

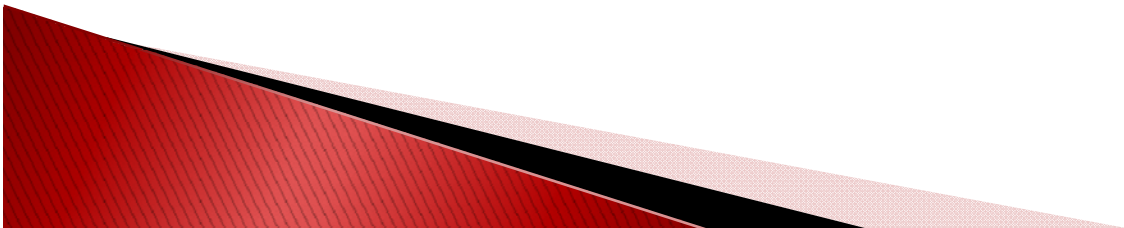
On social networks and collaborative recommendation

Social Networks WS09/10
Speaker: Julia Wolf

based on a paper by Ioannis Konstas, Vassilios Stathopoulos and Joemon M Jose

Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ Conclusion



Outline

- ▶ **The Network**
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ Conclusion

The Network: Last.fm

last.fm Music Radio Events Videos Charts Community Inbox Logout yzeal

Play Sony Fantasy Festival on Last.fm English | Paint it Black | Help Music search

Community » People

Find people on Last.fm

Search by username or real name

Looking for groups?
Go to the [Group Search](#)

Looking for music?
Go to the [Music Search](#) or [Event Search](#)

Browse People

Looking for people with music taste similar to yours? Check out your [neighbours](#), like-minded listeners recommended by Last.fm

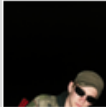

Country:

Music taste

Type in some artist names separated by commas.

[More options »](#)

Recently Active People





 <p>Cidero 22, Male, Austria Last track: Paramore – Ignorance</p>	 <p>ANNANN_LFC Annann Cantabile, 23, Female, Thailand Listening: Pink Floyd – Wish You Were Here</p>
---	--

You don't have any friends yet. Once you make some, you'll be able to see what they're listening to in real time! Try browsing for people with similar music taste.

Buy a subscription for a friend

When you subscribe to Last.fm you get uninterrupted radio listening, ad-free browsing and streaming plus more. You can buy a subscription for someone as a 1, 3, 6 or 12 months-long gift!

Recent Subscribers


 end_of_twilight	 HoneyGrrl
 bramblein	 mescalino

[See more](#)

Last.fm – recommendations

Music Recommended by Last.fm ▶ Play all Recommendations


All rock alternative pop pop punk alternative rock rnb emo female vocalists indie pop rock




▶ Go:Audio
483,518 plays (20,194 listeners)
+ Add to Your Library
pop punk, powerpop, indie rock, rock, alternative

James Matthews - Singer Josh Wilkinson - Keyboardist+Backing Vocals
Nick Tsang - Guitarist Andy Booth - Drummer [Read more](#)

Similar Artists from Your Library




Elliot Minor




▶ Simple Plan
18,339,530 plays (691,750 listeners)
+ Add to Your Library
rock, punk rock, pop punk, punk, emo

Simple Plan is a French Canadian pop/rock band formed in Montréal, Québec. They began with the formation of the Canadian punk band Reset in 1993, by high-school friends Pierre Bouvier and Charles-André (Chuck) Comeau. [Read more](#)


Similar Artists from Your Library



Elliot Minor




The All-American...



▶ Eve 6
4,511,764 plays (360,002 listeners)
+ Add to Your Library
rock, alternative, alternative rock, 90s, punk

Eve 6 is an alternative/post-grunge rock band which formed in 1995 in La Crescenta, California, United States. The band split in 2004 and reunited in October, 2007. Max Collins – lead vocals, bass Jon Siebels – guitar, vocals [Read more](#)

Similar Artists from Your Library



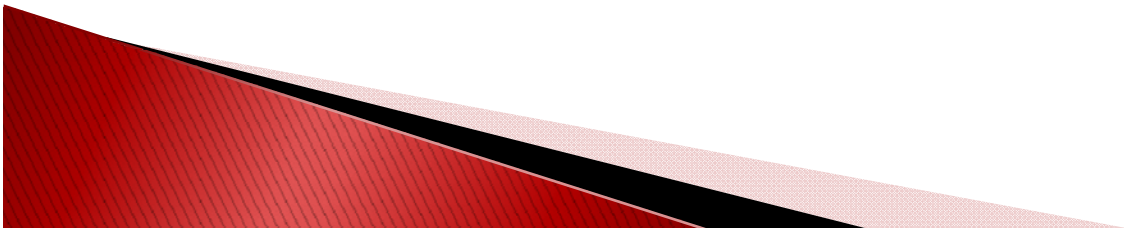
Third Eye Blind

Last.fm – recommendations (2)

The screenshot displays the Last.fm interface for a user named Yzeal. At the top, there's a navigation bar with a 'Start a new Station' button and the title 'Yzeal's Recommended Radio'. Below this is a 'Share Station' button and a photo credit 'Photos added by H_MM'. The main content area features a music player for 'The Pussycat Dolls - Wait A Minute' from the album 'PCD'. The player shows a progress bar at 00:22 and a total duration of -03:20. Below the player, there's a section for 'The Pussycat Dolls' with their profile picture, name, and statistics: '15,305,599 plays (872,454 listeners)'. The genres listed are 'pop, rnb, dance, female vocalists, hip-hop'. To the right of the artist name are buttons for 'Share Track', 'Tag Track', and 'Buy Track'. Below this is a description: 'The Pussycat Dolls are an US female dance-pop singing group, consisting of lead singer Nicole Scherzinger, Melody Thornton, Ashley Roberts, Jessica Sutta and Kimberly Wyatt. ... (read more)'. Further down, there are sections for 'Similar Artists' (featuring Nicole Scherzinger, Girlicious, and Britney Spears) and 'From the album' (featuring the album cover for 'PCD The Pussycat Dolls' with a 'Buy' button). On the right side of the page, there's a large advertisement for Sony with the text 'KANNST DU DAS GRÖßTE FESTIVAL ALLER ZEITEN ERSTELLEN?' and the Sony logo 'SONY make.believe'. Below the ad are sections for 'Your Recent Tracks' (showing 'No recent tracks saved'), 'Your Recent Stations' (listing 'Your Recommended Radio', 'Simple Plan Radio', and 'Go:Audio Radio'), and 'Your Stations' (listing 'Your Neighbourhood' and 'Your Recommended Radio'). At the bottom right, there's a notification box that says 'Get music recommendations based on the music in your library.' and provides instructions to download the Scrobbler.

