What’s Popular Amongst Your Friends?

Kishor Barman, Onkar Dabeer
School of Technology and Computer Science
Tata Institute of Fundamental Research
Mumbai, India
Introduction
Beyond Search
Beyond Search

Social Network

Media

Shopping

Events

Web Search:
Long list of related items

Travel

Services
Beyond Search

Social Network

Media

Shopping

Events

Services

Web Search:
Long list of related items

Travel

Recommendations:
Few “likable” items
Limited domain
Recommendations in Action
Recommendations in Action

- Amazon
  - People who bought this also bought...
Recommendations in Action

- Amazon
  - People who bought this also bought...
- RichRelevance
  - Commercial recommendation engine
Recommendations in Action

● Amazon
  – People who bought this also bought...

● RichRelevance
  – Commercial recommendation engine

● Netflix
  – Suggests movies using rating matrix
Recommendations in Action

- **Amazon**
  - People who bought this also bought...
- **RichRelevance**
  - Commercial recommendation engine
- **Netflix**
  - Suggests movies using rating matrix
- **LinkedIn**
  - Suggests connections
Recommendations in Action

- Amazon
  - People who bought this also bought...
- RichRelevance
  - Commercial recommendation engine
- Netflix
  - Suggests movies using rating matrix
- LinkedIn
  - Suggests connections
- Opinion estimation
Prior Art
Prior Art

- Different kinds of data
  - Content (example: cast of movies, plot)
  - Ratings (example: Movielens, Netflix)
  - Social connections (example: Facebook)
Prior Art

● Different kinds of data
  - Content (example: cast of movies, plot)
  - Ratings (example: Movielens, Netflix)
  - Social connections (example: Facebook)

● Several heuristic algorithms (Adomavicius et al, 2005; Su et al 2009)
  - The Netflix prize winner blends 100+ algorithms
  - RichRelevance blends 40 algorithms
Prior Art

- Different kinds of data
  - Content (example: cast of movies, plot)
  - Ratings (example: Movielens, Netflix)
  - Social connections (example: Facebook)

- Several heuristic algorithms (Adomavicius et al, 2005; Su et al 2009)
  - The Netflix prize winner blends 100+ algorithms
  - RichRelevance blends 40 algorithms

- Few *provably good* principles?
  - Focus: collaborative filters based on rating matrix
Matrix Algorithms

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>?</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
Matrix Algorithms

\[
\begin{array}{cccc}
2 & ? & 1 & 3 \\
4 & 4 & ? & ? \\
? & 3 & 1 & 5 \\
1 & 5 & 2 & ? \\
\end{array}
\]
### Matrix Algorithms

- **SVD inspired algorithms**
  - Koren *et al.* (2009) and earlier

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>?</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
Matrix Algorithms

- **SVD inspired algorithms**
  - Koren et al (2009) and earlier

- **Low-rank matrix completion**
  - Candes, Recht (2008)
  - Keshavan et al (2009)
  - Lee and Bresler (2009)

<table>
<thead>
<tr>
<th></th>
<th>?</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>?</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>?</td>
</tr>
</tbody>
</table>
In This Talk
In This Talk

• Popularity Amongst Friends (PAF) algorithm
  - Neighborhood method motivated by practice
In This Talk

• Popularity Amongst Friends (PAF) algorithm
  – Neighborhood method motivated by practice

• Competitive empirical performance
  – Comparison with OptSpace (2008) on MovieLens and Netflix datasets
In This Talk

- Popularity Amongst Friends (PAF) algorithm
  - Neighborhood method motivated by practice
- Competitive empirical performance
  - Comparison with OptSpace (2008) on Movielens and Netflix datasets
- Data model and analysis
  - Binary ratings, (lots of) user noise
  - 3 regimes of operation
In This Talk

- Popularity Amongst Friends (PAF) algorithm
  - Neighborhood method motivated by practice
- Competitive empirical performance
  - Comparison with OptSpace (2008) on Movielens and Netflix datasets
- Data model and analysis
  - Binary ratings, (lots of) user noise
  - 3 regimes of operation
- Discussion
The Popularity Amongst Friends (PAF) Algorithm
PAF for Binary Ratings
PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users
PAF for Binary Ratings

• Algorithm:
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users

• Motivated by practice (example: Amazon)
PAF for Binary Ratings

- **Algorithm:**
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users

- Motivated by practice (example: Amazon)
- Not matrix completion
PAF for Binary Ratings

- Algorithm:
  - For user 1, find top K similar users
  - Similarity = # agreements in available ratings
  - Recommend an unseen item that is most popular amongst these K users

- Motivated by practice (example: Amazon)
- Not matrix completion
- Low complexity
  - User node degree << Total number of users
  - Simple updates
Performance Metrics
Performance Metrics

- RMSE - popular since Netflix prize
Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works
Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works
- Probability that entire matrix is recovered
  - Candes, Recht (2008) and others
  - Aditya, D, Dey (2009)
Performance Metrics

