	**0.557**	**0.627**
LP*	**0.734**	**0.292**	**0.132**	0.469	0.121	**0.627**
Katz	0.719	0.290	0.063	0.456	0.368	0.623
LHN2	0	0.060	0.005	0	0	0.005

Methods or Models

数据挖掘实验室

Data Mining Lab

1. Similarity-based algorithms

1.3 Quasi-local indices

Local Path Index (LP): $S^{LP} = A^2 + \varepsilon A^3$

Local Random Walk (LRW): $s_{xy}^{LRW}(t) = q_x \pi_{xy}(t) + q_y \pi_{yx}(t)$

Superposed Random Walk (SRW): $s_{xy}^{SRW}(t) = \sum_{\tau=1}^{t} s_{xy}^{LRW}(\tau)$

a random walker is initially put on node x and thus the initial density vector $\vec{\pi}_x(0) = \vec{e}_x$,
this density vector evolves as $\vec{\pi}_x(t+1) = P^T \vec{\pi}_x(t)$
q *: the initial configuration function, such as* $q_x = \frac{k_x}{M}$ *(*M *is the total number of edges)*

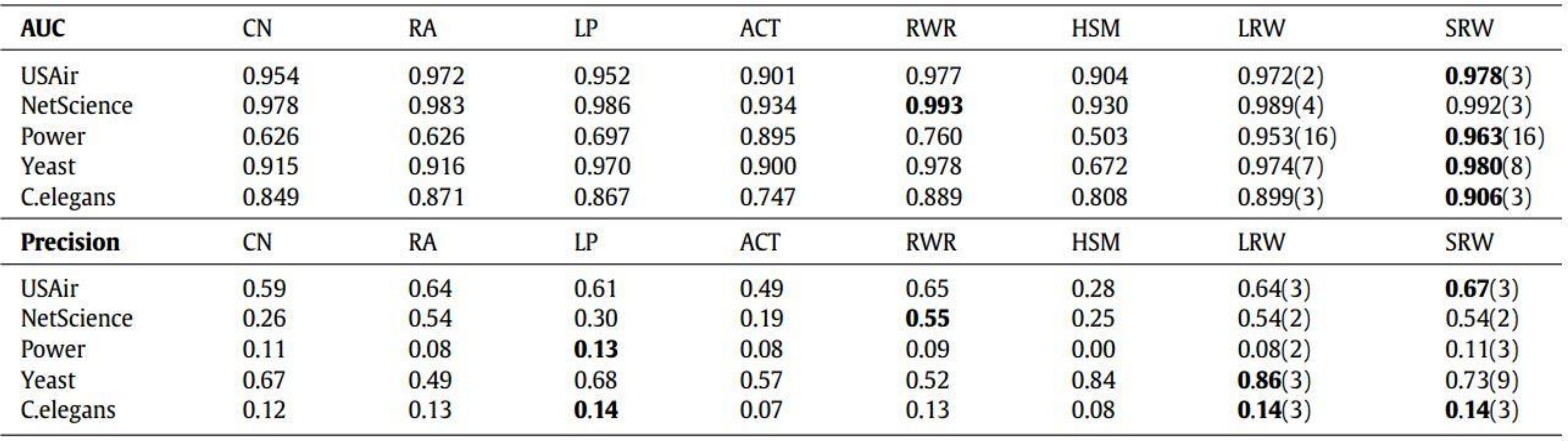
Methods or Models

数据挖掘实验室
Data Mining Lab

1. Similarity-based algorithms

1.3 Quasi-local indices

Accuracy (AUC & Precision):



AUC	CN	RA	LP	ACT	RWR	HSM	LRW	SRW
USAir	0.954	0.972	0.952	0.901	0.977	0.904	0.972(2)	**0.978**(3)
NetScience	0.978	0.983	0.986	0.934	**0.993**	0.930	0.989(4)	0.992(3)
Power	0.626	0.626	0.697	0.895	0.760	0.503	0.953(16)	**0.963**(16)
Yeast	0.915	0.916	0.970	0.900	0.978	0.672	0.974(7)	**0.980**(8)
C.elegans	0.849	0.871	0.867	0.747	0.889	0.808	0.899(3)	**0.906**(3)
Precision	CN	RA	LP	ACT	RWR	HSM	LRW	SRW
USAir	0.59	0.64	0.61	0.49	0.65	0.28	0.64(3)	**0.67**(3)
NetScience	0.26	0.54	0.30	0.19	**0.55**	0.25	0.54(2)	0.54(2)
Power	0.11	0.08	**0.13**	0.08	0.09	0.00	0.08(2)	0.11(3)
Yeast	0.67	0.49	0.68	0.57	0.52	0.84	**0.86**(3)	0.73(9)
C.elegans	0.12	0.13	**0.14**	0.07	0.13	0.08	**0.14**(3)	**0.14**(3)