Outline

- ▶ The Network
- ▶ **Collaborative Recommendation**
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ Conclusion

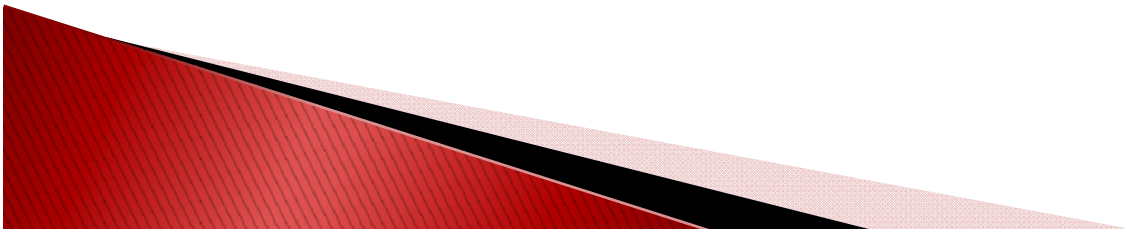


Collaborative Recommendation

▶ Content-based

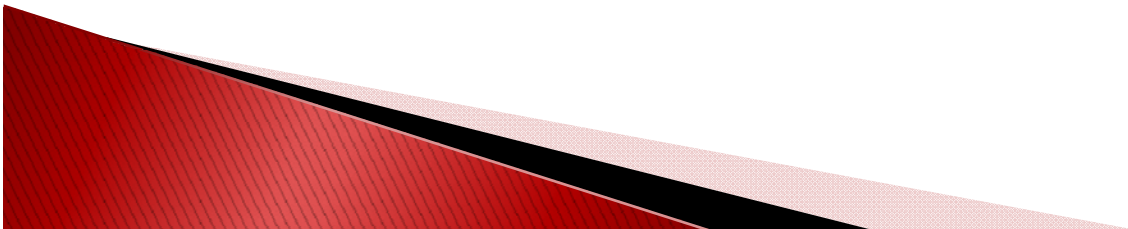
- based on correlation between **content** and a user's **preferences**
- limited to **dictionary-bound relations**

▶ Collaborative filtering (CF)



Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ **Collaborative Filtering**
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ Conclusion



Collaborative Filtering (CF)

▶ Memory-based:

- based on users' ratings of items
- Aggregate ratings of k nearest neighbours.

▶ Problems:

- integrating friendship and social tagging
- choice for k

Collaborative Filtering (2)


▶ Model-based:

- **clusters** based on similar rating behaviour
- **patterns** recognised inside the clusters

Problems:

- fine-tuning of parameters
- generalisation of models
- integrating additional information

Collaborative Filtering (3)

- ▶ User ratings are **bounded** and **discrete** (e.g. 0 to 5 stars on Amazon ).
- ▶ Memory-based CF systems:
 - **user based**
 - **item based**

CF – user based

► User based:

- Predict rating $p_{a,i}$ based on **users u with similar ratings** to those of the active user a .
- weighted combination of the ratings:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^N (r_{u,i} - \bar{r}_u) w_{a,u}}{\sum_{u=1}^N w_{a,u}}$$

a : active user
 i : item not yet rated by a
 $p_{a,i}$: predicted rating of a to i
 $r_{u,i}$: rating of user u to i
 \bar{r} : mean rating of user
 $w_{a,u}$: similarity weight between a and u

CF – item based

► Item based:

- Predict rating $p_{a,i}$ based on items k with similar ratings to item i .
- weighted combination of the ratings:

$$p_{a,i} = \bar{r}_i + \frac{\sum_{k=1}^M (r_{a,k} - \bar{r}_k) w_{i,k}}{\sum_{k=1}^M w_{i,k}}$$

a : active user

i : item not yet rated by a

$p_{a,i}$: predicted rating of a to i

$r_{a,k}$: rating of items k by a

\bar{r} : mean rating of item

$w_{i,k}$: similarity weight between i and k

CF on Last.fm

- ▶ **Item based** systems:
 - significantly **less items than users**
 - true for most **commercial** applications
- ▶ Last.fm:
 - significantly **more tracks than users**
 - ⇒ Consider a **user based** system.
- ▶ important if we think about the sizes of the item–item or user–user similarity matrices

CF – Pearson correlation

- ▶ Compute the **similarity weights** $w_{a,u}$.
- ▶ **Pearson correlation score** is used:

$$w_{a,u} = \frac{\sum_{i=1}^M (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sigma_a \sigma_u}$$

a : active user
 i : item not yet rated by a
 σ_a : standard deviation of a 's ratings
 $r_{u,i}$: rating of items i by user u
 \bar{r} : mean rating of item
 $w_{i,k}$: similarity weight between i and k

CF – integrating social components

- ▶ We rely on **playcount** instead of ratings.
- ▶ Integrate **friendship** and **tagging** into the model.
- ▶ ⇒ Compute **three similarity weights** based on
 - playcount,
 - tags and
 - friendships.

CF – integrating social components (2)

- ▶ Use the **Pearson correlations coefficient** for each.
- ▶ Replace the similarity weights with

$$W_{a,u} = \alpha W_{a,u}^{(I)} + \beta W_{a,u}^{(F)} + \gamma W_{a,u}^{(T)}$$

where

$$\alpha + \beta + \gamma = 1$$

and $W_{a,u}^{(I)}$, $W_{a,u}^{(F)}$ and $W_{a,u}^{(T)}$ are based on **user tracks**, **user friendships** and **user tags** respectively.