- RMSE - popular since Netflix prize
- MAE - popular in earlier works
- Probability that entire matrix is recovered
  - Candes, Recht (2008) and others
  - Aditya, D, Dey (2009)
- Our focus is on bit error rate (BER)
  - Probability that a recommendation made is incorrect
Empirical Performance
(Movielens and Netflix)
Setup
Setup

- Rating quantization
  - 4.5 mapped to 1 (yes), 1-3 mapped to 0 (no)
Setup

- Rating quantization
  - 4,5 mapped to 1 (yes), 1-3 mapped to 0 (no)
- Hide 30% of data per user; can compute metrics only when recommended item is in the hidden list
Setup

- Rating quantization
  - 4.5 mapped to 1 (yes), 1-3 mapped to 0 (no)
- Hide 30% of data per user; can compute metrics only when recommended item is in the hidden list
- Comparison with OptSpace (Keshavan et al, 2008)
  - Representative of matrix algorithms
  - Evaluated only on items recommended by local algorithm
  - Unquantized input; output quantized for BER
## Empirical Performance

### Movielens

<table>
<thead>
<tr>
<th></th>
<th>BER</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAF</td>
<td>0.103</td>
<td>0.627</td>
<td>0.748</td>
</tr>
<tr>
<td>OptSpace</td>
<td>0.108</td>
<td>0.581</td>
<td>0.733</td>
</tr>
</tbody>
</table>

**Naive Estimate:**
For Local Algorithm, to compute RMSE, MAE
1 is mapped to 4.5
## Empirical Performance

### Movielens

<table>
<thead>
<tr>
<th></th>
<th>BER</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAF</td>
<td>0.103</td>
<td>0.627</td>
<td>0.748</td>
</tr>
<tr>
<td>OptSpace</td>
<td>0.108</td>
<td>0.581</td>
<td>0.733</td>
</tr>
</tbody>
</table>

**Naive Estimate:**
For Local Algorithm, to compute RMSE, MAE 1 is mapped to 4.5

### Snapshot of Netflix (2000)

<table>
<thead>
<tr>
<th></th>
<th>BER</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAF</td>
<td>0.18</td>
<td>0.742</td>
<td>0.942</td>
</tr>
<tr>
<td>OptSpace</td>
<td>0.19</td>
<td>0.590</td>
<td>0.742</td>
</tr>
</tbody>
</table>

**Netflix has higher percentage of low ratings**
Empirical Performance (Contd.)

Movielens After Removing Popular Movies
(those with 60% or more 1’s)

<table>
<thead>
<tr>
<th></th>
<th>BER</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAF</td>
<td>0.335</td>
<td>0.709</td>
<td>1.010</td>
</tr>
<tr>
<td>OptSpace</td>
<td>0.327</td>
<td>0.718</td>
<td>0.901</td>
</tr>
</tbody>
</table>
Empirical Performance (Contd.)
Remarks
Remarks

- PAF competitive for BER
Remarks

- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information
Remarks

- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information
- 2-10 times faster than OptSpace
  - Also amenable to recursive update
Remarks

- PAF competitive for BER
- Are MAE, RMSE relevant?
  - Scale 1-5, RMSE 0.7+ - poor confidence intervals
  - Noisy data and difficult to squeeze out more than 1 bit information
- 2-10 times faster than OptSpace
  - Also amenable to recursive update
- Any provably good properties?
Analysis
Data Model
Data Model

Unknown Clusters

Diagram of data model with a process involving unknown clusters.
Data Model

Unknown Clusters

Errors
Data Model

Unknown Clusters

Errors

Erasures
Data Model

Unknown Clusters

? (in poly time)

Errors

Erasures
Data Model

- Unknown Clusters
- Errors
- Erasures
- ? (in poly time)

Channel Coding/
Estimation of rearranged, ‘smooth’ process, under noise and erasures
The Model in Words
The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
- **Errors**: i.i.d., binary symmetric channel, represent noisy user behavior
The Model in Words

- Underlying true block constant matrix
  - Cluster size determines degrees of freedom
- Clusters not known, but deterministic
- **Errors**: i.i.d., binary symmetric channel, represent noisy user behavior
- **Erasures**: i.i.d., model missing data
The Model in Words

• Underlying true block constant matrix
  - Cluster size determines degrees of freedom
• Clusters not known, but deterministic
• **Errors:** i.i.d., binary symmetric channel, represent noisy user behavior
• **Erasures:** i.i.d., model missing data
• **Diverse opinions:** i.i.d. Bernoulli(1/2) ratings across clusters
  - No information from self data; must use collaborative filtering
Some Assumptions

- Matrix: \( m \times n, m = \Theta(n) \)

- Erasure probability
  \[ \epsilon = 1 - \frac{c}{n^\alpha}, \quad 0 \leq \alpha \leq 1 \]

  \( \alpha < 1/2 \): Near-quadratic regime
  \( \alpha > 1/2 \): Near-linear regime

- All clusters of same order
  
  - Number of clusters = \( \Omega(\log n) \) to ensure
  \[ P(\text{cluster merging}) \] is vanishing
Recover Entire Matrix

Recover Entire Matrix


For cluster + majority
\[ P_e \to 0 \]
\[ \Theta(\alpha \log n + \log \log n) \]
Recover Entire Matrix


