Combinational Collaborative Filtering for Personalized Community Recommendation

WenYen Chen, Dong Zhang, Edward Y. Chang, KDD ’08

Presented by Shaddi Hasan
March 24, 2009

The University of North Carolina at Chapel Hill

Overview

- **Problem Statement**
- The CCF Model
  - Derivation
  - Training Technique
- Experimental Results
  - Training
  - Orkut Dataset
  - Scalability

“In recent articles, users complained they would soon require a full-time employee to manage their sizable social networks.”

Problem: Community Recommendations

facebook Home Profile Friends Inbox 157 Shaddi Hasan Settings Logout

Every Man Should Own A Trebuchet

Global

View Discussion Board
Invite People to Join
Leave Group

Share +

Basic Info

Type: Just for Fun - Outlandish Statements
Description: The Title Says It All

Members

Displaying 8 of 348 members See All


Dan Tedrick


Gregg Sandow


Jacob Max Morrill


Rick Anderson


Ondřej Špalek


N- d Schultz


Eric Downs


Jan Pavelka

Discussion Board

Displaying 3 of 4 discussion topics Start New Topic | See All

How much have you spent on your trebuchet?
2 posts by 2 people. Updated on January 8, 2009 at 12:04pm

Topic deleted on November 8, 2006 at 9:49pm

Who has the most efficient trebuchet?!?
7 posts by 6 people. Updated on December 21, 2008 at 10:27am

Officers

Magnus Alvestad (Norway)
Man

Applications

facebook Home Profile Friends Inbox 157 Shaddi Hasan Settings Logout

Every Man Should Own A Trebuchet

Global

Basic Info

Type: Just for Fun - Outlandish Statements

Description: The Title Says It All

Members

Displaying 8 of 348 members


Dan Tedrick


Gregg Sandow


Jacob Max Morrill


Rick Anderson


Ondřej Špalek


N- d Schultz


Eric Downs


Jan P

Discussion Board

Displaying 3 of 4 discussion topics Start New To

How much have you spent on your trebuchet?

2 posts by 2 people. Updated on January 8, 2009 at 12:04pm

Topic deleted on November 8, 2006 at 9:49pm

Who has the most efficient trebuchet?!?

7 posts by 6 people. Updated on December 21, 2008 at 10:27am

Applications

See All

Post

rth, TX) wrote

one day, and toss many a heavy item with it!!

H) wrote

a trebuchet - my husband made one for my birthday gift
it - we have a blast throwing pumpkins every fall.

ote

ideo of our whipping trebuchet that we entered into the
ace we tossed a 9lb pumpkin 1700'

ycbMg46jDwk

Arne Jæger (Norway)
Chief trebuchetist

Joe Bettridge (Barrie, ON)
also man

Jack Fuller (St. Bonaventure)
Man

Group Type

This is an open group. Anyone can join and invite others to join.

Admins

- Magnus Alvestad (Norway)

Related Groups

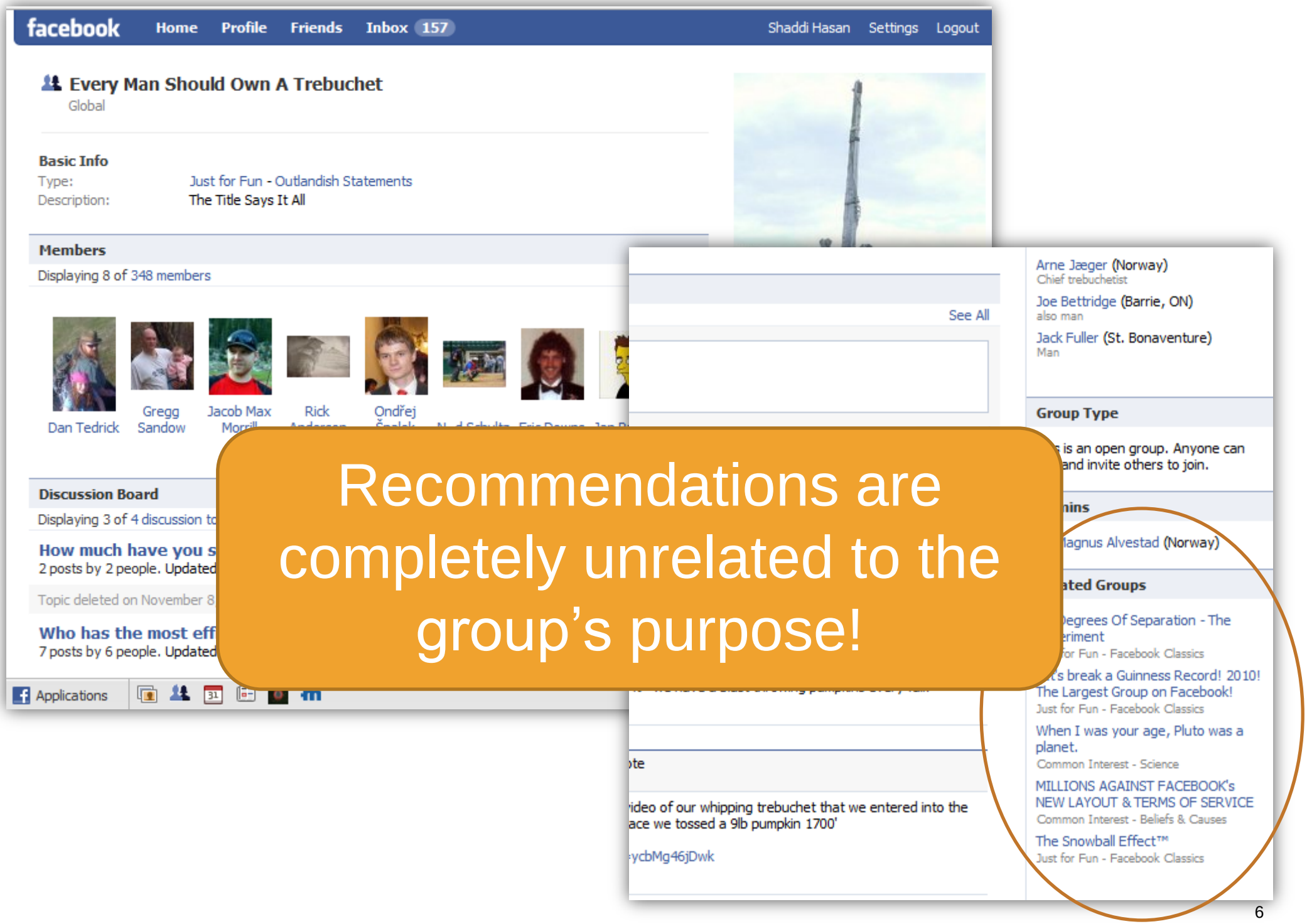
Six Degrees Of Separation - The Experiment
Just for Fun - Facebook Classics

