Iterative Algorithms for Collaborative Filtering with Mixture Models

Chalmers University of Technology

Eszter P. Windhager, Libertad Tansini, Istvan Biro & Devdatt Dubhashi

July 7, 2006

Outline

Outline

1. Introduction
   - Motivation
   - Mixture Model
2. Methods
3. Results
   - Tests
   - Generated Data
   - Real Data
4. Conclusions
5. Future Work

Motivation

- Millions of items to choose from
- Locate reliable experts?
- From individual to collective method of recommendation
  Collaborative Filtering – CF
- Amazon's purchase recommendations

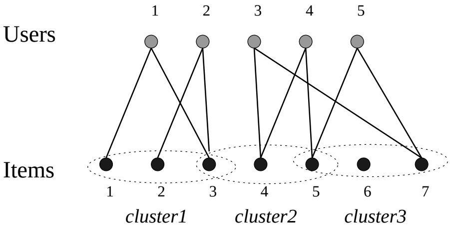
Outline

1. Introduction
   - Motivation
   - Mixture Model
2. Methods
3. Results
   - Tests
   - Generated Data
   - Real Data
4. Conclusions
5. Future Work

Mixture Model I

- Hofmann and Puzicha proposed a probabilistic model
- Hidden or latent structure in underlying data
- Items are grouped into topics – may overlap arbitrarily
- Kleinberg and Sandler gave first theoretical analysis of algorithms for CF in this setting

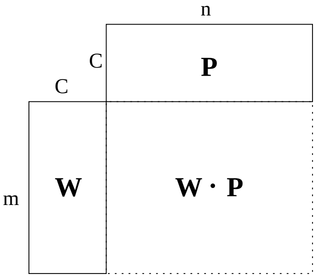
Mixture Model II



Mixture Model III

- n users and m items
- Underlying hidden structure of C clusters or topics of items
- Each user u distribution over clusters preference for cluster c is $p(u, c)$ – interest in several clusters
- Each cluster c probability distribution $w(a, c)$ over items a – overlapping

Mixture Model VI



Input Output

- Selection matrix S, where $s(u, a) = 1$ if item a was selected by user u, $s(u, a) = 0$ otherwise
- Suggest items to users
- With known P and W matrices – suggest the item a to user u with maximum value in uth column of $W \cdot P$ matrix
- Try to approximate P and W

Objective

- Suit of 4 algorithms tailored to the mixture model
- Simple – conceptually and implementation
- Efficient

Methods

1. Similarity based soft clustering – Pearson correlation – $w_0(a, c)$
2. Given $w_0(a, c)$ compute $p_0(u, c)$ $\forall u, c$ based on HITS idea
3. Given $w_i(a, c)$ and $p_i(u, c)$ compute intermediate values $\bar{p}_{i+1}(u, c)$ and $\bar{w}_{i+1}(a, c)$ $\forall u, a, c$ based on HITS idea
4. Finally calculate the average
$$p_{i+1}(u, c) = (1 - \theta)p_i(u, c) + \theta\bar{p}_{i+1}(u, c)$$
$$w_{i+1}(a, c) = (1 - \theta)w_i(a, c) + \theta\bar{w}_{i+1}(a, c),$$
where $0 \leq \theta \leq 1$ is a parameter of the algorithm
5. Until updates are not significant, less then $\epsilon \geq 0$.

First method

- Sum weights of items in the cluster selected by user and normalize

$$\bar{p}_{i+1}(u,c) = \frac{\sum_a w_i(a,c)\cdot s(u,a)}{\sum_{c'}\sum_a w_i(a,c')\cdot s(u,a)}$$

- Similarly sum preferences of users who selected the item for that cluster and normalize

$$\bar{w}_{i+1}(a,c) = \frac{\sum_u p_i(u,c)\cdot s(u,a)}{\sum_{a'\in c}\sum_u p_i(u,c)\cdot s(u,a')}$$

