Like like alike

Joint friendship and interest propagation in social networks

Shuang Hong Yang, Bo Long, Alex Smola
Nara Sadagopan, Zhaohui Zheng, Hongyan Zha

Georgia Institute of Technology Yahoo! Research

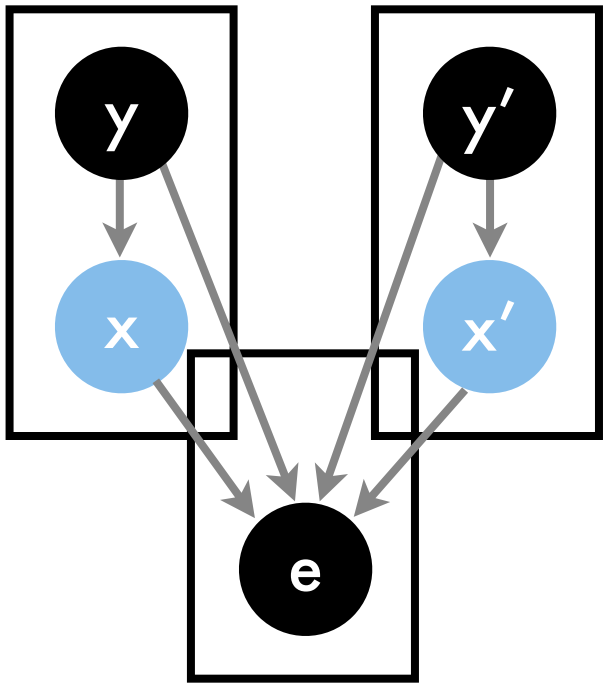
Outline

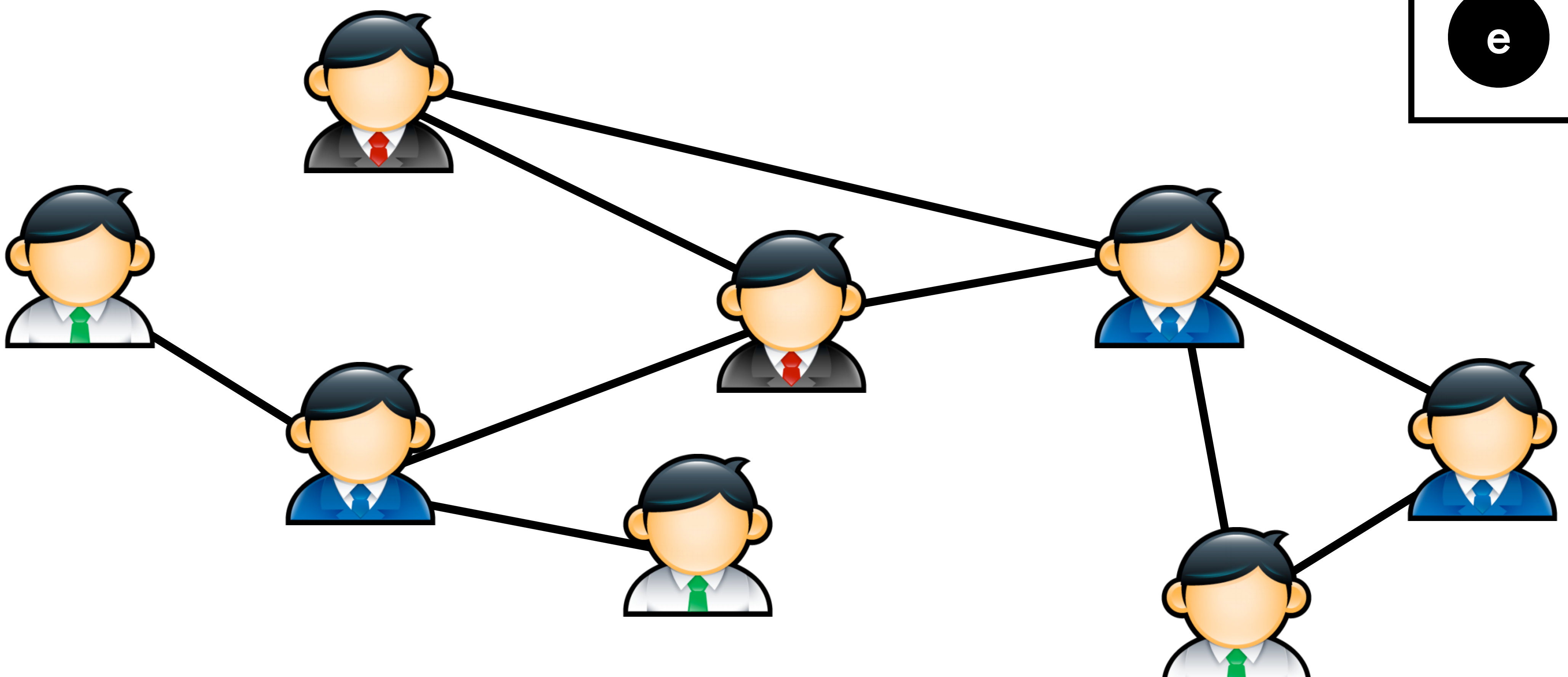
- Factorization models
- Friendship-Interest Propagation
- Experiments
- Summary

