The Art of Lemon's solution

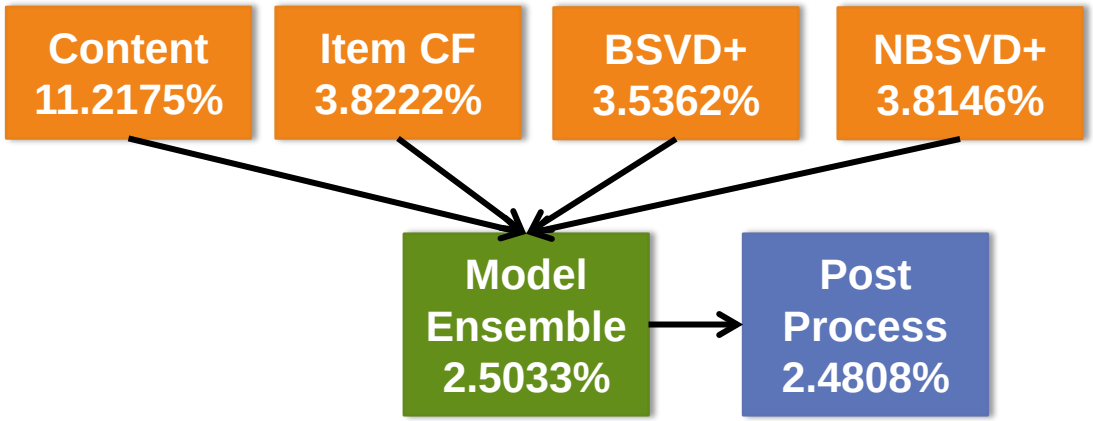
KDD Cup 2011 Track 2

hulu

Siwei Lai/ Rui Diao
Liang Xiang

Outline

- **Problem Introduction**
- **Data Analytics**
- **Algorithms**
  - Main Models
  - Model Ensemble
  - Post Process
- **Conclusion**
- **Future Work**



Problem Introduction

- **Two Tracks**
- **Track 2**
  - Classification Problem
    - Positive Samples : tracks users vote higher than 80
    - Negative Samples : popular tracks users have not voted
  - Data Set
    - User voting data
    - Taxonomy data
  - Comments
    - Similar to Top-N recommendation problem
    - Using negative samples to prevent Harry Potter problem

Data Analytics

- **User vote data may be ordered by time.**
  - Anchoring effect
  - Vote on artists and then vote on their tracks
- **This is main reason why we got 2nd position**

http://justaguyinagarage.blogspot.com/2011/06/recommendation-system-competitions.html

Data Analytics

- **If a user have voted on artist/album, she will have large probability to vote the tracks of the artist/album.**

45% 58% Artist ⇒ Artist's tracks
75% 75% Album ⇒ Album's tracks
45% 56% Item ⇒ Items with the same Artist
51% 52% Item ⇒ Items with the same Album

Data Analytics

- **User vote data may be ordered by time.**
  - Anchoring effect
  - Vote on artists and then vote on their tracks
- **If a user have voted on artist, she will have large probability to vote the tracks of the artist.**

Algorithm: Main Models

- **Content-based Model**
- **Item-based Collaborative Filtering Model**
- **Binary Latent Factor Model**
- **Neighborhood-based Binary SVD Model**

Content-based Model

- **If a user have voted on artist/album, she will have large probability to vote the tracks of the artist/album.**
  - Version 1. User will vote on a track if she have voted the same artist's item before. (Error rate ≈ 17%)

    P(u, i) = 1 if user u have voted tracks with same artist/album of track i

  - Version 2. Use the average score of some artist/album. (Error rate ≈ 11%)

    P(u, i) = average score user u assigned on artist/album of track i or tracks with same artist/ablum

Item-based Collaborative Filtering

- **Jaccard Index**

$$w_{ij} = \frac{\sum_{u \in N(i) \cap N(j)} 1}{|N(i) \cup N(j)|}$$

Error rate ≈ 9%

Item-based Collaborative Filtering

- **Our Similarity**

$$w_{ij} = \frac{\sum_{u \in N(i) \cap N(j)} f(u, i, j)}{|N(i) \cup N(j)|}$$

Item-based Collaborative Filtering Model

- **+ Temporal information**

$$f(u,i,j) = \frac{1}{\left|d_{ui} - d_{uj}\right|^{\gamma_3}}$$

$$w_{ij} = \frac{\sum_{u \in N(i) \cap N(j)} f(u,i,j)}{|N(i) \cup N(j)|}$$

$1/1406^{1.1}$

141|8573
...
251480 0
232699 50
132238 50
...
67376 50
3109 0
96153 30
...