Methods or Models

数据挖掘实验室
Data Mining Lab

2. *Maximum likelihood methods*

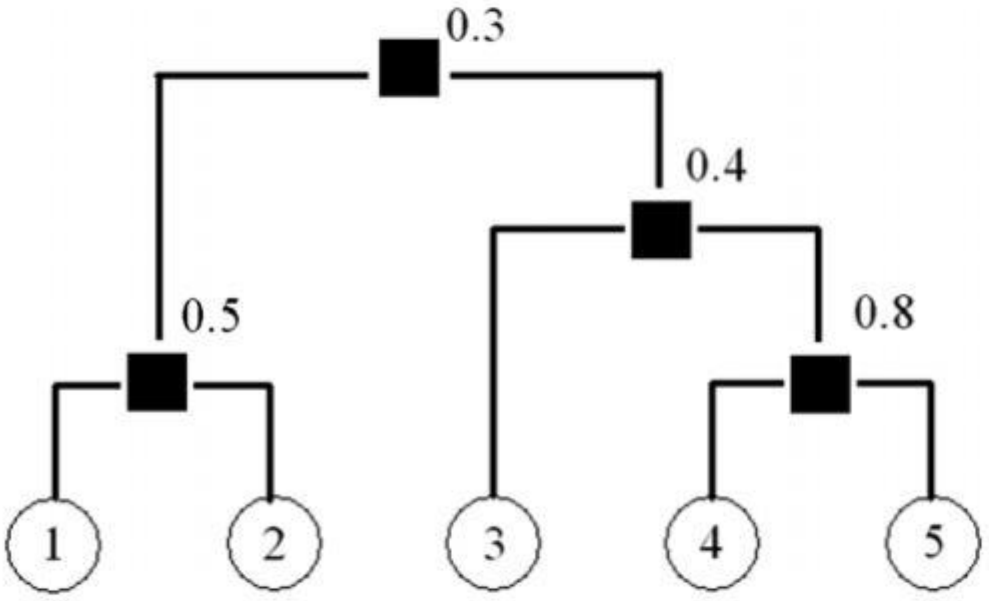
The algorithms presuppose some organizing principles of the network structure, with ***the detailed rules and specific parameters obtained by maximizing the likelihood of the observed structure****. Then, the likelihood of any non-observed link can be calculated according to those rules and parameters.*

Methods or Models

数据挖掘实验室
Data Mining Lab

2. *Maximum likelihood methods*

2.1 Hierarchical structure model



each internal node r is associated with a probability p_r and the connecting probability of a pair of nodes (leaves) is equal to $p_{r'}$, where r' is the lowest common ancestor of these two nodes.

Fig.1 Illustration of a dendrogram of a network with 5 nodes. Accordingly, the connecting probability of nodes 1 and 2 is 0.5, of nodes 1 and 3 is 0.3, of nodes 3 and 4 is 0.4.

Methods or Models

数据挖掘实验室

Data Mining Lab

2. *Maximum likelihood methods*

2.1 Hierarchical structure model

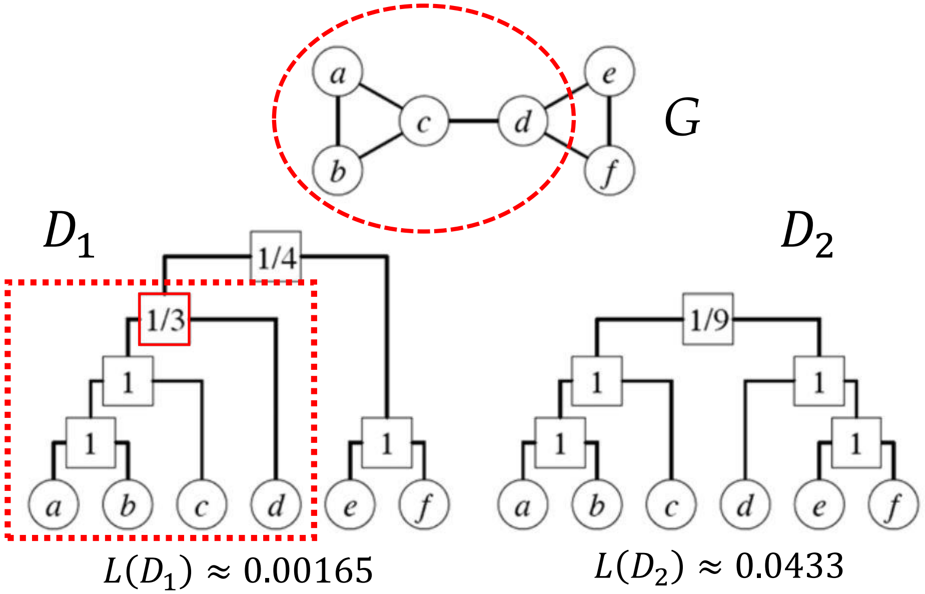
Given a real network G and a dendrogram D, let ***E_r be the number of edges in G whose endpoints have r as their lowest common ancestor in D****, and let L_r and R_r, respectively, be the number of leaves in the left and right subtrees rooted at r. Then the likelihood of the dendrogram D together with a set of p_r is*

$$L(D,\{p_r\}) = \prod_r p_r^{E_r}(1-p_r)^{L_rR_r-E_r}$$

$$p_r^* = \frac{E_r}{L_rR_r}$$

$$\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^{3*1-1}$$

maximizes L(D, {p_r}) for a fixed D



数据挖掘实验室
Data Mining Lab

2. *Maximum likelihood methods*

2.1 Hierarchical structure model

The algorithm to predict the missing links contains the following procedures:

a) *sample a large number of dendrograms with probability proportional to their likelihood;*

b) *for each pair of unconnected nodes i and j, calculate the mean connecting probability $\langle p_{ij} \rangle$ by averaging the corresponding probability p_{ij} over all sampled dendrograms;*

c) *sort these node pairs in descending order of $\langle p_{ij} \rangle$ and the highest-ranked ones are those to be predicted*

