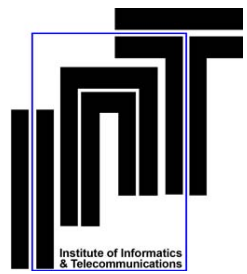
User community discovery: the transition from passive site visitors to active content contributors*

Georgios Paliouras

paliourg@iit.demokritos.gr

Institute of Informatics & Telecommunications,
NCSR "Demokritos", Greece



*Based on: G. Paliouras, "**Discovery of Web user communities and their role in personalization**," *User Modeling and User-Adapted Interaction*, v. 22, n. 1-2, pp. 151-175, 2012.

Goals of this tutorial

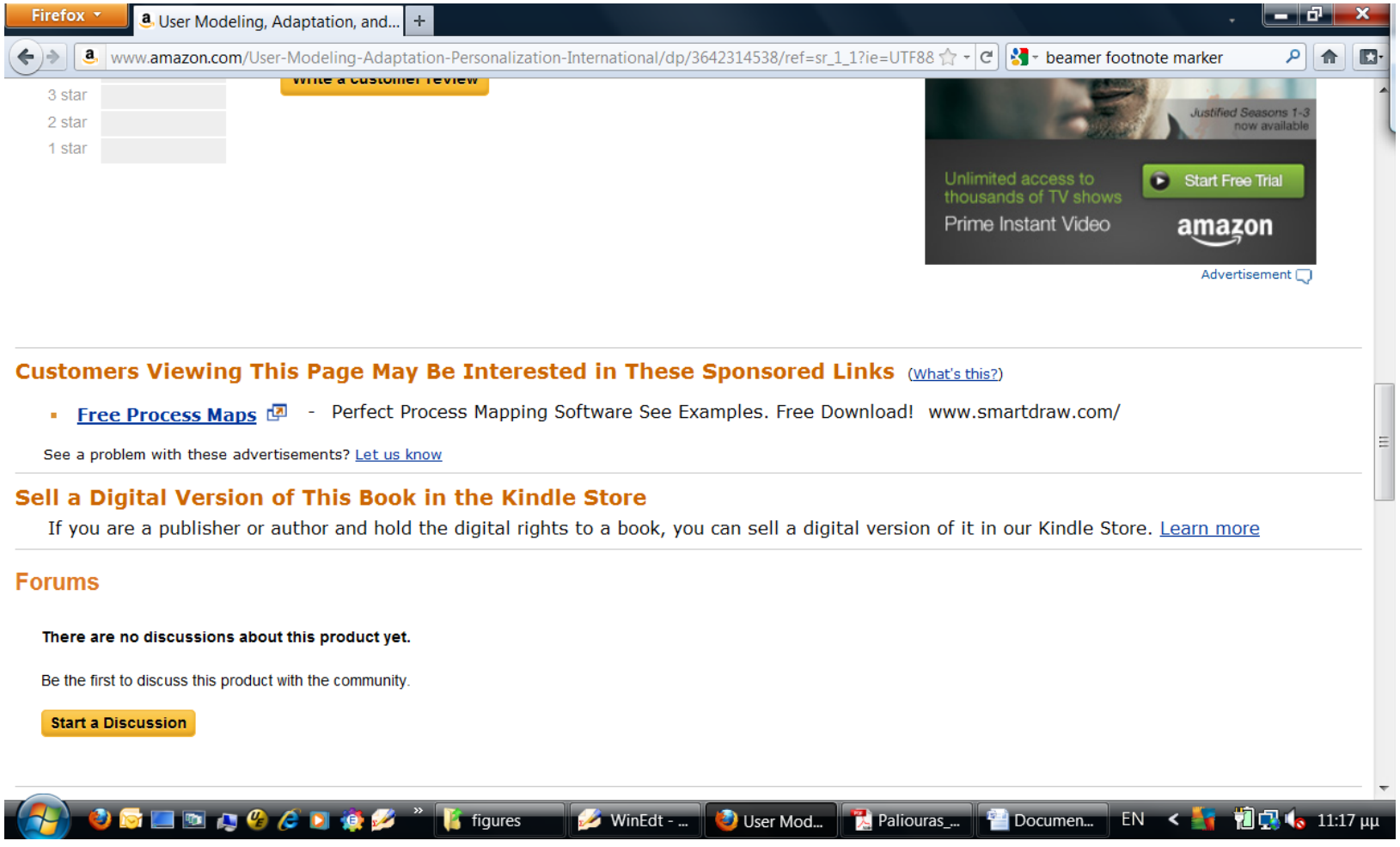
- Discuss what a community is and why it is useful for personalization.
- See what is needed to discover user communities from data.
- Discuss how this differs in the social Web.
- Raise a lot of issues and questions.
- NOT present methods and algorithms.

Why are communities useful?

Product recommendation . . .

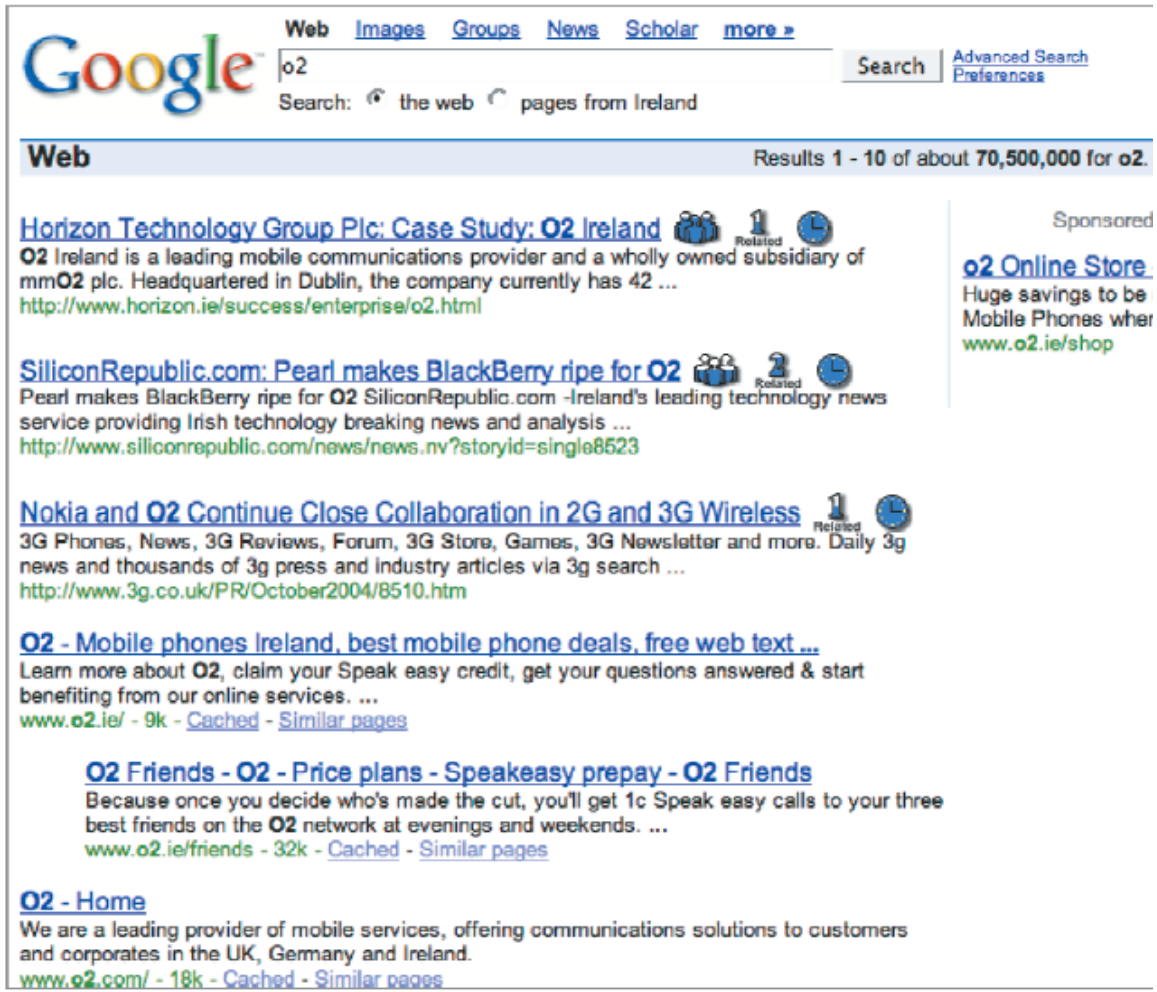
Why are communities useful?

Targeted Advertisement . . .



Why are communities useful?

Personalized search . . .



Why are communities useful?

Many other interesting applications, including:

- Personalized guidance through museums.
- Helping students, through collaboration communities.
- Style and fashion recommenders.

Community discovery and user modeling

- In the early '90s, we realized that *who* somebody is associated to is as important for personalization as *what* this person chooses to view or buy.
- At the same time recording user activity on the Web became easier and large datasets were generated.
- Web usage mining was born to make use of these data.
- Identifying user associations was an obvious choice . . . leading to the discovery of user communities.

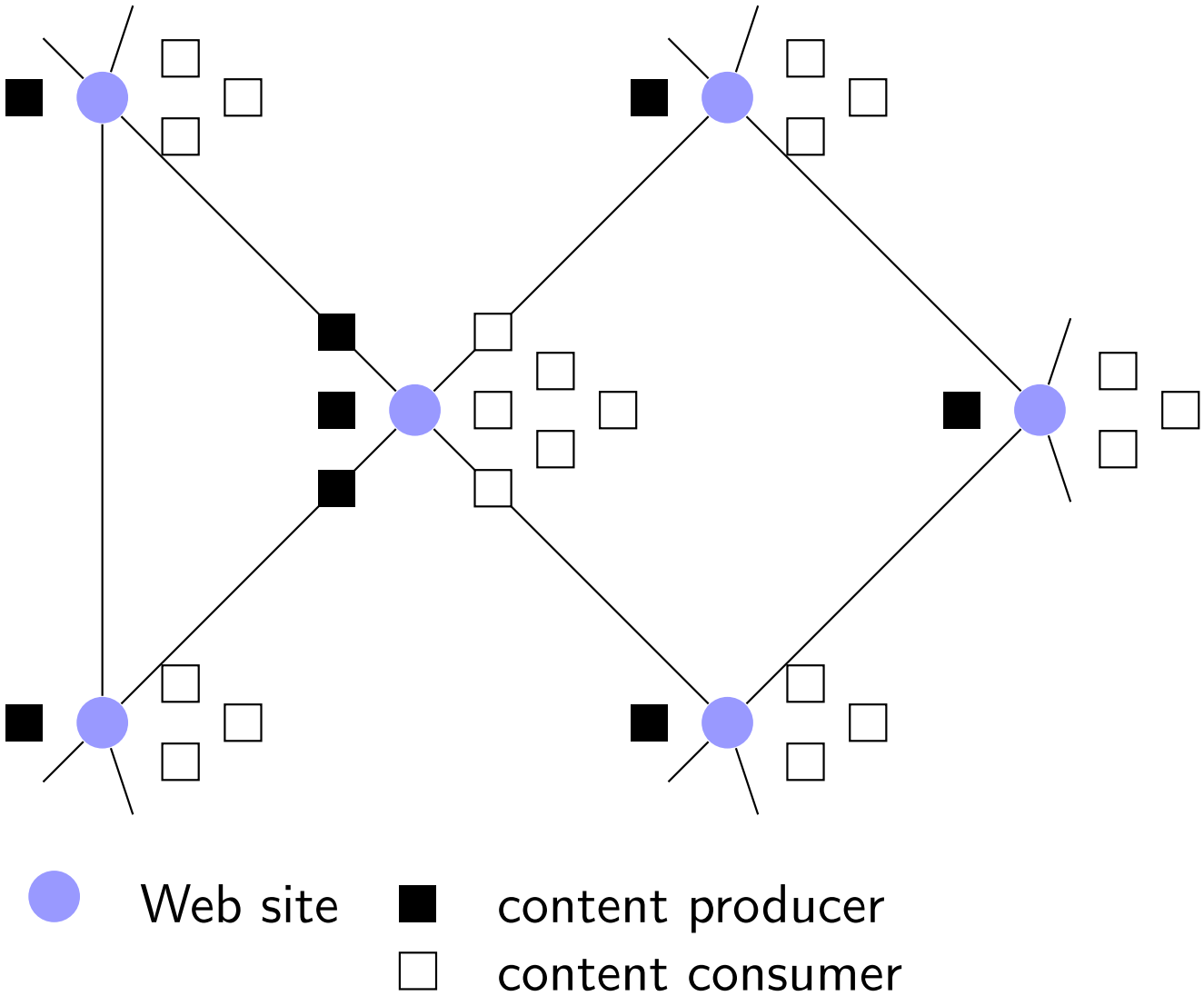
From the Web to the Social Web

But the Web is changing:

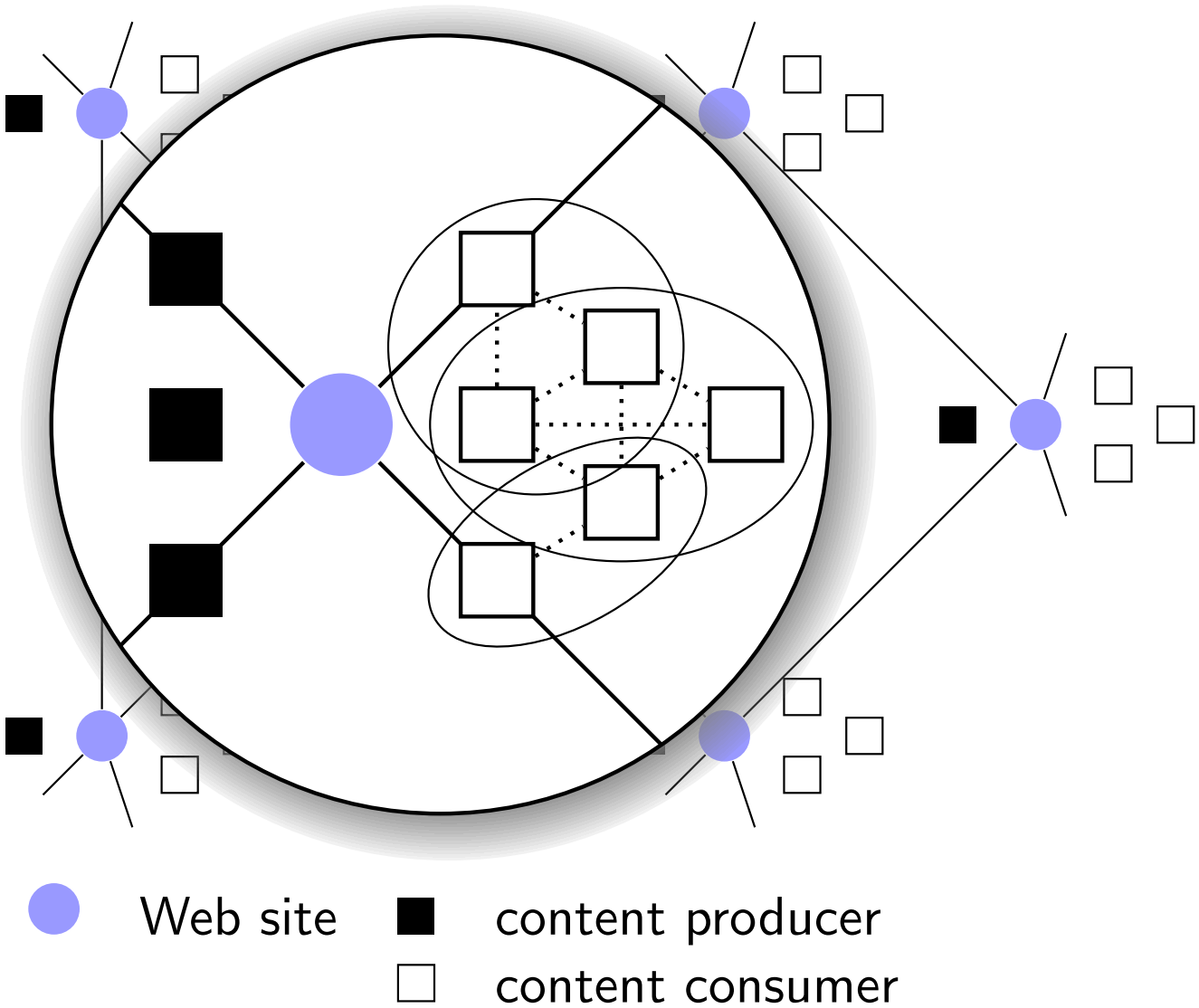
- Users are not content consumers, but active producers.
- They have a multitude of ways of expressing themselves: Making someone a friend or liking one's post is a more important social act than the selection of Web content was.
- Real communities are moving to the Web and new ones are being constructed.
- Discovering social relations and their motives is becoming increasingly important.
- Socialization itself has become the object of recommendation, i.e. friend recommendation.
- What is the role and nature of community discovery in this new environment?

