Recommendation Systems

part 2

School for advanced sciences of Luchon 2015

Debora Donato
debora@stumbleupon.com

Today presentation

- Similarity-based methods
  - User-similarity
  - Item-similarity
- Similarity score
  - Rating-based similarity
  - Structural similarity
- Serendipitous Rec
  - LDA

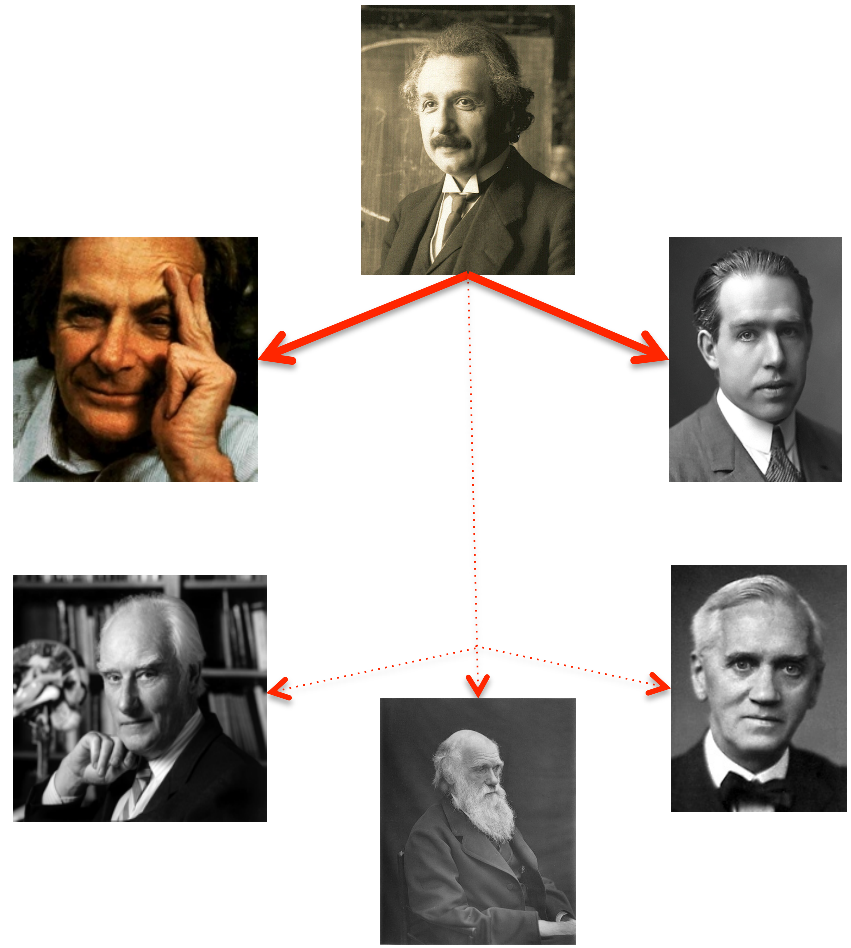
Similarity-based methods

- Also known as Memory-based collaborative filtering.
- Divided in two main classes
  - User similarity: people who agree in their past evaluations tend to agree again in their future evaluations
  - Item similarity: objects that are similar to what a user has collected before.

User similarity

- For a given user, find other similar users whose ratings strongly correlate with the current user.

- Recommend items rated highly by these similar users, but not rated by the current user.



- Weight all users with respect to similarity with the active user.
- Select a subset of the users (*neighbors*) to use as predictors.
- Normalize ratings and compute a prediction from a weighted combination of the selected neighbors' ratings.
- Present items with highest predicted ratings as recommendations.

- Let denote with s_{uv} the similarity score between user u and user v
- To select the set $\hat{U}_u$ of users that are most similar to user u, there are two neighborhood selection strategies:
  1. maximum number of neighbors consists of using the most similar *k* users to u based on similarity score
  2. correlation threshold is based on selecting all the users whose similarity weight is above a given threshold.