Let's break a Guinness Record! 2010! The Largest Group on Facebook!
Just for Fun - Facebook Classics

When I was your age, Pluto was a planet.
Common Interest - Science

MILLIONS AGAINST FACEBOOK's NEW LAYOUT & TERMS OF SERVICE
Common Interest - Beliefs & Causes

The Snowball Effect™
Just for Fun - Facebook Classics


facebook
Home
Profile
Friends
Inbox 157
Shaddi Hasan
Settings
Logout
Every Man Should Own A Trebuchet
Global
Basic Info
Type:
Just for Fun - Outlandish Statements
Description:
The Title Says It All
Members
Displaying 8 of 348 members
Dan Tedrick
Gregg Sandow
Jacob Max
Rick
Ondřej
Discussion Board
How much have you s
2 posts by 2 people. Updated
Topic deleted on November 8
Who has the most eff
7 posts by 6 people. Updated
Applications
See All
Arne Jæger (Norway)
Chief trebuchetist
Joe Bettridge (Barrie, ON)
also man
Jack Fuller (St. Bonaventure)
Man
Group Type
is an open group. Anyone can
and invite others to join.
Magnus Alvestad (Norway)
ideo of our whipping trebuchet that we entered into the
ace we tossed a 9lb pumpkin 1700'
ycbMg46jDwk
Degrees Of Separation - The
eriment
for Fun - Facebook Classics
t's break a Guinness Record! 2010!
The Largest Group on Facebook!
Just for Fun - Facebook Classics
When I was your age, Pluto was a
planet.
Common Interest - Science
MILLIONS AGAINST FACEBOOK's
NEW LAYOUT & TERMS OF SERVICE
Common Interest - Beliefs & Causes
The Snowball Effect™
Just for Fun - Facebook Classics
Recommendations are completely unrelated to the group's purpose!

Communities are…

- **Bags of Words**
  - Set of words that describes a community
  - Can’t provide personalized results
  - Similar to document-word co-occurance
- **Bag of Users**
  - Set of participating users
  - Can’t take advantage of content similarity
  - Similar to document-citation co-occurance

Overview

- Problem Statement
- **The CCF Model**
  - Derivation
  - Training Technique
- Experimental Results
  - Training
  - Orkut Dataset
  - Scalability

Combinatorial Collaborative Filtering (CCF)

CCF considers *both* bag of words and bag of users.

- Communities: $C = \{c_1, c_2, \ldots, c_n\}$
- Community Descriptions: $D = \{d_1, d_2, \ldots, d_v\}$
- Users: $U = \{u_1, u_2, \ldots, u_M\}$
- Latent Variables: $z \in Z = \{z_1, z_2, \ldots, z_K\}$

- If user *u* joins community *c*, *n(c,u)* = 1; else, *n(c,u)* = 0
- *n(c,d)* = *R* if community *c* contains word *d* for *R* times

The C-U Model

$$P(c,u) = \sum_z P(c,u,z)$$

$$= P(c)\sum_z P(u \mid z)P(z \mid c)$$

- *c*: community uniformly selected from C
- *z*: topic selected from P(z|c)
- *u*: user chosen by sampling P(u|z)

The C-D Model

$$P(c,d) = \sum_z P(c,d,z)$$

$$= P(c)\sum_z P(d \mid z)P(z \mid c)$$

- *c*: community uniformly selected from C
- *z*: topic selected from P(z|c)
- *d*: word chosen by sampling P(d|z)

The CCF Model

$$P(c,u,d) = \sum_z P(c,u,d,z)$$

$$= P(c)\sum_z P(u \mid z)P(d \mid z)P(z \mid c)$$

- Distribution over C, U, and D
- *c*: community uniformly selected from C
- *z*: topic selected from P(z|c)
- *u*: user chosen by sampling P(u|z)
- *d*: word chosen by sampling P(d|z)

Training the recommender

- Gibbs Sampling + Expectation Maximization
  - Gibbs: Too slow for large databases

$$P(z_{i,j} = k \mid u_i = m, d_j = n, \mathbf{z}_{-i,-j}, U_{-i}, D_{-j})$$

  - E-M: Faster, but sensitive to initialization

$$L = \sum_{c,u,d} n(c,u,d) \log P(c,u,d)$$

- Used to estimate *P(z|c)*, *P(u|z)*, *P(d|z)*
  - …which parameterize CCF.