1405 items

$1/9^{1.1}$

862|1455
...
232699 90
238869 90
271685 90
...
252580 90
3109 90
49451 90
...

9 items

$1/20^{1.1}$

2033|5396
...
81180 64
3109 54
26594 52
...
8830 26
232699 59
53396 57
...

20 items

...

Item-based Collaborative Filtering

- **+ Vote information**

$$f(u,i,j) = \frac{1}{\left|d_{ui} - d_{uj}\right|^{\gamma_3}\left(1 + \left|r_{ui} - r_{uj}\right|\right)^{\gamma_4}}$$

$1/1406^{1.1}/51^{0.2}$

141|8573
...
251480 0
232699 50
132238 50
...
67376 50
3109 0
96153 30
...

$1/9^{1.1}/1^{0.2}$

862|1455
...
232699 90
238869 90
271685 90
...
252580 90
3109 90
49451 90
...

$1/20^{1.1}/6^{0.2}$

2033|5396
...
81180 64
3109 54
26594 52
...
8830 26
232699 59
53396 57
...

...

Item-based Collaborative Filtering

- **Prediction**

$$\hat{r}_{ui} = \sum_{j \in S(N(u),k)} w_{ij}\left(1 + r_{uj}\right)^{\gamma_1}$$

Item-based Collaborative Filtering

- **+ Removing popular bias**

$$\hat{r}_{ui} = \sum_{j \in S(N(u),k)} w_{ij}\left(1 + r_{uj}\right)^{\gamma_1} \boxed{\frac{1}{\sqrt{n_i}}}$$

Item-based Collaborative Filtering

Factors	Error Rate (%)
initial model (Jaccard Index + KNN)	8.9992
+ removing popular bias	5.2953
+ using temporal information	3.9283
+ using vote information	3.8222
+ using taxonomy information	3.6578

Binary Latent Factor Model

prediction $\hat{r}_{ui} = p_u^T q_i$

Error rate ≈ 6%

minimize $\sum_{(u,i)\in\mathcal{K}^+\cup\mathcal{K}^-} (r_{ui} - p_u^T q_i) + \lambda(\|p_u\|^2 + \|q_i\|^2)$

$r_{ui} = 1 \quad (u,i) \in \mathcal{K}^+$
$r_{ui} = 0 \quad (u,i) \in \mathcal{K}^-$

- **Sampling**
  - Positive samples: items in train data.
  - Negative samples: nearly the same as sampling test data.
  - Positive samples and Negative samples have the **same** number for each user

Binary Latent Factor Model+

prediction $$\hat{r}_{ui} = b_u + b_i + b_{a(i)} + b_{b(i)} + \boxed{b_{u,I(a(i)\in N(u))}} + \boxed{b_{u,I(b(i)\in N(u))}} + p_u^T(q_i + x_{a(i)} + y_{b(i)})$$

minimize $$\sum_{(u,i)\in\mathcal{K}^+\cup\mathcal{K}^-} (r_{ui} - p_u^T q_i) + \lambda(\|p_u\|^2 + \|q_i\|^2)$$

Error rate ≈ 3.5%

Neighborhood-based Binary SVD Model

prediction

$$\hat{r}_{ui} = b_u + b_i + b_{a(i)} + b_{b(i)} + b_{u,I(a(i)\in N(u))} + b_{u,I(b(i)\in N(u))} + \frac{1}{\sqrt{|N(u)|}} q_i^T \left(\sum_{j\in N(u)} y_j \right)$$

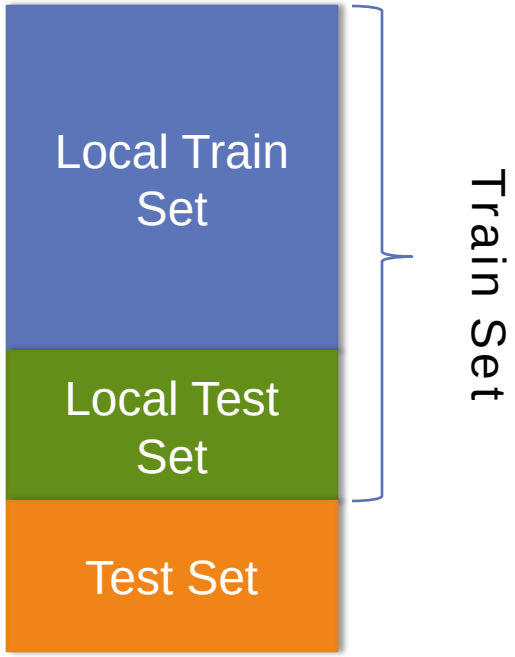
Features used

Features \ Models	Content	Item CF	BSVD+	NBSVD+
Collaborative filtering	×	√	√	√
Neighborhood info	×	√	×	√
Ratings	√	√	○	○
Time ordering	×	√	×	×
Artist/album	√	○	√	√
Genre structure	×	×	×	×

