Performance 2020

Infer User Preferences from Aggregate Measurements: A Novel Message Passing Algorithm for Privacy Attack

Du Su, Yi Lu

University of Illinois, Urbana-Champaign

Data Privacy

- Social media platforms
  - Identity, preference
  - Optimize algorithm
- Data security threats
  - Addiction, abuse

- Adobe. Date: October 2013. Impact: 153 million user records. ...
- Adult Friend Finder. Date: October 2016. Impact: 412.2 million accounts. ...
- Canva. Date: May 2019. ...
- **eBay**. Date: May 2014. ...
- Equifax. Date: July 29, 2017. ...
- Dubsmash. Date: December 2018. ...
- **Heartland Payment Systems**. Date: March 2008. ...
- LinkedIn. Date: 2012 (and 2016)

Traditional privacy protection

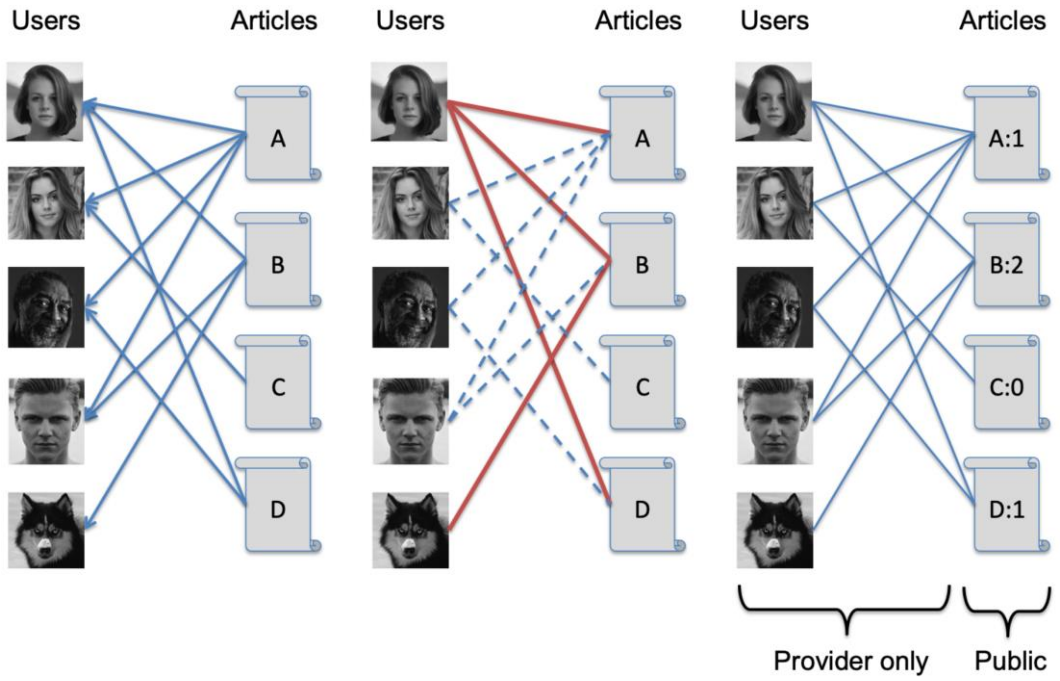
- Aggregation
  - No storage of sensitive individual record
  - Total reads / reviews

Traditional privacy protection

- Aggregation
- High degree of centralization
  - Actively choose how to aggregate
  - Reversable: recover user preference

Recover preferences: problem

- Content pushing
  - Push, read, aggregate
- Inferring
  - Recover user preference from aggregated reads