Social Network Data

Data: users, connections, features

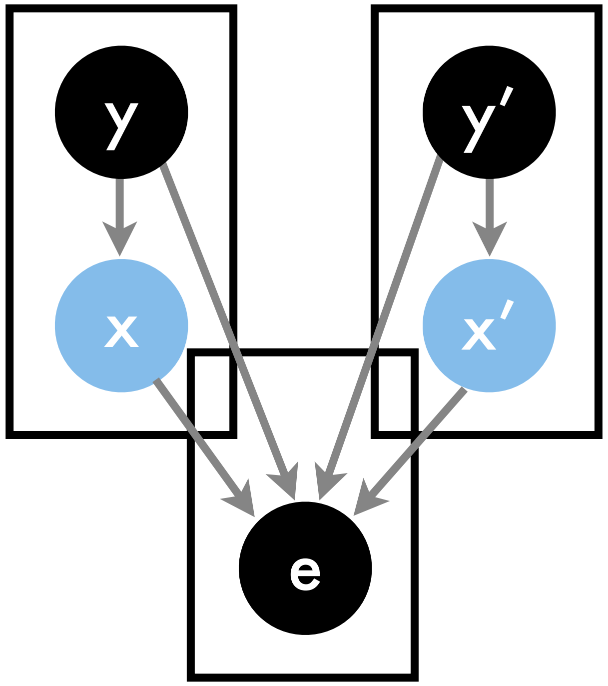
Goal: suggest connections





Social Network Data

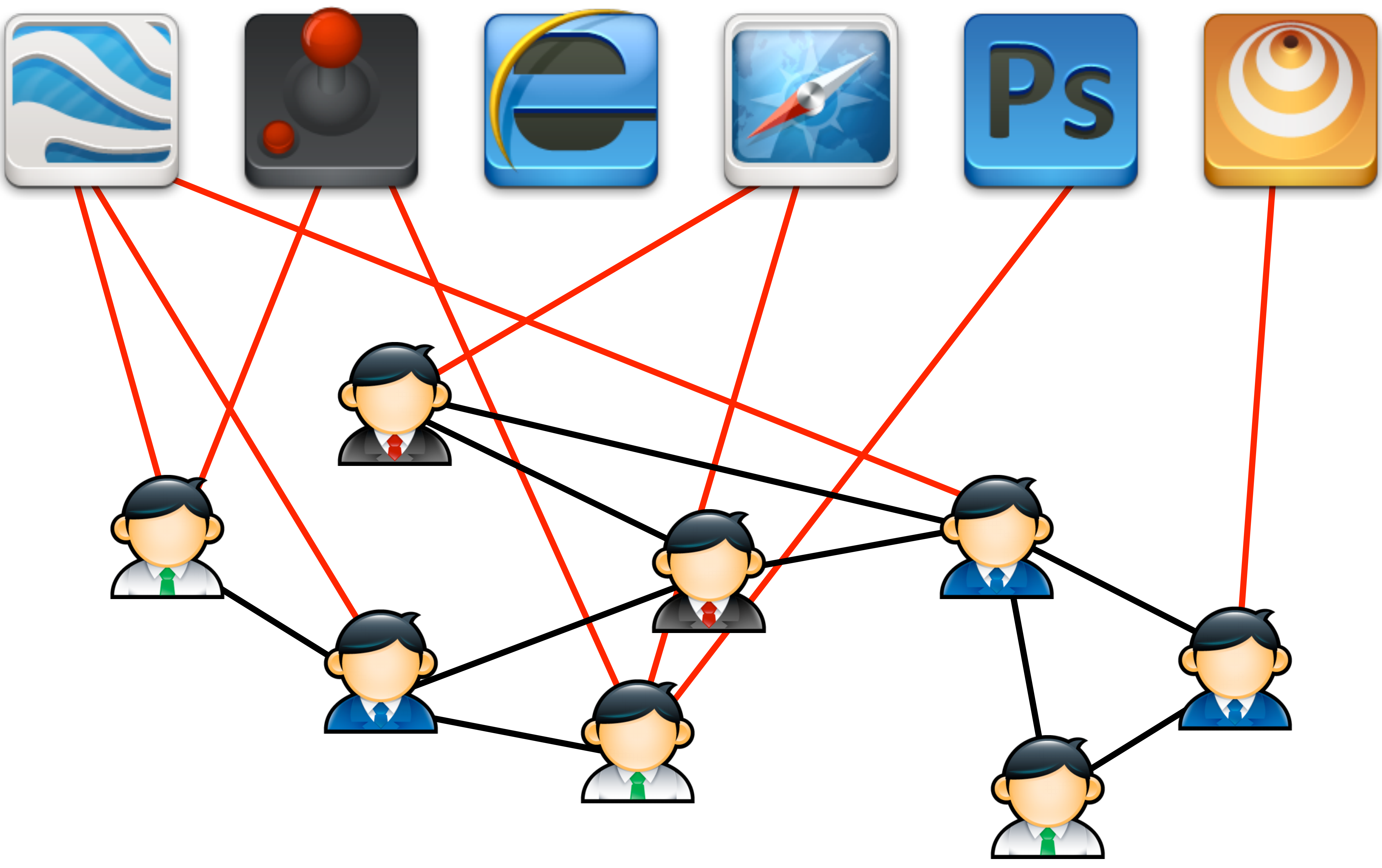
Data: users, connections, features
Goal: model/suggest connections



$$p(x, y, e) = \prod_{i \in \text{Users}} p(y_i) p(x_i | y_i) \prod_{i,j \in \text{Users}} p(e_{ij} | x_i, y_i, x_j, y_j)$$

Direct application of the Aldous-Hoover theorem.