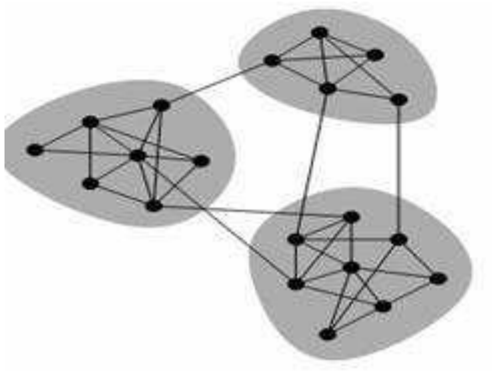
Methods or Models

数据挖掘实验室
Data Mining Lab

2. *Maximum likelihood methods*

2.2 Stochastic block model

- *nodes are partitioned into groups and the probability that* ***two nodes are connected depends solely on the groups to which they belong;***
- *capture the community structure, role-to-role connections and maybe other factors for the establishing of connections, especially when the* ***group membership plays a considerable role in determining how nodes interact with each other****.*



2. *Maximum likelihood methods*

2.2 Stochastic block model

*Given **a partition M** where each node belongs to one group and the connecting probability for two nodes respectively in groups α and β is denoted by $Q_{\alpha\beta}$ ($Q_{\alpha\alpha}$ represents the probability that two nodes within group α are connected), then the likelihood of the observed network structure is:*

$$L(A|M) = \prod_{\alpha\leq\beta} Q_{\alpha\beta}^{l_{\alpha\beta}} (1 - Q_{\alpha\beta})^{r_{\alpha\beta} - l_{\alpha\beta}}$$

$l_{\alpha\beta}$ *: the number of edges between nodes in groups α and β*

$r_{\alpha\beta}$ *: the number of pairs of nodes such that one node is in α and the other is in β*

2. *Maximum likelihood methods*

2.2 Stochastic block model

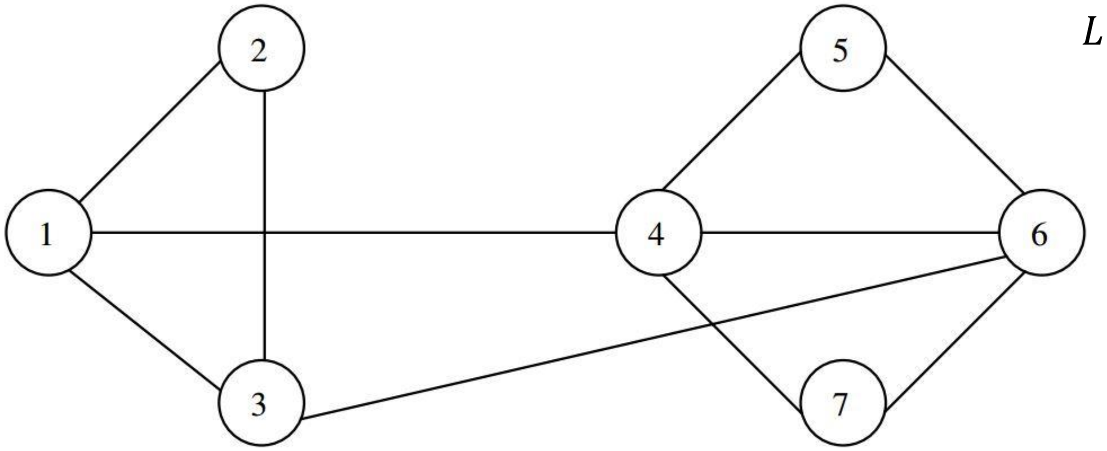
*Given **a partition M** where each node belongs to one group and the connecting probability for two nodes respectively in groups α and β is denoted by $Q_{\alpha\beta}$ ($Q_{\alpha\alpha}$ represents the probability that two nodes within group α are connected), then the likelihood of the observed network structure is:*

$$L(A|M) = \prod_{\alpha\leq\beta} Q_{\alpha\beta}^{l_{\alpha\beta}} (1 - Q_{\alpha\beta})^{r_{\alpha\beta} - l_{\alpha\beta}}$$

$Q^*_{\alpha\beta} = \frac{l_{\alpha\beta}}{r_{\alpha\beta}}$ *maximizes the likelihood L(A|M)*

2. *Maximum likelihood methods*

2.2 *Stochastic block model*



$$L(A|M) = \prod_{\alpha \leq \beta} Q_{\alpha\beta}^{l_{\alpha\beta}} (1 - Q_{\alpha\beta})^{r_{\alpha\beta} - l_{\alpha\beta}}$$

$M = \{ \{1, 2, 3\}, \{4, 5, 6, 7\}\}$, $Q_{11}^{*} = \frac{3}{3} = 1, Q_{12}^{*} = \frac{2}{12} = \frac{1}{6}, Q_{22}^{*} = \frac{5}{6}$

thus the likelihood is:

$$L = \boxed{1} \times \boxed{\left(\frac{1}{6}\right)^{2} \left(\frac{5}{6}\right)^{10}} \times \boxed{\left(\frac{5}{6}\right)^{5} \left(\frac{1}{6}\right)} \approx 3.005 \times 10^{-4}$$

2. Maximum likelihood methods

2.2 Stochastic block model

Denote by Ω the set of all possible partitions, the reliability of an individual link is:

$$R_{xy} = L(A_{xy} = 1|A) = \frac{\int_{\Omega} L(A_{xy} = 1|M)L(A|M)p(M)dM}{\int_{\Omega} L(A|M')p(M')dM'}$$

$p(M)$ *: a constant assuming no prior knowledge about the model*

Note: the number of different partitions of N elements grows faster than any finite power of N, and thus even for a small network, to sum over all partitions is not possible in practice. So usually use the Metropolis–Hastings algorithm.