Outline

A sketch of the old Web



Web usage communities within a site

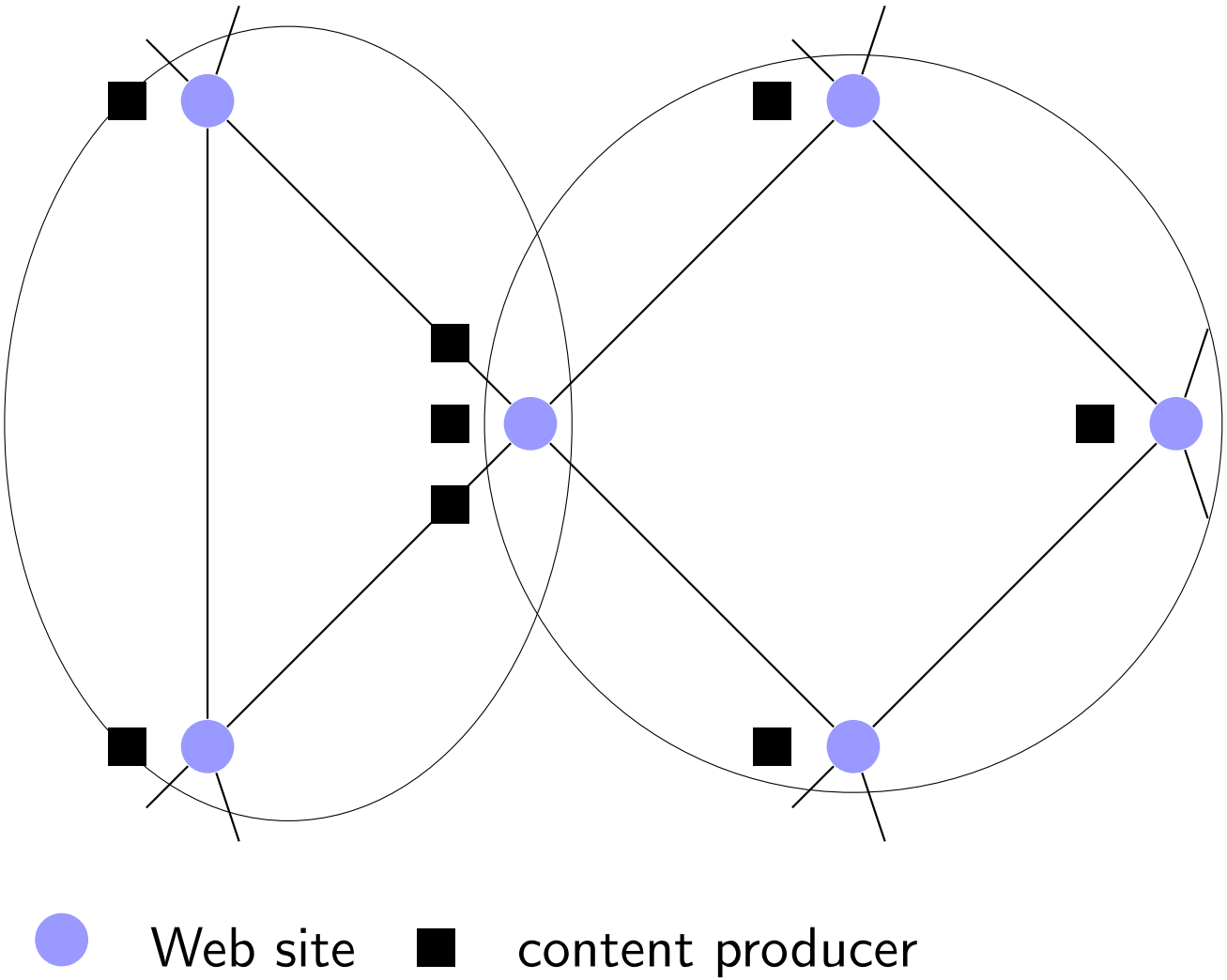


Web usage communities within a site

- User modeling focused on content consumers, site visitors:
  - Many more than producers.
  - Their activity is logged.
  - They are the customers.
- *Web user communities* segment users into groups, according to their preferences.
- Web usage mining focused on the discovery of such communities.

Web usage communities within a site

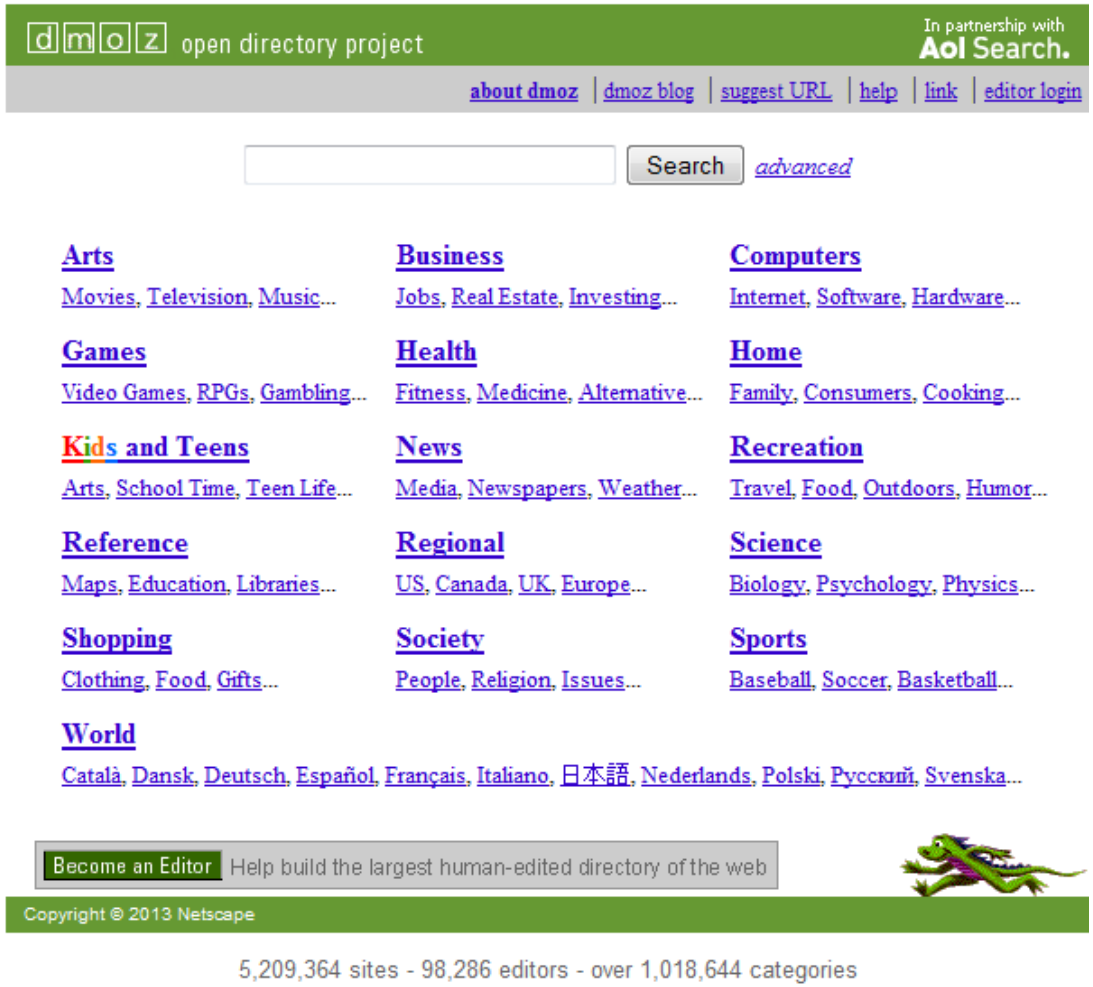
Web communities on the Web graph



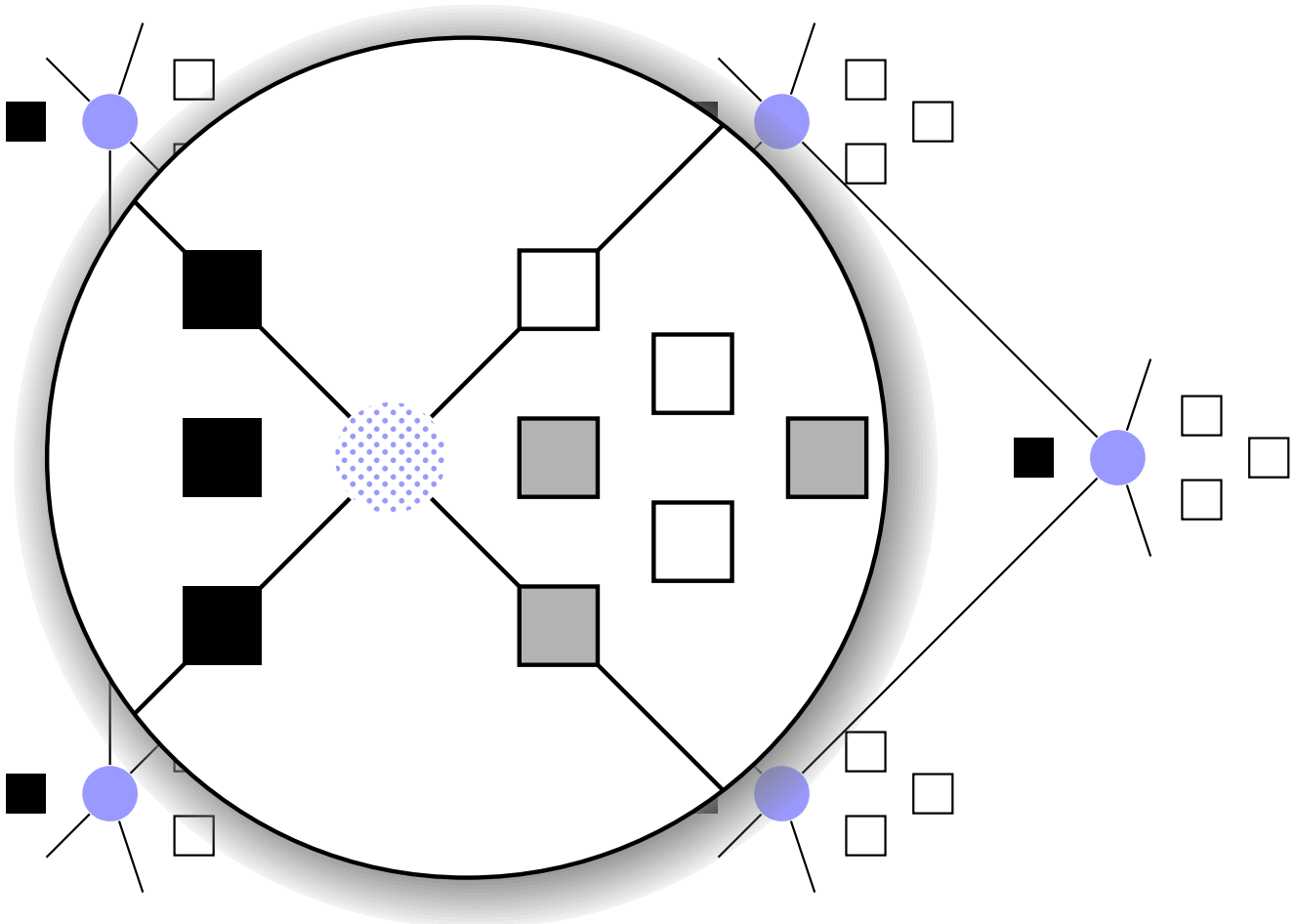
Web communities on the Web graph

- Not communities of users, but communities of sites or pages.
- Indirectly, communities of content producers.
- Web structure mining focused on the discovery of such communities.
- They represent usually themes or topics.
- Of interest to services that organize or index the Web:
  - Search engines (theme disambiguation).
  - Web directories (semi-automated expansion).
  - Web portals (focused crawling).

Web communities on the Web graph



Web-based community as a single site



Web site

Web-based community site

content producer

content consumer

active Web user

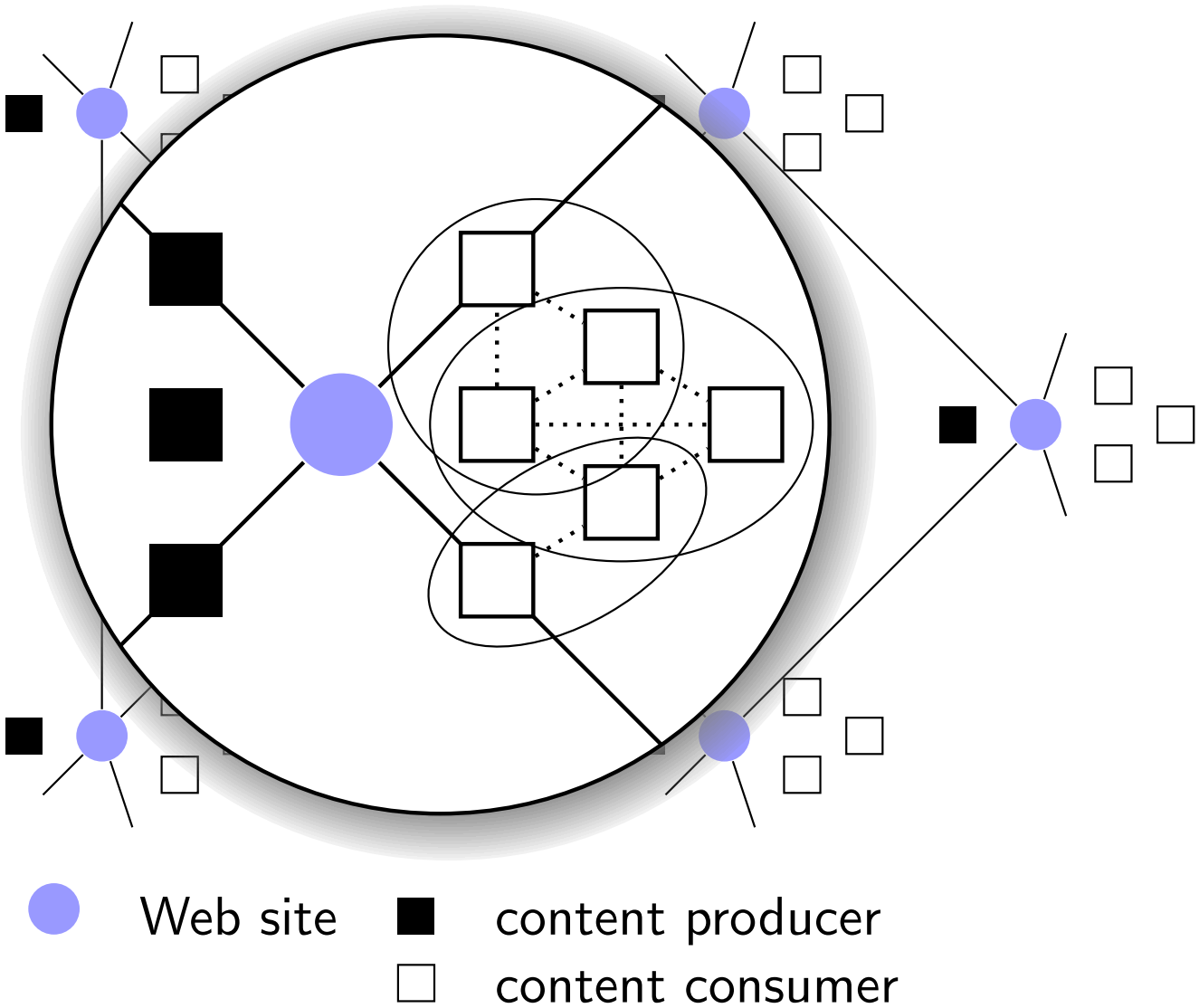
Web-based community as a single site

- Web sites where users explicitly join a community.
- Support and extend real-life communities.
- Typically, either local (e.g. University campus) or interest-driven (e.g. video games).
- Also known as *online* or *virtual communities*.
- Users start producing content, in addition to consuming it.
- Early steps of social networks.

Web-based community as a single site

Outline

Web usage communities



Usage data

- Users don't register for communities.
- When do users X and Y belong to the same community?
  - Typically: common interest on products and services.
  - Alternative: common level of knowledge, frequent collaboration, etc.
- What hints towards common interest?
  - Common views or purchases.
  - Similar ratings.