Second Method

Items can belong to several clusters

$$\hat{p}_{i+1}(u,c) = \frac{1}{S_u}\Big(\sum_a w_i(a,c)s(u,a) - \frac{1}{C-1}\sum_{a\in c}\sum_{c_r\neq c} w_i(a,c_r)s(u,a)\Big)$$

- S_u is number of samples for user u and C is number of clusters, obviously $\hat{p}_{i+1}(u,c) \in [-1,1]\ \forall i,u,c$.

$$\hat{w}_{i+1}(a,c) = \frac{1}{S_a}\Big(\sum_u p_i(u,c)\cdot s(u,a) - \frac{1}{C-1}\sum_{u\in c}\sum_{c_r\neq c} p_i(u,c_r)\cdot s(u,a)\Big)$$

- Item a was selected S_a times and $u \in c$ means that $p_i(u,c) \neq 0$

Second Method

- Linear transform to convert values into $(0, 1)$

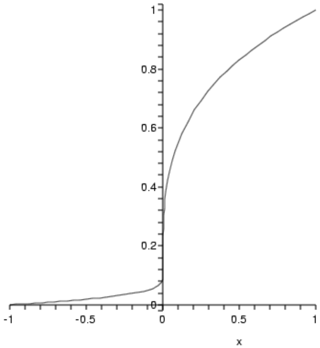
$$p'_{i+1}(u, c) = \frac{\hat{p}_{i+1}(u, c) + 1}{2}$$

- Values are highly concentrated around 1/2, so we re-scale

$$\bar{p}_{i+1}(u, c) = \frac{1}{N_u}(p'_{i+1}(u, c) - \frac{m_u}{2})$$

- normalizing factor for user u $\frac{1}{N_u}$ so $\sum_c \bar{p}_{i+1}(u, c) = 1$ and $m_u = \min_c p'_{i+1}(u, c)$
- Similarly for $\bar{w}_{i+1}(a, c)$

Too Simple



- Shape of the transform function, horizontal axis $\hat{p}_{i+1}(u, c)$ and vertical axis $\bar{p}_{i+1}(u, c)$

Third Method

$\hat{p}_{i+1}(u,c) \in [-1,1]$ values concentrated around zero

1. If no item was selected for cluster c then $\bar{p}_{i+1}(u,c) = 0$
2. if $\hat{p}_{i+1}(u,c) \geq 0$

$$\bar{p}_{i+1}(u,c) = \frac{1}{N_u}\Big((1-\frac{1}{C})(\hat{p}_{i+1}(u,c))^{0.3} + \frac{1}{C}\Big)$$

3. if $\hat{p}_{i+1}(u,c) < 0$

$$\bar{p}_{i+1}(u,c) = \frac{1}{N_u}\Big((-\frac{1}{C})(-\hat{p}_{i+1}(u,c))^{0.2} + \frac{1}{C}\Big)$$

Fourth Method

- Like third method without normalizing factor $1/N_u$
- Re-scaling formula of second method
- Typical spread of preferences and weights is wider

Outline

- Generated test data
- Real data collected from Hungarian web news cite – similar performance to generated data
- Previous benchmark studies
- "Semi-omniscient" algorithm of Kleinberg and Sandler

Outline

Generated Test Data

- *Disjoint Model* users like only one cluster, $p(u, c) = 1$ for exactly one c and $p(u, c') = 0$ for rest
  Disjoint clusters and random weights for items in clusters
- *Partially Mixed* users like one cluster more, $p(u, c) = 0.9$ for exactly one c, and $p(u, c') = \frac{0.1}{C-1}$ for the rest
  Disjoint clusters and random weights for items in clusters
- *Fully Mixed* items can belong to several clusters and users can be interested in several clusters
  Random preferences and weights
  Probability of non zero preference for $1, 2, 3, ..9$ clusters is $0.5, 0.3, 0.15, 0.04, 0.005, 0.003, 0.0015, 0.0004, 0.0001$