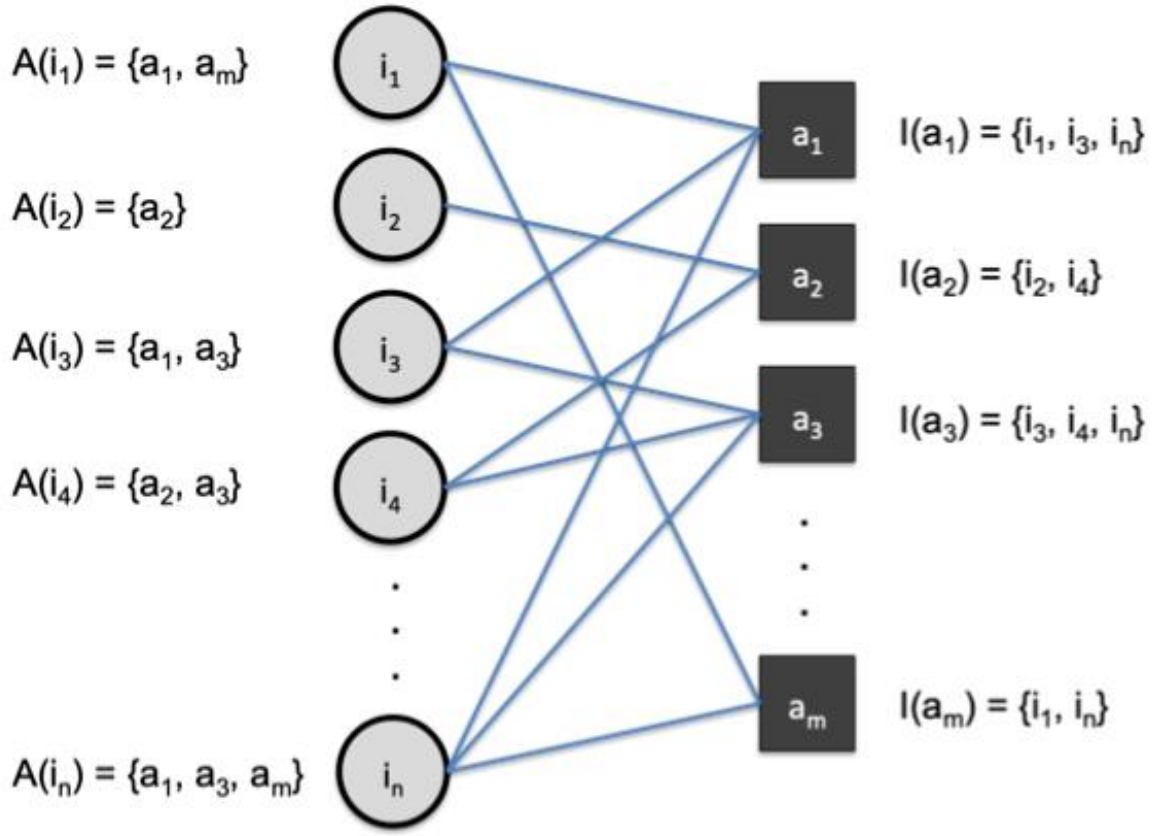


Recover preferences: problem

- Goal
  - Small number of articles pushed to user
  - Inference algorithm is of low complexity
  - As few articles as possible

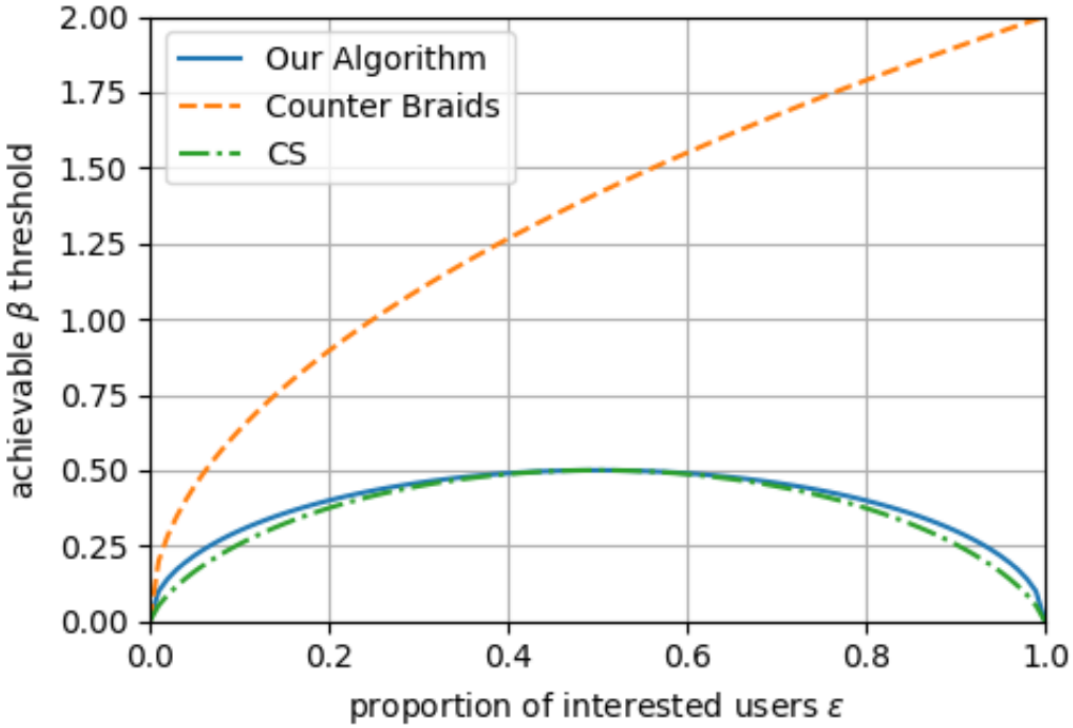
Recover preferences: formulation

- m articles: a = 1, 2, …, m
- n users: i = 1, 2, …, n
  - Preferences: $p_i \in \{0,1\}$
  - ϵ are interested
- a pushed to a set of users $\mathrm{I}(a)$
  - User view if $p_i = 1$.
  - Aggregate reads $r_a = \sum_{i \in \mathrm{I}(a)} p_i$
- Goal:
  - Recover p_i
  - Minimize $\beta = \frac{m}{n}$