Different view on the problem

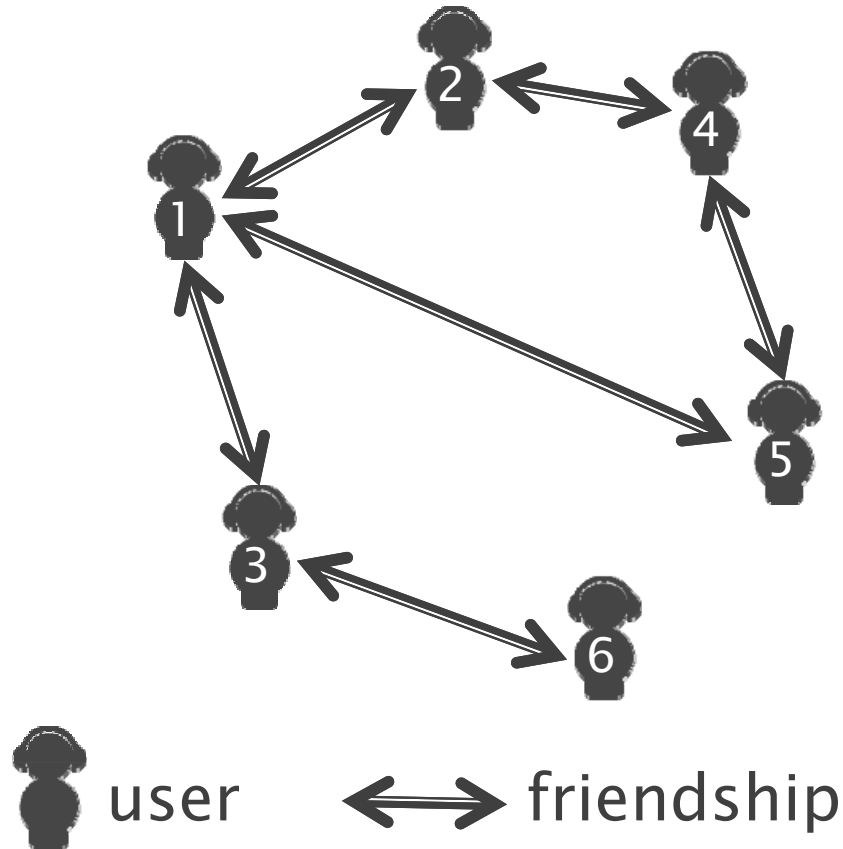
- ▶ View different **similarities** as a **graph** with different node types.
 - ▶ Starting from a user: through friendship and preferences, where will he end up?
- ⇒ Random Walk model over the graph

Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ **The Social Graph**
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ Conclusion

Last.fm social graph – friends

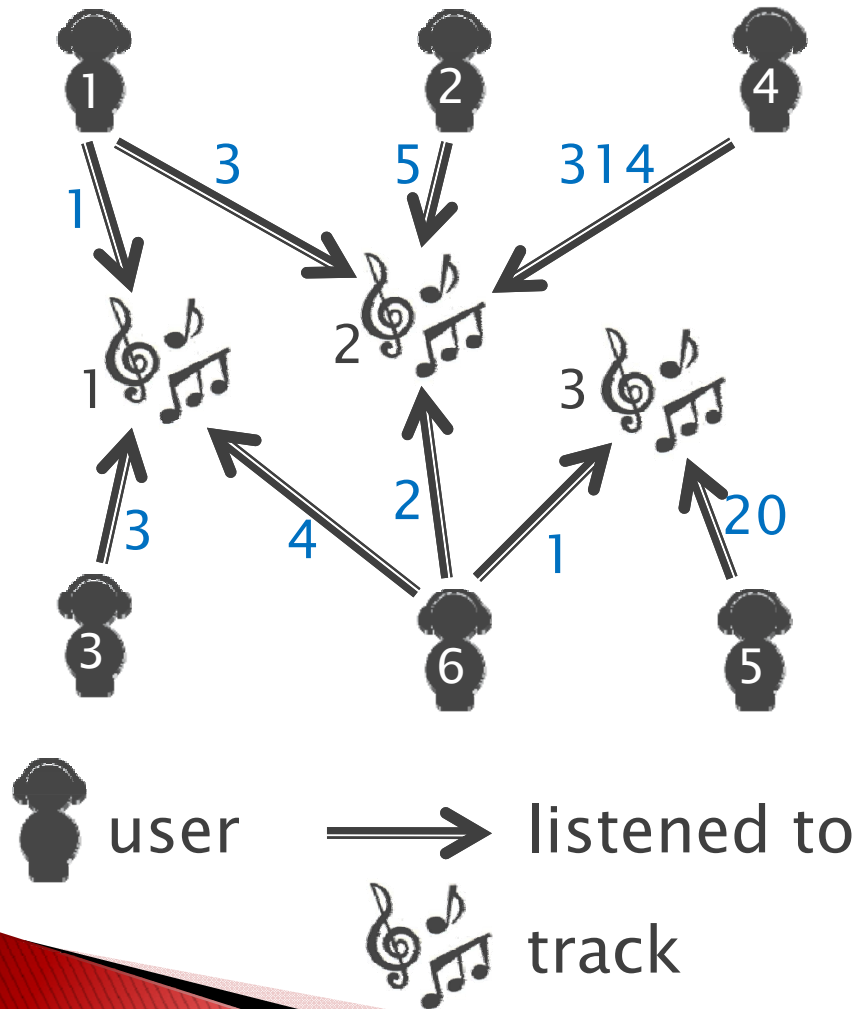
Corresponding **adjacency matrix**:



$$UU := \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

User–User matrix (UU)

Social graph – tracks



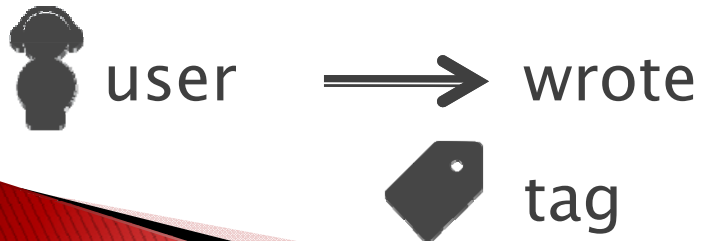
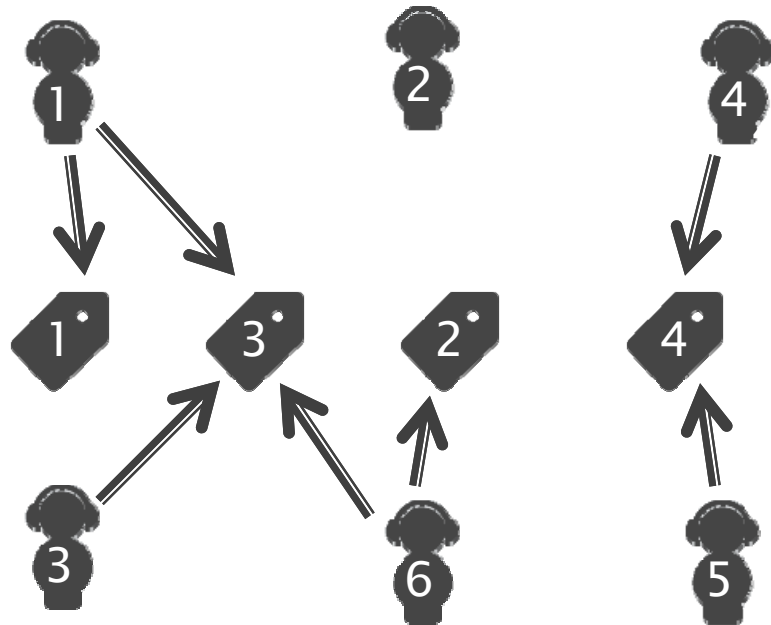
Corresponding **adjacency matrix**:

$$UTr := \begin{bmatrix} 1 & 3 & 0 \\ 0 & 5 & 0 \\ 3 & 0 & 0 \\ 0 & 314 & 0 \\ 0 & 0 & 20 \\ 4 & 2 & 1 \end{bmatrix}$$

User-Track matrix (UTr)

The **numbers** correspond to the times a user has listened to a track. They can be **very high**.