Model Ensemble

- **Local test set**
- **Linear combination**
- **Simulated Annealing**
- **8-fold cross validation**

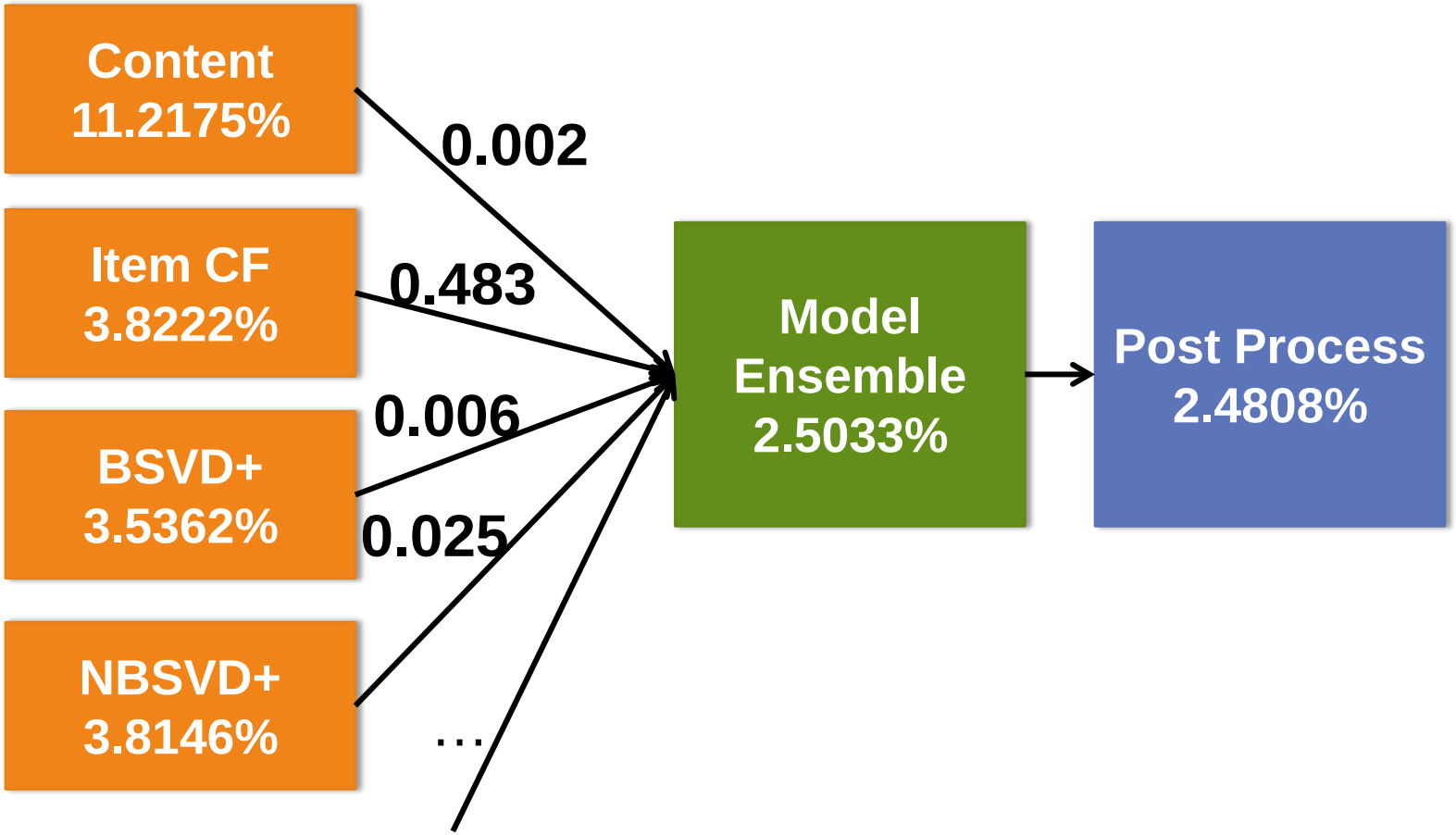
Model	Error Rate (%)	weight
Content	11.2175	0.002
Item CF	3.8222	0.438
BSVD+	3.5362	0.006
NBSVD+	3.8146	0.025