The predicted rating of user u on object α is

$$\tilde{r}_{u\alpha} = \bar{r}_u + k \sum_{v \in \hat{U}_u} s_{uv} (r_{u\alpha} - \bar{r}_v)$$

where

- $r_{u\alpha}$: rating from user u on object α
- Γ_u : set of objects that user u has evaluated
- $\bar{r}_u = \frac{1}{|\Gamma_u|} \sum_{\alpha \in \Gamma_u} r_{u\alpha}$: average rating given by u
- $k = \frac{1}{\sum_v |s_{uv}|}$: normalization factor

The predicted rating of user u on object α is

$$\tilde{r}_{u\alpha} = \frac{\sum_{\beta \in \Gamma_u} s_{\alpha\beta} r_{u\beta}}{\sum_{\beta \in \Gamma_u} \left| s_{\alpha\beta} \right|}$$

where

- $s_{\alpha\beta}$: item-item similarity score
- Γ_u : set of objects that user u has evaluated

Similarity score

- Similarity of users/objects is the key problem
- Two scenarios:
  - Available ratings -> correlation metrics
  - No ratings available -> structural properties of the input data
- external information such as users' attributes, tags and objects' content meta information can be utilized

Cosine index

- When explicit information is available (5 levels from 1 to 5)

$$s_{xy}^{\cos} = \frac{r_x \cdot r_y}{\left|r_x\right| \cdot \left|r\right|}$$

Where

- For users similarity r_x and r_y are rating vectors in the N-dimensional object space.
- For items similarity r_x and r_y are rating vectors in the N-dimensional user space.

Important to keep into consideration ‘tendencies’

- Pearson coefficient for measuring rating correlation between users u and v:

$$s_{uv}^{PC} = \frac{\sum_{\alpha \in O_{uv}} (r_{u\alpha} - \bar{r}_u)(r_{v\alpha} - \bar{r}_v)}{\sqrt{\sum_{\alpha \in O_{uv}} (r_{u\alpha} - \bar{r}_u)^2} \sqrt{\sum_{\alpha \in O_{uv}} (r_{v\alpha} - \bar{r}_v)^2}}$$

Where

– $O_{uv} = \Gamma_u \cap \Gamma_v$ is the set of items rated by both u and v

- Pearson coefficient for measuring rating correlation between items α and β:

$$s_{\alpha\beta}^{PC} = \frac{\sum_{u \in U_{\alpha\beta}} (r_{u\alpha} - \bar{r}_\alpha)(r_{u\beta} - \bar{r}_\beta)}{\sqrt{\sum_{u \in U_{\alpha\beta}} (r_{u\alpha} - \bar{r}_\alpha)^2} \sqrt{\sum_{u \in U_{\alpha\beta}} (r_{u\beta} - \bar{r}_\beta)^2}}$$

Where

– $U_{\alpha\beta}$ is the set of users who rated both α and β

Correlation coefficients properties

- Used also for binary vectors
  - Amazon use case: *"User who bought this also bought"*
- Constrained Pearson coefficient
  - To take into consideration positive and negative rates
  - $\overline{r}_x$ is substituted by the "central rating" (3 stars)
- Weighted Pearson coefficient
  - To capture confidence in the correlation

$$S_{uv}^{WPC} = \begin{cases} s_{uv}^{PC} \dfrac{|O_{uv}|}{H} & for \;\; |O_{uv}| \leq H \\ s_{uv}^{PC} & otherwise \end{cases}$$

Structural similarity

- Similarity can be defined using the external attributes such as tag and content information (difficult to obtain)
- structural similarity only exploit data network structure
- For sparse data, structural similarity outperforms correlation
- Computed by projecting the rating bipartite network into a monopartite user-user or item-item network

Node-dependent similarity

The node similarity is given by the number of Common Neighbors (CN)

Many possible variations:

- Salton Index, Jaccard Index, Sørensen Index, Hub Promoted Index (HPI), Hub Depressed Index (HDI) and Leicht-Holme-Newman Index (LHN1)
- Variations to reward less-connected neighbors with a higher weight: Adamic- Adar Index (AA) and Resource Allocation Index (RA)
- Preferential Attachment Index (PA) builds on the classical preferential attachment rule in network science

Index	Definition
CN	$s_{xy} = \|\Gamma_x \cap \Gamma_y\|$
Salton	$s_{xy} = \|\Gamma_x \cap \Gamma_y\| / \sqrt{k_x k_y}$
Jaccard	$s_{xy} = \|\Gamma_x \cap \Gamma_y\| / \|\Gamma_x \cup \Gamma_y\|$
Sørensen	$s_{xy} = 2\|\Gamma_x \cap \Gamma_y\| / (k_x + k_y)$
HPI	$s_{xy} = \|\Gamma_x \cap \Gamma_y\| / \min\{k_x, k_y\}$
HDI	$s_{xy} = \|\Gamma_x \cap \Gamma_y\| / \max\{k_x, k_y\}$
LHN1	$s_{xy} = \|\Gamma_x \cap \Gamma_y\| / (k_x k_y)$
AA	$s_{xy} = \sum_{z \in \Gamma_x \cap \Gamma_y} 1 / \ln k_z$
RA	$s_{xy} = \sum_{z \in \Gamma_x \cap \Gamma_y} 1 / k_z$
PA	$s_{xy} = k_x k_y$

Path-dependent similarity

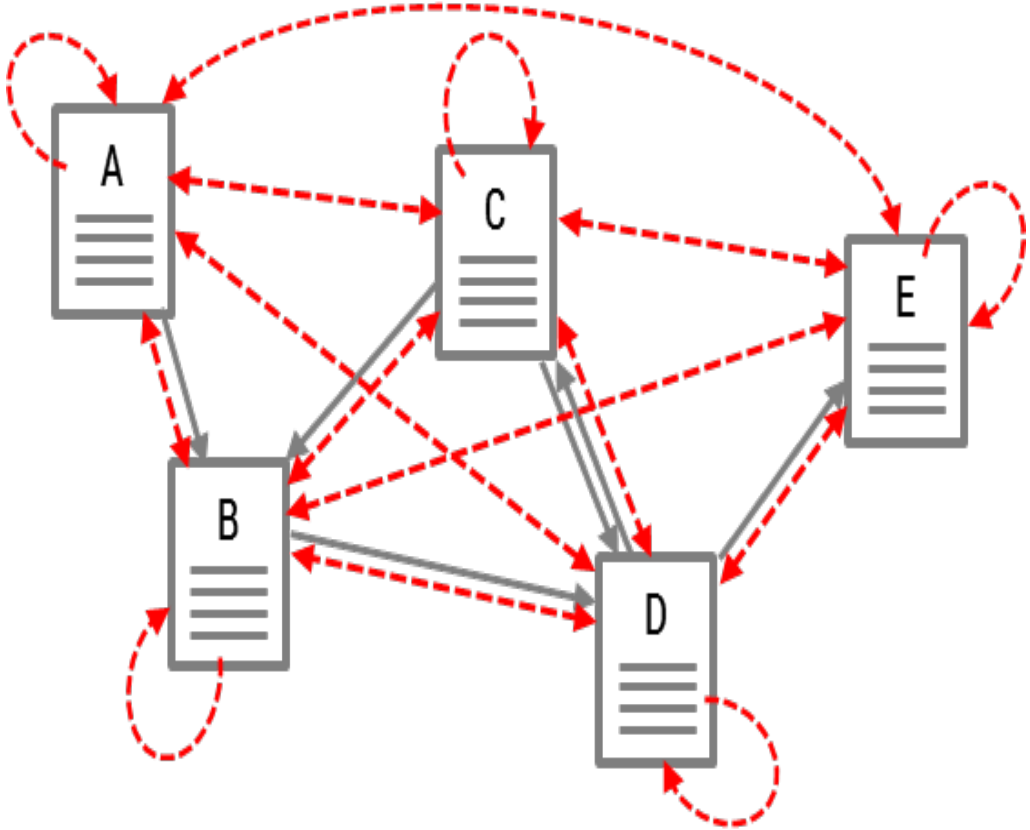
- Two nodes are similar if they are connected by many paths
- $\left[A^n\right]_{ij}$: number of paths between nodes i and j
- Local Path Index:

$$s_{xy}^{LP} = \left(A^2\right)_{xy} + \varepsilon\left(A^3\right)_{xy}$$

- Katz similarity:

$$s_{xy}^{Katz} = \beta A_{xy} + \beta^2\left(A^2\right)_{xy} + \beta^3\left(A^3\right)_{xy} + \ldots$$

Random-walk-based similarity.



$$R_{i+1} = (1-\alpha)(S^T + \mathbf{w} \times \mathbf{d}^T)R_i + \alpha \mathbf{w}$$

Image courtesy: http://parkcu.com/blog/pagerank/

Topic Sensitive or Personalized Pagerank

$$R_{i+1} = (1-\alpha)(S^T + w \times d^T)R_i + \alpha \begin{bmatrix} \frac{1}{N} \\ \vdots \\ \frac{1}{N} \end{bmatrix} \Rightarrow R_{i+1} = (1-\alpha)(S^T + w \times d^T)R_i + \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

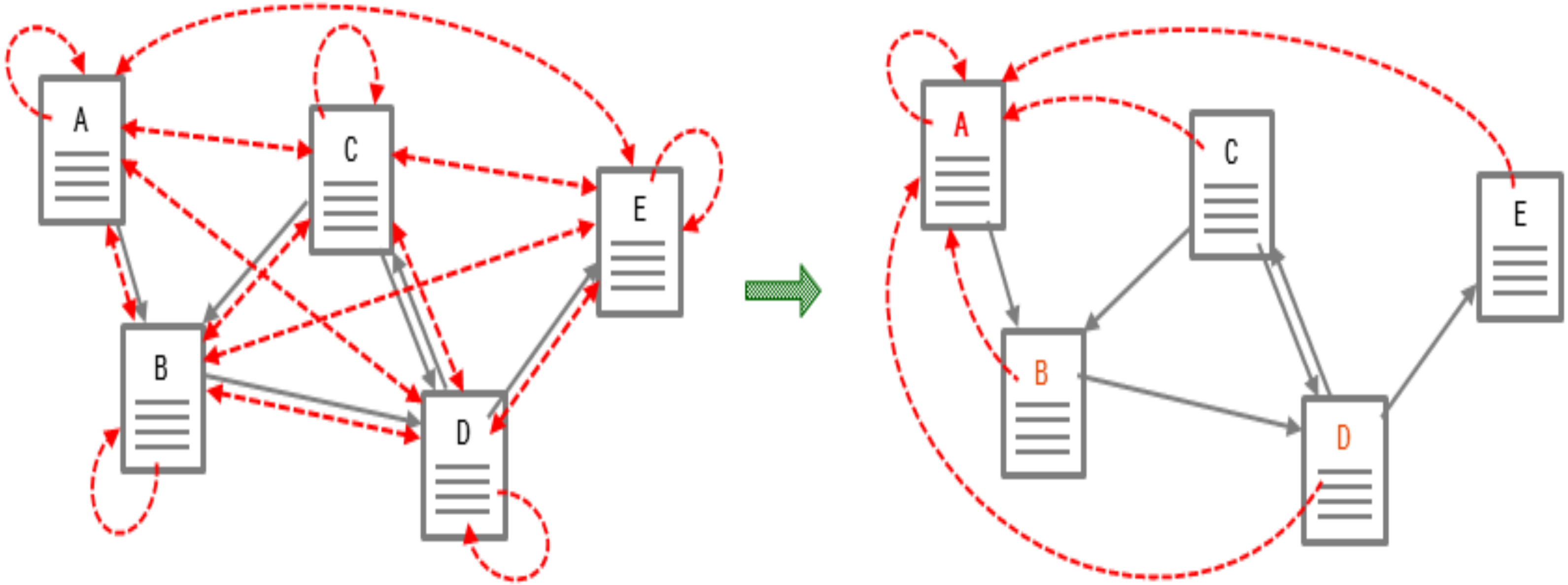


Image courtesy: http://parkcu.com/blog/pagerank/

Many other variations

– SimRank: based on the assumption that two nodes are similar if they are connected to similar nodes