Gibbs sampling

$$P(z_{i,j} = k \mid u_i = m, d_j = n, \mathbf{z}_{-i,-j}, U_{-i}, D_{-j}) \propto$$
$$P(d_n \mid z_k) P(u_m \mid z_k) P(z_k \mid c_c)$$

where

$$P(d_n \mid z_k) = \frac{C_{nk}^{DZ} + 1}{\sum_{n'} C_{n'k}^{UZ} + V'}$$

$$P(u_m \mid z_k) = \frac{C_{mk}^{UZ} + 1}{\sum_{m'} C_{m'k}^{UZ} + M'}$$

$$P(z_k \mid c_c) = \frac{C_{ck}^{CZ} + 1}{\sum_{k'} C_{ck'}^{UZ} + K'}$$

Expectation-Maximization

- *P(z|c)*, *P(u|z)*, *P(d|z)* initialized by Gibbs sampling
- Expectation step

$$P(z \mid c,u,d) = \frac{P(u \mid z)P(d \mid z)P(z \mid c)}{\sum_{z'} P(u \mid z')P(d \mid z')P(z' \mid c)}$$

- Maximization step

$$P(u \mid z) = \frac{\sum_{c,d} n(c,u,d)P(z \mid c,u,d)}{\sum_{c,u',d} n(c,u',d)P(z \mid c,u',d)}$$

$$P(d \mid z) = \frac{\sum_{c,u} n(c,u,d)P(z \mid c,u,d)}{\sum_{c,u,d'} n(c,u,d')P(z \mid c,u,d')} \qquad P(z \mid c) = \frac{\sum_{u,d} n(c,u,d)P(z \mid c,u,d)}{\sum_{u,d,z'} n(c,u,d)P(z' \mid c,u,d)}$$

Parallel Computing

- Gibbs sampling
  - Each machine handles subset of communities
  - Master machine merges counts after each iteration
- Expectation
  - Each machine computes posterior of latent variables for subset of communities
- Maximization
  - Each machine updates $P(z|c_i)$, $P(u|z)$, $P(d|z)$
  - Communication needed to coordinate update values of $P(u|z)$ and $P(d|z)$

Inferring Recommendations

- User-community:

$$P(c_j \mid u_i) = \frac{\sum_z P(c_j, u_i, z)}{P(u_i)}$$

$$\propto \sum_z P(u_i \mid z) P(z \mid c_j)$$

- Community similarity

$$P(c_j \mid c_i) = \frac{\sum_z P(c_j, c_i, z)}{P(c_i)}$$

$$\propto \sum_z \frac{P(z \mid c_i) P(z \mid c_j)}{P(z)}$$

Overview

- Problem Statement
- The CCF Model
  - Derivation
  - Training Technique
- **Experimental Results**
  - Training
  - Orkut Dataset
  - Scalability