Social graph – tags

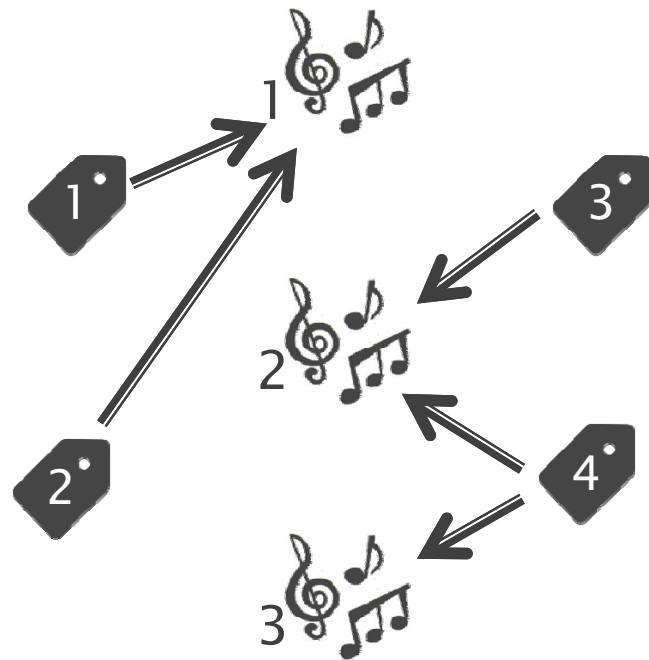


Corresponding **adjacency matrix**:

$$UTg := \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

User-Tag matrix (UTg)

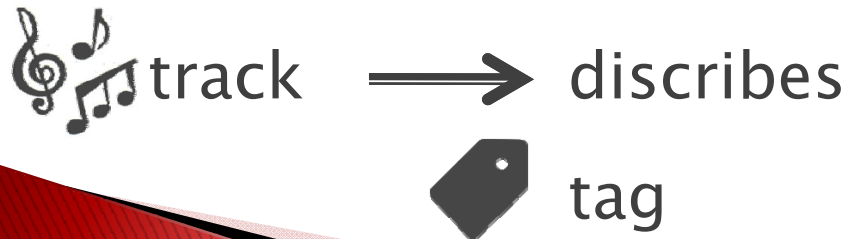
Social graph – tags/tracks



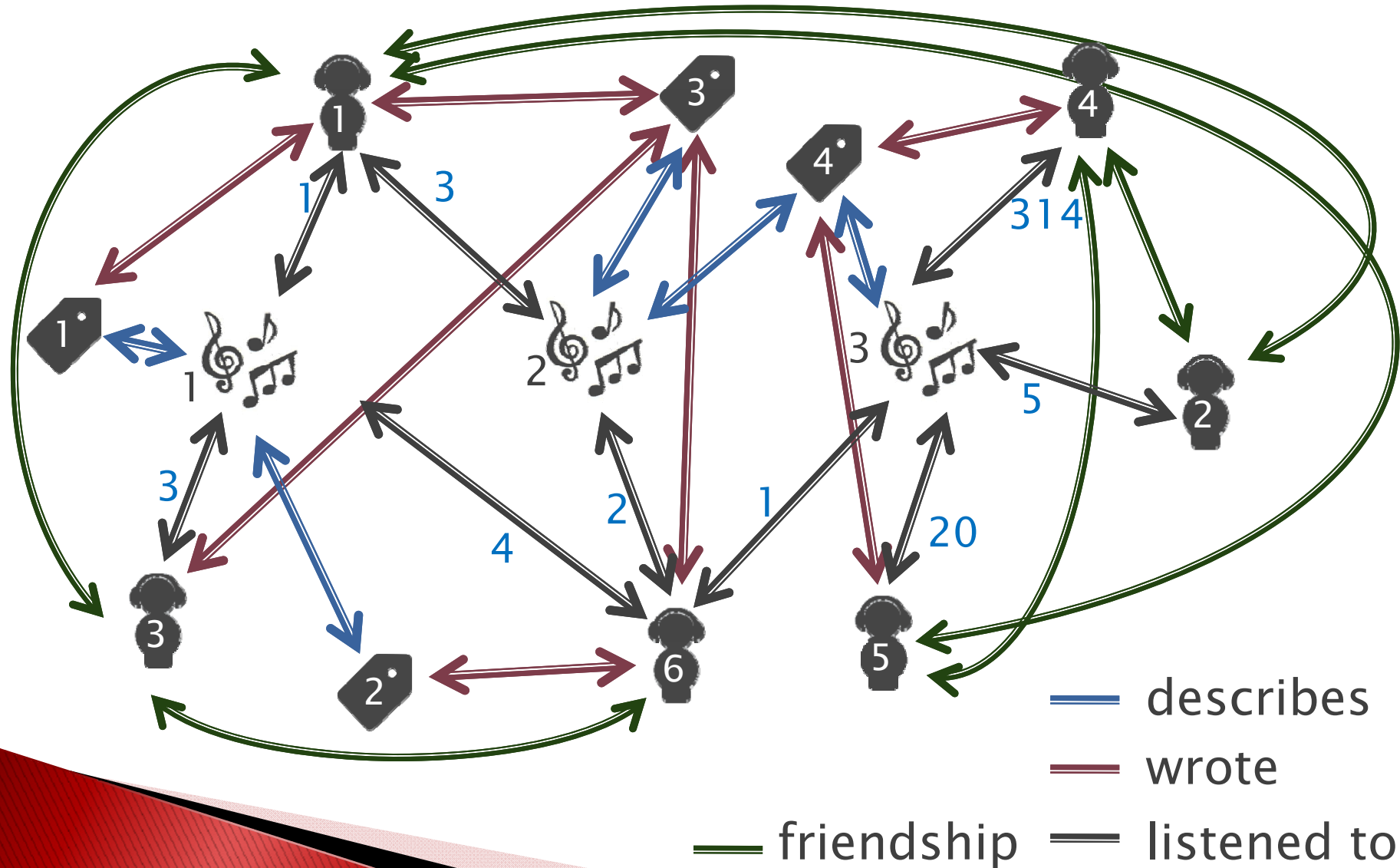
Corresponding **adjacency matrix**:

$$\text{TrTg} := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Track-Tag matrix (TrTg)



Complete social graph



— describes
 — wrote
 — friendship — listened to

Complete social graph – matrix

What we call the **social graph S** is now made up of UU , UTr , UTg and $TrTg$ as follows:

	Users	Tracks	Tags
Users	UU	UTr	UTg
Tracks	UTr^T	0	$TrTg$
Tags	UTg^T	$TrTg^T$	0

$=: S$

Social graph – normalisation

- ▶ **playcount** in UTr for each ranges from 1 to **11640**
⇒ **Normalise** the other sub-matrices of S:
- ▶ Replace each bond of **friendship** in UU with the **average user playcount**.
- ▶ Apply an **exponential decay function** to the popularity values in UTg and TrTg.
- ▶ Most **popular** tags get the **average user playcount**.