2. *Maximum likelihood methods*

2.2 Stochastic block model

$$R_{xy} = L(A_{xy} = 1|A) = \frac{\int_{\Omega} L(A_{xy} = 1|M)L(A|M)p(M)dM}{\int_{\Omega} L(A|M')p(M')dM'}$$

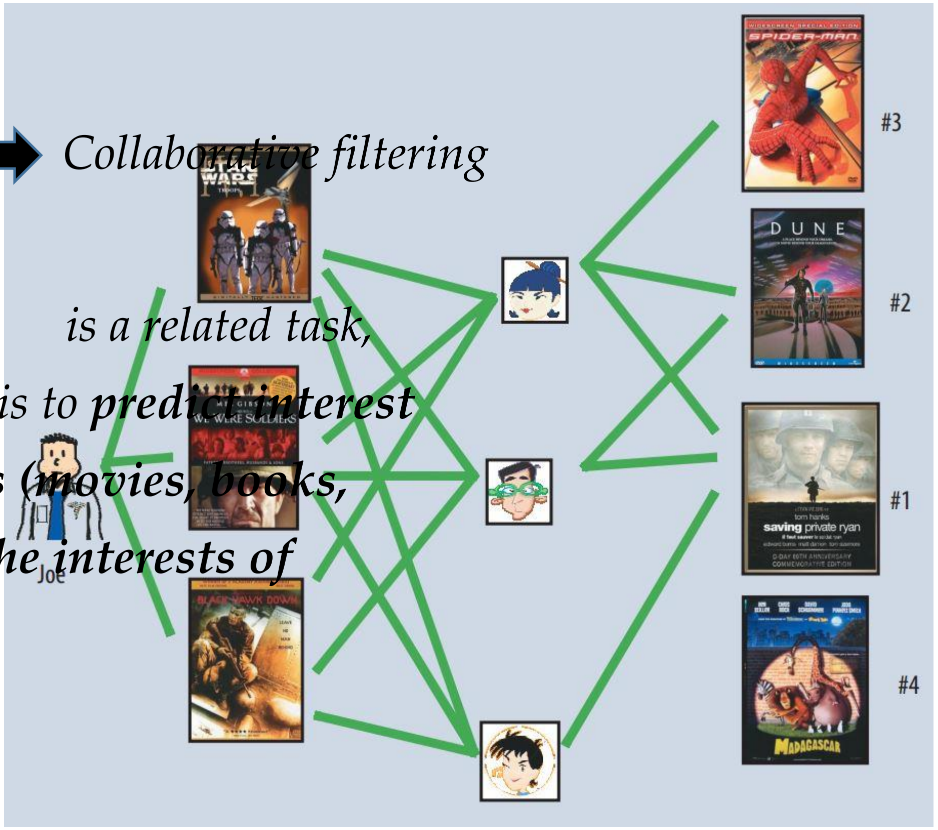
Reliability describes the likelihood of the existence of a link (i.e., the probability that the link "truly" exists) given the observed structure, which can be used to:

- *predict missing links (the nonexistent links in the observed network yet with the highest reliabilities)*
- *identify possible spurious links (the existent links with the lowest reliabilities)*

3. Matrix and tensor factorizations

*Now we focus on **bipartite graphs**, which are the targets of Recommender Systems.*

Link prediction ⟷ *Collaborative filtering is a related task, where the objective is to **predict interest of users to objects (movies, books, music) based on the interests of similar users**.*



3. *Matrix and tensor factorizations*

3.1 Matrix factorizations

- *Suppose that our data set consists of matrices* Z_1 *through* Z_T *of size M × N and the goal is to predict* Z_{T+1}, *where Z is:*

$$Z(i,j,t) = \begin{cases} 1 & \text{if object } i \text{ links to object } j \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

- *Collapse the data into a single M × N matrix X:*

$$X(i,j) = \sum_{t=1}^{T} (1-\theta)^{T-t} Z_t(i,j)$$

the link structure is damped backward in time

3. *Matrix and tensor factorizations*

3.1 Matrix factorizations

- *Truncated SVD:*

suppose that the compact SVD of X is given by $X = U\Sigma V^T$

suppose that the compact SVD of X is given by $X \approx \underline{U_K \Sigma_K V_K^T}$

U_K, V_K*: comprise the first K columns of U and V and* Σ_K *is the K × K principal submatrix of Σ (ordered by the corresponding singular value)*

A matrix of scores for predicting future links can then be calculated as $S \approx U_K \Sigma_K V_K^T$

3. Matrix and tensor factorizations

3.1 Matrix factorizations

- *Truncated SVD:*

suppose that the compact SVD of X is given by $X = U\Sigma V^T$

suppose that the compact SVD of X is given by $X \approx U_K\Sigma_K V_K^T$

This is one reason for using of SVD: ***low-rank approximation for scalability***