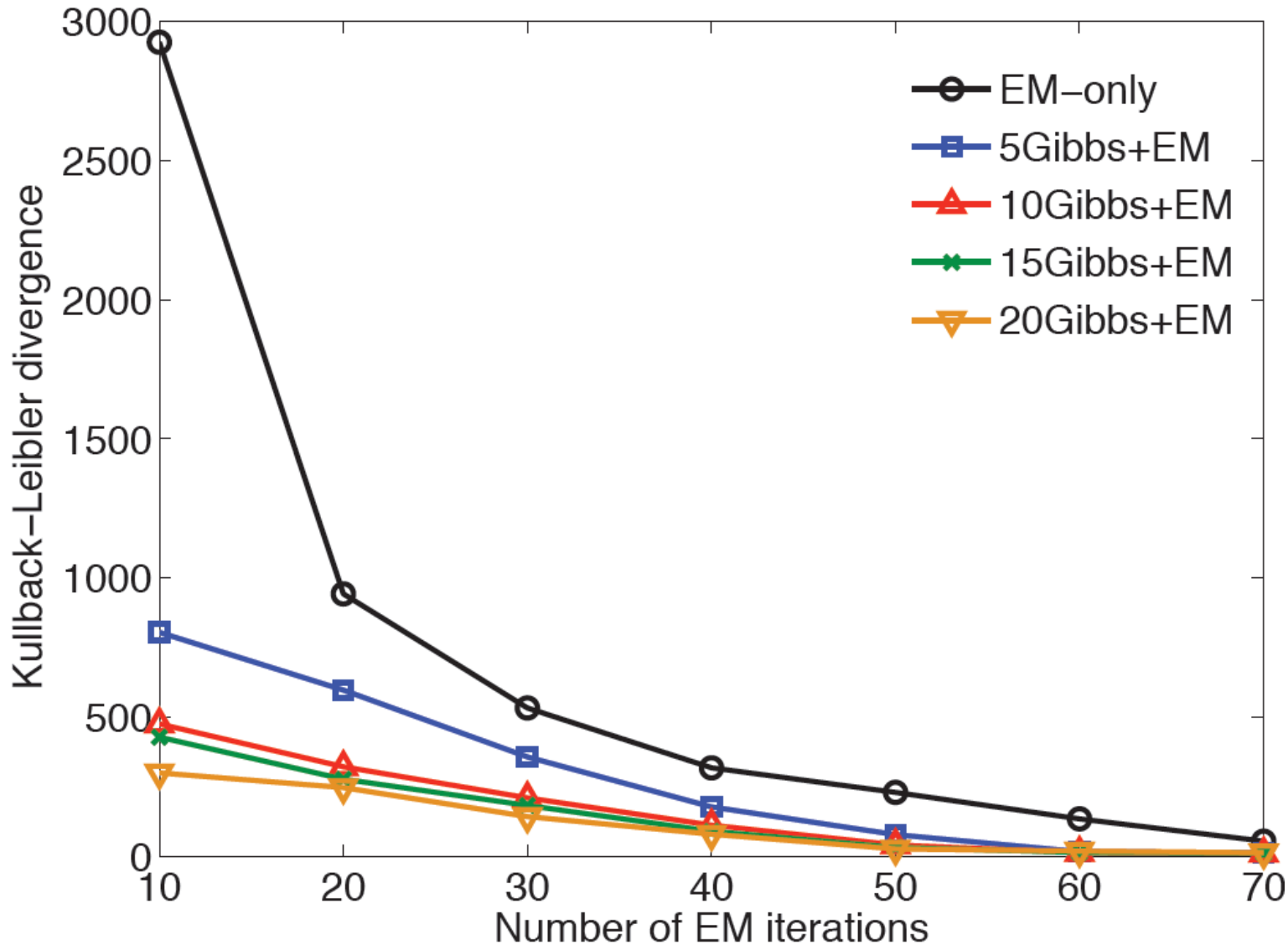
Evaluation of Training

- Synthetic dataset
  - 5,000 documents, 10 topics
  - Vocabulary size 10,000
  - 50,000,000 word tokens
- EM-only training vs. Gibbs+EM training
- Kullback-Leibler divergence:

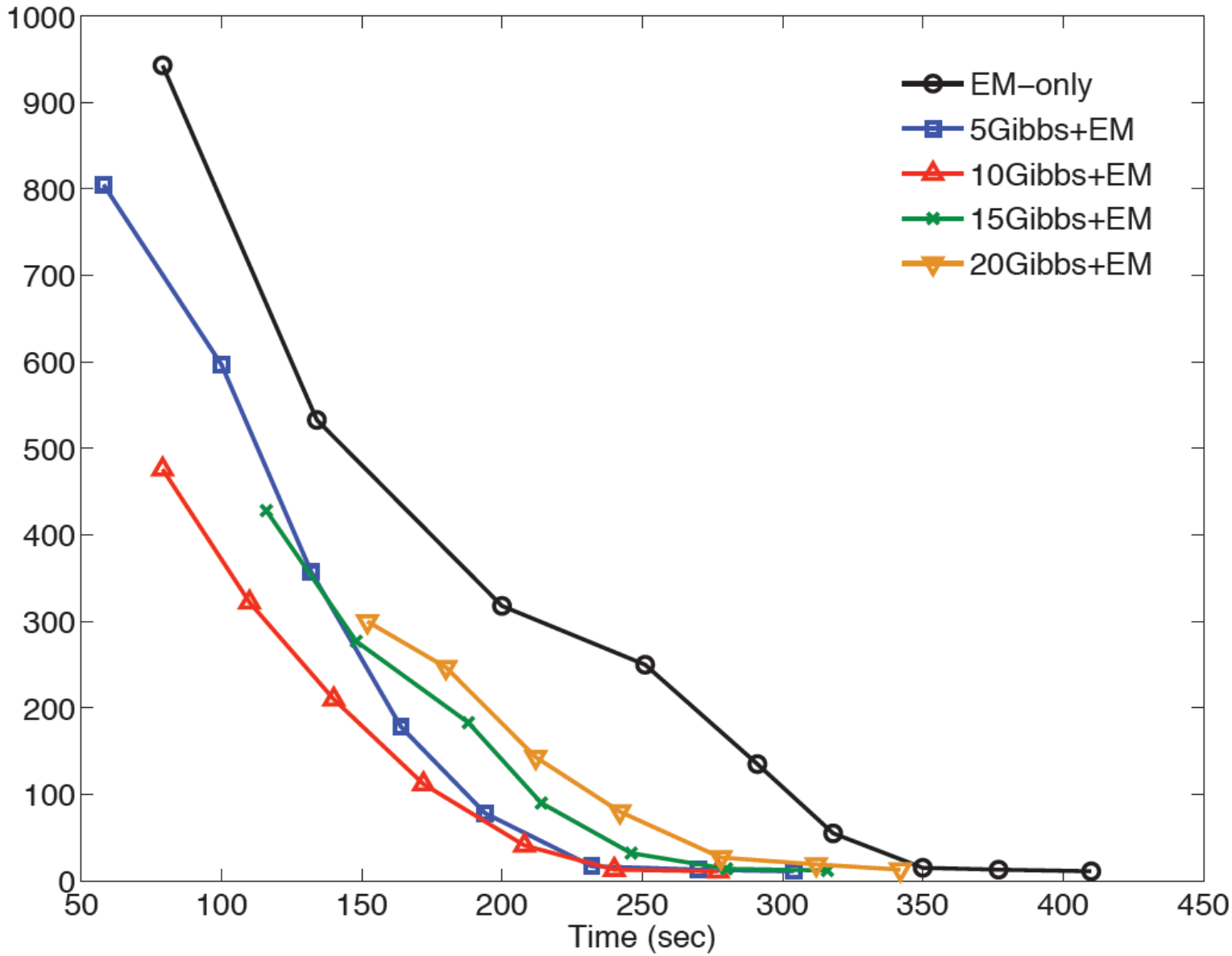
$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