Web server logs

127.1.2.2	[10/Sep/2010:21:15:05]	"GET /index.html HTTP/1.1"	200	1043
127.1.2.2	[10/Sep/2010:21:15:06]	"GET /main.html HTTP/1.1"	200	954
127.1.2.2	[10/Sep/2010:21:15:07]	"GET /books.html HTTP/1.1"	200	837
127.1.2.2	[10/Sep/2010:21:15:08]	"GET /books/p13.html HTTP/1.1"	200	568
204.0.0.1	[10/Sep/2010:21:15:10]	"GET /index.html HTTP/1.1"	200	1043
204.0.0.1	[10/Sep/2010:21:15:11]	"GET /cars.html HTTP/1.1"	200	1235
127.1.2.2	[10/Sep/2010:21:15:12]	"GET /books/p14.html HTTP/1.1"	200	2037
204.0.0.1	[10/Sep/2010:21:15:12]	"GET /cars/p38.html HTTP/1.1"	200	8923
204.0.0.1	[10/Sep/2010:21:15:15]	"GET /cars/p97.html HTTP/1.1"	200	9478
127.1.2.2	[10/Sep/2010:21:15:17]	"GET /books/p15.html HTTP/1.1"	200	4056
204.0.0.1	[10/Sep/2010:21:15:20]	"GET /extra/p29.html HTTP/1.1"	200	3459

Web server logs

```
127.1.2.2  [10/Sep/2010:21:15:05]  "GET /index.html HTTP/1.1"        200  1043
127.1.2.2  [10/Sep/2010:21:15:06]  "GET /main.html HTTP/1.1"         200  954
127.1.2.2  [10/Sep/2010:21:15:07]  "GET /books.html HTTP/1.1"        200  837
127.1.2.2  [10/Sep/2010:21:15:08]  "GET /books/p13.html HTTP/1.1"    200  568

127.1.2.2  [10/Sep/2010:21:15:12]  "GET /books/p14.html HTTP/1.1"    200  2037

127.1.2.2  [10/Sep/2010:21:15:17]  "GET /books/p15.html HTTP/1.1"    200  4056
```

Web server logs

```
204.0.0.1  [10/Sep/2010:21:15:10]  "GET /index.html HTTP/1.1"       200  1043
204.0.0.1  [10/Sep/2010:21:15:11]  "GET /cars.html HTTP/1.1"        200  1235

204.0.0.1  [10/Sep/2010:21:15:12]  "GET /cars/p38.html HTTP/1.1"    200  8923
204.0.0.1  [10/Sep/2010:21:15:15]  "GET /cars/p97.html HTTP/1.1"    200  9478

204.0.0.1  [10/Sep/2010:21:15:20]  "GET /extra/p29.html HTTP/1.1"   200  3459
```

Web server logs

127.1.2.2	[10/Sep/2010:21:15:05]	"GET /index.html HTTP/1.1"	200	1043
127.1.2.2	[10/Sep/2010:21:15:06]	"GET /main.html HTTP/1.1"	200	954
127.1.2.2	[10/Sep/2010:21:15:07]	"GET /books.html HTTP/1.1"	200	837
127.1.2.2	[10/Sep/2010:21:15:08]	"GET /books/p13.html HTTP/1.1"	200	568
204.0.0.1	[10/Sep/2010:21:15:10]	"GET /index.html HTTP/1.1"	200	1043
204.0.0.1	[10/Sep/2010:21:15:11]	"GET /cars.html HTTP/1.1"	200	1235
127.1.2.2	[10/Sep/2010:21:15:12]	"GET /books/p14.html HTTP/1.1"	200	2037
204.0.0.1	[10/Sep/2010:21:15:12]	"GET /cars/p38.html HTTP/1.1"	200	8923
204.0.0.1	[10/Sep/2010:21:15:15]	"GET /cars/p97.html HTTP/1.1"	200	9478
127.1.2.2	[10/Sep/2010:21:15:17]	"GET /books/p15.html HTTP/1.1"	200	4056
204.0.0.1	[10/Sep/2010:21:15:20]	"GET /extra/p29.html HTTP/1.1"	200	3459

User identification

- Issue: Correspondence of users to IP addresses.
- Attempted solutions:
  - Registered users.
  - Extended log (user id, referring page, browser, etc.)
  - Cookies, Javascript, etc.
- Open issues:
  - Privacy?
  - Changing Web technology (e.g. HTML5)?

User sessions

```
127.1.2.2  [10/Sep/2010:21:15:05]  "GET /index.html HTTP/1.1"        200  1043
127.1.2.2  [10/Sep/2010:21:15:06]  "GET /main.html HTTP/1.1"         200  954
127.1.2.2  [10/Sep/2010:21:15:07]  "GET /books.html HTTP/1.1"        200  837
127.1.2.2  [10/Sep/2010:21:15:08]  "GET /books/p13.html HTTP/1.1"    200  568

127.1.2.2  [10/Sep/2010:21:15:12]  "GET /books/p14.html HTTP/1.1"    200  2037

127.1.2.2  [10/Sep/2010:21:15:17]  "GET /books/p15.html HTTP/1.1"    200  4056
```

User sessions

- Sessions associate selected items.
- Essential when users not registered.
- Non-obvious when a session stops (time-out threshold).
- Longer user sessions more informative, but typically short.

User profiles

- Statistical aggregate of user history.
- Measure user interest on an item:
  - Frequency of selection of the item.
  - Viewing time of the item.
  - Frequency of purchase of the item.
  - Explicit rating of the item.
- Can be short-term or long-term model of the user.
- Can incorporate forgetting mechanisms.
- May be limited to session profiles if no registered users.
- Can generalize to item categories, if a taxonomy is provided.

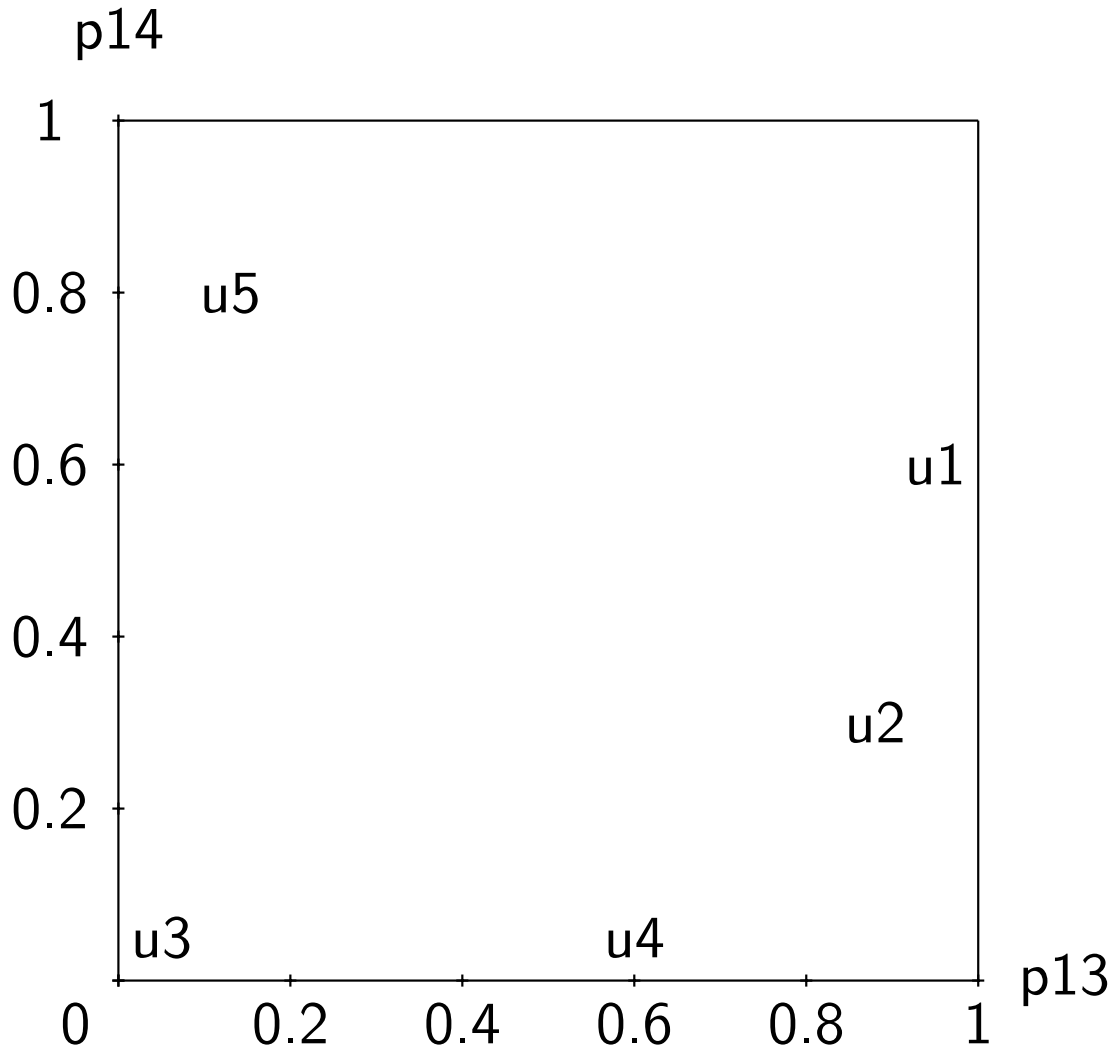
User-item matrix

	p13	p14	p15	p29	p38
u1	1.0	0.6	0.8	0.0	0.0
u2	0.9	0.3	0.0	0.8	0.0
u3	0.0	0.0	0.0	0.8	0.9
u4	0.6	0.0	0.0	1.0	0.0
u5	0.1	0.8	1.0	0.0	0.0

Binarized user profiles (interest threshold 0.5)

	p13	p14	p15	p29	p38
u1	1	1	1	0	0
u2	1	0	0	1	0
u3	0	0	0	1	1
u4	1	0	0	1	0
u5	0	1	1	0	0

User profiles in the item space



User similarity

- Cosine similarity:

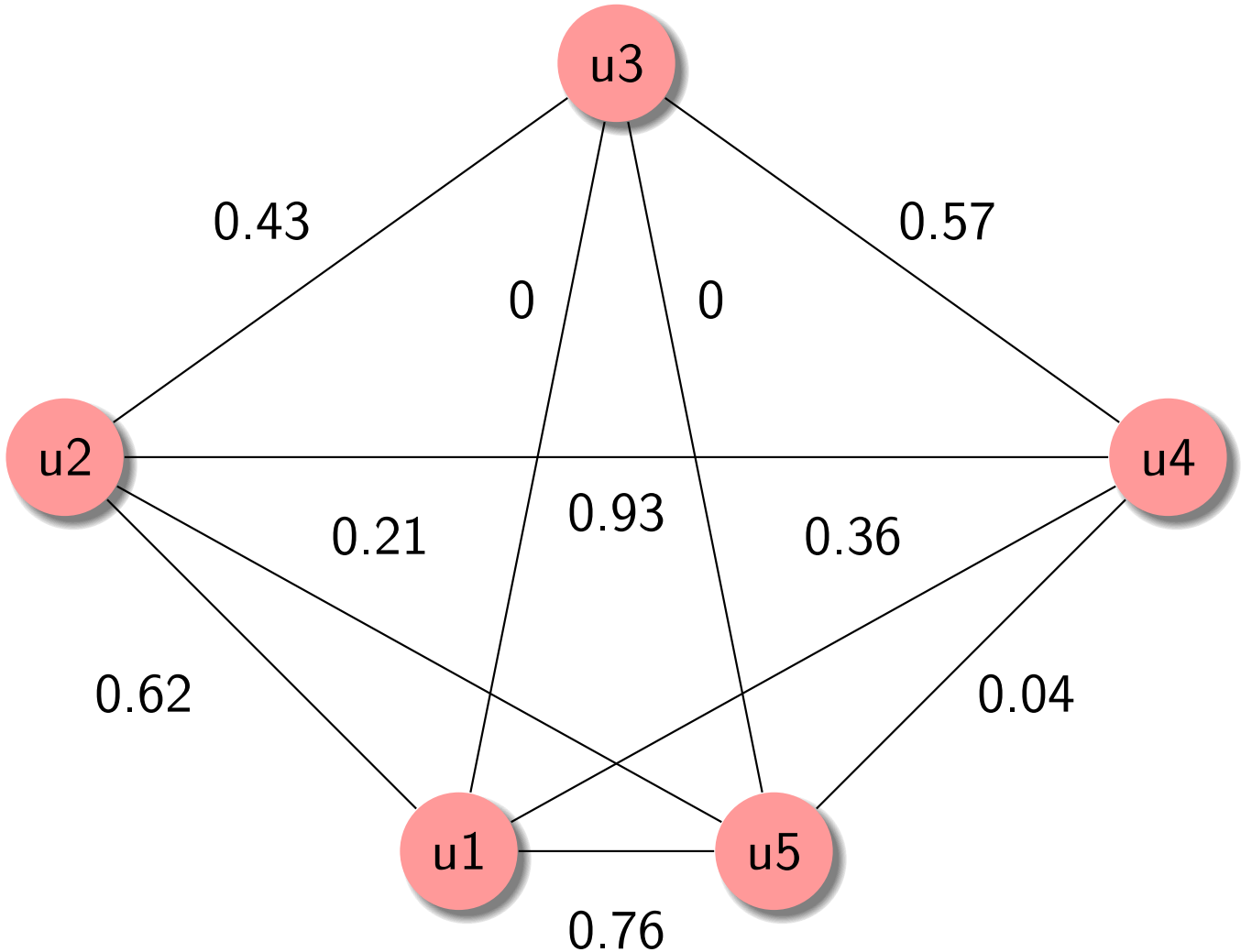
$$R(\mathbf{u}_i, \mathbf{u}_j) = \frac{\sum_{k=1}^{T} u_{ik} \times u_{jk}}{\sqrt{\sum_{k=1}^{T} u_{ik}^2} \times \sqrt{\sum_{k=1}^{T} u_{jk}^2}}$$

- Pearson correlation:

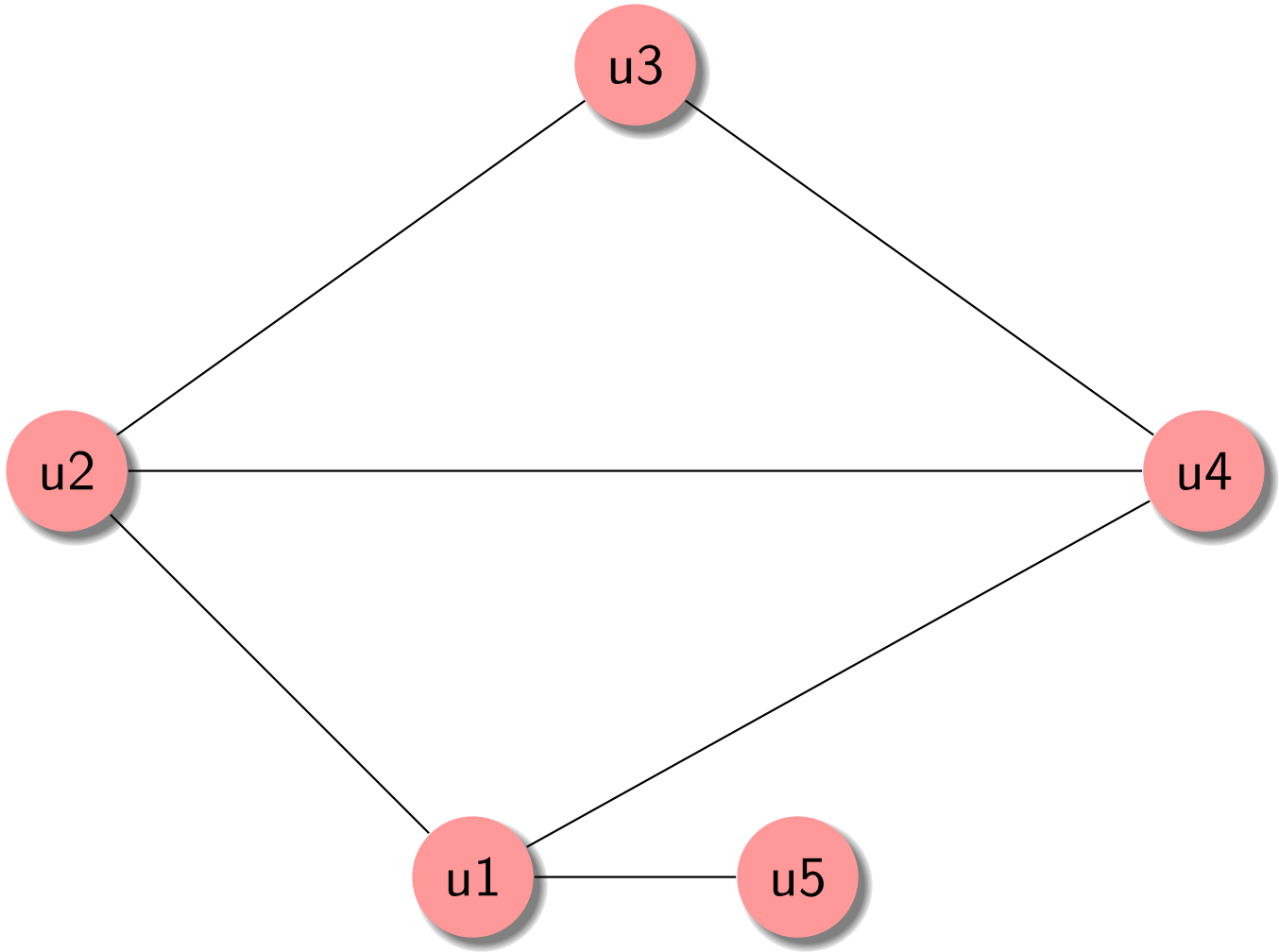
$$R(\mathbf{u}_i, \mathbf{u}_j) = \frac{\sum_{k=1}^{T} (u_{ik} - \overline{\mathbf{u}_i}) \times (u_{jk} - \overline{\mathbf{u}_j})}{\sqrt{\sum_{k=1}^{T} (u_{ik} - \overline{\mathbf{u}_i})^2} \times \sqrt{\sum_{k=1}^{T} (u_{jk} - \overline{\mathbf{u}_j})^2}}$$

- Many more, some suitable for numeric and other for binary data.
- Used to create user-user similarity matrix and graph.