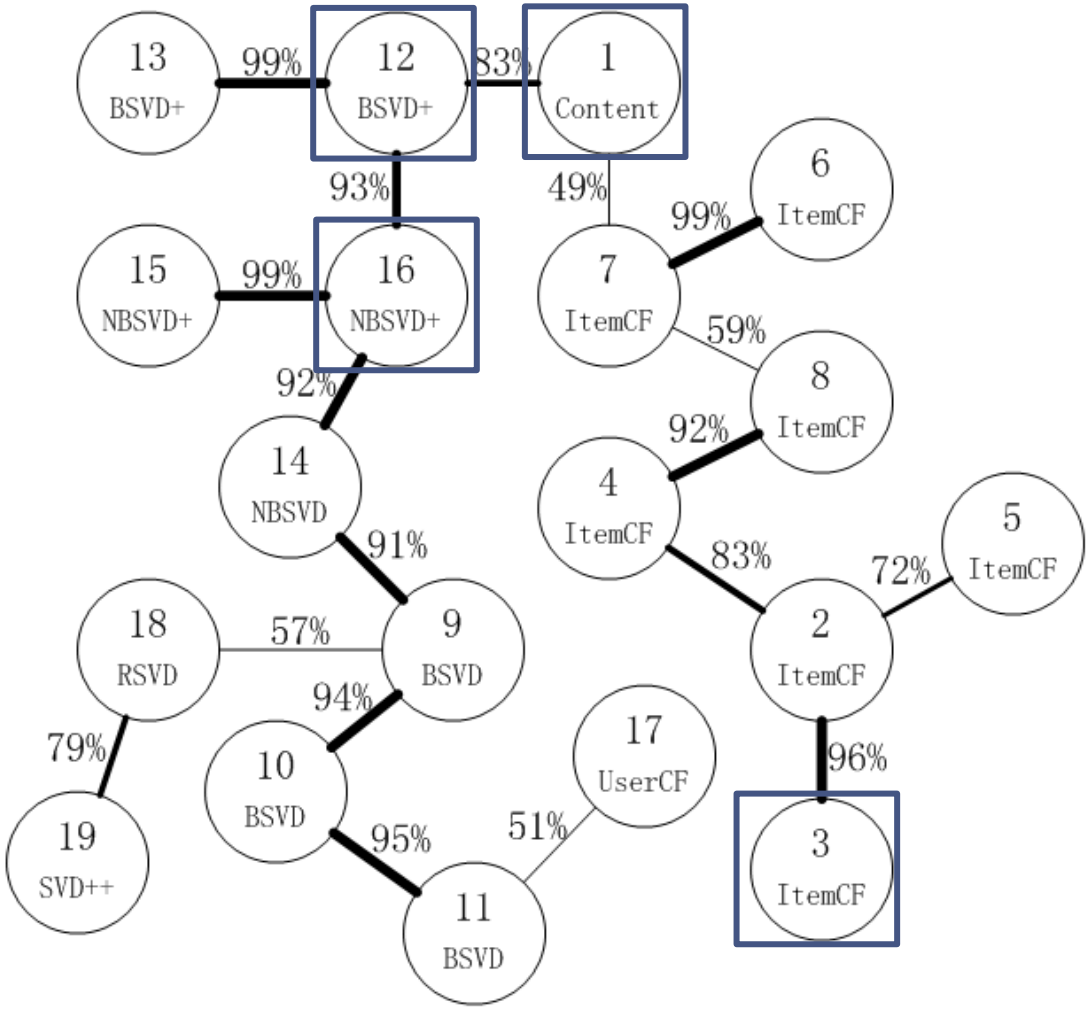
Post Process

- **Some special features can not be modeled well**
- **Find special user-item pairs.**
  - The most popular items.
  - Vote high on track’s album but vote low on it’s artist.
  - …
- **Multiply a factor**

Algorithms


Content
11.2175%
Item CF
3.8222%
BSVD+
3.5362%
NBSVD+
3.8146%
…
0.002
0.483
0.006
0.025
Model
Ensemble
2.5033%
Post Process
2.4808%

Model Similarities



Conclusion

- **Data Analysis is very important**
  - User behavior data is ordered by time
  - Artist/Album data can improve accuracy a lot
- **Team members number and model numbers is very important**
- **Useful algorithms:**
  - Content-based
  - Neighborhood-based
  - Matrix Factorization

Future Work

- **How to add temporal information into Binary SVD Model?**
- **Apply Binary SVD into real production**
  - How to make explanation
  - How to make real-time on-line recommendation

Q&A

Thanks!

xlvector@gmail.com