  - Smaller the K-L, better estimated distribution matches actual

K-L Divergence vs. Iterations



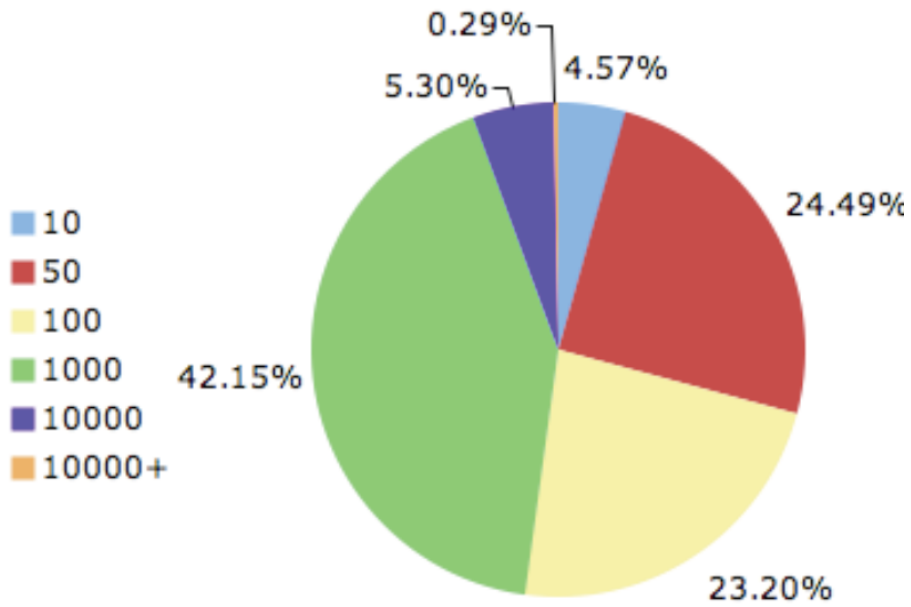
K-L Divergence vs. Time



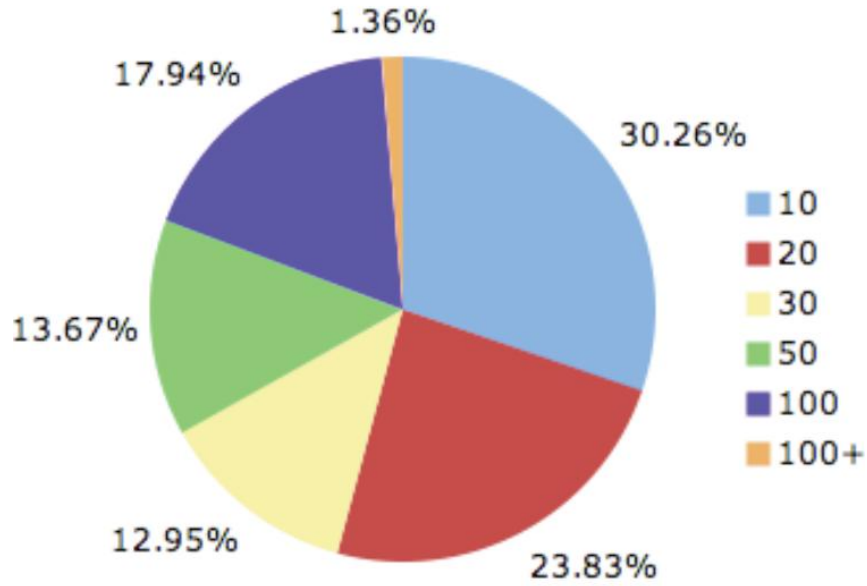
The Orkut Dataset

- 109,987 communities
  - English only
  - Membership and description information
- 312,385 users
- 35,932,001 entries in community-user matrix
  - Density: *0.001045*
- Data collected July 26, 2007

The Orkut Dataset



(A) User-per-community



(B) Word-per-community

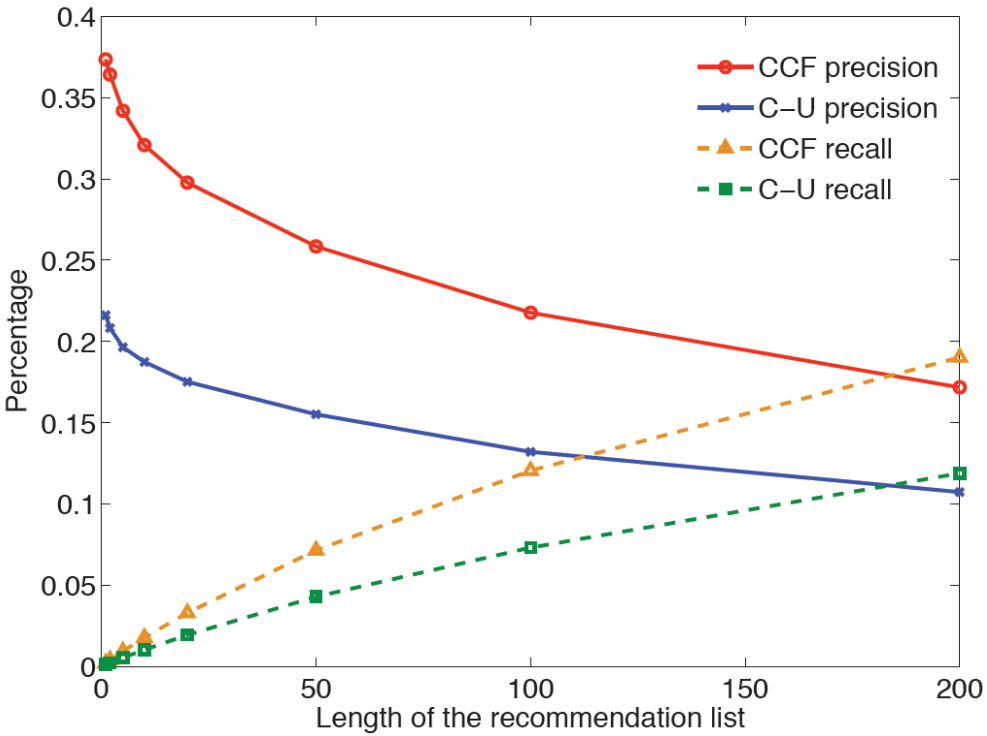
- 52% have <100 users
- 42% have 100-1000 users