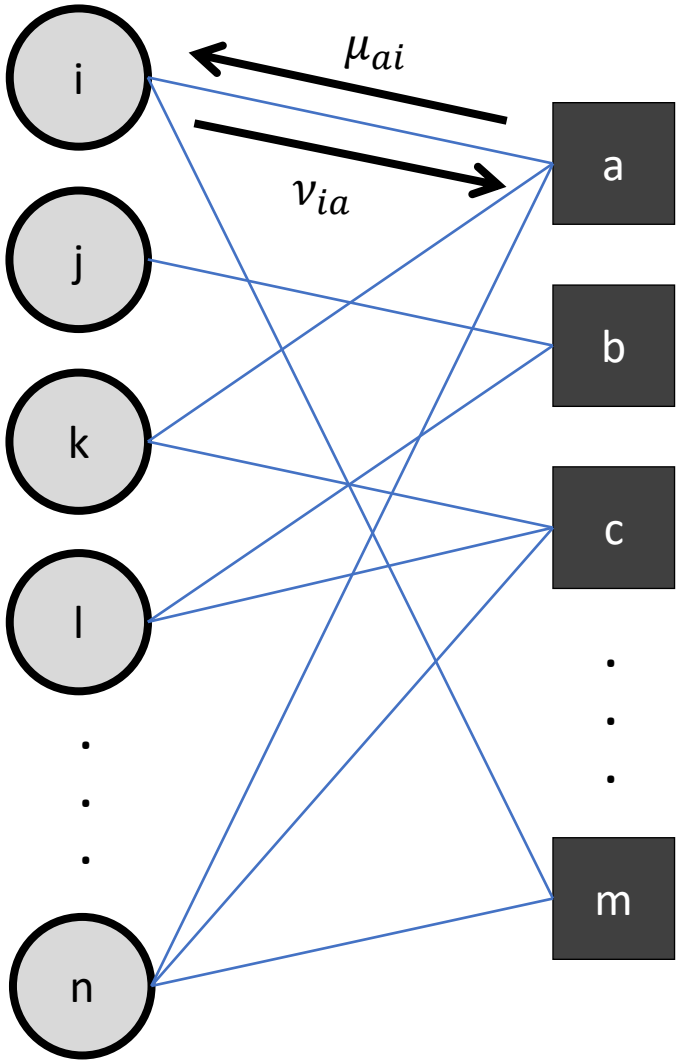
Inferring algorithm: related work

- Compressing sensing
  - $O(n^3)$ complexity
- Counter Braids
  - $O(n)$ complexity
  - Optimal ratio: $\beta = 2\sqrt{\epsilon}$
- Our problem
  - $O(n)$ complexity
  - Optimal ratio: $\beta = \sqrt{\epsilon(1-\epsilon)}$



Inferring algorithm: related work

- Counter Braids
  - low complexity message passing decoding
- Our problem
  - ν_{ia}: guess user i's preference on user side
  - μ_{ai}: guess user i's preference on article side