$$s_{xy}^{SimRank} = C\frac{\sum_{z\in\Gamma_x}\sum_{z'\in\Gamma_x} s_{zz'}^{SimRank}}{k_x k_y}$$

– Local Random Walk: To measure similarity between nodes x and y, a random walker is introduced in node x

- the initial occupancy vector is $\pi_x(0) = e_x$
- At each t: $\pi_x(t+1) = P^T\pi_x(t)$

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

- q is the initial configuration function and t denotes the time step
- q may be detrmined by the node degree $q_x = k_x / M$

Similarity based on external information

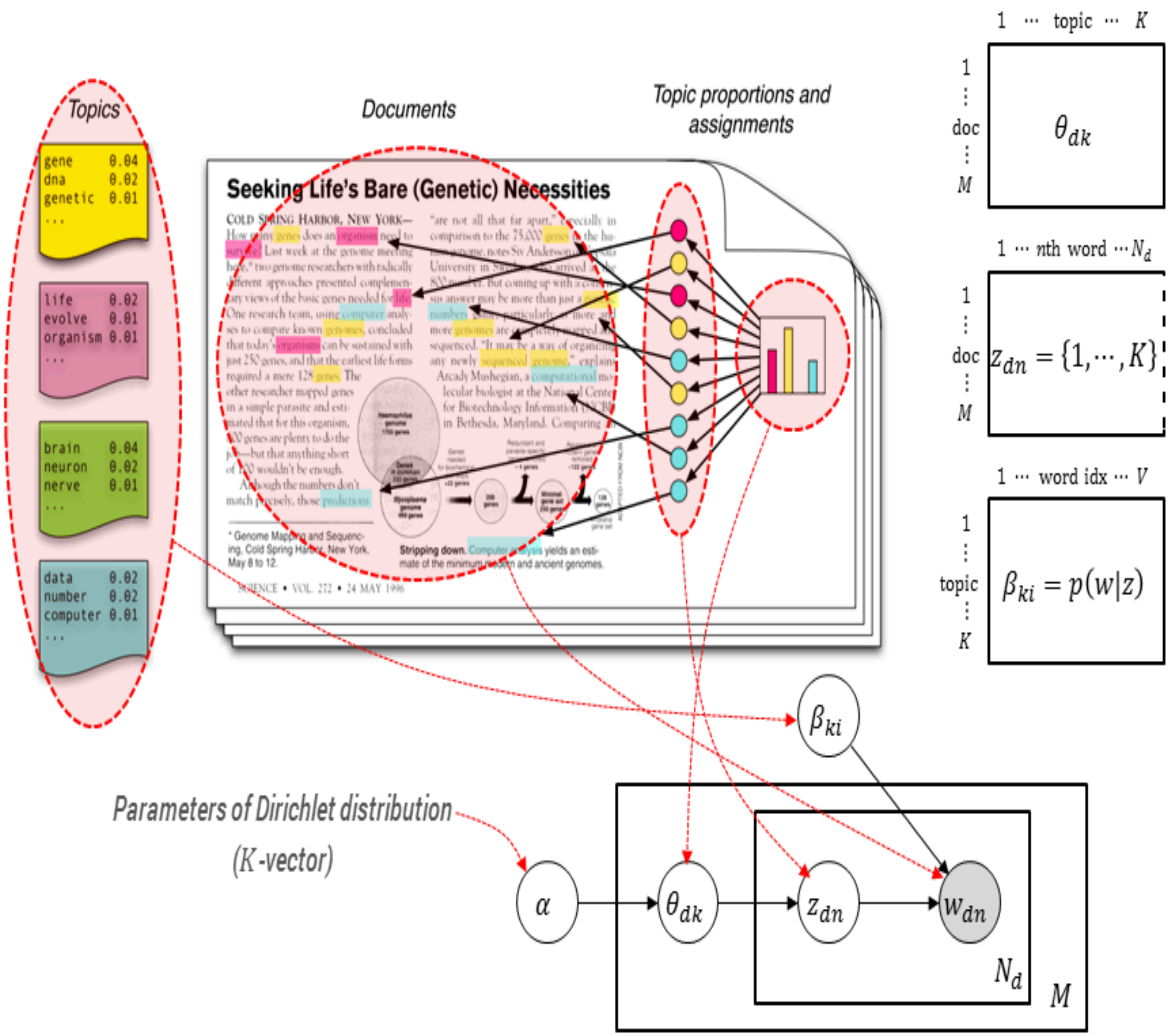
- User attributes:
  - u: <age,gender, location, career,…>
- Content meta information
  - Information retrieval
- User-generated tags