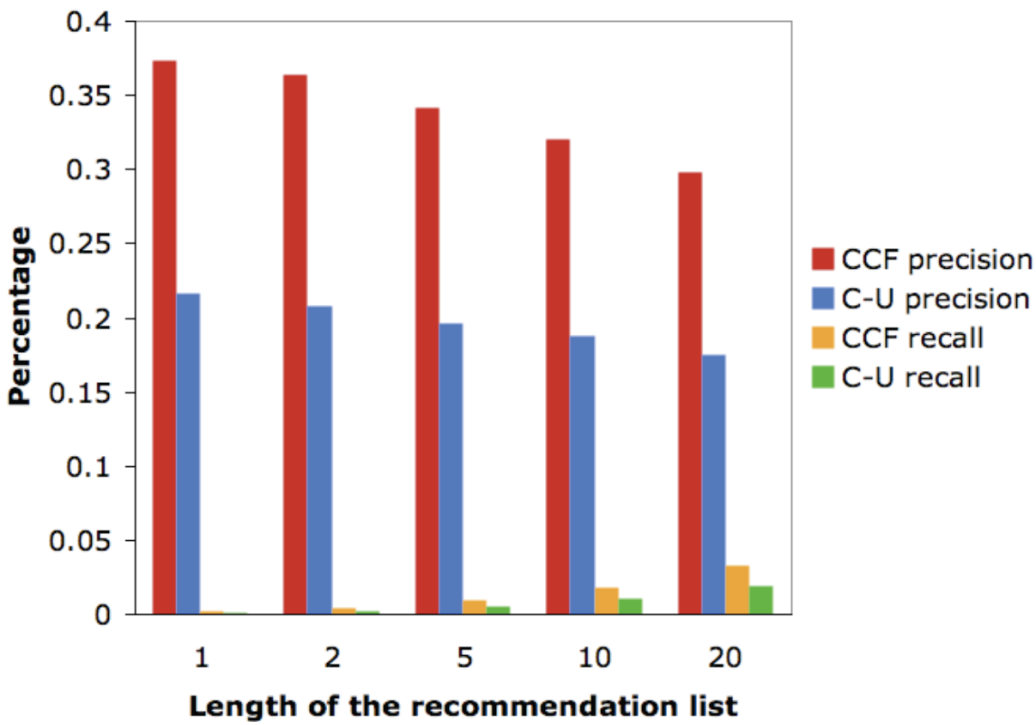
- 191,034 unique words
- 27.64 words/community

Methodology

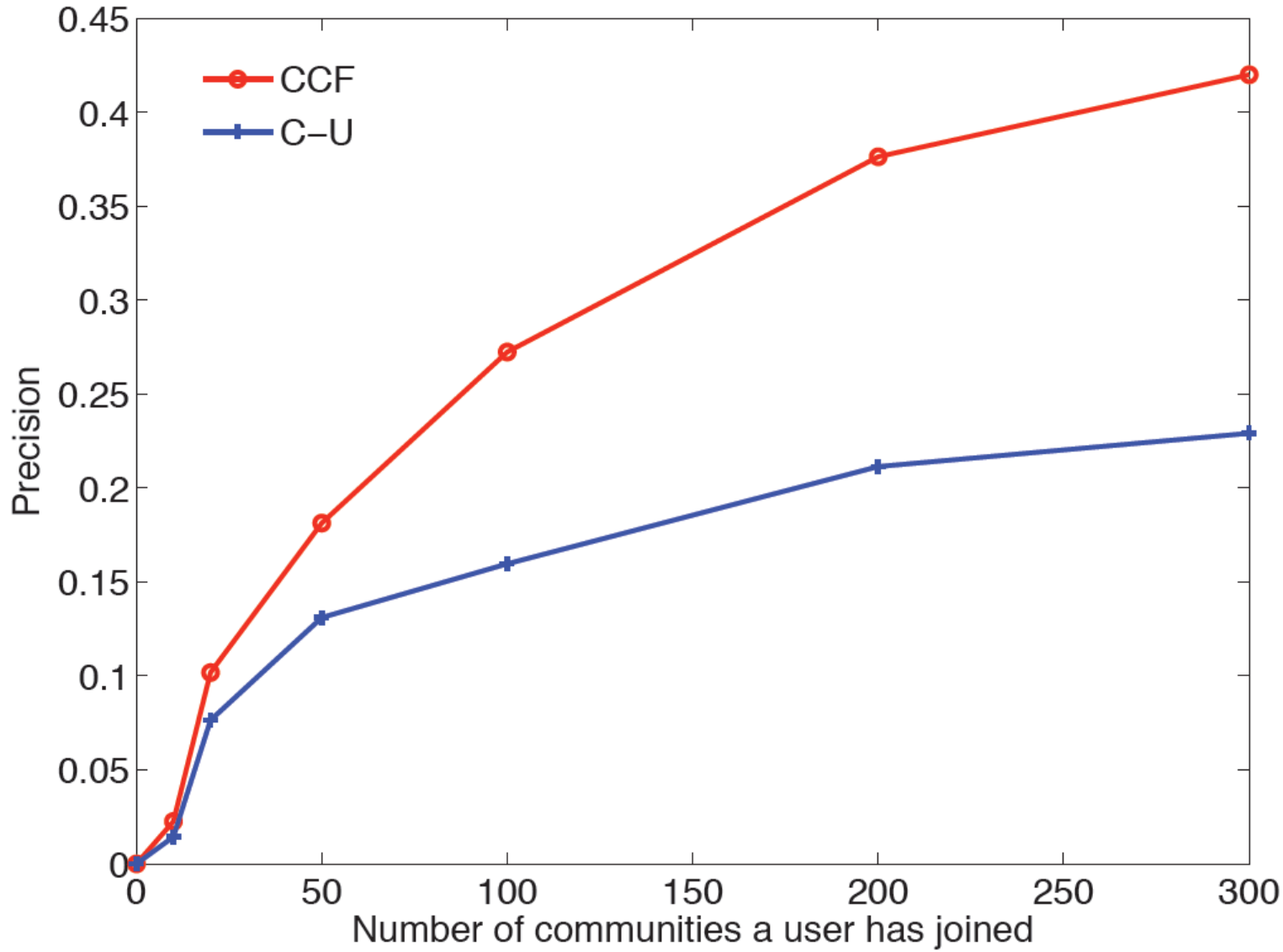
- Randomly delete one entry per user in C-U matrix
- See if deleted group can be recommended
- Each experiment repeated 10 times
- Metrics
  - $\text{Precision} = \frac{\left|\{\text{recommendation list}\} \cap \{\text{joined list}\}\right|}{\left|\{\text{recommendation list}\}\right|}$
  - $\text{Recall} = \frac{\left|\{\text{recommendation list}\} \cap \{\text{joined list}\}\right|}{\left|\{\text{joined list}\}\right|}$
  - Size of recommendation list limited to 200