There is another one interpretation: ***realizations of latent factor models***

Methods or Models

3. Matrix and tensor factorizations

3.1 Matrix factorizations

Latent factor models

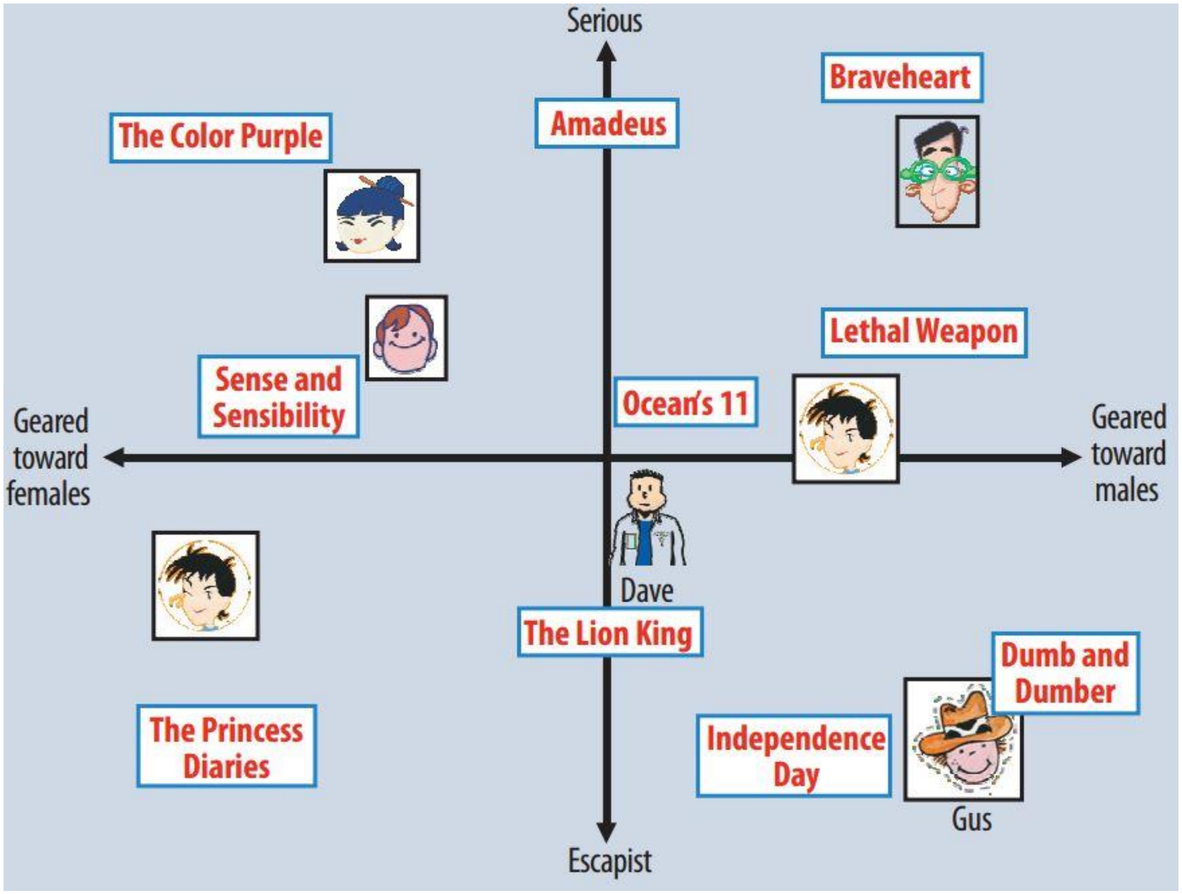


Fig.2 A simplifed illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Methods or Models

3. *Matrix and tensor factorizations*

3.1 Matrix factorizations

Latent factor models

$$\hat{R} = \boxed{U\Sigma}V^T$$

$$\Downarrow$$

$$\hat{R} = PQ \quad \text{or} \quad \hat{r}_{ui} = p_u^T q_i$$

But this often raises difficulties due to the ***high portion of missing values*** *caused by sparseness in the user-item ratings matrix.*

Carelessly addressing only the relatively few known entries is highly prone to ***overfitting****.*

3. *Matrix and tensor factorizations*

3.1 Matrix factorizations

Latent factor models

- *Solution1: earlier systems relied on imputation to fill in missing ratings and make the rating matrix dense;*
- *Solution2: modeling directly the observed ratings only, while avoiding overfitting through a regularized model:*

$$\min_{p*,q*} \sum_{(u,i)\in\kappa} (r_{ui} - p_u^T q_i)^2 + \lambda(||p_u||^2 + ||q_i||^2)$$

3. ***Matrix and tensor factorizations***

3.1 Matrix factorizations

Latent factor models

More inputs, or other considerations:

- ***implicit feedback****, including purchase history, browsing history, search patterns, or even mouse movements;*
- ***systematic tendencies*** *for some users to give higher ratings than others, and for some items to receive higher ratings than others;*
- ***ratings to the similar items****, which is called neighborhood models conventionally.*

Methods or Models

数据挖掘实验室
Data Mining Lab

3. *Matrix and tensor factorizations*

3.1 Matrix factorizations

An integrated model

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T\left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j\right) + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j\in R^k(i;u)} \left(r_{uj} - b_{uj}\right) w_{uj} + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j\in N^k(i;u)} c_{ij}$$

μ: *the overall average rating;*

b_u, b_i: *the observed deviations of user u and item i, respectively, from the average;*