Applications

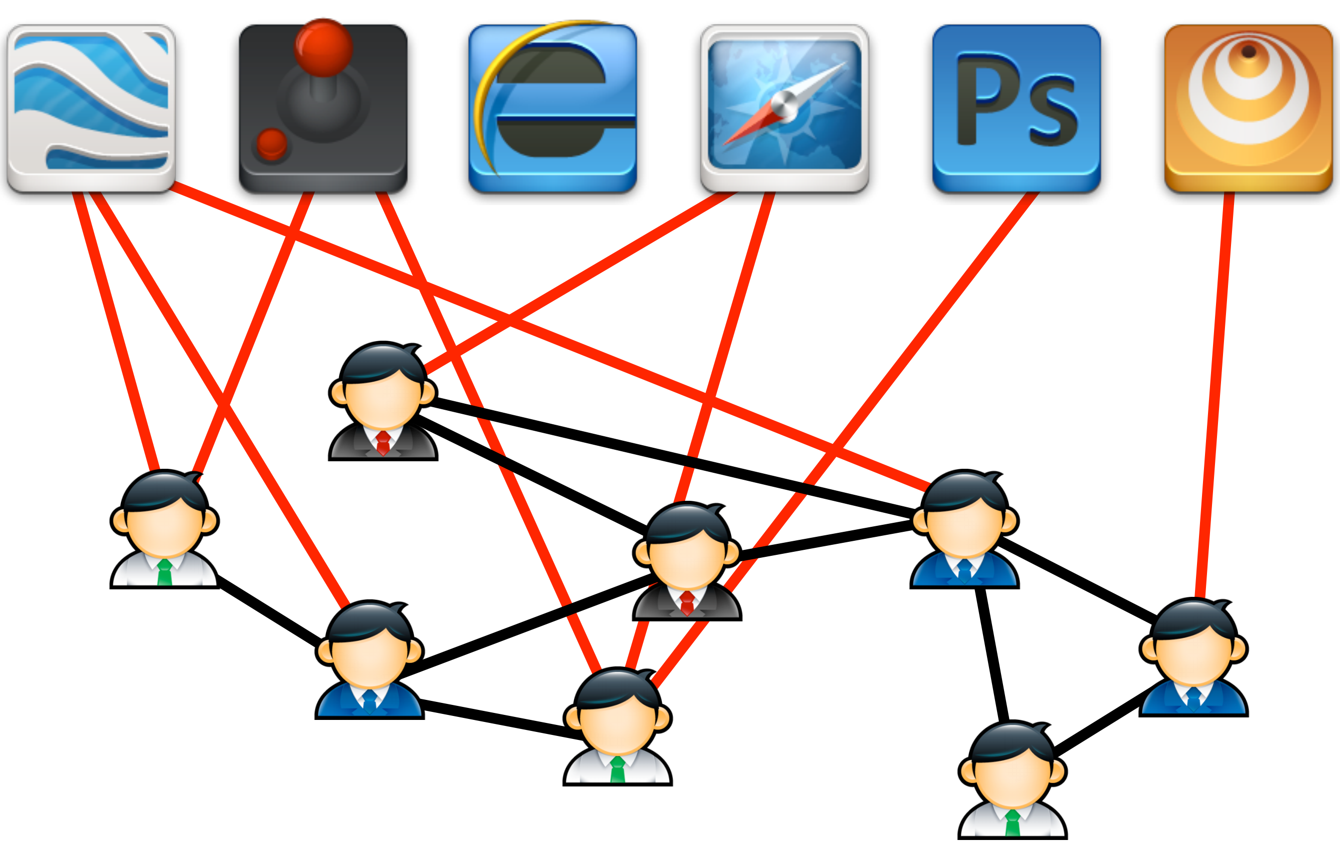


Applications

social network = friendship + interests

recommend users based on friendship & interests

recommend apps based on friendship & interests



Social Recommendation

recommend users based on friendship & interests

- boost traffic
- make the user graph more dense
- increase user population
- stickiness

recommend apps based on friendship & interests

- boost traffic
- increased revenue
- increased user participation
- make app graph more dense

... usually addressed by separate tools ...

Homophily

recommend users based on friendship & interests

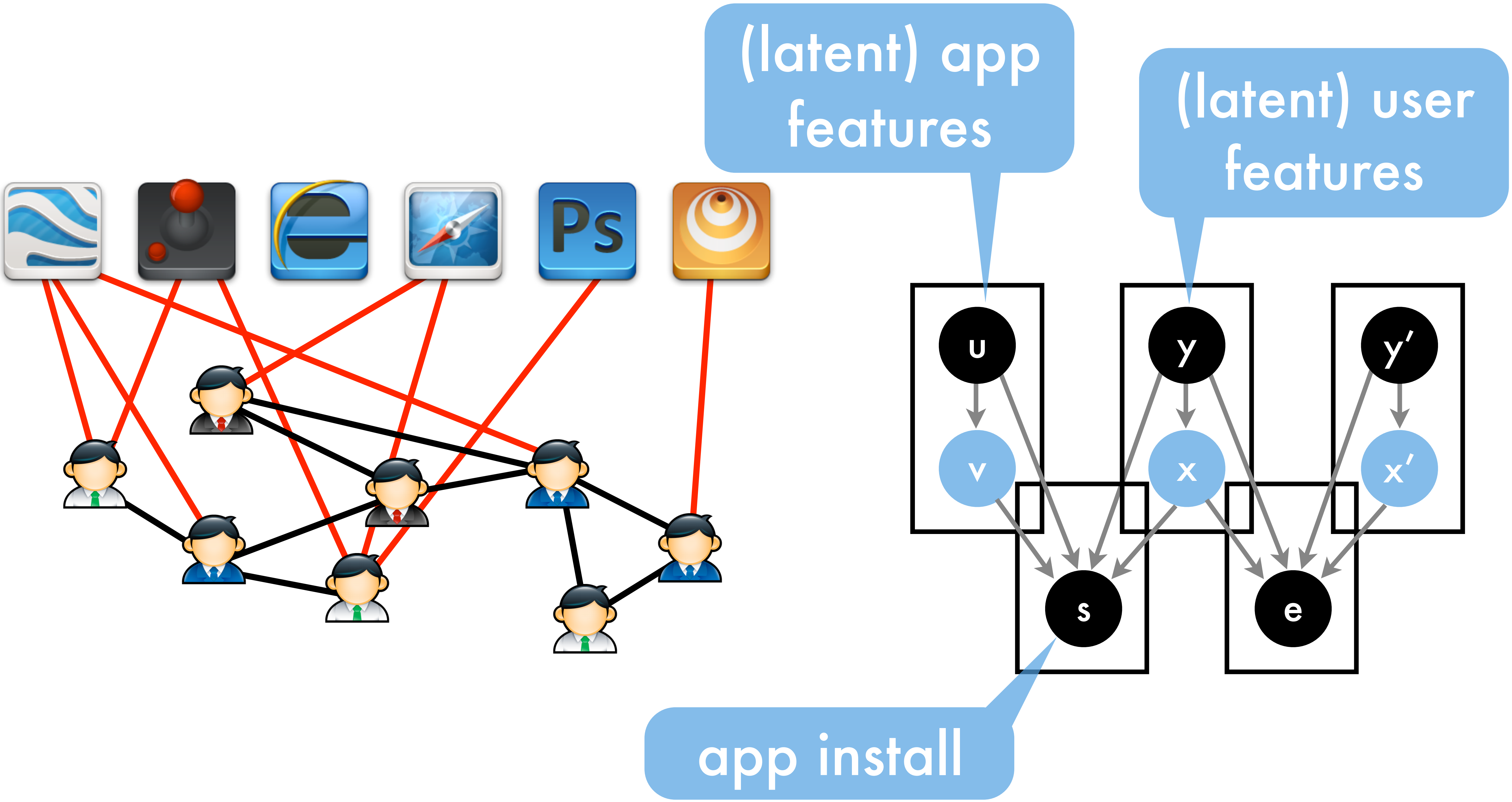
recommend apps based on friendship & interests