Precision, Recall vs. Length





Precision vs. Number of Communities



Community Similarity

Model	C-U	C-D	CCF
NMI	0.4508	0.3127	0.4526

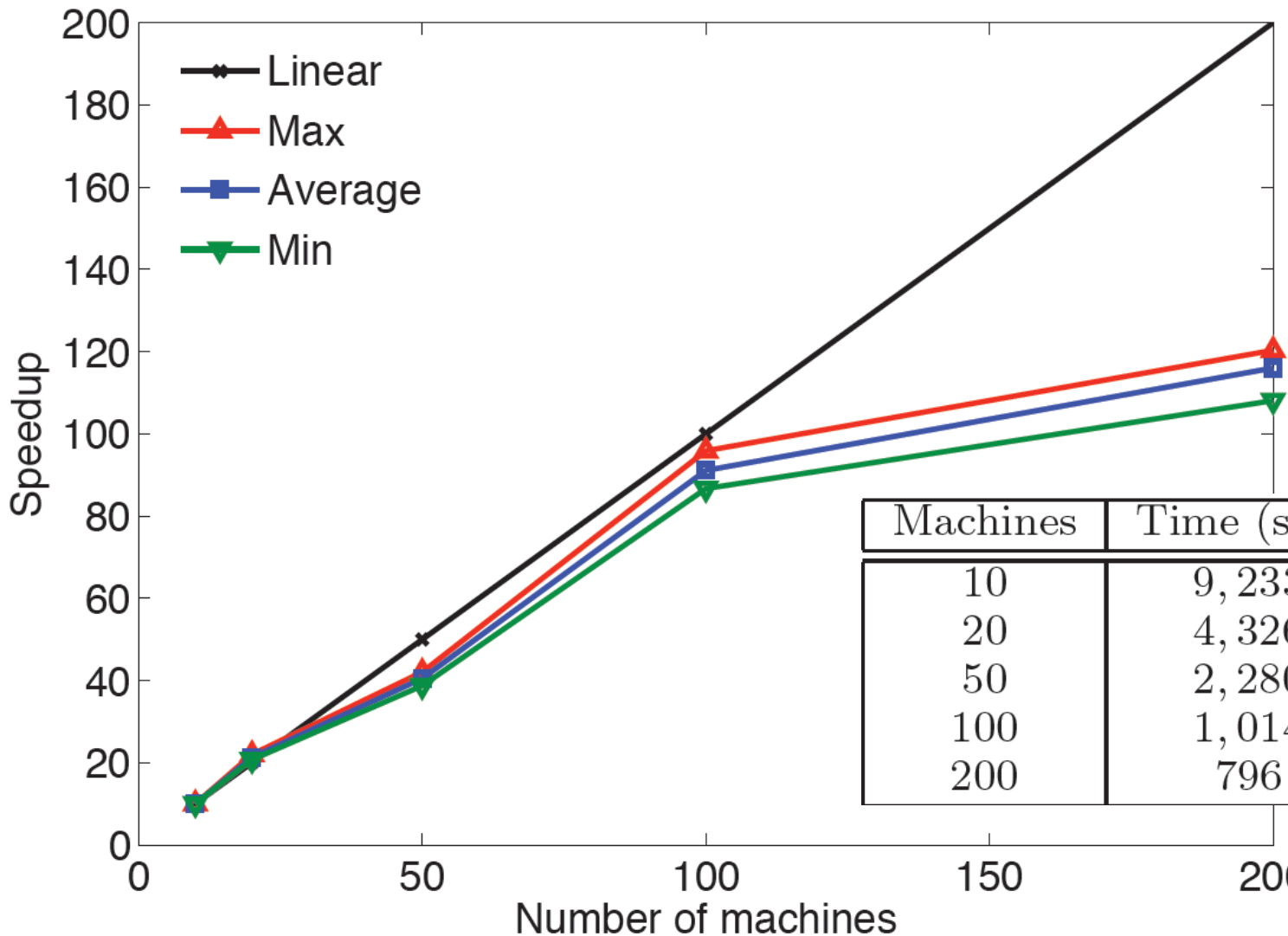
- Used P(cj|ci) to cluster communities based on topic (latent aspect)
- Used “community labels” as ground truth
- Compared clusters using Normalized Mutual Information
  - 1 indicates perfect match with ground truth, 0 is random pairing