Weighted user graph



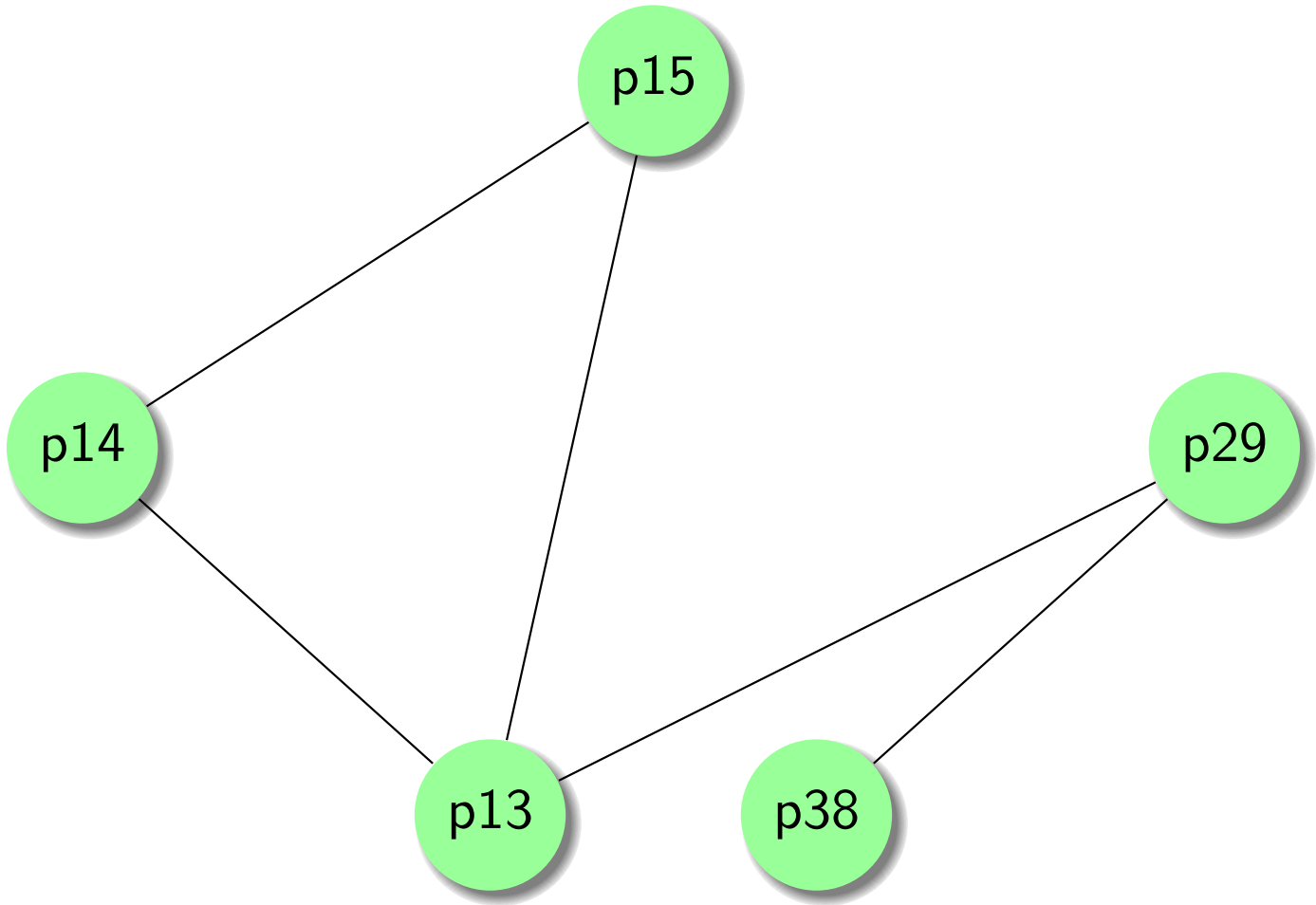
Unweighted user graph (threshold 0.3)



Item similarity

- Based on usage, not content or prior knowledge!
- Particularly important when no registered users and short sessions.
- Calculating usage-based item similarity:
  - Transpose user-item matrix.
  - Calculate similarity in multi-dimensional user space.
  - Equivalent measures to user similarity.

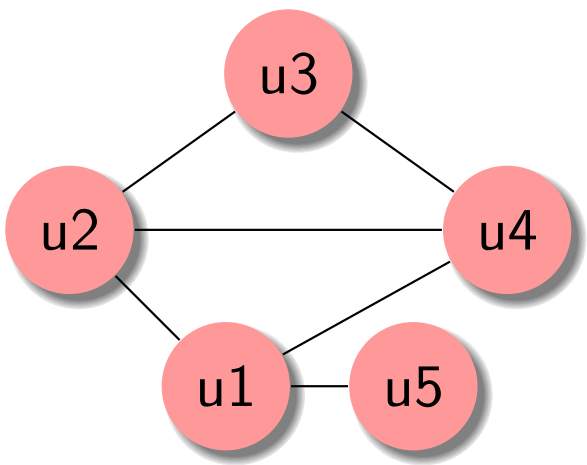
Unweighted graph of items



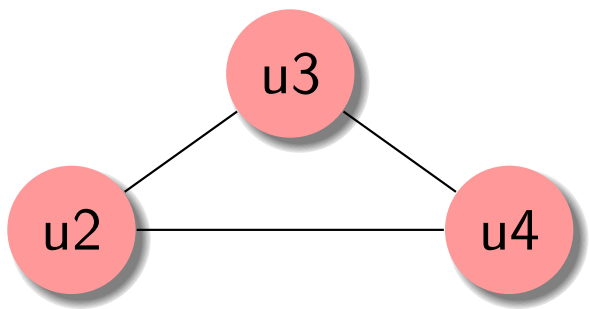
Communities and community models

- Typically communities are densely-connected subgraphs of the user graph.
- Clustering and graph partitioning methods are useful for finding them.
- Once identified a community can be characterized in terms of user interests, i.e. items.
- The description of a community is the *community model*.

User Graph



Community

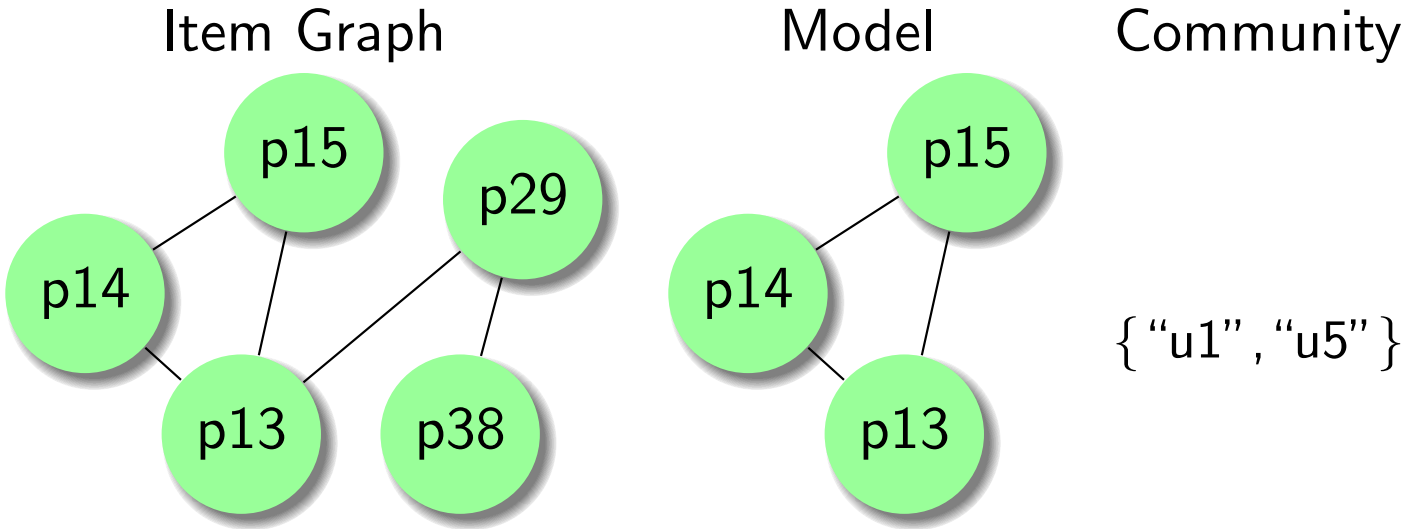


Model

{“p29”, “p38”}

Communities and community models

- What if subgraphs are sought on the item graph?
- What are the "models" of item subgraphs? User communities?
- Are item subgraphs valid community models?



Communities and community models

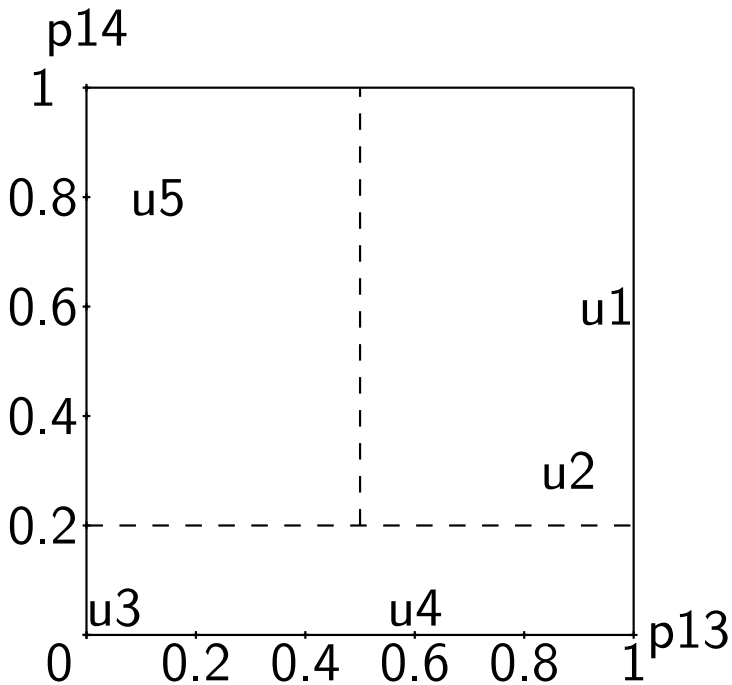
Item-based community models are very useful in practice . . .

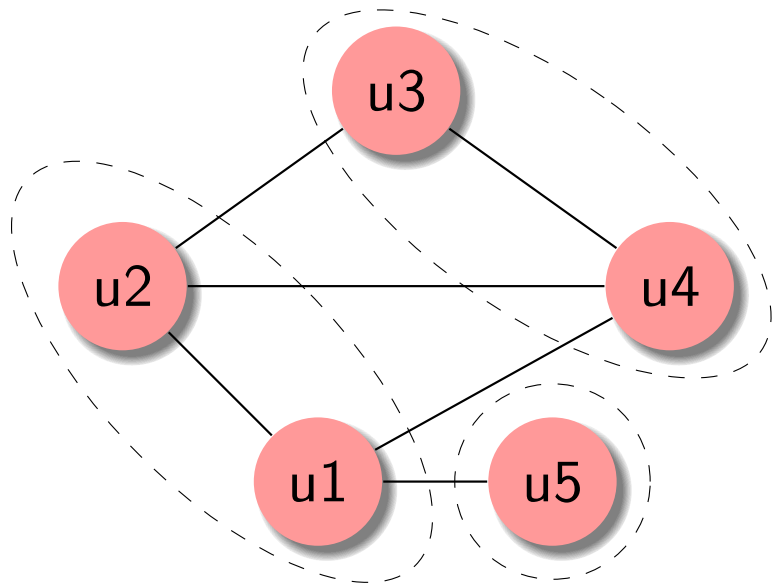
Discovering communities - objective

- Aim at a set of user clusters $\mathbf{U} = \bigcup_{j=1..K} U_k$ that are
  - maximally cohesive, i.e. maximize $R(\mathbf{u}_i, \mathbf{u}_j), u_i, u_j \in U_k$, within each community.
  - minimally redundant, i.e. minimize $R(\mathbf{u}_i, \mathbf{u}_j), u_i \in U_k, u_j \in U_l, k \neq l$, across communities.
- That's what all traditional clustering methods do.

Discovering communities - hard partitioning

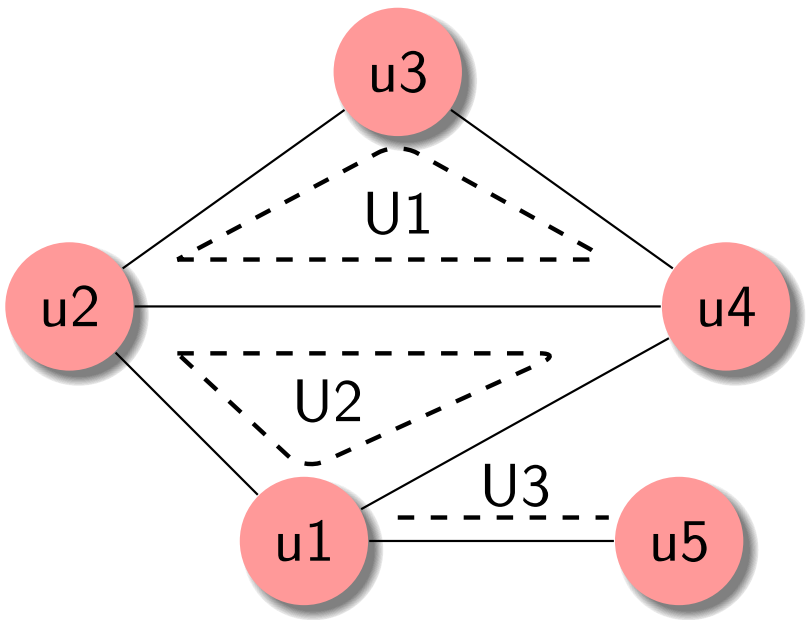
- Users as objects, described by attributes (item preferences).
- Common clustering methods can and have been used, including k-means, density-based methods, etc.
- Partitioning of the user graph, e.g. based on modularity criteria, leads to similar clustering.
- Have also been applied (less often) to the item graph.

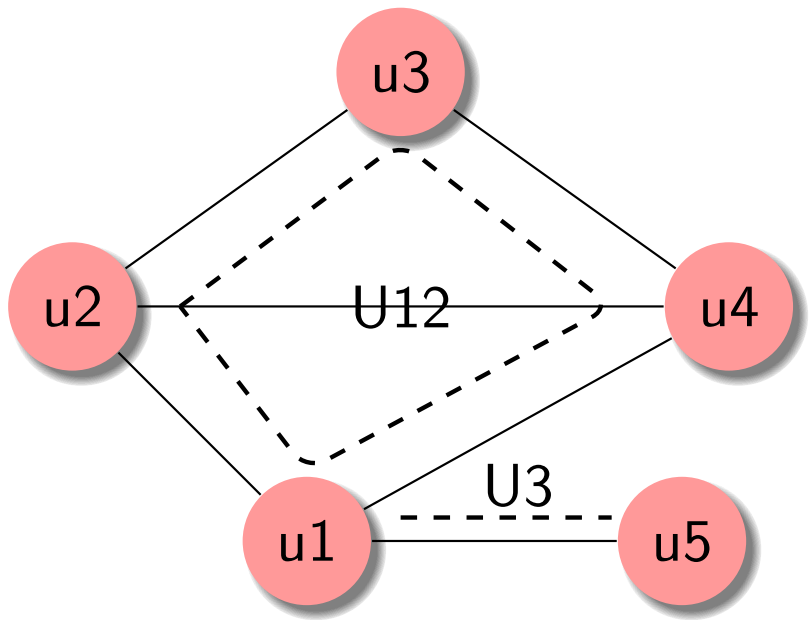




Discovering overlapping communities

- Users communities and their models naturally overlapping.
- Users and items belong in many communities.
- Graph analysis methods produce overlapping communities, e.g. cliques, connected components.
- Common on item graphs (resemble frequent itemsets).
- Have been used on user graphs, too.
- Hard to control overlap. Percolation of cliques.