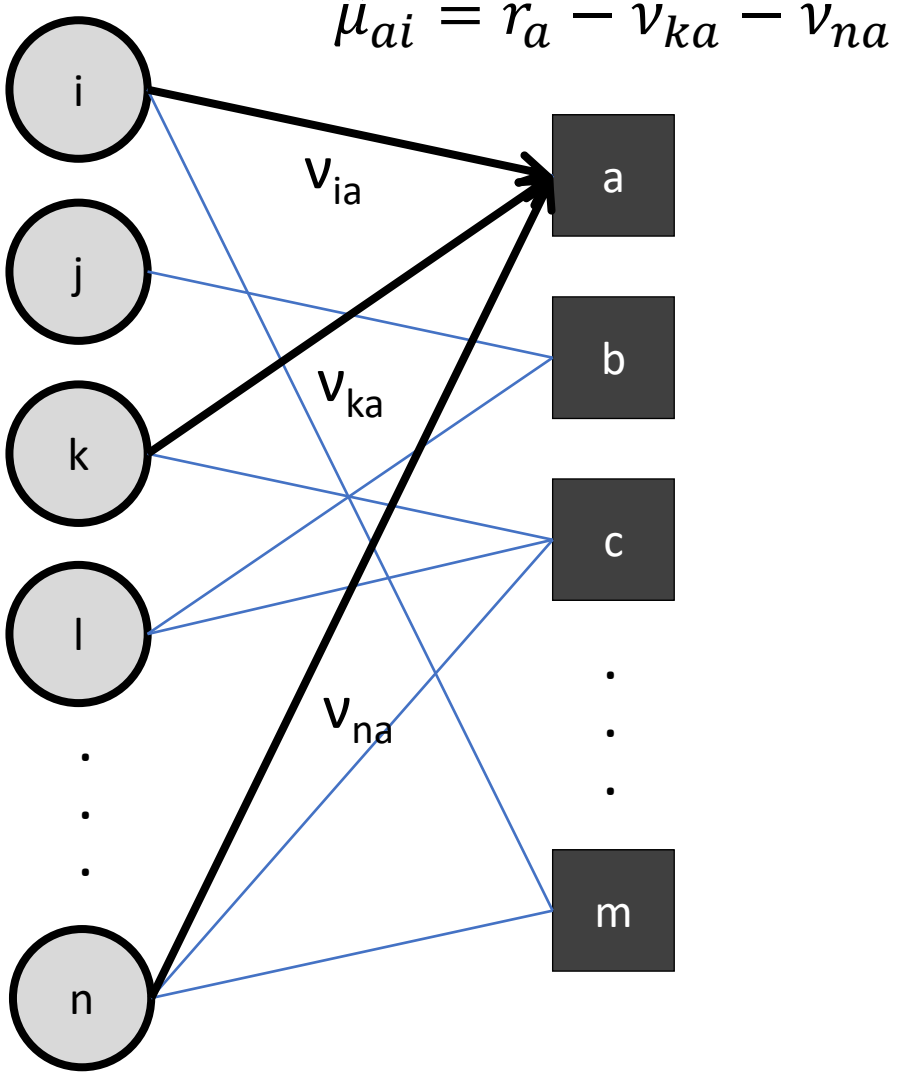


Inferring algorithm: message passing

- ν_{ia}: passed to article node a
  - Initialize to 0
- Article node a infer user i's preference
  - based on other users in $\mathrm{I}(a)$