$R(u)$: *all the items for which ratings by u are available*

$N(u)$: *all items for which u provided an implicit preference*

$S^k(i)$: *the set of k items most similar i*

$R^k(i;u)$: $R(u) \cap S^k(i)$

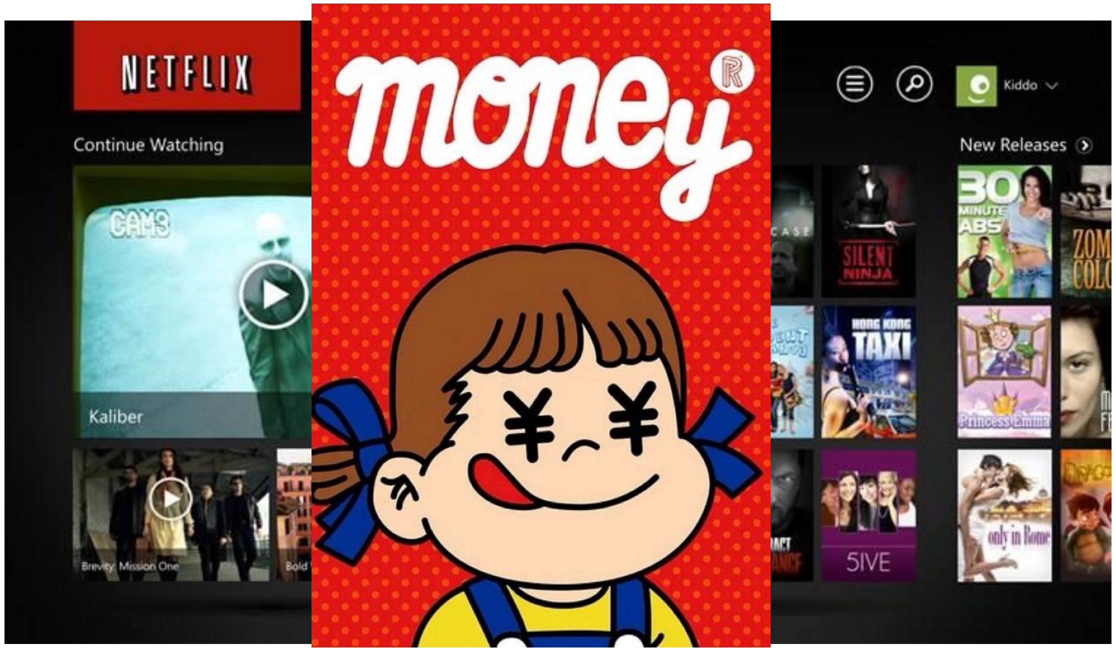
y_j, w_{uj}, c_{ij}: *offsets which can be learnt from the data through optimization*

Methods or Models

3. *Matrix and tensor factorizations*

NetfLix Prize Competition

In 2006, the online DVD rental company Netflix announced a contest to improve the state of its recommender system. To enable this, the company released a training set of more than 100 million ratings spanning about 500,000 anonymous customers and their ratings on more than 17,000 movies, each movie being rated on a scale of 1 to 5 stars. The first participating team that can improve on the Netflix algorithm's RMSE(Root-Mean-Square) performance by 10 percent or more wins a $1 million prize.



3. Matrix and tensor factorizations

3.2 Tensor factorizations

Instead of collapsing the data, tensor factorization can explicitly model the time dimension.

CP(CANDECOMP/PARAFAC) Decomposition:

Given a three-way tensor Z of size M × N × T, its K-component CP decomposition is given by

$$Z = \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k$$

$\circ$ denotes the outer product

3. Matrix and tensor factorizations

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

$$Z = \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k$$

An N-way tensor $X \in R^{I_1} \times R^{I_2} \times \cdots \times R^{I_N}$ is rank one if it can be written as the outer product of N vectors

$$X = a^{(1)} \circ a^{(2)} \circ \cdots \circ a^{(N)}$$

This means that each element of the tensor is the product of the corresponding vector elements:

$$x_{i_1 i_2 \cdots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}$$

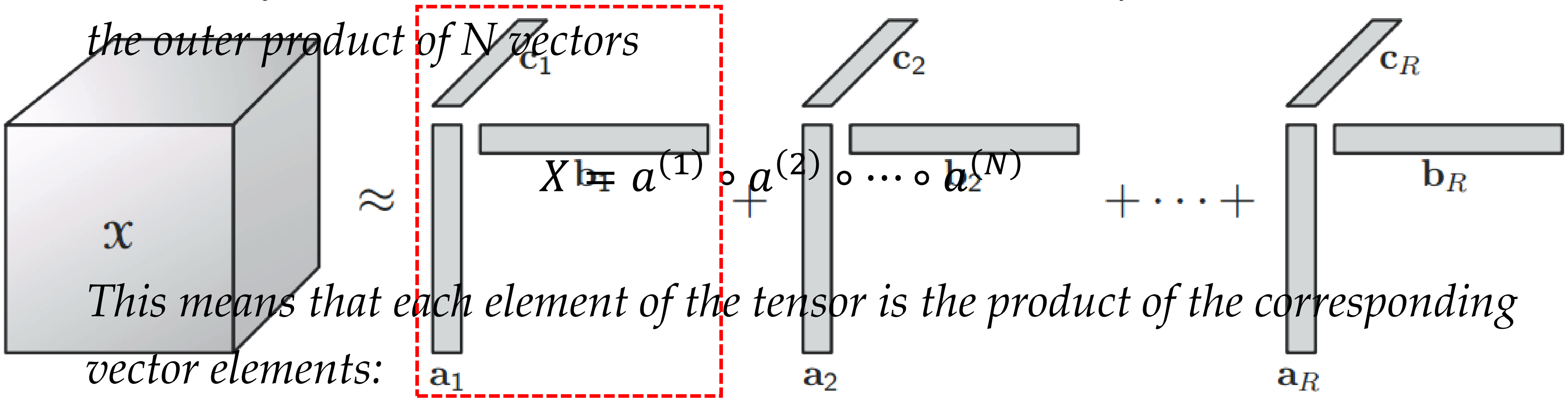


Fig.3 CP decomposition of a three-way array: factorizes a tensor into a sum of component ***rank-one tensors.***