- users with similar interests are more likely to connect

- friends install similar applications

Highly correlated. Estimate both jointly

Model


(latent) app features
(latent) user features
u
y
y′
v
x
x′
s
e
app install

Model

- Social interaction

$$x_i \sim p(x|y_i)$$

$$x_j \sim p(x|y_j)$$

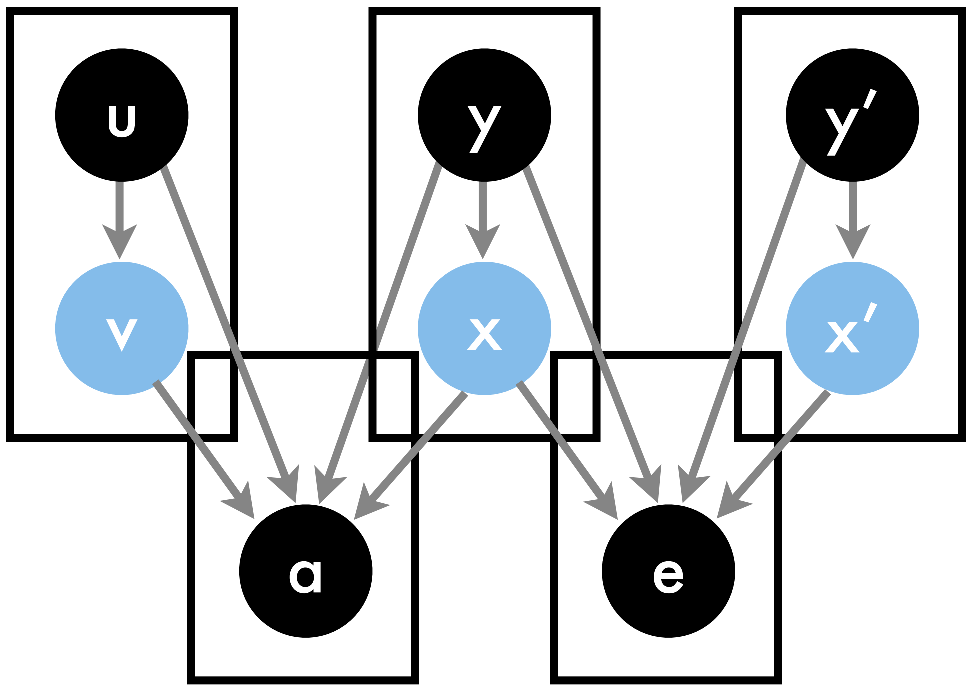
$$e_{ij} \sim p(e|x_i, y_i, x_j, y_j, \Phi)$$

- App install

$$x_i \sim p(x|y_i)$$

$$v_j \sim p(v|u_j)$$

$$a_{ij} \sim p(a|x_i, y_i, u_j, v_j, \Phi)$$



Model

- **Social interaction**

$$x_i \sim p(x|y_i)$$
$$x_j \sim p(x|y_j)$$
$$e_{ij} \sim p(e|x_i, y_i, x_j, y_j, \Phi)$$

cold start

latent features

$$x_i = Ay_i + \epsilon_i$$
$$v_j = Bu_j + \tilde{\epsilon}_j$$

- **App install**

$$x_i \sim p(x|y_i)$$
$$v_j \sim p(v|u_j)$$
$$a_{ij} \sim p(a|x_i, y_i, u_j, v_j, \Phi)$$

$$e_{ij} \sim p(e|x_i^\top x_j + y_i^\top W y_j)$$
$$a_{ij} \sim p(a|x_i^\top v_j + y_i^\top M u_j)$$