$$\hat{\mu}_{ai}(t) = r_a - \sum_{j \in I(a), j \neq i} \nu_{ja}(t-1)$$

$$\hat{\mu}_{ai} = r_a - \nu_{ka} - \nu_{na}$$



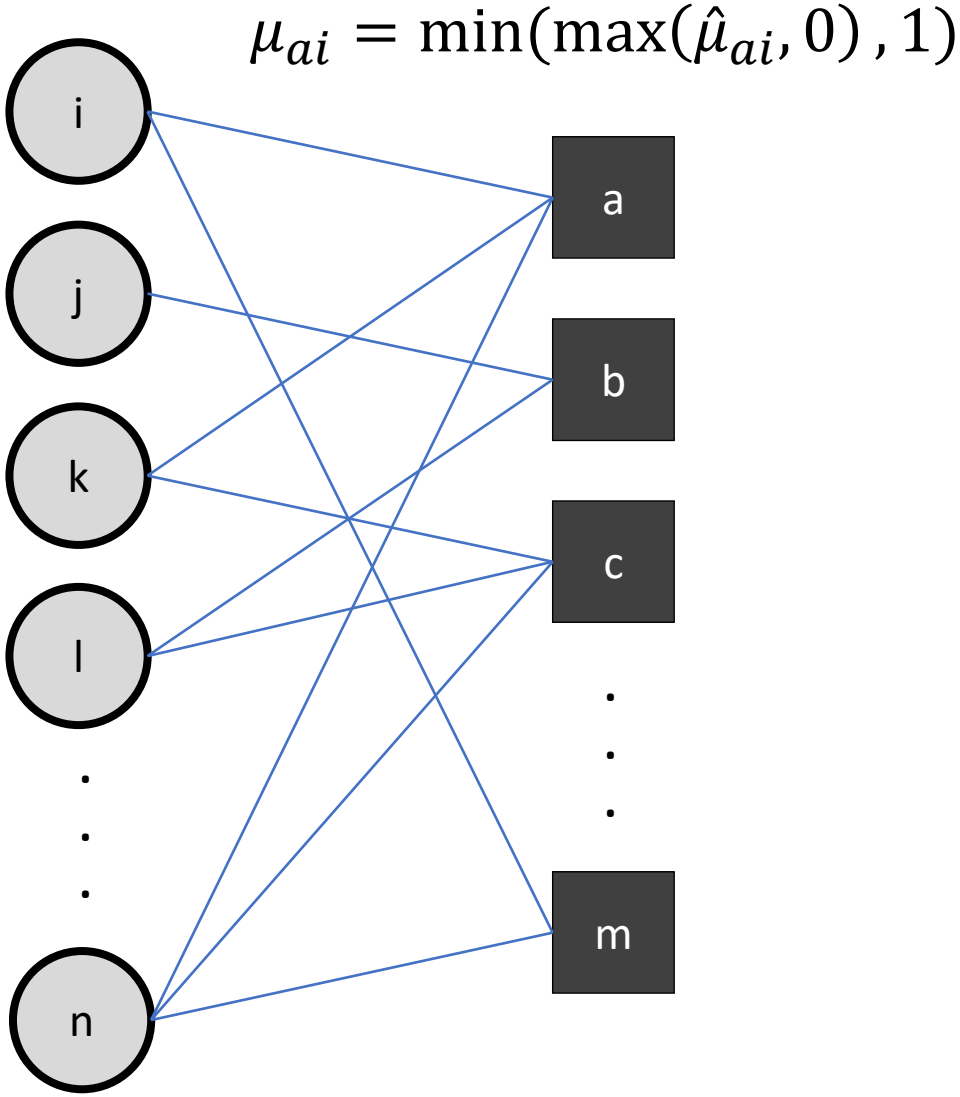
Inferring algorithm: message passing

- v_{ia}: passed to article node a
  - Initialize to 0
- Article node a infer user i's preference
  - based on other users in I(a)

$$\hat{\mu}_{ai}(t) = r_a - \sum_{j \in I(a), j \neq i} v_{ja}(t-1)$$

  - Apply bounds

$$\mu_{ai} = \min(\max(\hat{\mu}_{ai}, 0), 1)$$

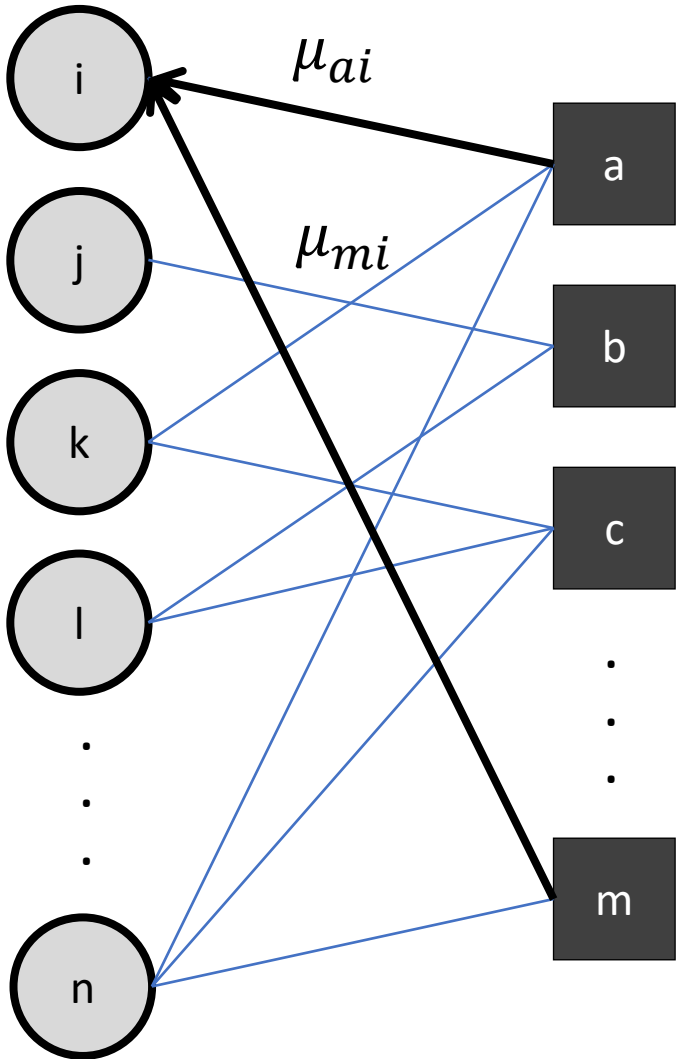


Inferring algorithm: message passing

- μ_{ai} : passed to user node i
- user node i infer its preference
  - based on articles in $A(i)$
  - Odd round is upper bound, even round is lower bound

$$\nu_{ia}(t) = \begin{cases} \min_{b \in A(i),} \mu_{bi}(t) & \text{if } t \text{ is odd,} \\ \max_{b \in A(i),} \mu_{bi}(t) & \text{if } t \text{ is even.} \end{cases}$$

$$\nu_{ia} = \min(\mu_{ai}, \mu_{mi})$$



Push algorithm: Construct bipartite graph

- Degree distribution
  - Edge-perspective degree distribution

$$\lambda(x) = \sum_{k=1}^{l_{max}} \lambda_k x^{k-1} \qquad \rho(x) = \sum_{k=1}^{r_{max}} \rho_k x^{k-1}$$

  - Average number of articles

$$\beta = \frac{m}{n} = \frac{l_{avg}}{r_{avg}} = \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$$