SERENDIPITOUS RECS

Hibrid methodology

- Content features extraction
  - Dimensionality Reduction
  - Build LDA model using “Head” URLs
  - Use the model to classify “Tail” URLs in Latent Topic Space
- Document Graph
  - Compute pairwise similarity between documents with topic overlaps Cosine Similarity, Weighted Jaccard
  - Build a graph where documents make up the nodes and the similarity score make up the edge weights.
- Page Rank
  - Run topic sensitive page rank over the document graph.
  - Spot influential documents per topic and index for fast retrieval

Content Categorization: Discovering Semantic Groups



$$P(\mathbf{W},\mathbf{Z},\theta|\alpha,\eta) = \prod_{d=1}^{M} P(\theta|\alpha) \prod_{n=1}^{N_d} P(w_{d,n}|\theta_d) P(w_{dn}|z_{dn},\beta_{1:V,z_{dn}})$$

Properties

- Unsupervised (Classic LDA) and generative
- Well suited for domain adaptation (taxonomy shift)
- Allows making topic clusters as loose/tight as needed
  – α controls the peak-ness of the per-document topic distributions
  – β controls the peak-ness of the per-topic word distributions
- Can be extended to discover relations, hierarchies, etc.,

Evaluation + Relearning

- Periodically evaluate the model
  - Perplexity $2^{Entropy} = 2^{-\sum p \log p}$
    - Measure of how surprised the model is on an average when having to guess between k equally probable choices.
    - The average log probability of the trained model having seen the test samples
  - Use human judgment from word intrusion and topic intrusion tasks
- Good topic associations can be initialized from previous trainings or from separate topic clustering

Topic Mixtures

recipe
sugar
baking
cooking
oven
butter
food
vanilla
cream
black_pepper

photography
art
design
photograph
artist
camera
digital
architecture
illustration
painting

nasa
earth
moon
planet
sun
space
solar_system
universe
astronomer
science

israel
jews
egypt
bible
jesus
syria
iran
russia
new_testament
god

facebook
internet
twitter
advertising
youtube
marketing
myspace
google
social_media
mass_media

cannabis_(drug)
police
crime
law
prison
cannabis
drug
murder
gun
federal_bureau_of_investigation

energy
water
carbon_dioxide
sustainability
gas
nuclear_weapon
electricity
sun
climate_change
solar_energy

- Given an initial document d, we can pick similar document i.e., document with a similar distribution on the topic space.
- Using topical page rank to control serendipity

		T1	T2	T3		T4	T5
D1		1	1	0		0	1
D2		1	1	0		1	1
D3		1	1	0		1	0

- A/B Testing
  - Measure the difference in user behavior (implicit/explicit signals and retention):
    - "A Recommended item" vs. "Randomly picked item from the set"
    - "Serendipity free stumbling session" vs. "Sessions with serendipitous recommendations"