3. Matrix and tensor factorizations

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

*The CP tensor decomposition can be considered an analogue of the SVD because **it decomposes a tensor as a sum of rank-one tensors just as the SVD decomposes a matrix as a sum of rank-one matrices**.*

Computing the CP Decomposition:

*Assuming the number of components is fixed, there is the "workhorse" algorithm for CP: **the alternating least squares (ALS)** method.*

3. Matrix and tensor factorizations

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

ALS method:

The goal is to compute a CP decomposition with K components that best approximates Z:

$$\min_{\hat{Z}} ||Z - \hat{Z}|| \quad \textit{with} \quad \hat{Z} = \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k = [\lambda; A, B, C\,]$$

The ALS approach fixes B and C to solve for A, then fixes A and C to solve for B, then fixes A and B to solve for C, and continues to repeat the entire procedure until some convergence criterion is satisfied.

3. *Matrix and tensor factorizations*

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

ALS method:

```
procedure CP-ALS($\mathcal{X}$,$R$)
    initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \dots, N$
    repeat
        for $n = 1, \dots, N$ do
            $\mathbf{V} \leftarrow \mathbf{A}^{(1)\mathsf{T}}\mathbf{A}^{(1)} * \cdots * \mathbf{A}^{(n-1)\mathsf{T}}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\mathsf{T}}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\mathsf{T}}\mathbf{A}^{(N)}$
            $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})\mathbf{V}^{\dagger}$
            normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
        end for
    until fit ceases to improve or maximum iterations exhausted
    return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$
end procedure
```

3. *Matrix and tensor factorizations*

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

Define the similarity score for objects i and j using a K-component CP model as the (i, j) entry of the following matrix:

$$S = \sum_{k=1}^{K} \gamma_k \lambda_k a_k b_k^T, \qquad \text{where} \quad \gamma_k = \sum_{t=T-2}^{T} c_k(t)$$

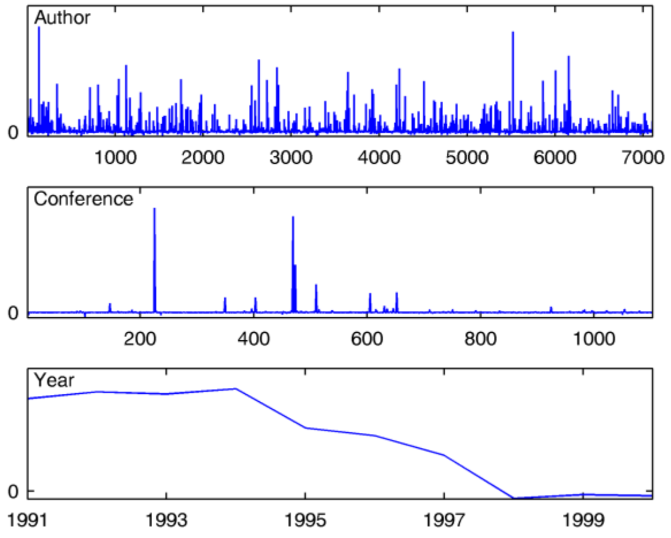
Obviously, this treatment loses much information for prediction!

3. Matrix and tensor factorizations

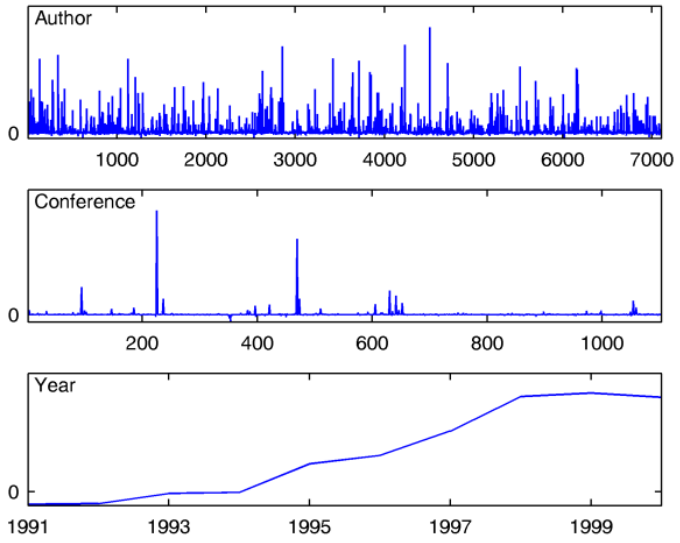
3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

Fig.4 Examples from 50-component CP model of publications from 1991 to 2000 (DBLP data, it contains publications from 1936 through the end of 2007).



(a) Factors from component 3: Top authors are Alberto L. Sangiovanni Vincentelli, Robert K. Brayton, Sudhakar M. Reddy, and Irith Pomeranz. Top conferences are DAC, ICCAD, and ICCD.



(b) Factors from component 4: Top authors are Miodrag Potkonjak, Massoud Pedram, Jason Cong, and Andrew B. Kahng. Top conferences are DAC, ICCAD, and ASPDAC.