Social graph – example

- ▶ In our example from before, where the average playcount is **39**, that yields:

$$UU = \begin{bmatrix} 0 & 39 & 39 & 0 & 39 & 0 \\ 39 & 0 & 0 & 39 & 0 & 0 \\ 39 & 0 & 0 & 0 & 0 & 39 \\ 0 & 39 & 0 & 0 & 39 & 0 \\ 39 & 0 & 0 & 39 & 0 & 0 \\ 0 & 0 & 39 & 0 & 0 & 0 \end{bmatrix} \quad UTg = \begin{bmatrix} 14 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 39 \\ 0 & 0 & 0 & 39 \\ 0 & 14 & 14 & 0 \end{bmatrix} \quad TrTg = \begin{bmatrix} 14 & 14 & 0 & 0 \\ 0 & 0 & 14 & 39 \\ 0 & 0 & 0 & 39 \end{bmatrix}$$

Here I simply assumed $f(x) = 39 * e^{-x}$ as the exponential decay function and truncated the values.

Social graph – example (2)

- ▶ We want to use S as a transition probability table.
- ▶ \Rightarrow **Normalise S .**

$$UU = \begin{bmatrix} 0 & 39 & 39 & 0 & 39 & 0 \\ 39 & 0 & 0 & 39 & 0 & 0 \\ 39 & 0 & 0 & 0 & 0 & 39 \\ 0 & 39 & 0 & 0 & 39 & 0 \\ 39 & 0 & 0 & 39 & 0 & 0 \\ 0 & 0 & 39 & 0 & 0 & 0 \end{bmatrix}$$

$$UU' = \begin{bmatrix} 0 & \frac{39}{117} & \frac{39}{117} & 0 & \frac{39}{117} & 0 \\ \frac{39}{78} & 0 & 0 & \frac{39}{78} & 0 & 0 \\ \frac{39}{78} & 0 & 0 & 0 & 0 & \frac{39}{78} \\ 0 & \frac{39}{78} & 0 & 0 & \frac{39}{78} & 0 \\ \frac{39}{78} & 0 & 0 & \frac{39}{78} & 0 & 0 \\ 0 & 0 & \frac{39}{39} & 0 & 0 & 0 \end{bmatrix}$$

$$UU' = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

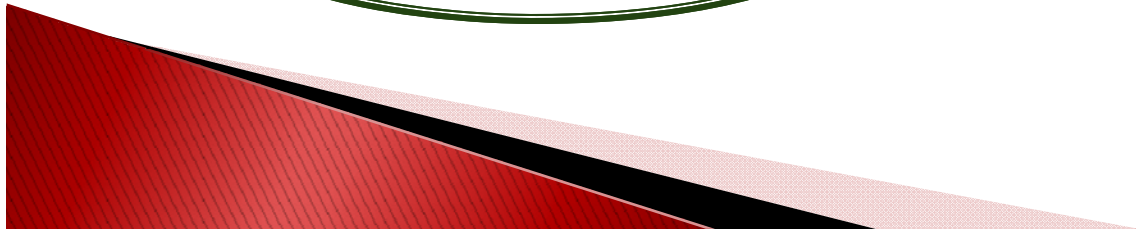
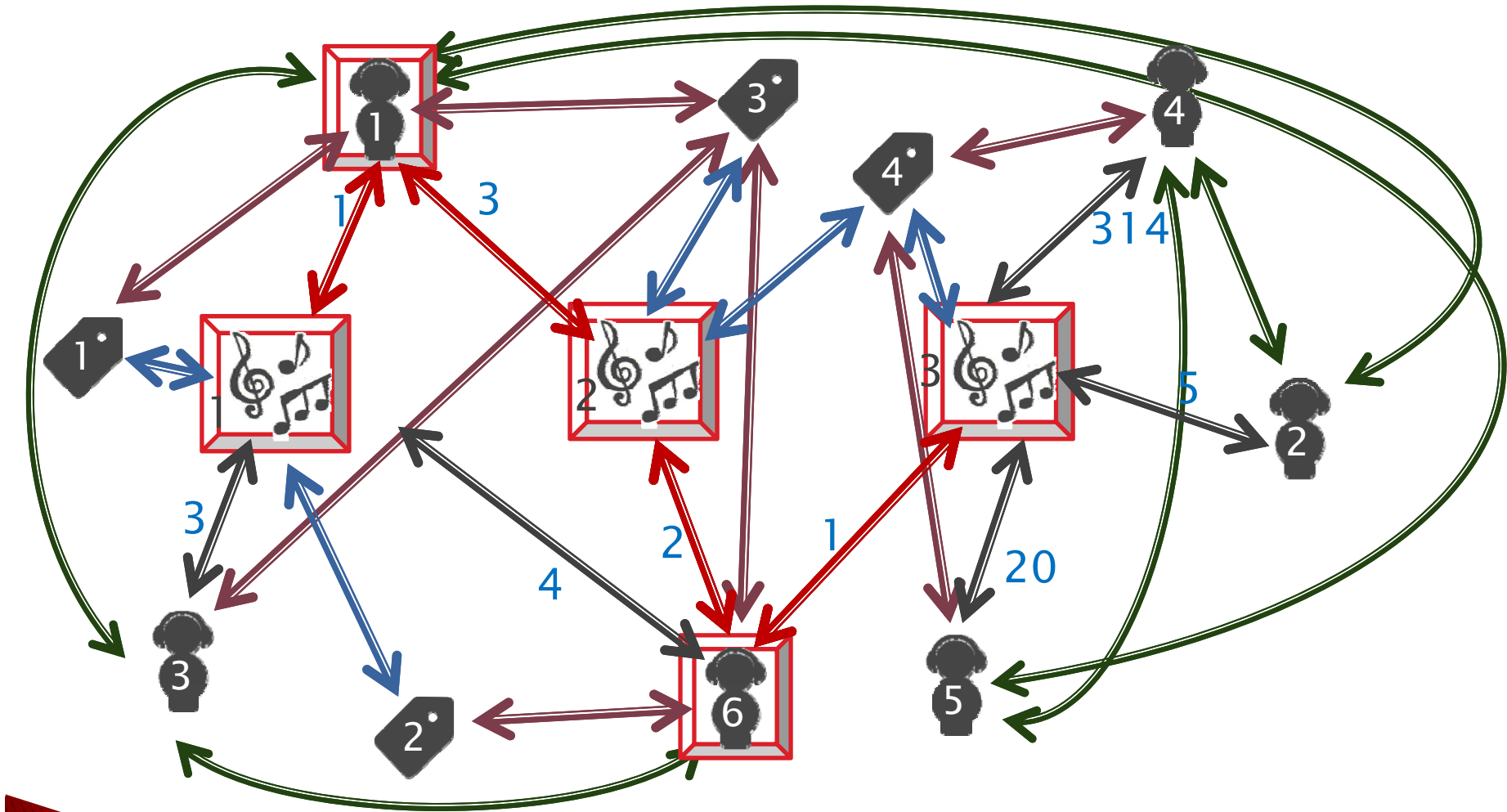
Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ **Random Walks with Restarts**
- ▶ Experiments and Results
- ▶ Conclusion