bilinear features

Optimization Problem

minimize $\lambda_e \sum_{(i,j)} l(e_{ij}, x_i^\top x_j + y_i^\top W y_j) +$

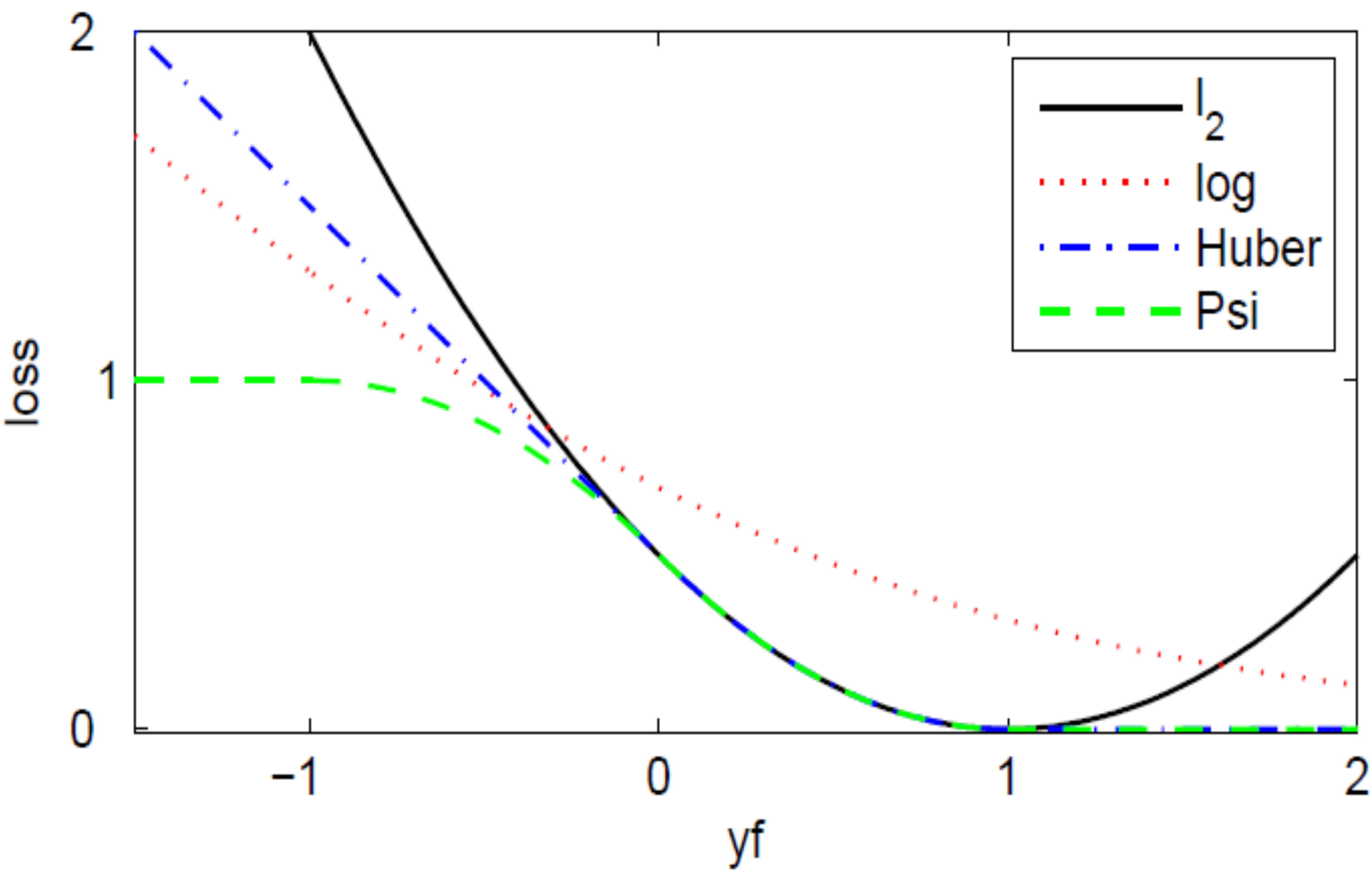
social

app

reconstruction

regularizer

Loss Function

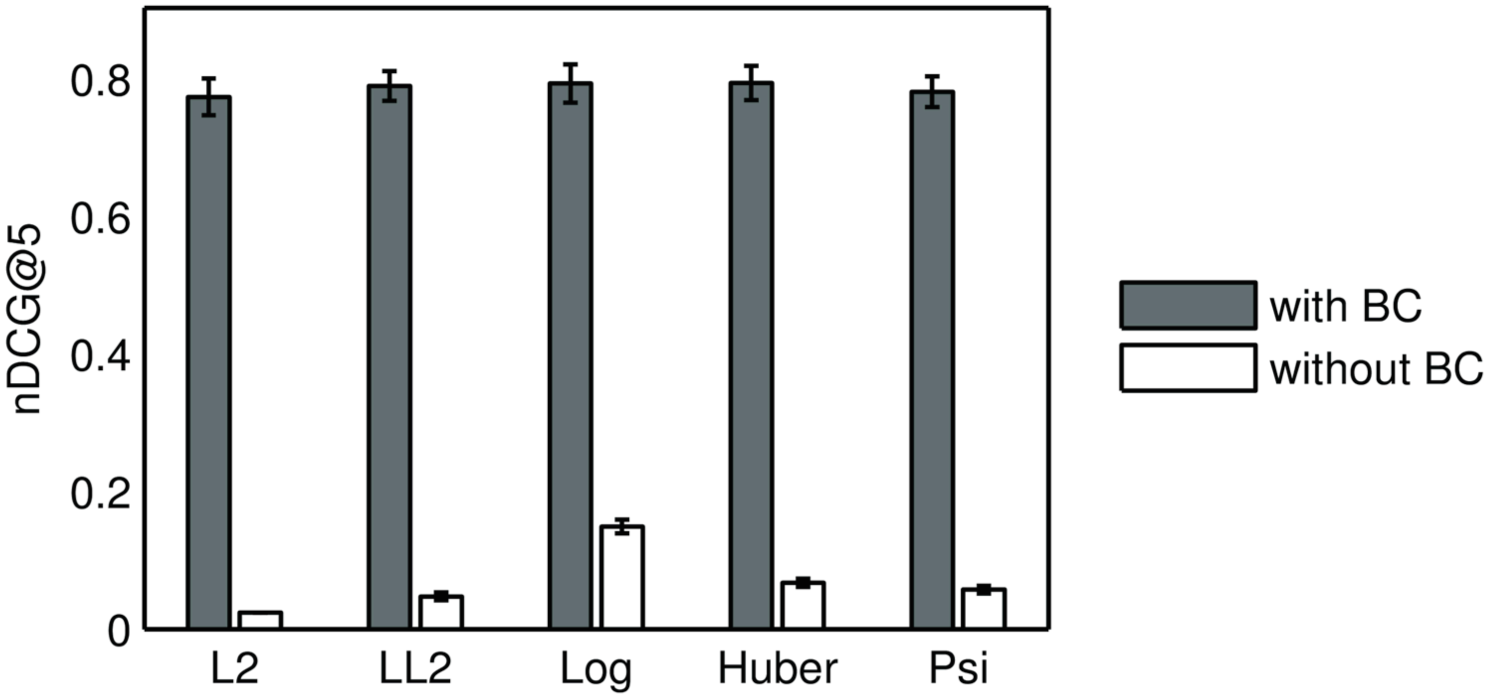


Loss

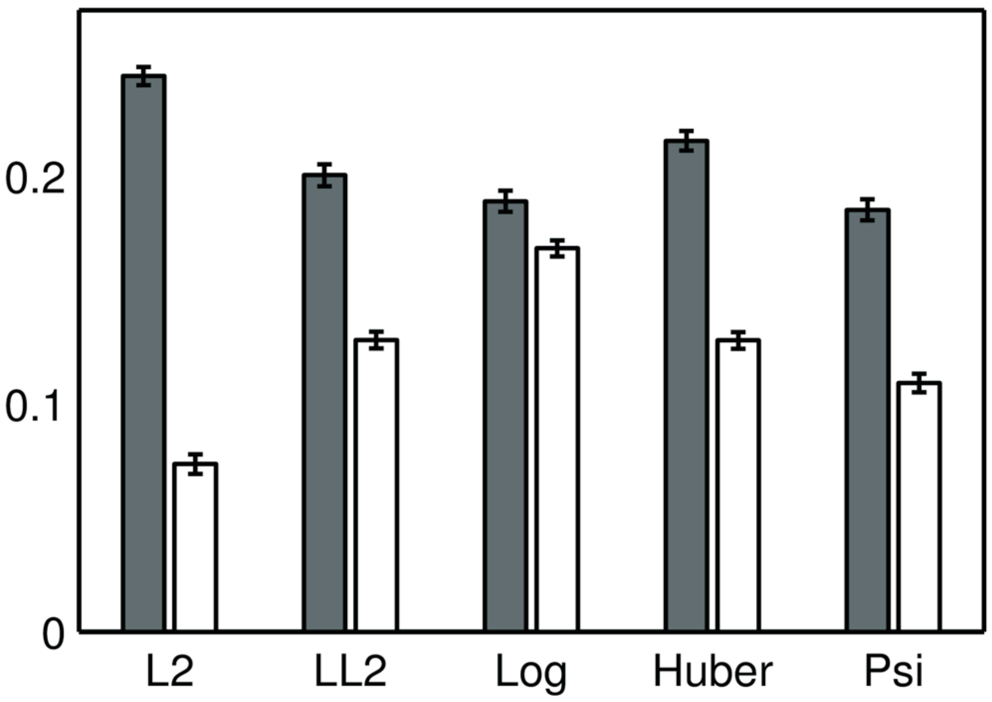
- Much more evidence of application non-install (i.e. many more negative examples)
- Few links between vertices in friendship network (even within short graph distance)