Methods or Models

数据挖掘实验室
Data Mining Lab

3. Matrix and tensor factorizations

3.2 Tensor factorizations

CP(CANDECOMP/PARAFAC) Decomposition:

Link Prediction Results:

- Predicting All Links: pos links in the test set.
- Predicting New Links: th links that have not been previously seen at any ti the training set.

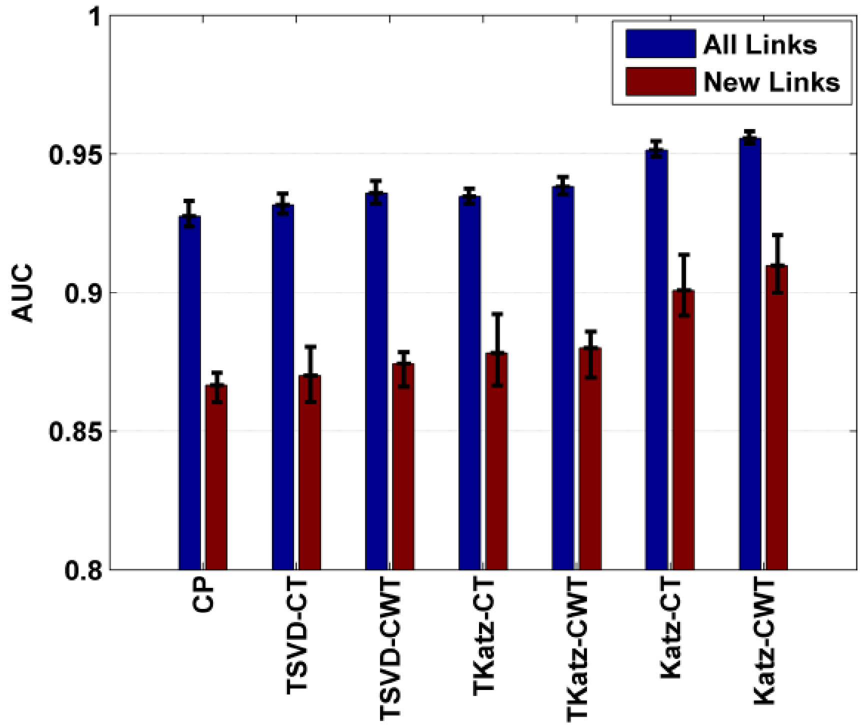


Fig.5 Average link prediction performance of each method across all seven training/test set pairs (black bars show absolute range).

Methods or Models

数据挖掘实验室
Data Mining Lab

4. ***Probabilistic models***

Probabilistic models aim at abstracting the underlying structure from the observed network, and then predicting the missing links by using the learned model. Given a target network G = (V, E), the probabilistic model will optimize a built target function to establish a model composed of a group of parameters Θ, which can best fit the observed data of the target network. Then the probability of the existence of a nonexistent link (i, j) is estimated by the conditional probability $P(A_{ij}= 1 | \Theta)$.

03 *Applications*

Applications

1. *Reconstruction of networks*

the reconstruction of networks from the observed networks with missing and spurious links.

$$R(A) = \prod_{A_{xy}=1, x<y} R_{xy} = \prod_{A_{xy}=1, x<y} L(A_{xy} = 1 | A^0)$$

A straightforward idea is to find out the network A that maximizes the reliability $R(A)$.

Applications

数据挖掘实验室
Data Mining Lab

1. ***Reconstruction of networks***

2. ***Evaluation of network evolving mechanisms***

   an algorithm for link prediction makes a guess about the factors resulting in the existence of links, which is actually what an evolving model wants to show. In other words, an evolving model in principle can be mapped to a link prediction algorithm.

Applications

数据挖掘实验室
Data Mining Lab

1. ***Reconstruction of networks***

2. ***Evaluation of network evolving mechanisms***

3. ***Classification of partially labeled networks***

   *Given a network with partial nodes being labeled, the problem is to predict the labels of these unlabeled nodes based on the known labels and the network structure. An underlying assumption is that **two nodes are more likely to be categorized into the same class if they are more similar to each other**.*

Applications

1. ***Reconstruction of networks***

2. ***Evaluation of network evolving mechanisms***

3. ***Classification of partially labeled networks***

$$p(l_i|x) = \frac{\sum_{\{y|y\neq x, lable(y)=l_i\}} s_{xy}}{\sum_{\{y|y\neq x, lable(y)\neq 0\}} s_{xy}}$$

The nodes without labels are labeled by 0

04 Conclusion

Conclusion

Definition and Motivation[AUC & Precision]

↓

Simple yet effective methods: Similarity-based algorithms

↓

Maximum likelihood methods

&

Matrix and tensor factorizations

↓

Three applications

Conclusion

数据挖掘实验室
Data Mining Lab

Reference

- ✓ *L. Lü, T. Zhou. Link prediction in complex networks: A survey. Physica A, 390 (2011), pp. 1150-1170.*
- ✓ *Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009) 32.*
- ✓ *E. Acar, D.M. Dunlavy, T.G. Kolda. Link prediction on evolving data using matrix and tensor factorizations Proc. of the 2009 IEEE International Conference on Data Mining Workshops, ICDMW '09, IEEE Computer Society, Washington, DC, USA (2009), pp. 262-269*
- ✓ *T. Kolda, B. Bader, Tensor decompositions and applications, SIAM Rev. 51, 455–500 (2009).*
- ✓ …

Thanks!