Random Walks with Restarts (RWR)

- ▶ We use Random Walks to compute the **relatedness of two nodes** in a graph.
- ▶ Start with node x .
- ▶ Perform a Random Walk by **randomly** following links to other nodes.
- ▶ There's also a **probability a** to **restart** at x in each step.

A Random Walk



RWR – definitions

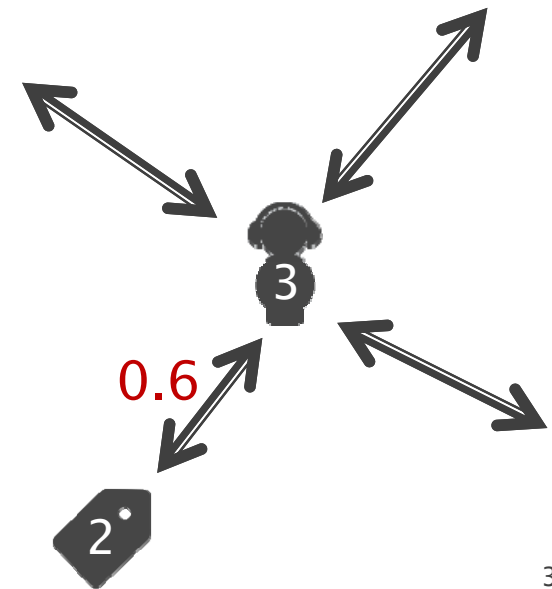
- ▶ $\mathbf{p}^{(t)}$ is a column vector.
 $p_i^{(t)}$ is **probability** that we are at **node i** in **step t**
- ▶ q corresponds to the starting setup.
- ▶ S (column normalised) is the **transition probability table**. Its elements $S_{i,j}$ give the probability of j being the next state if we are in state i .

$$S = \begin{bmatrix} 0.3 & 0.5 & 0.1 & 0.1 \\ 0.4 & 0.1 & 0.2 & 0.3 \\ 0.1 & 0.6 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{bmatrix}$$

$j=2$ (above the second column)

$i=3$ (to the left of the third row)

The element $S_{3,2} = 0.6$ is highlighted with a red box.



RWR – probabilities

- ▶ **steady-state probabilities** for each node:
Apply

$$\mathbf{p}^{(t+1)} = (1 - \alpha)\mathbf{S}\mathbf{p}^{(t)} + \alpha\mathbf{q}$$

until convergence.

- ▶ \Rightarrow **long-term visit rates** of each node based on the starting node x .

RWR – relatedness

- ▶ $p_i^{(l)}$ is the **measure of relatedness** between x and i .
- ▶ By taking into account users'
 - **music taste**,
 - **tagging behaviour** and
 - **friendships**

the Random Walks method allows us to **predict preferences** of users to particular tracks.

Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ **Experiments and Results**
- ▶ Conclusion

Experiments: Last.fm – data

- ▶ The following amount of **data** was extracted from the Last.fm social network:
 - ▶ 3148 users
 - ▶ 30520 tracks
 - ▶ 12565 tags
 - ▶ 5616 bonds of friendship
- ▶ In order to make the resulting matrices **less sparse**, only very **active users** were selected. The rest of the dataset was populated with respect to those users.

Experiments – ground truth

- ▶ 20% of the tracks each user has listened to will be set to 0 in UTr.
- ▶ We can later compare our results to this **ground truth**.

Experiments – CF

- ▶ Predict the **playcount** of the tracks in S for each user.
 - ⇒ **ranked vector** of the best 1000 tracks in descending order of playcount
- ▶ Playcount variables have a **wide range** (1 – 11640) .
 - ⇒ **Standardise** the playcount and the mean of the nearest neighbours.

Experiments – CF (2)

- ▶ Subtract the mean and divide with the standard deviation:

$$p_{a,i} = \frac{\sum_{u=1}^n \left(\frac{r_{u,i} - \bar{r}_a}{\sigma_a} - \frac{\bar{r}_u - \bar{r}_a}{\sigma_a} \right) W_{a,u}}{\sum_{u=1}^n W_{a,u}}$$

σ_a : standard deviation of a 's ratings
 a : active user
 i : item not yet rated by a
 $p_{a,i}$: predicted rating of a to i
 $r_{u,i}$: rating of user u to i
 \bar{r} : mean rating of user
 $W_{a,u}$: similarity weight between a and u