Disjoint Model

- 1000 users, 300 items and 10 clusters
- 10 or 20 random samples for each user

		Sample 10		Sample 20			Sample 10		Sample 20	
M	θ	F 10	F 20	F 10	F 20	θ	F 10	F 20	F 10	F 20
1	0	4.7	14.9	6.3	16.1	0.4	**8.7**	**19.0**	**8.7**	**19.4**
2		4.0	14.3	5.9	15.8		3.7	14.0	6.2	16.2
3		4.2	14.5	6.0	15.8		3.7	13.9	6.2	16.1
4		4.3	14.6	6.1	15.9		3.7	14.0	6.2	16.1
1	0.2	**8.7**	**19.1**	**8.7**	**19.3**	1	**8.7**	**19.1**	**8.7**	**19.3**
2		4.4	15.0	5.8	15.9		4.0	13.8	3.7	13.9
3		4.5	15.1	5.8	16.0		4.0	14.1	3.4	13.9
4		4.4	15.1	5.8	16.0		3.8	14.8	3.5	14.5

Partially Mixed Model

		Sample 10		Sample 20			Sample 10		Sample 20	
M	θ	F 10	F 20	F 10	F 20	θ	F 10	F 20	F 10	F 20
1	0	**4.1**	**14.2**	**4.3**	**14.9**	0.4	1.1	2.2	1.2	2.3
2		0.0	0.9	0.1	0.6		0.0	0.2	0.0	0.3
3		2.7	12.1	3.2	13.2		1.9	13.1	0.8	12.1
4		2.8	12.4	3.4	13.6		2.1	13.3	0.8	12.2
1	0.2	1.1	2.2	1.1	2.2	1	1.1	2.2	1.1	2.2
2		0.0	0.2	0.1	0.4		0.3	1.3	0.3	1.4
3		1.9	13.0	1.0	12.1		1.9	13.1	0.8	11.7
4		1.9	13.1	1.0	12.1		1.9	13.3	0.7	11.9

Fully Mixed Model

		Sample 10		Sample 20			Sample 10		Sample 20	
M	θ	F 10	F 20	F 10	F 20	θ	F 10	F 20	F 10	F 20
1	0	1.1	6.4	1.5	7.6	0.4	1.7	4.7	1.7	4.7
2		0.0	0.2	0.0	0.1		0.0	0.1	0.0	0.1
3		0.7	5.6	1.3	7.1		**1.8**	**7.5**	**1.8**	**7.4**
4		0.7	5.7	1.3	7.1		**1.8**	**7.6**	**1.8**	**7.4**
1	0.2	1.7	4.6	1.6	4.5	1	1.6	4.6	1.6	4.5
2		0.0	0.2	0.0	0.1		0.3	1.1	0.4	1.5
3		**1.9**	**7.7**	**1.9**	**7.7**		**1.9**	**7.7**	**1.8**	**7.7**
4		**1.9**	**7.7**	**1.9**	**7.7**		**1.9**	**7.7**	**1.8**	**7.7**

Outline

1. Introduction
   - Motivation
   - Mixture Model
2. Methods
3. Results
   - Tests
   - Generated Data
   - Real Data
4. Conclusions
5. Future Work

Real Data

- Log file of a Hungarian news portal (http://origo.hu)
- 1.07 million users and 368000 items from one month
- First test with 1000 users and 8321 selected items
- Without iteration ($\theta = 0$)
- Similar to results on generated data

Method	F10	F20	F30
1	8.13	13.77	18.49
2	0.42	1.35	3.18
3	3.71	8.13	11.15
4	3.50	7.73	10.71

Conclusions

- Disjoint clusters – method 1 works the best with iterations
- Fully mixed model – methods 3 and 4 with iteration – small resources then method 1 without iterating acceptable
- Partially mixed model – not clear – method 1 without iteration works well
- Open question to explain different performance characteristics by a theoretical analysis
- Real data – method 1 best results, as we expected

Future Work

- Expect real data to behave like fully mixed model – complex
- Tests on larger real data sets and with iteration
- Refined evaluation by looking into results manually
- Include multiple selections
- High-order Markov-chains combined with collaborative filtering – meaningful suggestions

Questions?