Error analysis: density evolution

- Degree distribution: $\lambda(x) = \sum_{k=1}^{l_{max}} \lambda_k x^{k-1}$ $\rho(x) = \sum_{k=1}^{r_{max}} \rho_k x^{k-1}$
- Article message error: $\sum_k \rho_k (1 - (1 - x_{t-1})^{k-1}) = 1 - \rho(1 - x_{t-1})$
- User message: $\lambda[1 - \rho(1 - x_{t-1})]$
- Projection
  - Odd round: $(1 - \epsilon)\lambda[1 - \rho(1 - x_{t-1})]$
  - Even round: $\epsilon\lambda[1 - \rho(1 - x_{t-1})]$
  - Two rounds: $f(\epsilon, x) = \epsilon\lambda\{1 - \rho[1 - (1 - \epsilon)\lambda(1 - \rho(1 - x))]\}$

Prove of optimal beta

- Binary erasure channel
  - Error evolution $f(\epsilon, x) = \epsilon\lambda(1 - \rho(1 - x))$
  - Capacity-achieving: $\hat{\lambda}_\alpha(x) \triangleq 1 - (1 - x)^\alpha = \sum_{i=1}^{\infty} \binom{\alpha}{i} (-1)^{i-1} x^i, \quad \rho_\alpha(x) \triangleq x^{\frac{1}{\alpha}}$
  - Optimal ratio $\beta = \epsilon$: $\frac{N}{\alpha}\binom{\alpha}{N}(-1)^{N-1} = 1 - \epsilon$
- Our problem
  - Error evolution: $f(\epsilon, x) = \epsilon\lambda\{1 - \rho[1 - (1 - \epsilon)\lambda(1 - \rho(1 - x))]\}$

Prove of optimal beta

- Binary erasure channel
  - Error evolution: $f(\epsilon, x) = \epsilon\lambda(1 - \rho(1 - x))$
- Our problem
  - Error evolution: $f(\epsilon, x) = \epsilon\lambda\{1 - \rho[1 - (1 - \epsilon)\lambda(1 - \rho(1 - x))]\}$.
  - Optimal ratio: $\beta = \sqrt{\epsilon(1 - \epsilon)}$

Phase transition

- $\epsilon \leq \epsilon^*$, user preference can be recovered
- $\epsilon > \epsilon^*$, some user preference cannot be inferred
- Lack of monotonicity:

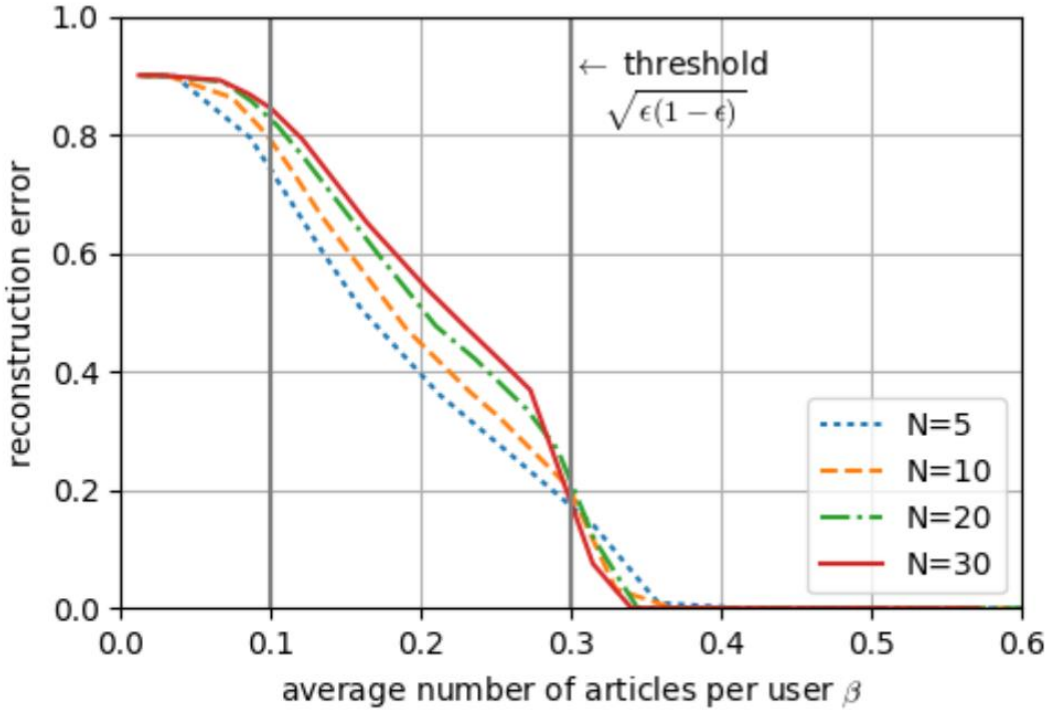
$$f(\epsilon, x) = \epsilon\lambda\{1 - \rho[1 - (1 - \epsilon)\lambda(1 - \rho(1 - x))]\},$$