Discovering communities - probabilistic modeling

- Item clustering can be viewed as dimensionality reduction.
- Typically soft, e.g. Principle Component Analysis.
- Probabilistic variants particularly interesting, e.g. Probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation.
- Can be applied to the user-item matrix.
- Estimate latent dimensions D, based on $Pr(p_j|d_k), Pr(u_i|d_k)$.

	d1	d2	d3
u1	0.10	0.40	0.50
u2	0.45	0.40	0.05
u3	0.85	0.10	0.05
u4	0.45	0.40	0.05
u5	0.05	0.10	0.85

	d1	d2	d3
p13	0.30	0.60	0.10
p14	0.10	0.20	0.70
p15	0.10	0.30	0.60
p29	0.70	0.25	0.05
p38	0.9	0.05	0.05

Discovering communities - probabilistic modeling

- $Pr(u_i|d_k)$ soft clusters users u_i into communities.
- $Pr(p_j|d_k)$ soft clusters items p_j into community models.
- Each user and each item belongs to each community to some degree (aka community indicator).
- Latent dimensions model overlapping communities.
- Discrete communities and models can be formed by MAP (partitioning) or thresholding (overlapping).

	d1	d2	d3
u1	0	0	1
u2	1	0	0
u3	1	0	0
u4	1	0	0
u5	0	0	1

	d1	d2	d3
p13	0	1	0
p14	0	0	1
p15	0	0	1
p29	1	0	0
p38	1	0	0

Discovering communities - probabilistic modeling

- $Pr(u_i|d_k)$ soft clusters users u_i into communities.
- $Pr(p_j|d_k)$ soft clusters items p_j into community models.
- Each user and each item belongs to each community to some degree (aka community indicator).
- Latent dimensions model overlapping communities.
- Discrete communities and models can be formed by MAP (partitioning) or thresholding (overlapping).

	d1	d2	d3
u1	0	1	1
u2	1	1	0
u3	1	0	0
u4	1	1	0
u5	0	0	1

	d1	d2	d3
p13	1	1	0
p14	0	0	1
p15	0	0	1
p29	1	0	0
p38	1	0	0

Discovering communities - back to individuals

- Given the latent models, construct a new model for a user u_i.
- Differs from user profile, incorporates community knowledge.
- Various ways to do this:

Discovering communities - back to individuals

- Given the latent models, construct a new model for a user u_i.
- Differs from user profile, incorporates community knowledge.
- Various ways to do this:
  - Choose the model of the "closest" community.

	d1	d2	d3
u1	0.10	0.40	0.50
u2	0.45	0.40	0.05
u3	0.85	0.10	0.05
u4	0.45	0.40	0.05
u5	0.05	0.10	0.85

	d1	d2	d3
p13	0.30	0.60	0.10
p14	0.10	0.20	0.70
p15	0.10	0.30	0.60
p29	0.70	0.25	0.05
p38	0.9	0.05	0.05

Discovering communities - back to individuals

- Given the latent models, construct a new model for a user u_i.
- Differs from user profile, incorporates community knowledge.
- Various ways to do this:
  - Choose the model of the "closest" community.
  - Calculate a weighted mixture of all community models.

	p13	p14	p15	p29	p38
u1	0.32	0.44	0.43	0.19	0.13
u2	0.38	0.16	0.19	0.41	0.42
u3	0.30	0.13	0.13	0.64	0.81
u4	0.38	0.16	0.19	0.41	0.42
u5	0.16	0.62	0.54	0.10	0.09

Discovering communities - back to individuals

- Given the latent models, construct a new model for a user u_i.
- Differs from user profile, incorporates community knowledge.
- Various ways to do this:
  - Choose the model of the "closest" community.
  - Calculate a weighted mixture of all community models.
  - Calculate a weighted mixture of the k closest models.

	d1	d2	d3
u1	0.10	0.40	0.50
u2	0.45	0.40	0.05
u3	0.85	0.10	0.05
u4	0.45	0.40	0.05
u5	0.05	0.10	0.85

	p13	p14	p15	p29	p38
u1	0.32	0.47	0.46	0.13	0.05
u2	0.44	0.14	0.19	0.48	0.50
u3	0.31	0.10	0.11	0.67	0.85
u4	0.44	0.14	0.19	0.48	0.50
u5	0.15	0.64	0.56	0.07	0.05

Note: Popular k-nearest neighbor is a special case of the third approach, where each community consists of a single user.

Discovering communities - extended itemsets

- Assumed so far that items are plain labels.
- In practice items can come with rich information:
  - Explicit features, e.g. the actors of a movie.
  - Features extracted from content, e.g. from text or multimedia.
  - Meta-data annotations, esp. for multimedia.
  - Taxonomies and ontologies, e.g. music genres or relations among book authors.
- Including such information in the community discovery process is essential, but non-trivial.
- Hint: Discussion to follow about communities in the social Web ...

Summary of section

- Usage data are recorded but their use is non-trivial.
- Communities are sets of users, their models are sets of items.
- Communities are naturally overlapping.
- We can discover interesting models directly from item graphs.
- Probabilistic methods model communities as latent variables.
- Personal models can be constructed from communities.
- Items are not just labels.

Outline

What is the social Web

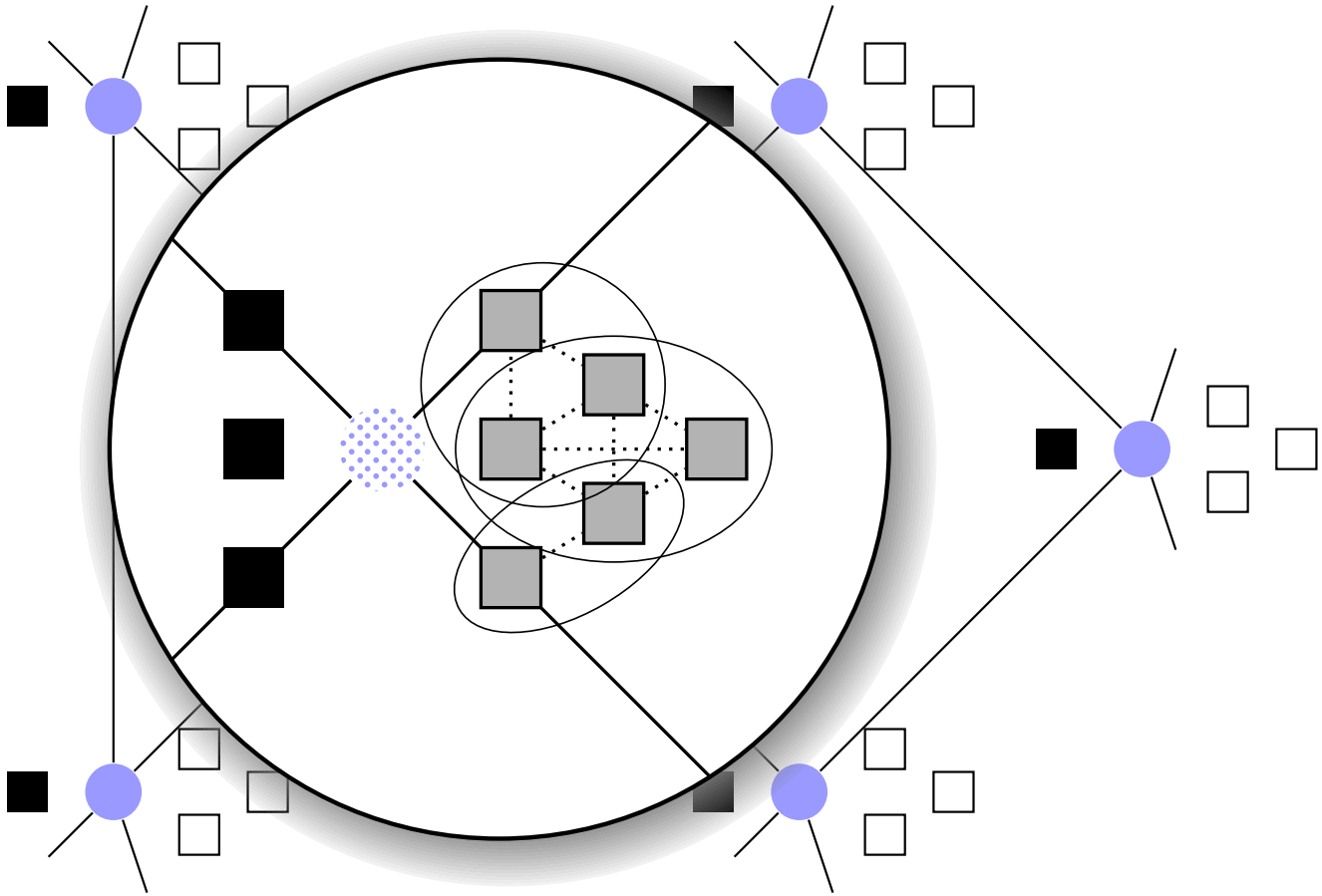
- Social media, aka user-generated content: blogs, wikis, etc.
- Social networks, i.e. Web sites that allow people to connect to each other. Similar to Web-based communities, but also different (scale, scope, decentralization, etc.)
- Social networks supporting user-generated content.

- Technologies enabling the above (Web 2.0, cloud storage and computation, mobile Web, etc.)
- Innovative Web applications.
- Revolutionary mixture of all of these.

Social aspects of the social Web

- Motivated people to join the Web.
- Motivated people to start using a computer.
- Motivated people to contribute.
- People move, create, re-establish social relations on the Web.
- The Web has become part of everyday routine.
- The role of Web communities has changed, e.g. communities reveal different types of relation among users (my schoolmates, my colleagues, etc.)

Communities of active users



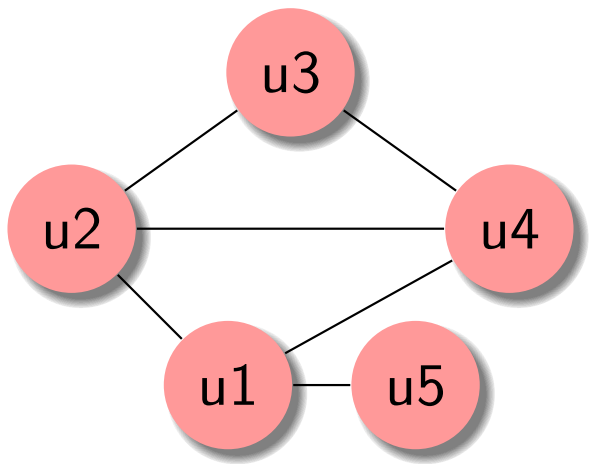
Web site
Social Web site
content producer
content consumer
active Web user

Discovering active user communities

- Users of social Web sites differ from traditional Web users:
  - They belong to explicit communities.
  - They provide rich personal information.
  - They collaborate and interact in many ways.
  - They generate content that characterizes them.
  - Their content is used by others.
- User-item matrices are not a sufficient representation.
- The network is richer, multi-modal (many types of node) and multi-relational (many types of relation).
- How can we discover communities?

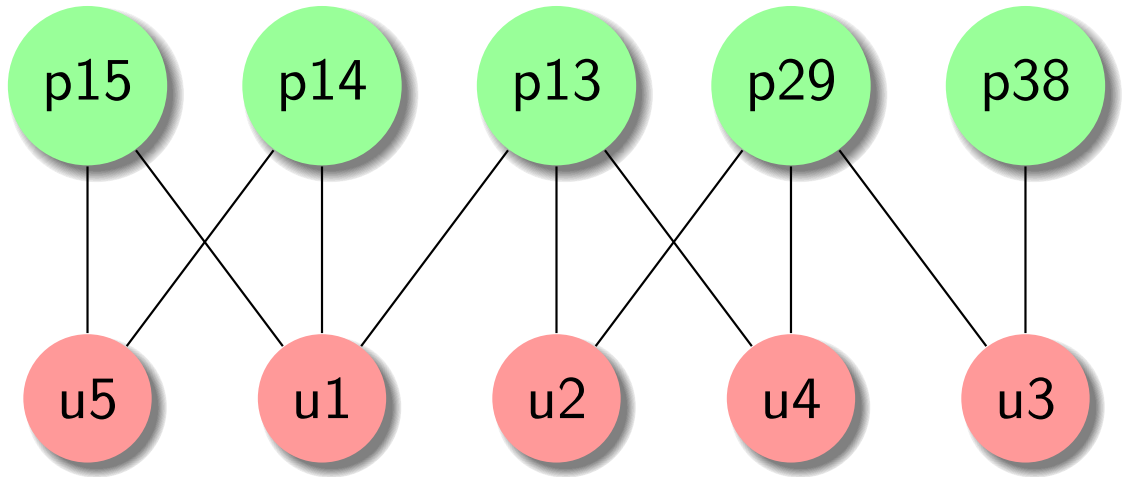
A multi-modal, multi-relational network

Our simple user graph . . .

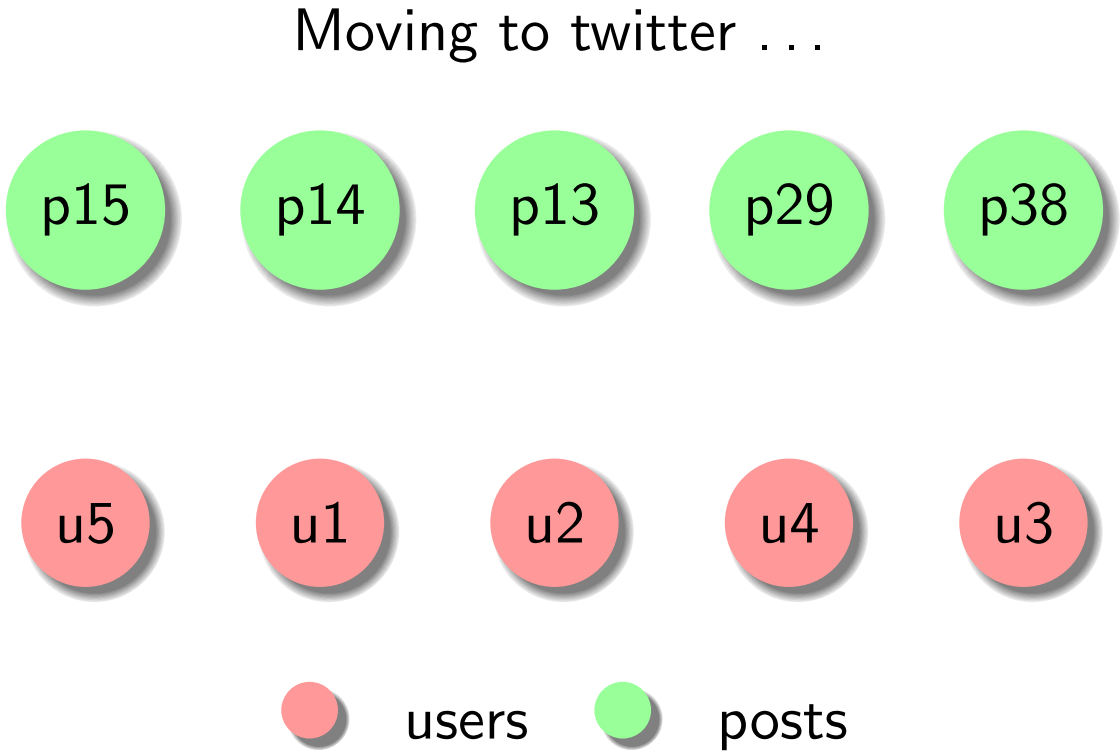


A multi-modal, multi-relational network

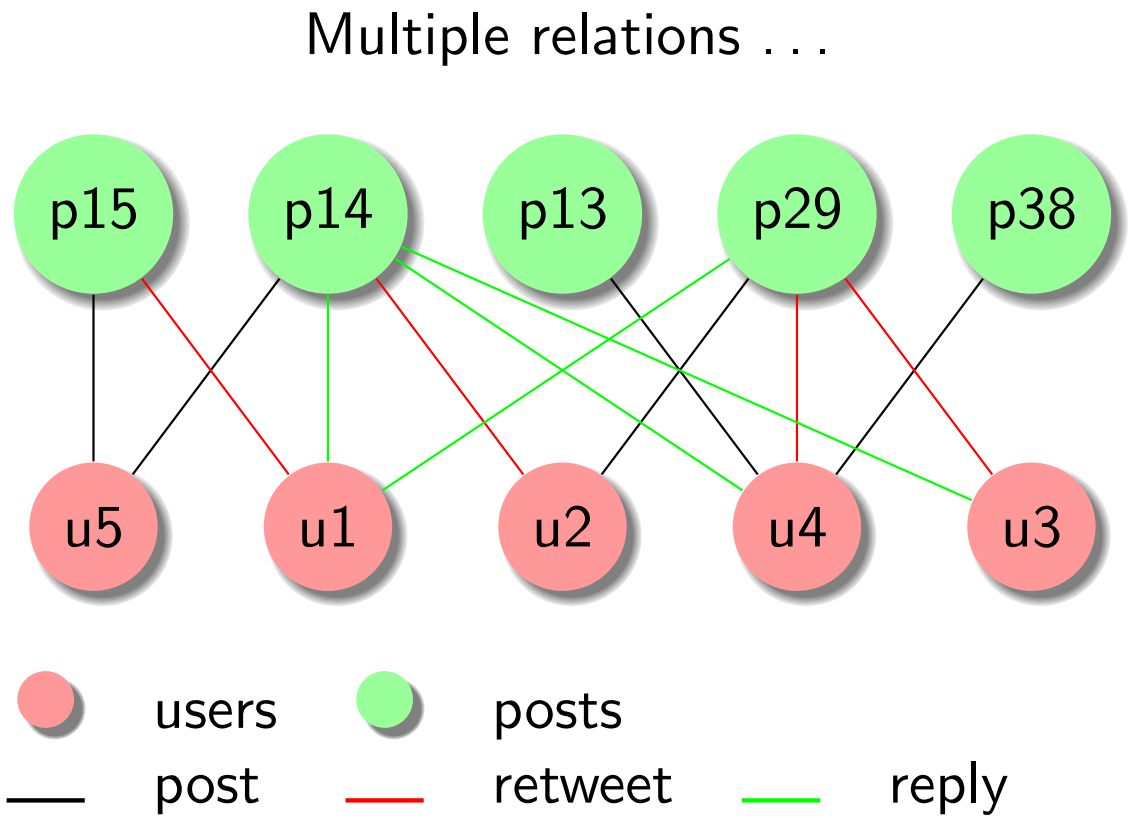
Its bi-partite equivalent . . .