$\alpha$

$log(\text{cluster size})$

$\Theta(\alpha \log n + \log \log n)$

For cluster + majority

$P_e \to 0$

For any scheme

$P_e \to 1$

Threshold determined by majority decoding
Recover Entire Matrix


$\log(\text{cluster size}) = \Theta(\alpha \log n + \log \log n)$

For cluster + majority
$P_e \to 0$

For any scheme
$P_e \to 1$

Threshold determined by majority decoding
Recover Entire Matrix


For cluster + majority
$P_e \to 0$

$\Theta(\alpha \log n + \log \log n)$

Threshold determined by majority decoding

For any scheme
$P_e \to 1$

Clustering algo fails
Limits not known
Asymptotic BER of PAF

\[ \log(\text{cluster size}) \]

\[ \alpha \]

\[ 0.5 \]

\[ 1 \]
Asymptotic BER of PAF

For $K = \# \text{ friends}$

$\text{BER} = 0$

$2\alpha \log n$
Asymptotic BER of PAF

\[ BER = p^{\frac{1}{\gamma}} - (1 - p)^{\frac{1}{\gamma}} \]

For \( K = \# \text{ friends} \),
\[ BER = 0 \]

\[ 2\alpha \log n \]

\[ 2(\alpha - \gamma) \log n \]

log(cluster size)

\( \alpha \)

\( p \)
Asymptotic BER of PAF

\[ \text{BER} = \frac{p^{\frac{1}{\gamma}}}{p^{\frac{1}{\gamma}} + (1-p)^{\frac{1}{\gamma}}} \]

For \( K = \# \text{ friends} \)
\[ \text{BER} = 0 \]

For \( \alpha = 0.5 \)
\[ 2\alpha \log n \]

For \( \alpha = 1 \)
\[ 2(\alpha - \gamma) \log n \]

PAF fails even with no noise
\[ \text{BER} = \frac{1}{2} \]
Three Phases
Three Phases

- **Phase 1**: Large cluster, near-quadratic samples, BER=0
  - Top neighbors good, large cluster averages out noise
Three Phases

- **Phase 1**: Large cluster, near-quadratic samples, $\text{BER}=0$
  - Top neighbors good, large cluster averages out noise

- **Phase 2**: Small cluster, near-quadratic samples, $0 < \text{BER} < 1/2$
  - Top neighbors good
  - But cluster too small to average out noise
  - Optimum list size $= \# \text{ friends}$
Three Phases

- **Phase 1**: Large cluster, near-quadratic samples, $BER=0$
  - Top neighbors good, large cluster averages out noise
- **Phase 2**: Small cluster, near-quadratic samples, $0 < BER < 1/2$
  - Top neighbors good
  - But cluster too small to average out noise
  - Optimum list size = # friends
- **Phase 3**: Near-linear samples
  - Most neighbors picked are bad
Phase 2: Finding Good Neighbors

- Similarity between row 1 and another row in its cluster:
  \[ \text{Binomial} \left( n, c^2[p^2 + (1 - p)^2]n^{-2\alpha} \right) \]

- Similarity between row 1 and a row in a different cluster:
  \[ \text{Binomial} \left( n, c^2n^{-2\alpha}/2 \right) \]

- Above marginals concentrate for \( \alpha < 1/2 \). So we hope to find good neighbors.

- Detailed calculations (accounting for dependence) confirm the hope.
Phase 2: Filtering Noise

- $K = \# \text{ friends} - \text{w.h.p. all neighbors are good}$

- Most popular column: w.h.p. $\# 1's = \lceil 1/\gamma \rceil$, and $\# 0's = 0$
  - For an arbitrary column, $E[\#\text{samples}] = \Theta(1/n^\gamma)$

- Aposteriori probability:

$$P(X(1, j^*) = 0|Y_K(:, j^*)) = \frac{p^{\#1-\#0}}{p^{\#1-\#0} + (1 - p)^{\#1-\#0}}$$
Where Does Real Data Live?
The Movielens Matrix

Optimum $K = 100-150$
The Movielens Matrix

Optimum K = 100-150

Phase 2?

Phase 3?
The Movielens Matrix

Optimum K = 100-150

Phase 2?
Side information for regression

Phase 3?
Side information for clustering
Discussion
Discussion

- PAF algorithm is scalable and competitive
Discussion

- PAF algorithm is scalable and competitive
- Provably good in near-quadratic regime
  - BER bounded away from $1/2$
Discussion

• PAF algorithm is scalable and competitive
• Provably good in near-quadratic regime
  – BER bounded away from 1/2
• Near-linear regime: Blend side information
Discussion

• PAF algorithm is scalable and competitive
• Provably good in near-quadratic regime
  - BER bounded away from $1/2$
• Near-linear regime: Blend side information
• Refining our simple model
  - Sampling - account for heavy tails
  - Incorporate item correlations
  - Rearranged general ‘smooth’ processes?
  - Incorporating side information?
More Details...

• ISIT’10
• Journal submission - arXiv: 1006:1772

Thank you