- Capacity achieving graph:

$$\lim_{N\to\infty} f^{(N)}(\epsilon, x) = \frac{\epsilon(1-\epsilon)}{\epsilon^*(1-\epsilon^*)}x$$

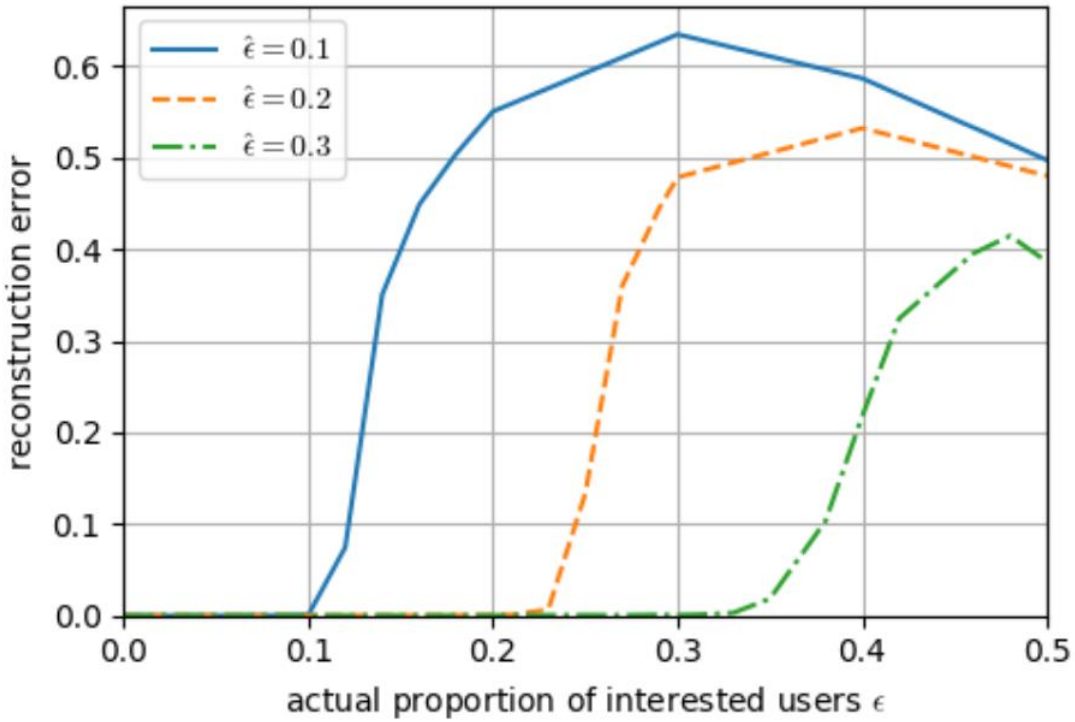
Performance of capacity achieving graph

- Setup
  - N=5, 10, 20, 30
  - 10000 users
  - Proportion of interested users: 0.1
- Optimality
  - Converges at $\beta = 0.34$
  - Optimal ratio: $\beta = \sqrt{\epsilon(1-\epsilon)} = 0.3$



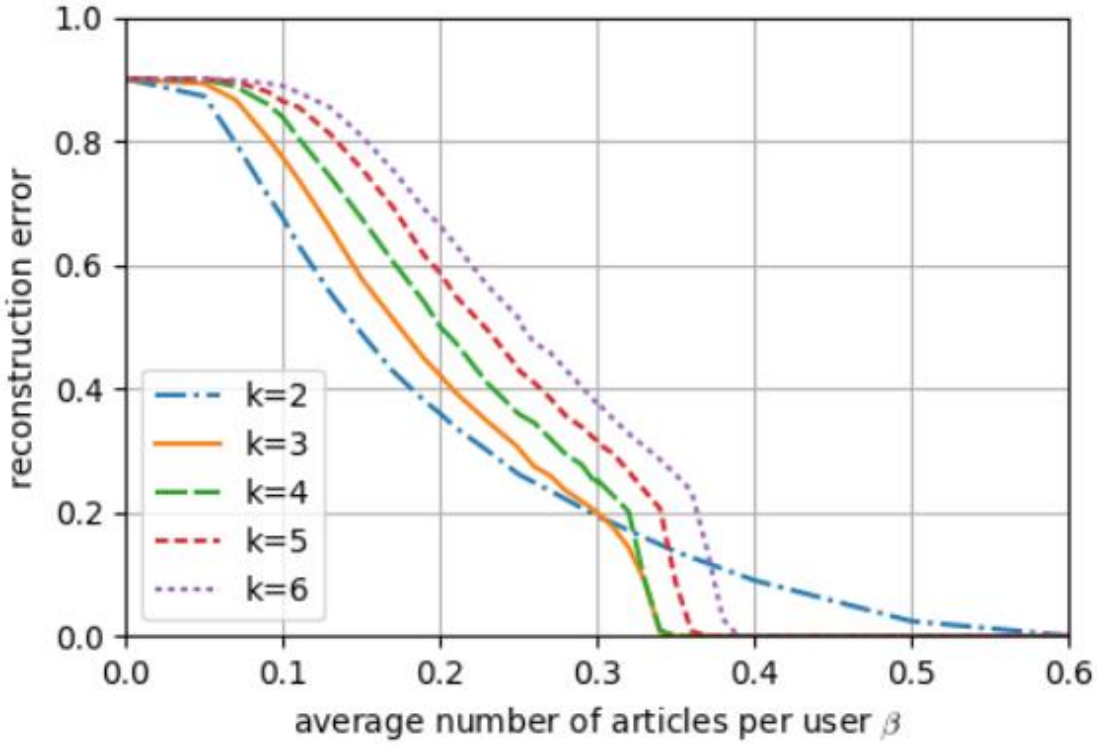
Phase transition in capacity achieving graph