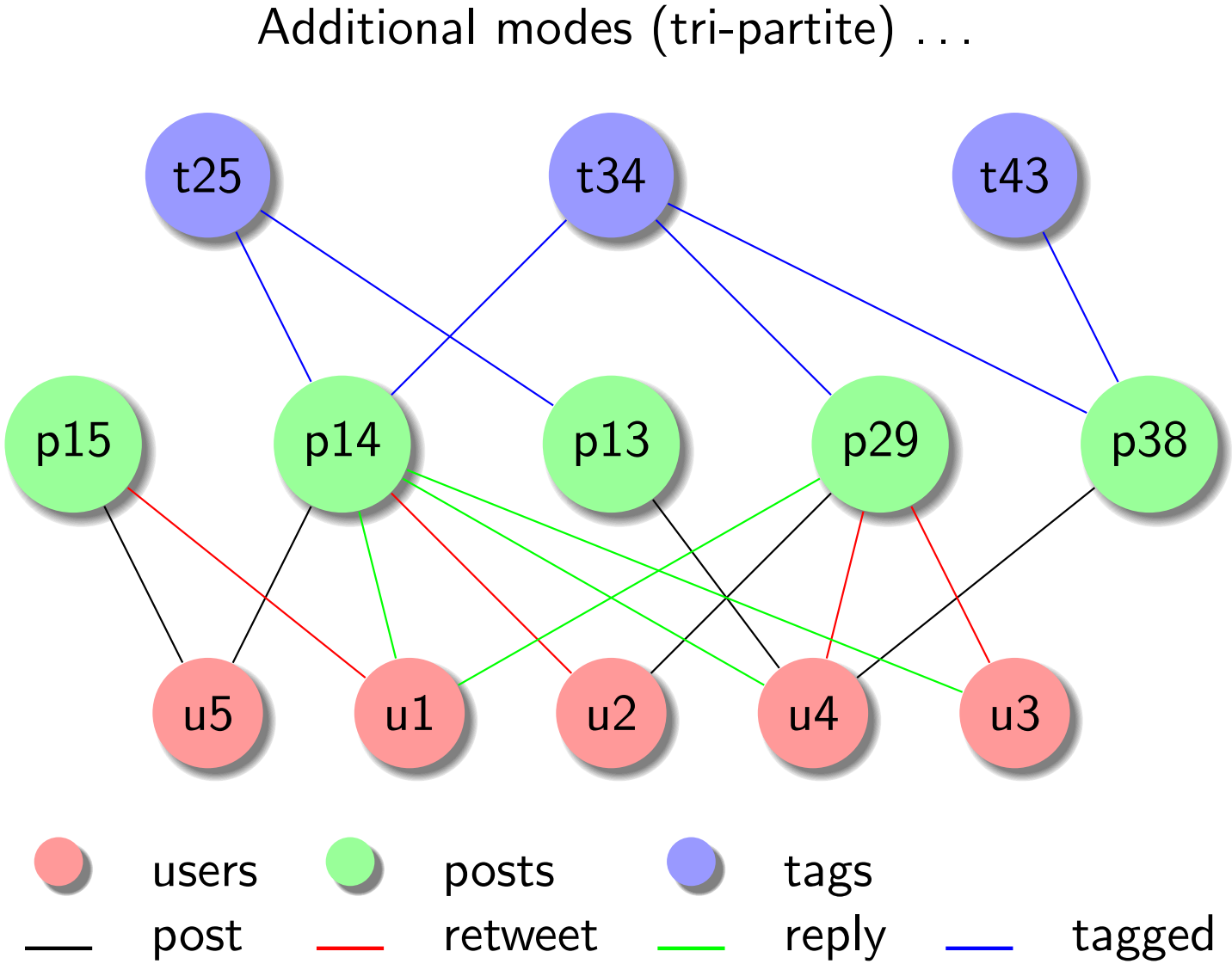
A multi-modal, multi-relational network



A multi-modal, multi-relational network

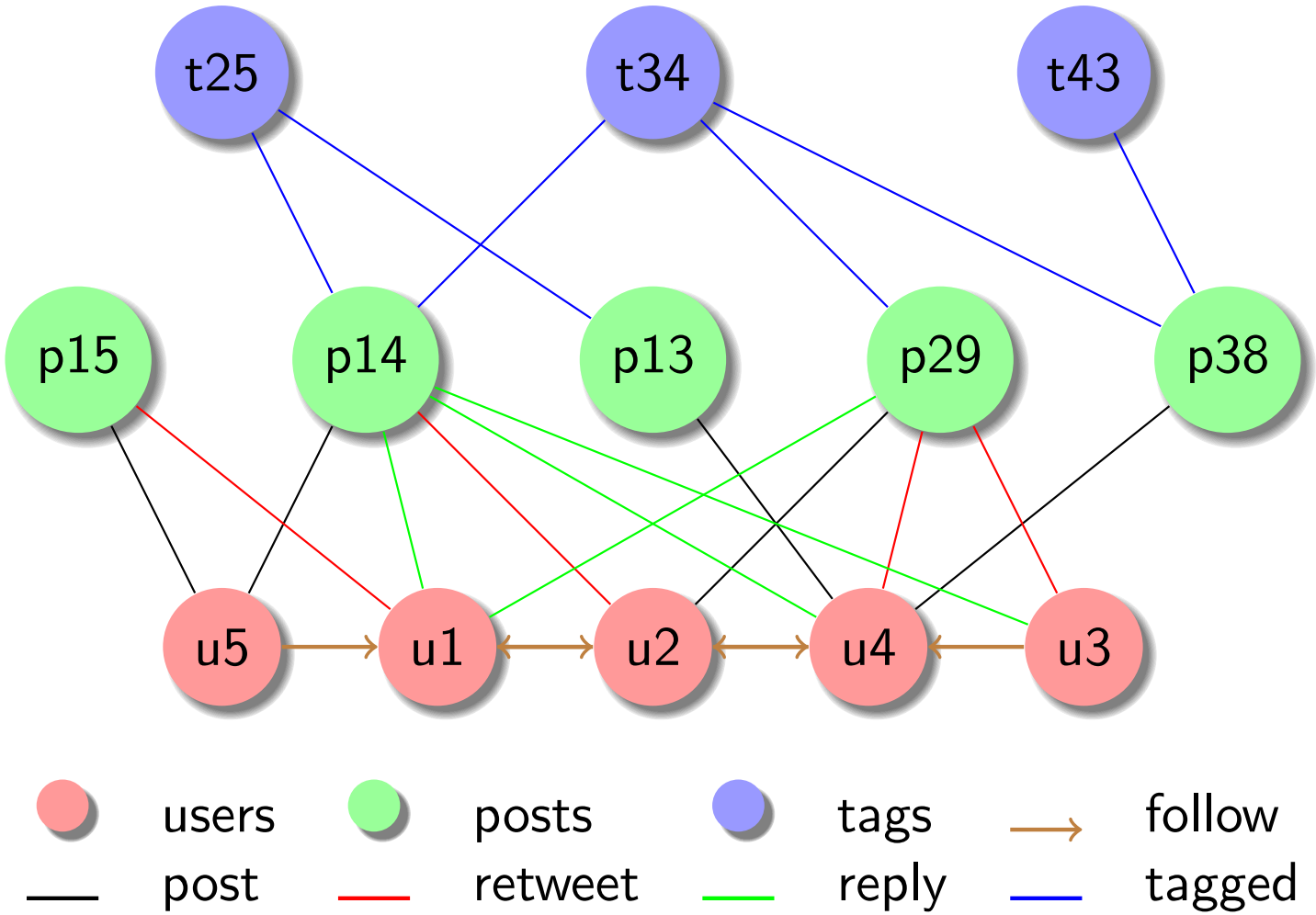


A multi-modal, multi-relational network



A multi-modal, multi-relational network

Relations among same-type nodes (no more multi-partite) . . .

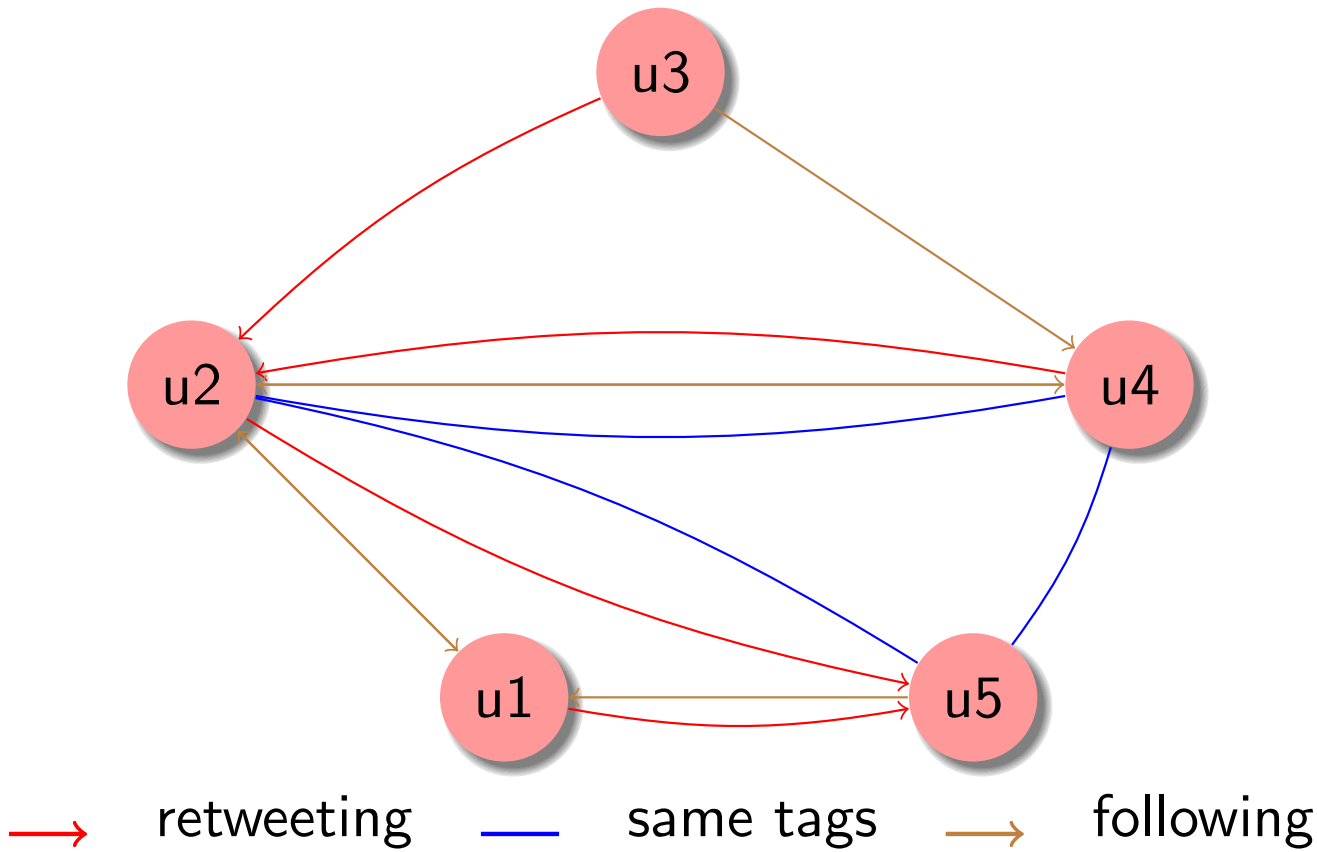


A multi-modal, multi-relational network

- Many more modes and relations, even in a simple social network like twitter.
- Many different definitions of community.
- For personalization, we are interested in user communities.
- Many different views to user communities:
  - Subgroups of explicitly connected users (latent factors).
  - Users connected through content (e.g. similar posts).
  - Users connected through activity (e.g. similar searches).
- We want to use all information hinting towards a community, e.g. retweeting posts of users who use similar tags.
- Extra requirement: scalable methods.

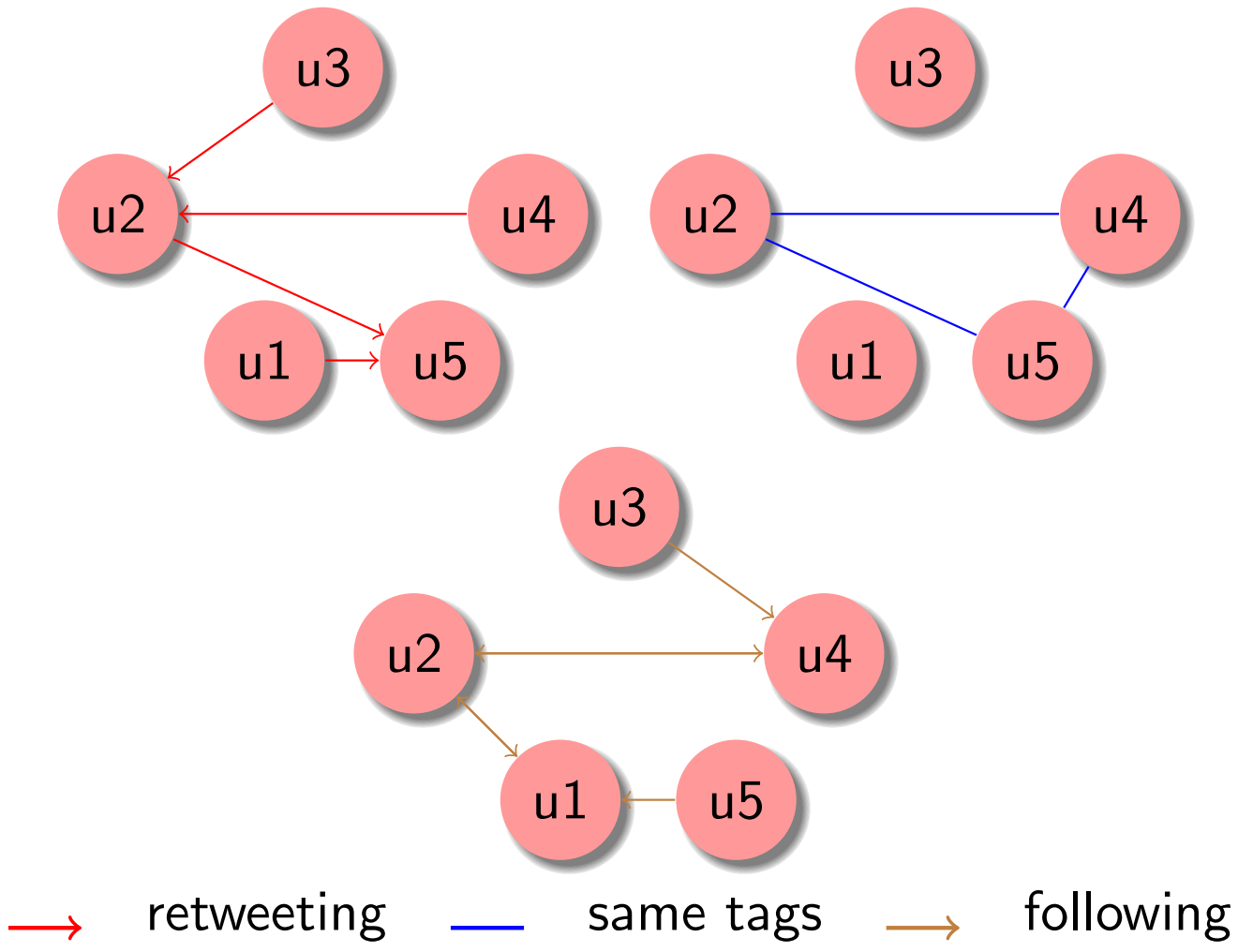
Discovering communities in multi-relational networks

- Assume nodes of the same type, i.e. users.
- Multiple relations among them.
- Different types of node can also be mapped onto relations, e.g. frequency of use of the same tag.