- Generate ranking problems (link, non-link) with non-links drawn from background set

Loss



application recommendation



social recommendation

Optimization

- Nonconvex optimization problem
- Large set of variables

$$x_i = Ay_i + \epsilon_i$$
$$v_j = Bu_j + \tilde{\epsilon}_j$$

- Stochastic gradient descent on x, v, ε for speed

$$e_{ij} \sim p(e|x_i^\top x_j + y_i^\top W y_j)$$
$$a_{ij} \sim p(a|x_i^\top v_j + y_i^\top M u_j)$$

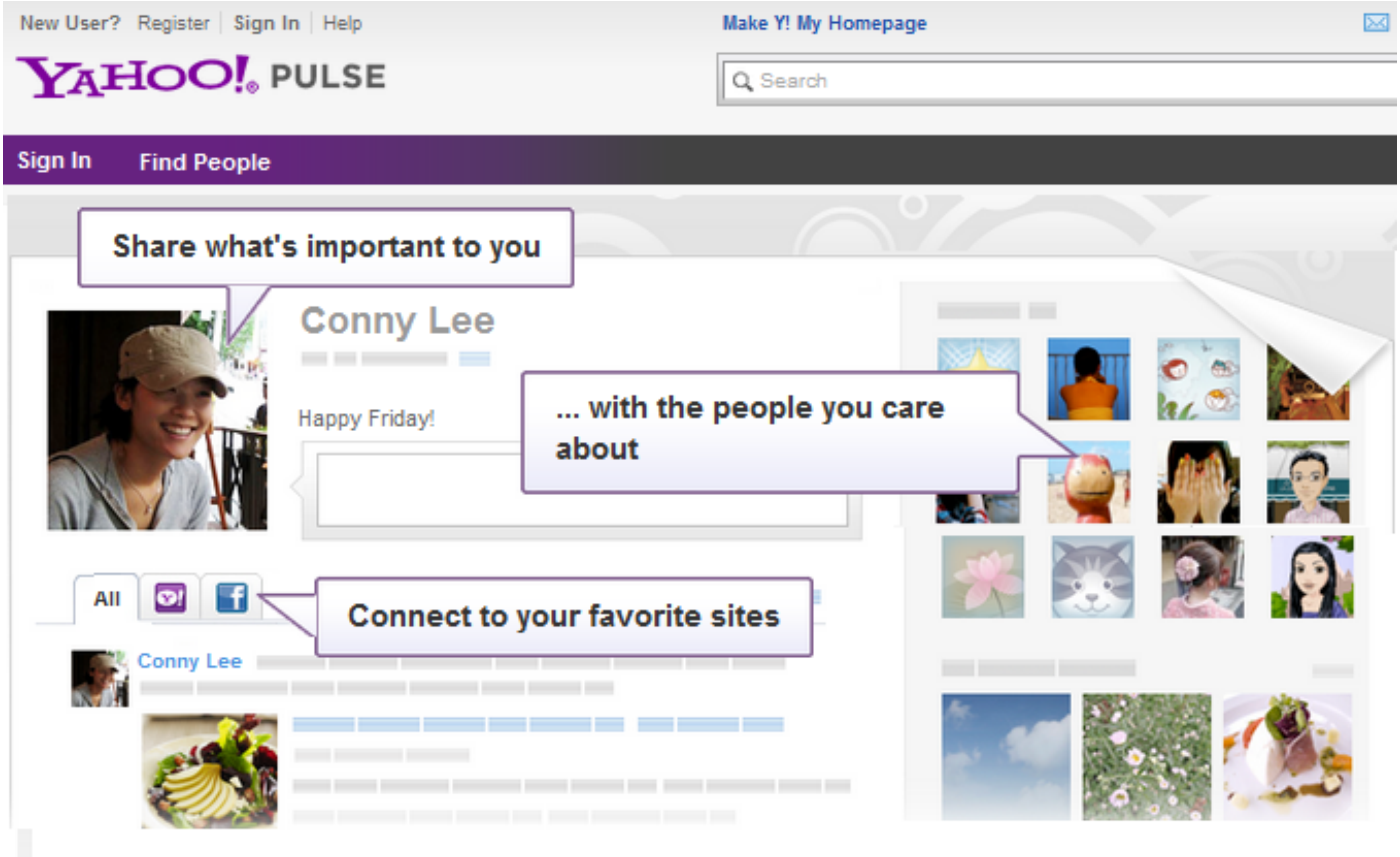
- Use hashing to reduce memory load, i.e.