- Setup
  - $\hat{\epsilon}$= 0.1, 0.2, 0.3
  - N = 30
- Phase transition



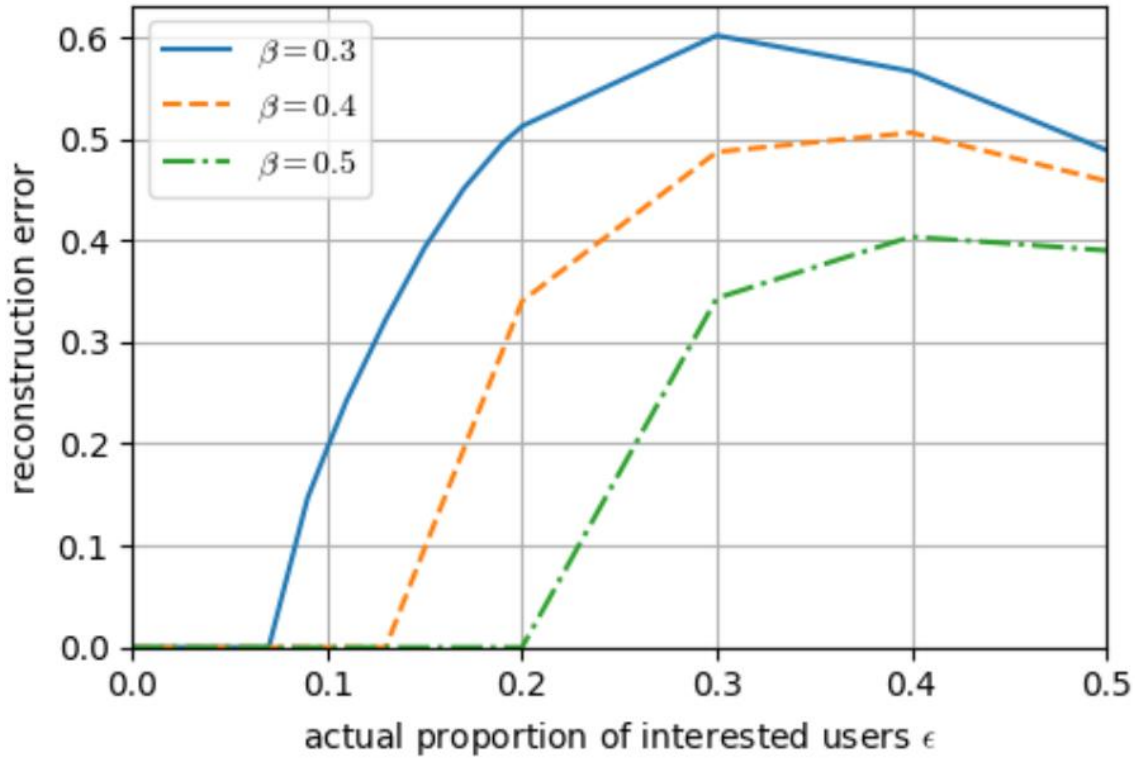
Performance of k-regular graph

- Setup
  - k=2, 3, 4, 5, 6
  - 10000 users
  - Proportion of interested users: 0.1
- Best performance with k = 3, 4
  - Converges at $\beta = 0.34$
  - Optimal ratio: $\beta = \sqrt{\epsilon(1-\epsilon)} = 0.3$
  - Practically good



Phase transition in k-regular graph

- Setup
  - β= 0.1, 0.2, 0.3
  - k = 3
- Phase transition
  - Estimate ϵ
  - Select β



Conclusion

- Pushing algorithm and inference algorithm
- Optimal ratio $\beta = \sqrt{\epsilon(1-\epsilon)}$
- Phase transition

Thanks!