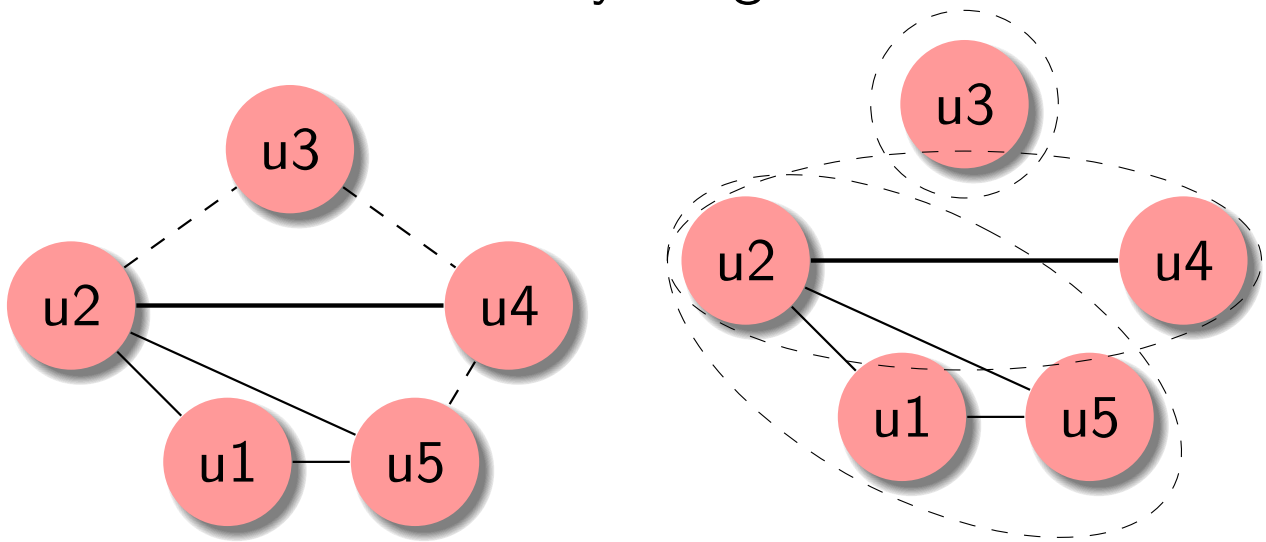
Discovering communities in multi-relational networks

Different network for each relation . . .



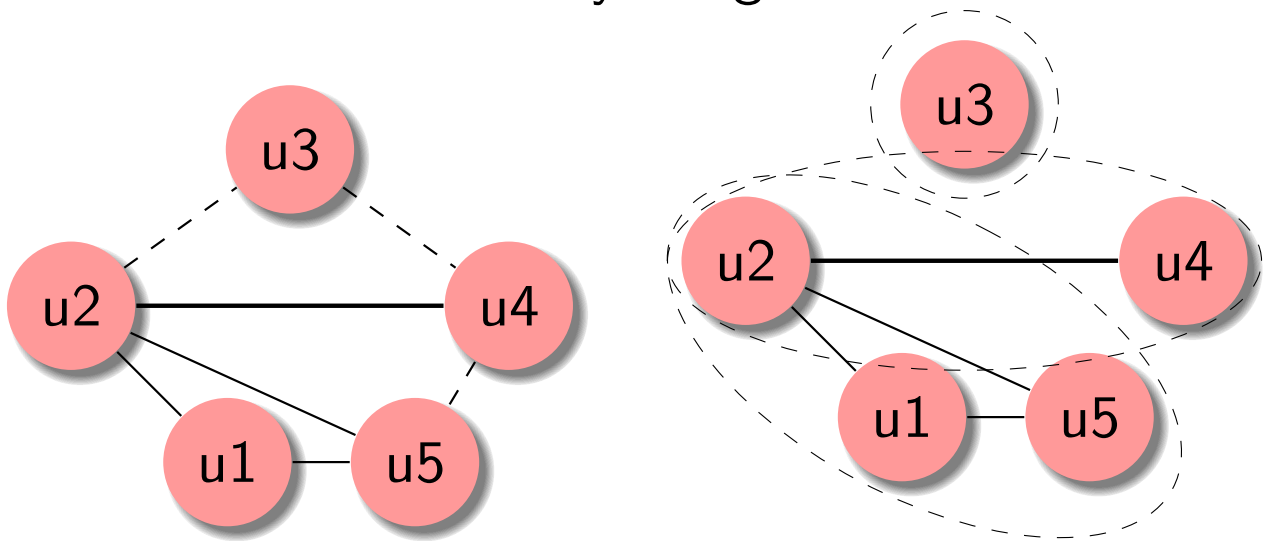
Discovering communities in multi-relational networks

Naive early integration:



Discovering communities in multi-relational networks

Naive early integration:



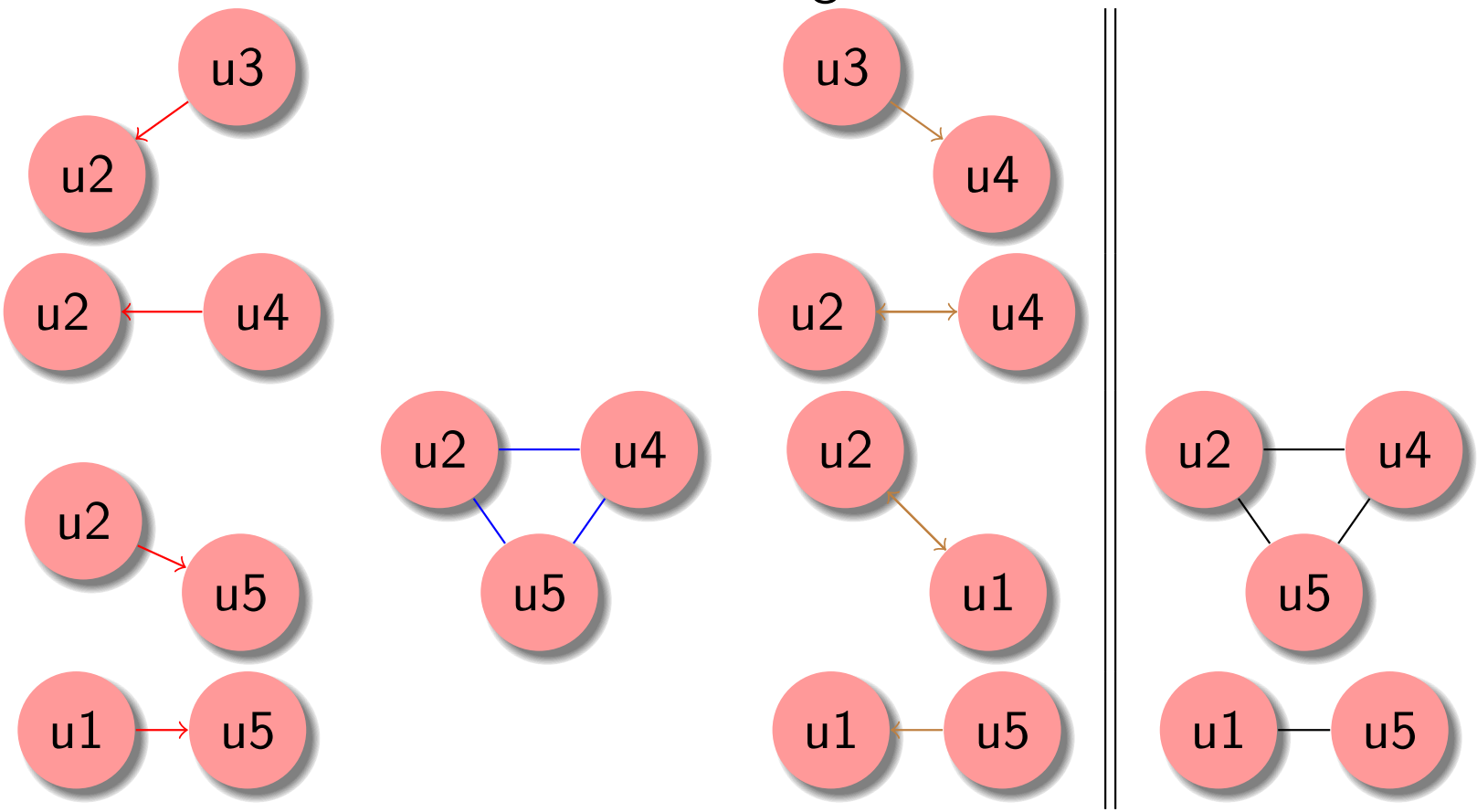
Alternative early integration (extended user-item matrix):

	rt(u5)	rt(u2)	f(u1)	f(u2)	f(u4)	t25	t34	t43
u1	1	0	0	1	0	0	0	0
u2	1	0	1	0	1	0	1	0
u3	0	1	0	0	1	0	0	0
u4	0	1	0	1	0	1	1	1
u5	0	0	1	0	0	1	1	0

Increased dimensionality and loss of information about features.

Discovering communities in multi-relational networks

Naive late integration:



Discovering communities in multi-relational networks

Less naive late integration (community indicators):

	u23	u24	u25	u15	u245	u34	u21
u1	0	0	0	1	0	0	1
u2	1	1	1	0	1	0	1
u3	1	0	0	0	0	1	0
u4	0	1	0	0	1	1	0
u5	0	0	1	1	1	0	0

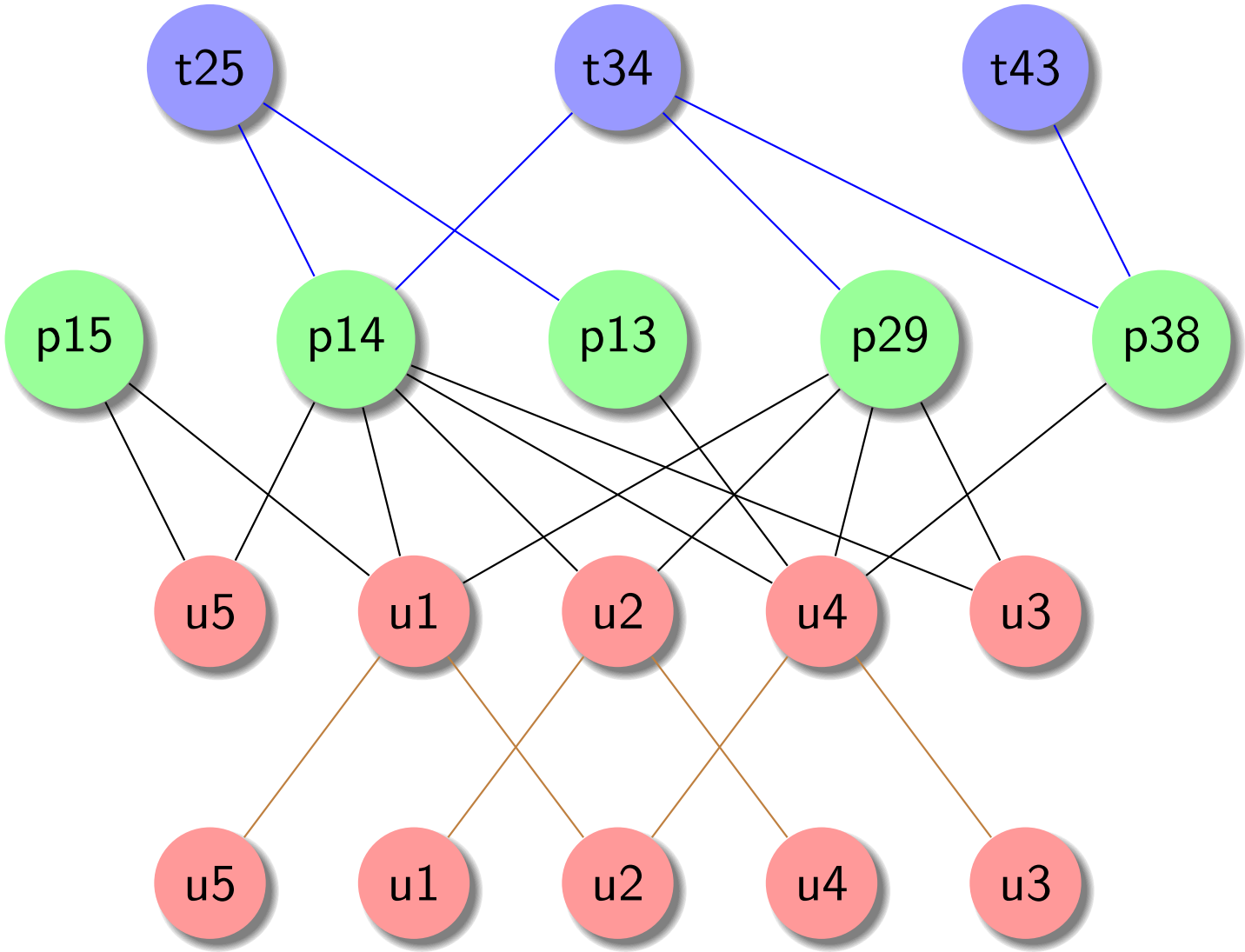
	u23	u24	u25	u15	u245	u34	u21
u1	0.0	0.0	0.0	0.7	0.0	0.0	0.5
u2	0.4	0.5	0.7	0.0	0.5	0.0	0.4
u3	0.7	0.0	0.0	0.0	0.0	0.5	0.0
u4	0.0	0.7	0.0	0.0	0.5	0.4	0.0
u5	0.0	0.0	0.6	0.6	0.5	0.0	0.0

- Tricky calculation of probabilities, due to overlapping communities in different base networks.
- Need to perform clustering on these data.

Discovering communities in multi-modal networks

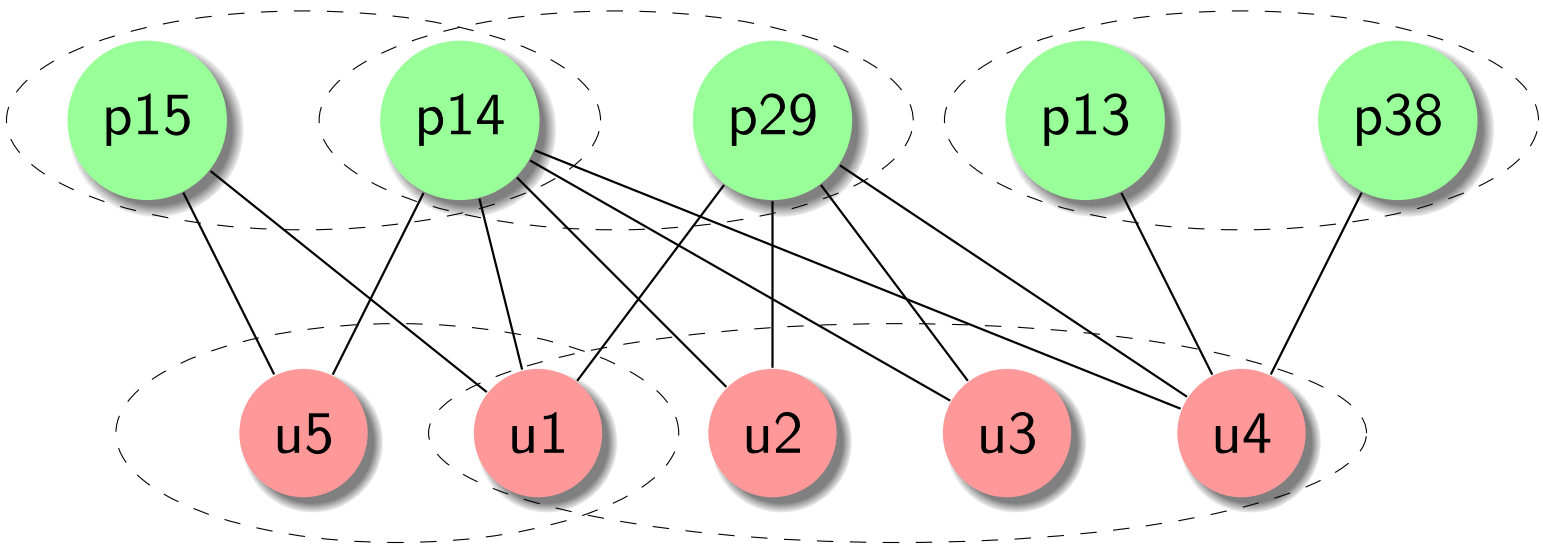
- Assume relations only across node types (multi-partite networks).
- Relations among nodes of the same type can be handled by node duplication, e.g. two sets of user nodes.
- Assume one relation type among two node types.
- Simplest case bi-partite network.