$$x_{ij} = \sigma(i,j) X[h(i,j)]$$

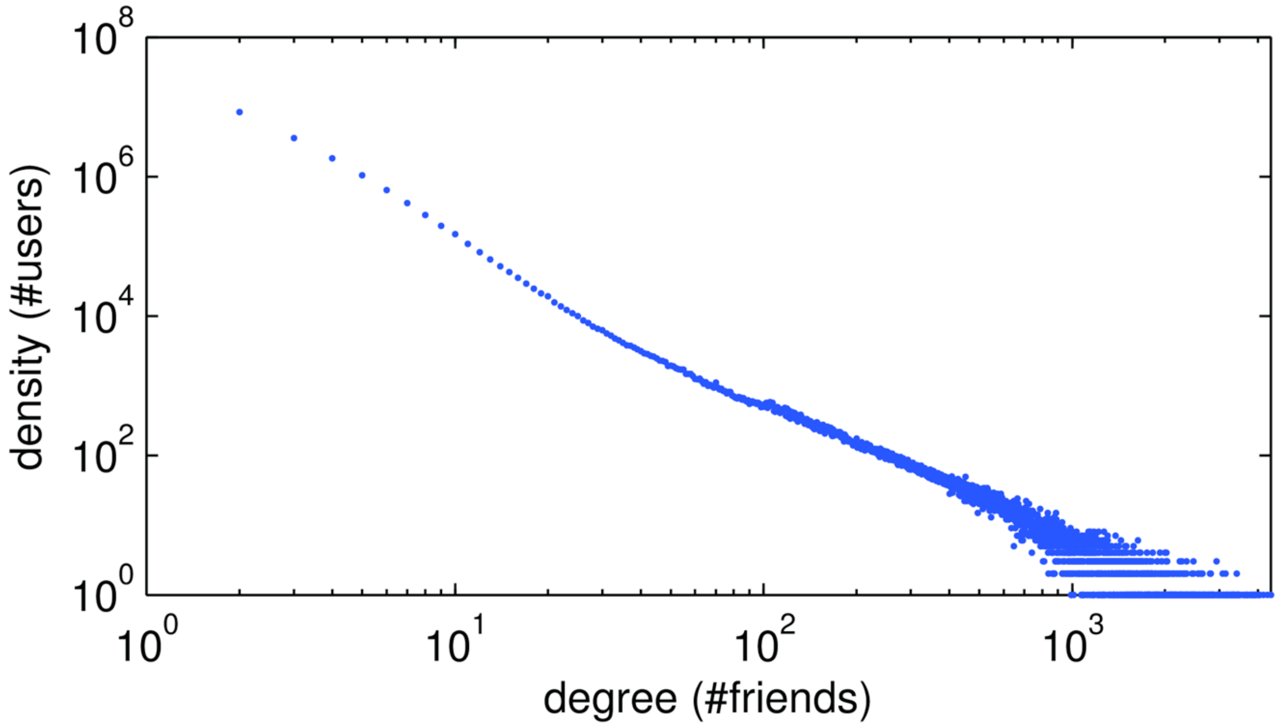
binary hash

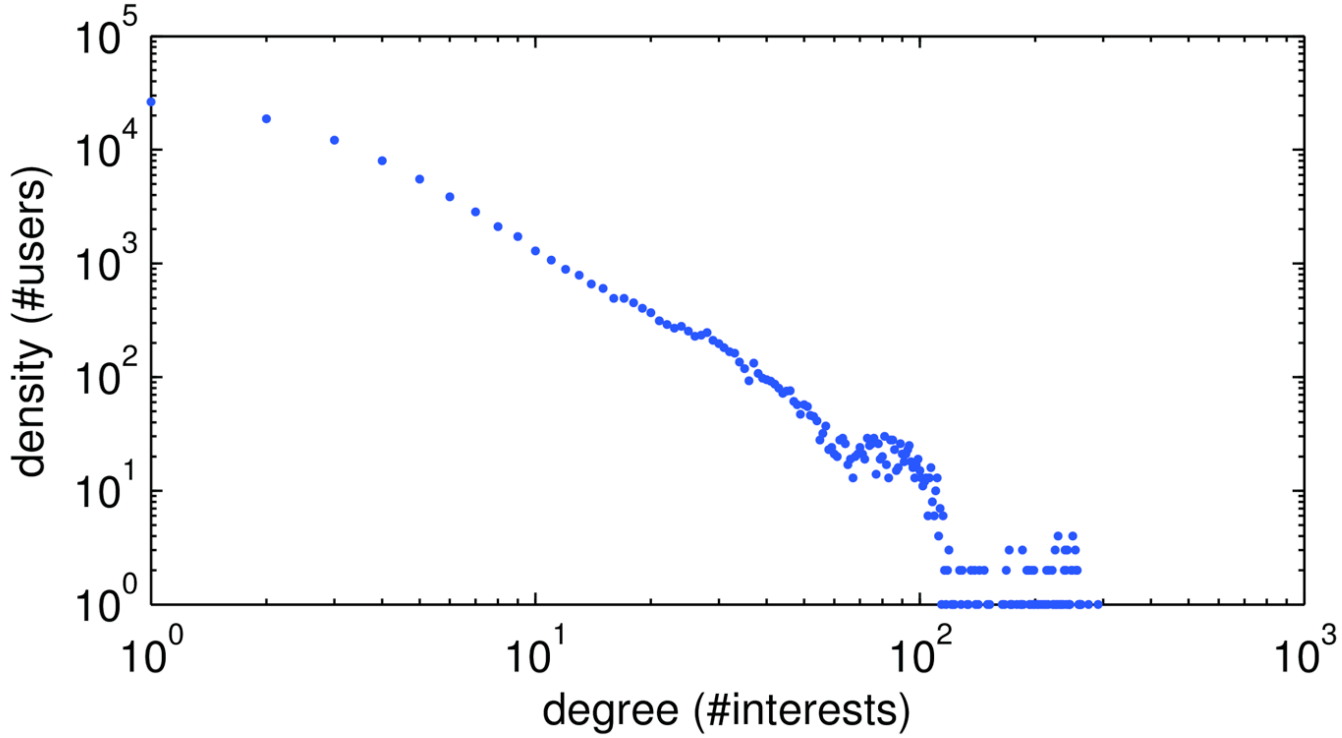
hash

Y! Pulse


New User? Register | Sign In | Help
Make Y! My Homepage
YAHOO! PULSE
Search
Sign In
Find People
Share what's important to you
Conny Lee
Happy Friday!
... with the people you care about
All
Connect to your favorite sites
Conny Lee