Experiments – CF – parameters

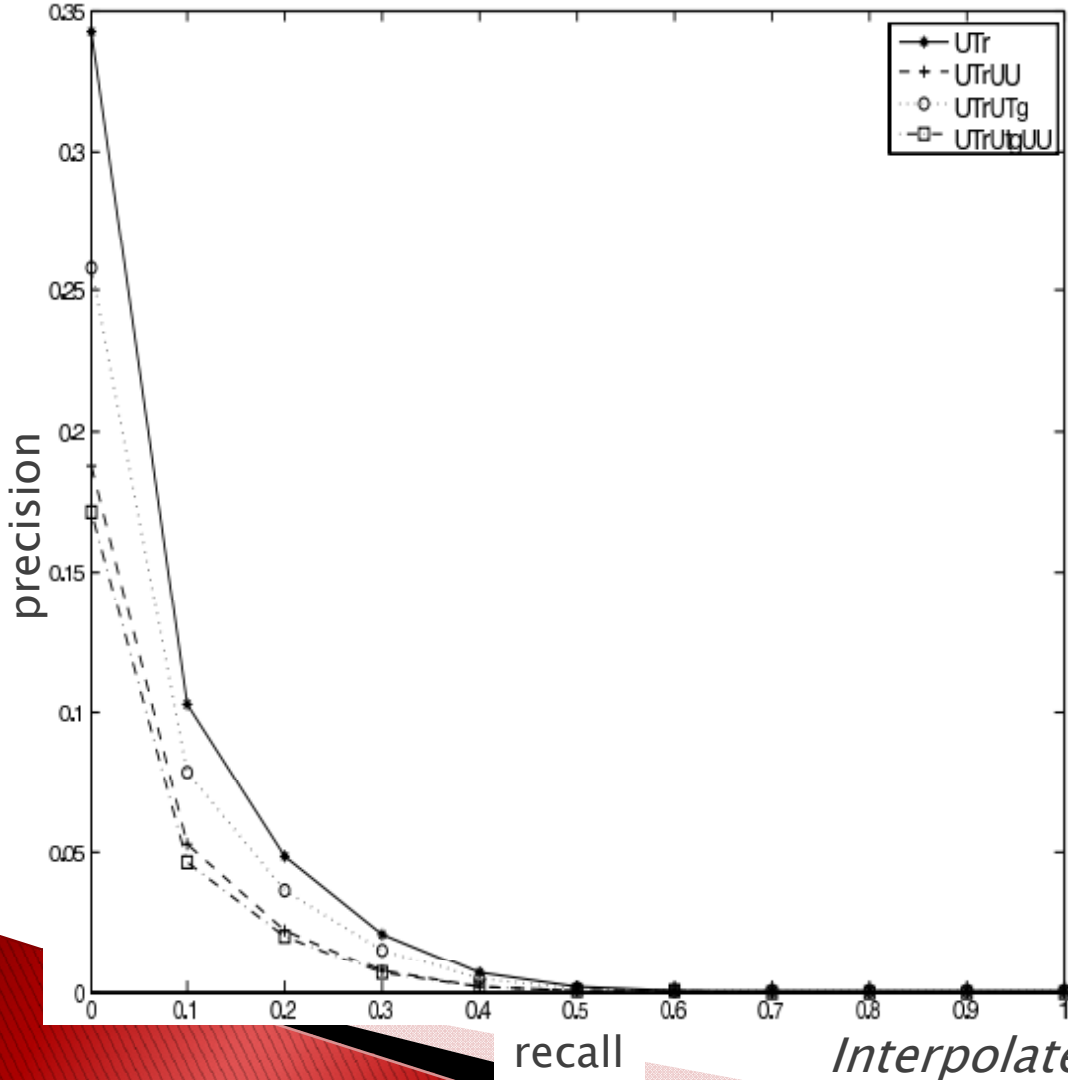
- ▶ The CF method adopted in these experiments uses **two parameters**:
 - The **threshold number** of common tracks between users T_r , empirically estimated to 20.
 - The number of **nearest neighbours** k with the highest correlation to a user, again estimated to 15.

Both were estimated using a withheld part of the graph.

Experiments – CF – matrices

- ▶ Three **user–user similarity matrices** were computed:
 - $W_{a,u}^{(I)}$: based on UTr, the **standard** used in CF systems
 - $W_{a,u}^{(F)}$: based on UU, measuring correlation based on **friendship**
 - $W_{a,u}^{(T)}$: based on UTg, measuring correlation based on **collaborative annotation**

Experiments - CF - results



Based on:

UTr: tracks

UTrUU: tracks and friendships

UTrUTg: tracks and tags

UTrUTgUU: tracks, tags and friendship

Interpolated Recall-Precision Curves

Experiments – CF – results (2)

- ▶ **Additional information** about friendship and social tagging makes the performance **worse**.
- ▶ Possible reason:
Weights and **parameters**, **have not been tuned finely enough**.
- ▶ So the **memory based CF** method **cannot incorporate social knowledge** in a trivial way.

Experiments – RWR

- ▶ Top 1000 tracks in descending order for each user:
- ▶ N : number of columns in S
- ▶ u_a : user currently under evaluation
- ▶ Create a query vector q with $q_i = 1$ if $S_{u,i} > 0$ and $q_{u_a} = 1, i=1..N$.

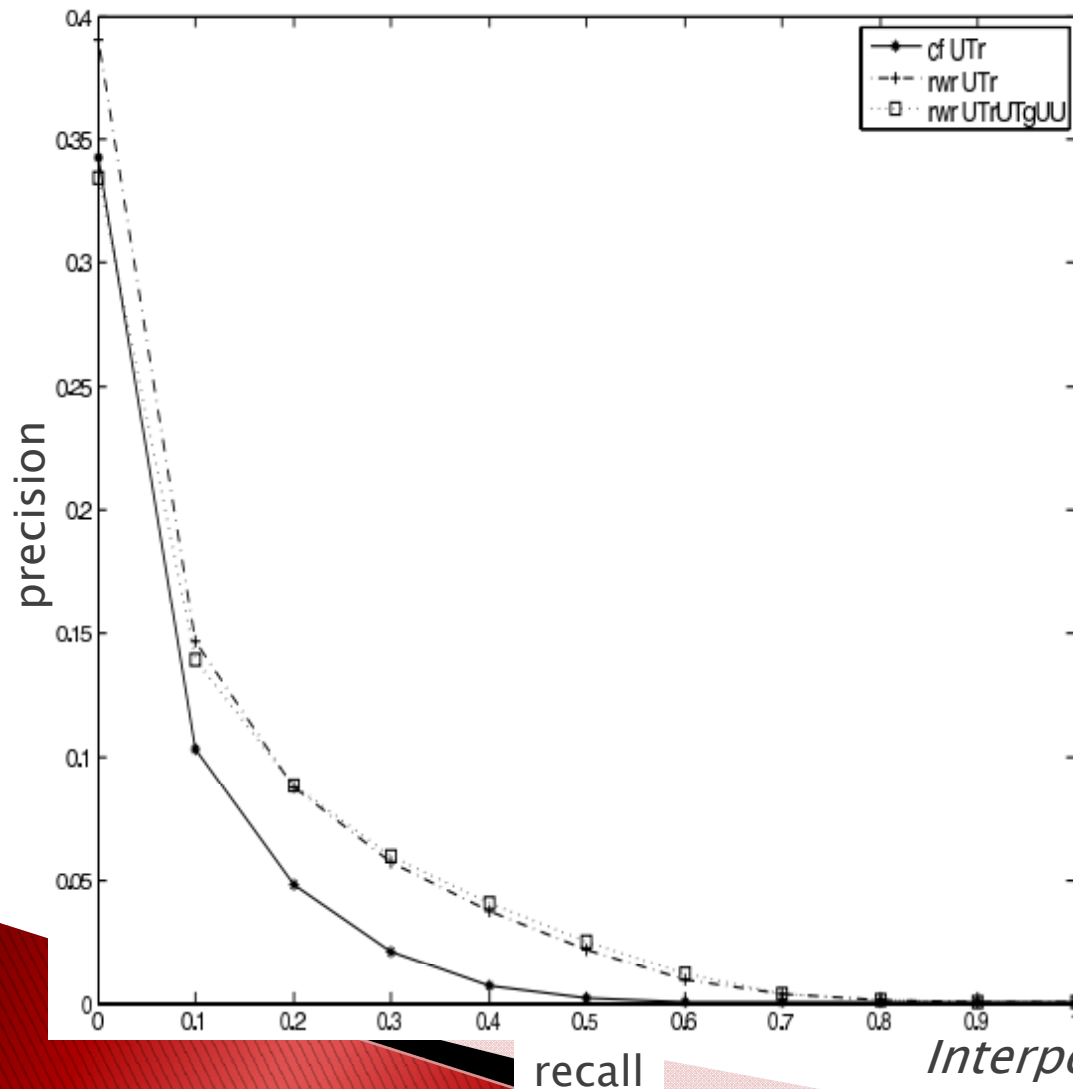
Experiments – RWR (2)