Discovering communities in multi-modal networks



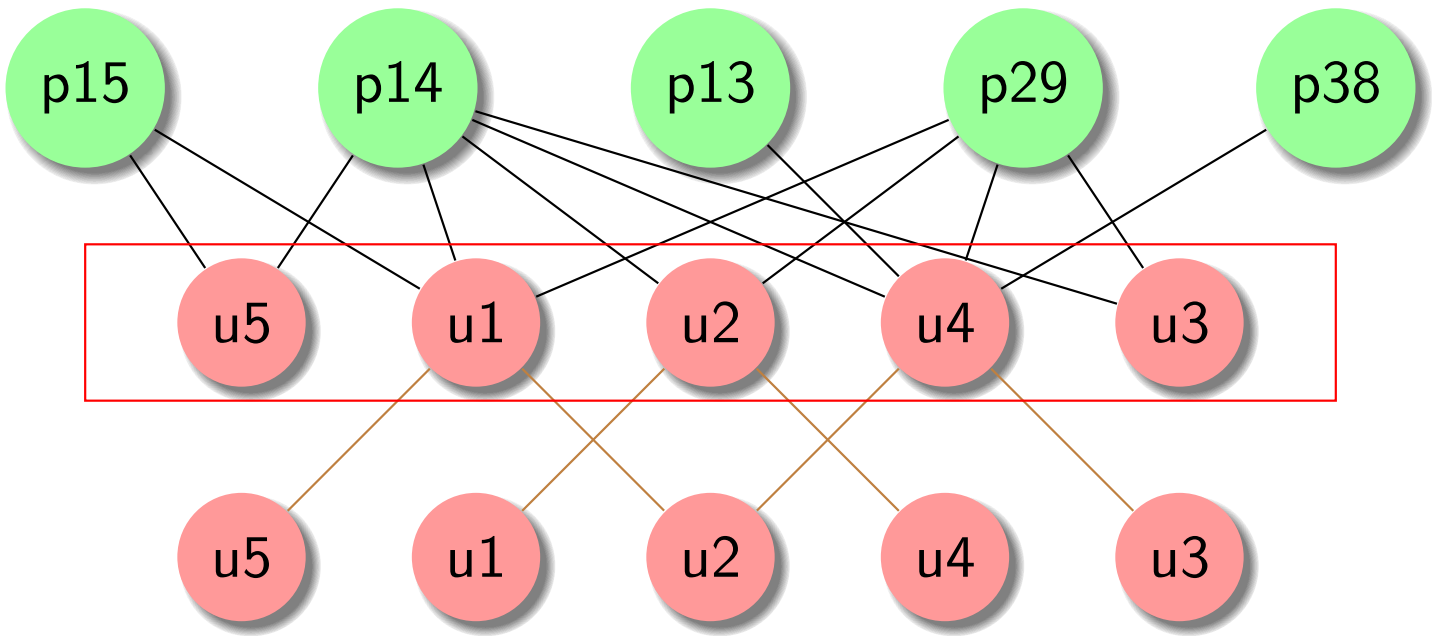
Discovering communities in multi-modal networks

- Simplest solution is to treat each bi-partite network separately.
- Either translate it into user-item matrix and calculate similarities or perform latent community modeling.
- Co-clustering, i.e. clustering simultaneously both node types, has also been used.
- Aggregating communities of bi-partite networks is tricky.



Discovering communities in multi-modal networks

- Are we interested in all types of node? Can we focus on users?
- Use methods for star-structured networks, e.g. spectral co-clustering.
- Missing higher-order relations, e.g. use of same tags.
- Can create intermediate communities, e.g. of tags, and incorporate community indicators, e.g. of posts.



Discovering communities in multi-modal networks

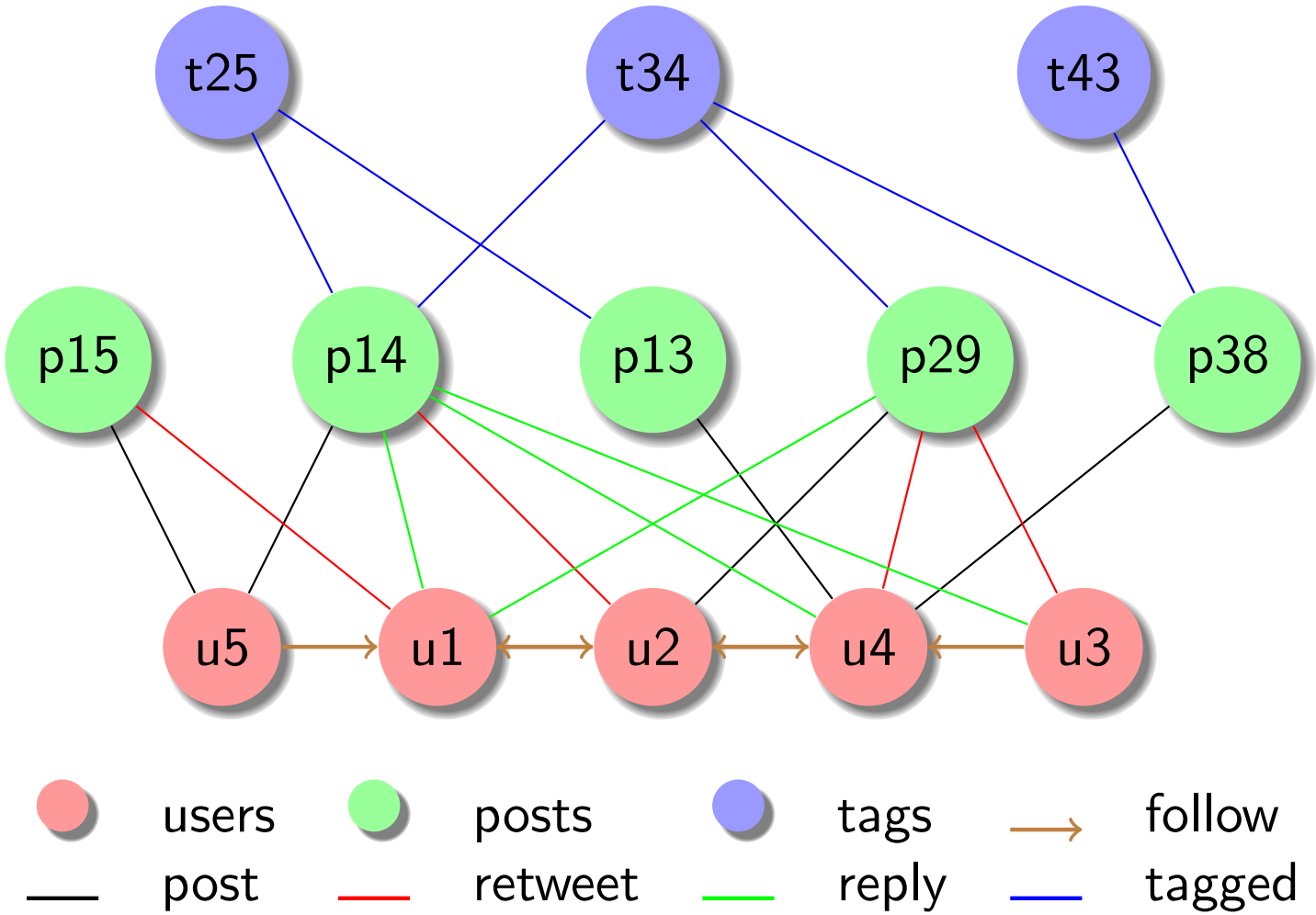
- A different approach: Extend the user-item matrix to a multi-dimensional tensor.
- Extend dimensionality reduction to tensors (decomposition), leading to multi-dimensional community models.
- Assumes everything can be related to anything, which is:
  - Not always applicable, e.g. we have assumed that tags are related to posts, but not directly to users.
  - Inefficient if the sparsity of the tensor is not used.
- Does not naturally handle:
  - multiple relation types between node types, posts and users in our example,
  - relations between entities of the same type, e.g. follows relation.

Relational data mining

- All methods for multi-relational and multi-modal networks make strong simplifying assumptions.
- Treating all relations between two node types as the same or focusing on a single node type.
- Lost information can be valuable for community discovery.
- In practice, simplifications can be informed, due to our knowledge of the network.
- Are there knowledge discovery methods that can handle the complexity of the social Web?
- First-order logic lends itself for relational data.

Relational data mining

Our sketch social network . . .



Relational data mining

$user(u1)$	$follows(u5, u1)$	$posts(u5, p15)$
$user(u2)$	$follows(u1, u2)$	$posts(u5, p14)$
$user(u3)$	$follows(u2, u1)$	$posts(u2, p29)$
$user(u4)$	$follows(u2, u4)$	$posts(u4, p13)$
$user(u5)$	$follows(u4, u2)$	$posts(u4, p38)$
$post(p13)$	$follows(u3, u4)$	$retweet(u1, p15)$
$post(p14)$	$tagged(p14, t25)$	$retweet(u2, p14)$
$post(p15)$	$tagged(p14, t34)$	$retweet(u4, p29)$
$post(p29)$	$tagged(p13, t25)$	$retweet(u3, p29)$
$post(p38)$	$tagged(p29, t34)$	$reply(u1, p14)$
$tag(t25)$	$tagged(p38, t34)$	$reply(u4, p14)$
$tag(t34)$	$tagged(p38, t43)$	$reply(u3, p14)$
$tag(t43)$		$reply(u1, p29)$

Relational data mining

What would communities look like?

$$
\begin{aligned}
&\mathit{belongs}(u_i, c_1) \text{ if } && \mathit{user}(u_i) \wedge \mathit{follows}(u_i, u_2) \wedge \mathit{follows}(u_i, u_3) \\
&\mathit{belongs}(u_i, c_1) \text{ if } && \exists p_k,\ \mathit{user}(u_i) \wedge \mathit{post}(p_k) \wedge \mathit{follows}(u_i, u_2) \wedge \\
& && \mathit{posts}(u_2, p_k) \wedge \mathit{retweet}(u_i, p_k) \\
&\mathit{belongs}(u_i, c_1) \text{ if } && \exists p_k,\ \mathit{user}(u_i) \wedge \mathit{post}(p_k) \wedge \mathit{posts}(u_i, p_k) \wedge \\
& && \mathit{tagged}(p_k, t43)
\end{aligned}
$$

Relational data mining

What would communities look like?

$$
\begin{array}{lll}
(1.5) & \mathit{belongs}(u_i, c1) \text{ if } & \mathit{user}(u_i) \wedge \mathit{follows}(u_i, u2) \wedge \mathit{follows}(u_i, u3) \\
(0.8) & \mathit{belongs}(u_i, c1) \text{ if } & \exists p_k,\ \mathit{user}(u_i) \wedge \mathit{post}(p_k) \wedge \\
 & & \mathit{follows}(u_i, u2) \wedge \mathit{posts}(u2, p_k) \wedge \\
 & & \mathit{retweet}(u_i, p_k) \\
(1.3) & \mathit{belongs}(u_i, c1) \text{ if } & \exists p_k,\ \mathit{user}(u_i) \wedge \mathit{post}(p_k) \wedge \\
 & & \mathit{posts}(u_i, p_k) \wedge \mathit{tagged}(p_k, t43)
\end{array}
$$

Can also be associated with weights, corresponding to soft community indicators, as in the simpler probabilistic community models.

Relational data mining

We can also add background knowledge, e.g. (hard) constraints:

$$
\begin{aligned}
tagged(p_k, t_s) \rightarrow \quad & post(p_k) \wedge tag(t_s) \\
posts(u_i, p_k) \rightarrow \quad & post(p_k) \wedge user(u_i) \wedge \\
& \nexists u_j,\ user(u_j) \wedge posts(u_j, p_k)
\end{aligned}
$$

Relational data mining

We can also add background knowledge, e.g. (hard) constraints:

$$\begin{aligned} tagged(p_k, t_s) \rightarrow \quad & post(p_k) \wedge tag(t_s) \\ posts(u_i, p_k) \rightarrow \quad & post(p_k) \wedge user(u_i) \wedge \\ & \nexists u_j,\ user(u_j) \wedge posts(u_j, p_k) \end{aligned}$$

or even hint (softly) towards communities:

$$\begin{aligned} (1.5) \quad belongs(u_i, c_w) \text{ if } \quad & user(u_i) \wedge \exists u_j,\ user(u_j) \wedge \\ & belongs(u_i, c_w) \wedge follows(u_i, u_j) \wedge \\ & \exists p_k,\ post(p_k) \wedge posts(u_j, p_k) \wedge \\ & retweet(u_i, p_k) \end{aligned}$$

Deductive inference of communities?
Can it be combined with learning (inductive inference) beyond weight learning?

Relational data mining

- Various approaches to representing such kind of knowledge, i.e. combining logic and uncertainty:
  - Probabilistic logic programming.
  - Markov logic networks.
  - Probabilistic soft logic.
- Statistical Relational Learning methods have also been developed (not many for unsupervised learning).
- Computationally too expensive still.

Relational data mining

- An alternative: combine tensors with relations (Metagraph approach).
- Provide simple background knowledge, i.e. what can be related with what.
- Use it to break down a large tensor into smaller ones, e.g. one for relating posts with tags, one for user with posts, etc.
- Modify tensor decomposition to account for interdependencies, e.g. posts connecting users with tags.
- Realistically fast.
- Inherits some of the problems of tensors, e.g. relations among nodes of the same type.

Community evolution

- Communities change over time: they disappear, they merge, they split, new ones appear.
- Particularly important in the social Web, due to its dynamic nature.
- Many methods try to capture the evolution of communities over time.
- Based on sliding time windows, forgetting mechanisms, etc.
- Predicting a change is important and hard.
- The recognition of events that are associated with changes can help, e.g. the formation of communities before or after riots.

Summary of the section

- Community discovery in the social Web is very different.
- In terms of user modeling research, much more interesting.
- Social networks are multi-dimensional, multi-modal graphs.
- New methods are being developed and existing ones are extended.
- Can we have truly relational community discovery?
- Scalability is an important obstacle.
- Many challenges ahead of us: community evolution, handling of uncertainty, multi-site communities, etc.

Outline

Goals of this tutorial

- Discuss what a community is and why it is useful for personalization.
- See what is needed to discover user communities from data.
- Discuss how this differs in the social Web.
- Raise a lot of issues and questions.
- NOT present methods and algorithms.

Active user communities

- Users are active and our community discovery methods need to adapt.
- The Web is becoming the Web of people, taking the form of human society.
- Web sites and the computer itself are disappearing.
- Active user communities are “traditional” communities of people.
- The medium amplifies both the good and the bad things of our society.

Short break for a commercial: PServer

- Generic server (+ Web server) for personalizing applications.
- Developed for many years at NCSR "Demokritos".
- Incorporates modeling methods, including communities.
- Has been used for a range of applications, from personalized news reading to style advise.
- Now open source (Apache licence).
- Documented and promoted with the help of Scify SCIENCE FOR YOU
- Download from: `https://code.google.com/p/pserver/`

Teasers

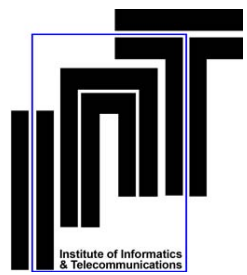
- Is simplification necessary in discovering active communities?
- Tensors, logic, both or none?
- Generic methods or ad-hoc solutions?
- What should personalization be used for in social networks?
- Is cross-site logging dangerous for our privacy?
- Are our methods dangerous in the hands of the wrong people?
- Can we help solve societal problems, e.g. crime, social isolation?
- Is the social Web a good thing?

User community discovery: the transition from passive site visitors to active content contributors[†]

Georgios Paliouras

paliourg@iit.demokritos.gr

Institute of Informatics & Telecommunications,
NCSR "Demokritos", Greece



[†]Based on: G. Paliouras, "**Discovery of Web user communities and their role in personalization**," *User Modeling and User-Adapted Interaction*, v. 22, n. 1-2, pp. 151-175, 2012.