Y! Pulse Data





1.2M users, 386 items
6.1M friend connections
29M interest indications

App Recommendation

Models	loss	$\Omega[\cdot]$	MAP@5	MAR@5	nDCG@5
SIM			0.630	0.186	0.698
RLFM			0.729	0.211	0.737
NLFM			0.748	0.222	0.761
FIP	ℓ_2	ℓ_2	0.768	0.228	0.774
FIP	lazy ℓ_2	ℓ_2	0.781	0.232	0.790
FIP	logistic	ℓ_2	0.781	0.232	0.793
FIP	Huber	ℓ_2	0.781	0.232	0.794
FIP	Ψ	ℓ_2	0.777	0.231	0.771
FIP	ℓ_2	ℓ_1	0.778	0.231	0.787
FIP	lazy ℓ_2	ℓ_1	0.780	0.231	0.791
FIP	logistic	ℓ_1	0.779	0.231	0.792
FIP	Huber	ℓ_1	**0.786**	**0.233**	**0.797**
FIP	Ψ	ℓ_1	0.765	0.215	0.772

SIM: similarity based model;
RLFM: regression based latent factor model (Chen&Agarwal); NLFM: SIM&RLFM

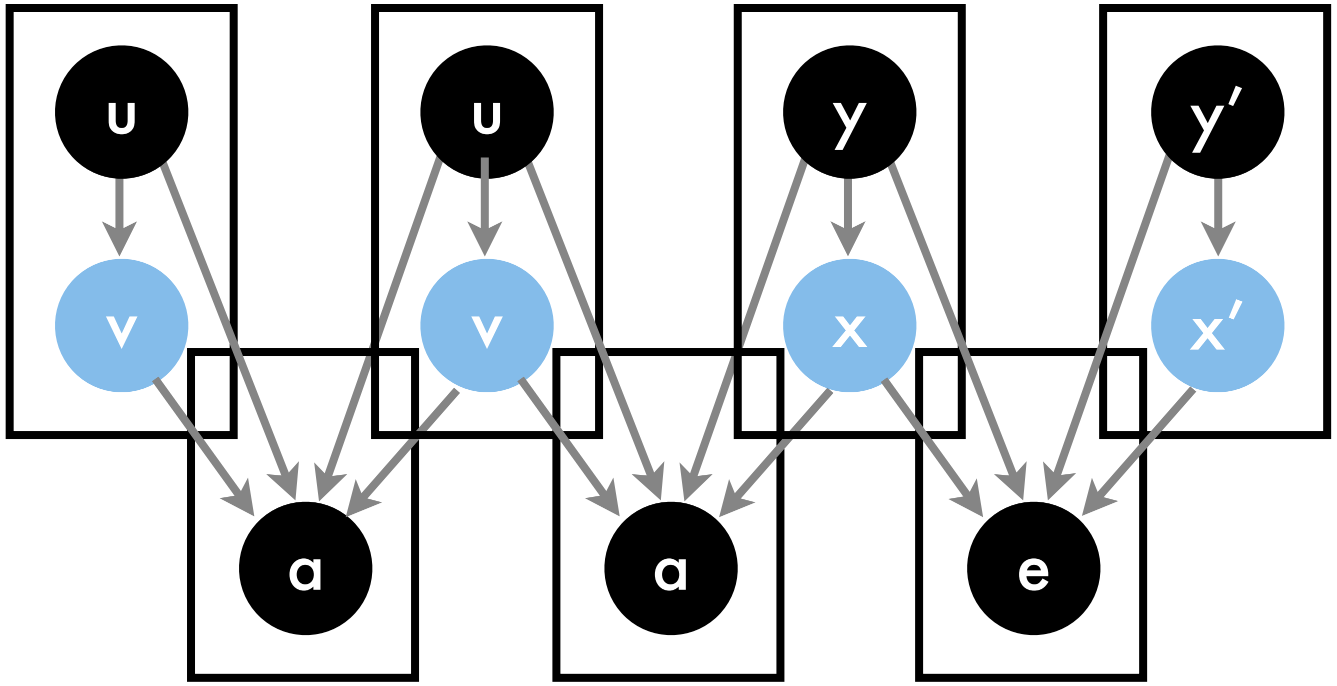
Social recommendation

Models	loss	$\Omega[\cdot]$	MAP@5	MAR@5	nDCG@5
RLFM			0.164	0.202	0.174
FIP	ℓ_2	ℓ_2	**0.359**	**0.284**	**0.244**
FIP	lazy ℓ_2	ℓ_2	0.193	0.269	0.200
FIP	logistic	ℓ_2	0.174	0.220	0.189
FIP	Huber	ℓ_2	0.210	0.234	0.215
FIP	Ψ	ℓ_2	0.187	0.255	0.185
FIP	ℓ_2	ℓ_1	0.186	0.230	0.214
FIP	lazy ℓ_2	ℓ_1	0.180	0.223	0.194
FIP	logistic	ℓ_1	0.183	0.217	0.189
FIP	Huber	ℓ_1	0.188	0.222	0.200
FIP	Ψ	ℓ_1	0.178	0.208	0.179

Extensions

- Multiple relations

  (user, user)
  (user, app)
  (app, advertisement)



Summary

- Factorization model for users
- Integration of preferences helps both social and app recommendation
- Simple stochastic gradient descent algorithm
- Hashing for memory compression
- Extensible framework