- ▶ Normalise \mathbf{q} so that $\|\mathbf{q}\|_1 = 1$.
- ▶ A series of experiments was performed to determine the restart probability $\alpha = 0.8$.
- ▶ high restart probability α
 - ⇒ Model goes back to the initial \mathbf{q} more frequently.
 - ⇒ It stays in the neighbourhood of u_α .

Experiments – RWR (3)

- ▶ Perform a Random Walk on S and get the
⇒ **stationary probability vector** for u_a .
- ▶ Order the tracks in descending order and select the first 1000.
- ▶ Compare them to the ground truth.

Experiments – RWR – results



cf UTr:
CF method based on **tracks**

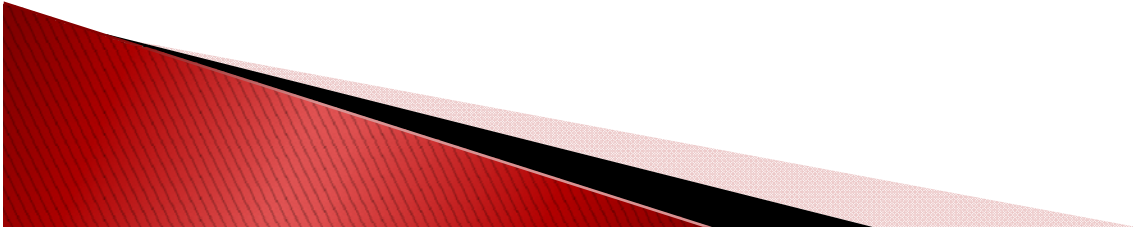
rwr UTr:
Random Walks based on **tracks**

rwr UTrUTgUU:
Random Walks based on the **whole social graph S**

Interpolated Recall-Precision Curves

Experiments – results

- ▶ Random Walks **outperforms** the baseline CF method.
- ▶ Random Walks using S is **not significantly better** than Random Walks using only UTr .



Experiments – results – numbers

- ▶ Let's take a closer look at the numbers:

		P5	P20	P1000	num_rel_ret
cf	UTr	0.1472	0.0934	0.0144	45268
	UTrUU	0.0719	0.0470	0.0104	32862
	UTrUTg	0.1046	0.0698	0.0129	40546
	UTrUTgUU	0.0628	0.0416	0.0102	32106
rwr	UTr	0.1747	0.1229	0.0221	69514
	UTrUU	0.1726	0.1229	0.0222	69742
	UTrUTg	0.1486	0.1145	0.0227*	71404*
	UTrUTgUU	0.1483	0.1139	0.0228	71645

- ▶ Bold numbers indicate **statistical significance** (at $p < 0.001$).

Experiments – results – numbers (2)

		P5	P20	P1000	num_rel_ret
cf	UTr	0.1472	0.0934	0.0144	45268
	UTrUU	0.0719	0.0470	0.0104	32862
	UTrUTg	0.1046	0.0698	0.0129	40546
	UTrUTgUU	0.0628	0.0416	0.0102	32106
rwr	UTr	0.1747	0.1229	0.0221	69514
	UTrUU	0.1726	0.1229	0.0222	69742
	UTrUTg	0.1486	0.1145	0.0227*	71404*
	UTrUTgUU	0.1483	0.1139	0.0228	71645

- ▶ The number of relevant retrieved (num_rel_ret) tracks using UTrUTg is **significantly higher** than that using UTrUU.

Experiments – results

- ▶ In Last.fm users usually spend
 - **more time** looking for music they like and **annotating**
 - **than looking for friends.**
- ⇒ UU matrix is **relatively sparse** compared to the UTg matrix.
 - ⇒ It doesn't influence the results as significantly.
- ▶ This is typical for **special interest** social networks like Last.fm, Flickr or YouTube (unlike Facebook).

Experiments – results – reasons

- ▶ Random Walks outperforms CF because
 - the graph model **captures relationships better**
 - we can recognise more **elaborate patterns** and rules.

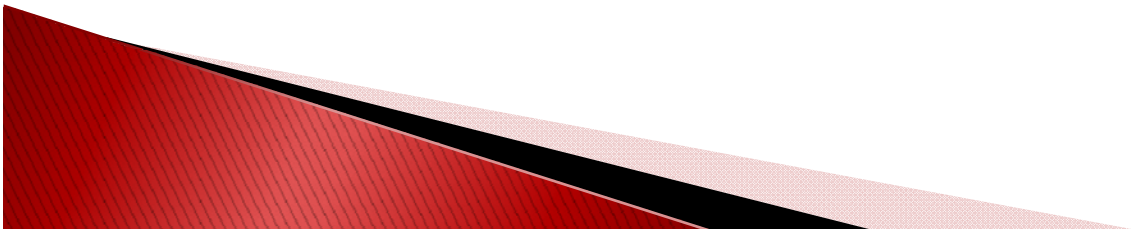
Outline

- ▶ The Network
- ▶ Collaborative Recommendation
- ▶ Collaborative Filtering
- ▶ The Social Graph
- ▶ Random Walks with Restarts
- ▶ Experiments and Results
- ▶ **Conclusion**

Conclusion

- ▶ Random Walk with Restarts doesn't need so much **fine-tuning** of parameters
 - ⇒ it is **more robust** than CF.
- ▶ lack of scalability:
 - On-line response time is not acceptable when applied to large social graphs in situations that need real-time response, like searching.
- ⇒ Use an approximation of RWR instead (with >90% quality).

Thank you for your
attention!



Resources

- ▶ Ioannis Konstas, Vassilios Stathopoulos, Joemon M. Jose: On social networks and collaborative recommendation, SIGIR Conference 2009, pp. 195–202.