Runtime Speedup – Methodology

- 20 Latent Aspects
- 10 Gibbs samplings
- 20 E-M iterations

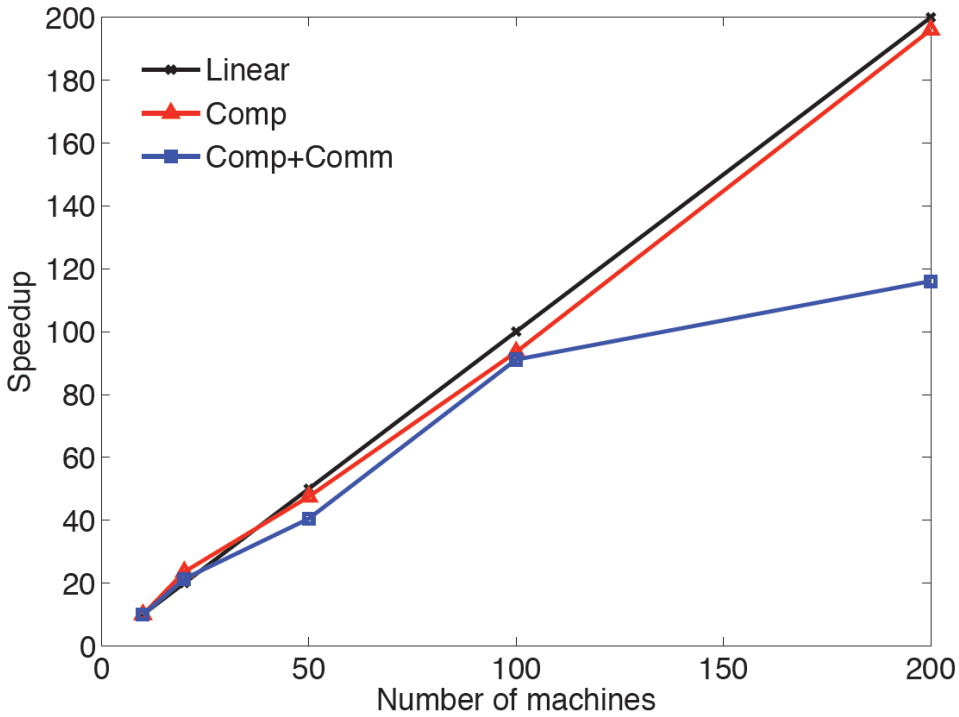
- Minimum: 10 Machines
- Maximum: 200 Machines

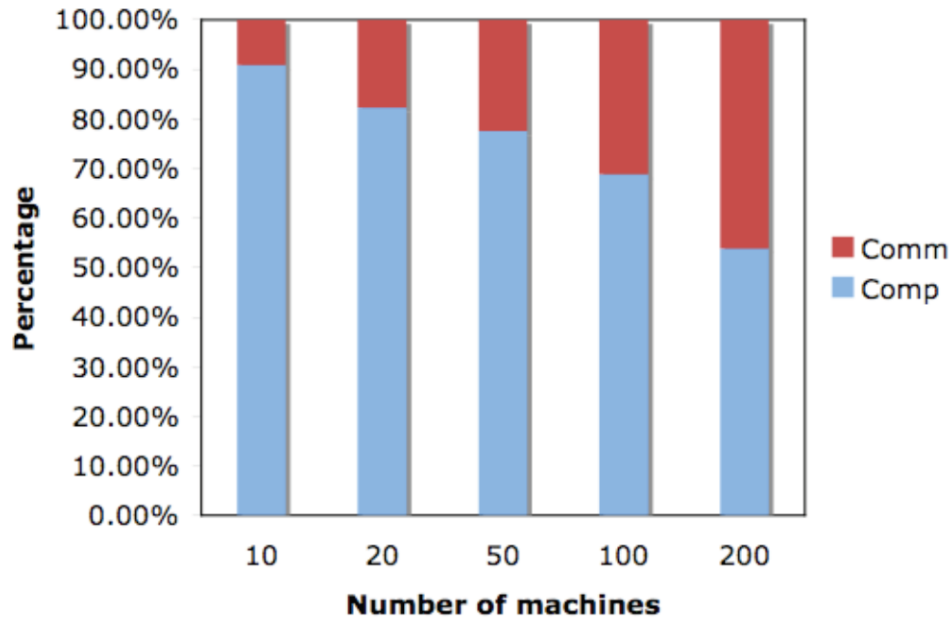
Runtime Speedup – Results



Machines	Time (sec.)	Speedup
10	9,233	10
20	4,326	21.3
50	2,280	40.5
100	1,014	91.1
200	796	116

Runtime Speedup – Overhead Analysis





- Communication > Computation @ 200 Machines
- “Saturation” point deferred for larger datasets
- Parallel CCF enables near-real-time recommendations (~14 min update)

Conclusions

- By combining user and description models of communities, CCF produces better quality recommendations than other methods.
- Gibbs sampling provides better initialization values for E-M than random seeding.
- CCF can be parallelized to handle large data sets.