Recommender Systems

Scott Sanner

IR, ML, and Recommendation

- IR
  - Find documents relevant to a query
  - Long tail of queries
  - No labels
- ML
  - Predict on future data given training data
  - Fixed task (spam, topic classification)
  - Requires a lot of labeled data

IR, ML, and Recommendation

- Recommendation
  - "Personalized" machine learning
    - Predict differently for every user (row)
    - Rather than train per user (sparse data)…
      - Leverage similar users (transfer learning)
  - Like ML, have lot's of labeled data
  - Like IR, large output space **y** to recommend
    - Not often query-driven

Recommendation

- Predict missing from observed ratings?

$$\begin{array}{r} \text{Joseph} \\ \text{Nguyen} \\ \\ \text{Scott} \end{array} \quad R = \begin{pmatrix} 1 & 1 & 1 & 0 & ? & 0 \\ 1 & 0 & ? & 0 & 0 & \\ & \vdots & & \vdots & & \vdots \\ & 0 & 0 & & 1 & ? \end{pmatrix}$$

Canonical Example: Netflix Competition

…1-5 ratings, here: like (1), dislike (0)

Recommendation = matrix completion. Once matrix completed… how to recommend item to user?

Recommend many Bipartite Relations

- Bipartite Relations
  - Movies, books, store products, news articles ➔ users
  - Questions ➔ students (automated tutoring)
  - Points of interest ➔ tourists
  - Products ➔ stores, vending machines
  - Tags ➔ documents

Note: users here are vending machines, docs… "personalization" is relative

- Not just binary relations
  - Classes (binary, k-ary), ratings (ordinal, real)
  - Combinatorial objects (product quantities)
  - Ternary, k-ary relations (tensors)

Fundamental Methods

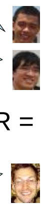
Types of Recommendations

- Editorial and hand curated
  - List of favorites
  - Lists of "essential" items
- Simple aggregates
  - Top 10, Most Popular, Recent Uploads
- Tailored to individual users
  - Amazon, Netflix, …

7

Content-based Filtering (CBF)

- Predict like / dislike directly from features

Sci-Fi, Director: Mel Brooks

Romance, Starring: Julia Roberts, Richard Gere

29, Male, Sydney

24, Male, Canberra

33, Male, Canberra

R =

1	1	1	0	?	0
1	0	?	0	0	
⋮		⋮		⋮	
	0	0		1	?

R(user x, movie y) = f(Φ_x, Φ_y)

E.g., SVM or any trained classifier.

Φ are features

Koren et al

Collaborative Filtering (CF): KNN

- No features? k-nearest neighbor, e.g., k=2


This is user-based (user-user sim), can also do item-based (item-item sim)

Use any similarity metric you want

Error per prediction or ranking error.

R =

1	1	1	0	?	0
1	0	?	0	0	
⋮					
	0	0		1	?

$\frac{.71*1 + .50*0}{.71 + .50} = 0.59$

0.71

0.50

Salakhutdinov et al

Collaborative Filtering: PMF

- Or low k-rank matrix factorization, e.g. k=2


R =

1	1	1	0	?	0
1	0	?	0	0	
⋮		⋮		⋮	
	0	0		1	?

= U^T V

U^T	
.7	.3
.5	-.2
⋮	
-.1	.9

V		
.9	…	.1
.3		-.7

Latent Item Features

Latent User Features

Probabilistic matrix factorization (PMF) objective (reg. not shown).

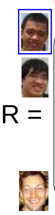
$$\min_{U,V} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - U_{\mathbf{x}}^T V_{\mathbf{y}})^2$$

Extensions to Standard Recommendation Methods

- User and item side information
- Social (and other) side information
- Cold-start
- Implicit Feedback

Stern, Herbrich, Graepel, WWW-09

Side Information in CF: Matchbox

R =

1	1	1	0	?	0
1	0	?	0	0	
⋮		⋮		⋮	
	0	0		1	?

= $(Ux)^T$ Vy

$(Ux)^T$	
.7	.3
.5	-.2
⋮	
-.1	.9

Vy		
.9	…	.1
.3		-.7

Project features into latent space – helps cold-start problem.

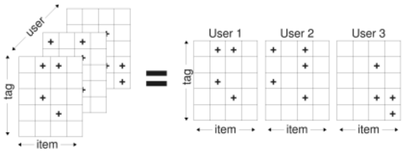
Reduces to previous PMF CF if x, y are indicators.

$$\min_{U,V} \sum_{(\mathbf{x},\mathbf{y})\in D} \frac{1}{2}(R_{\mathbf{x},\mathbf{y}} - [\sigma]\mathbf{x}^T U^T V \mathbf{y})^2$$

Rendle et al, WSDM-10

Tensor Factorization

- Multirelational recommendation (user, tag, documents)

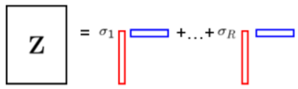


- Many ways to do tensor factorization
  - PARAFAC
  - Tucker (dense core tensor version of PARAFAC)
  - See slides by Tamara Kolda for intro:
    http://www.cs.cornell.edu/cv/tenwork/Slides/Kolda.pdf
    http://www.mat.uniroma2.it/~tvmsscho/Rome-Moscow_School/2012/files/kolda_2008.pdf

Kolda: http://www.cs.cornell.edu/cv/tenwork/Slides/Kolda.pdf

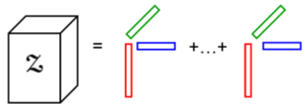
Tensor Factorization: PARAFAC

Singular Value Decomposition (SVD) expresses a matrix as the sum of rank-1 factors.



$$\mathbf{Z} = \sum_{r=1}^{R} \sigma_r \, \mathbf{u}_r \circ \mathbf{v}_r$$

CANDECOMP/PARAFAC (CP) expresses a tensor as the sum of rank-1 factors.



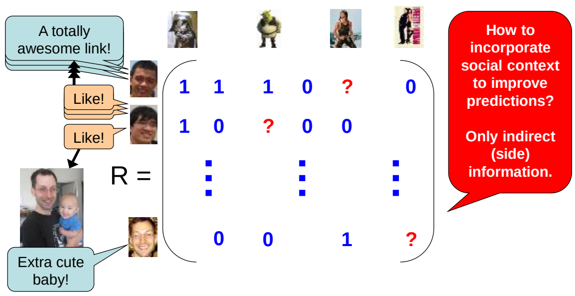
$$\mathcal{Z} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

Can you think of any uses of tensor factorization in recommendation?

Noel, Sanner et al WWW-12

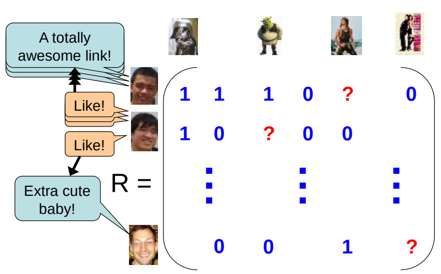
Social Recommendation

- Adds indirect social context to users



Noel, Sanner et al WWW-12

Social Collaborative Filtering



$$Int_{\mathbf{x},\mathbf{z}} = \frac{\#\text{ interactions b}\ldots}{\frac{1}{N(N-1)}\sum_{\mathbf{x}',\mathbf{z}'\neq\mathbf{x}'} \#\text{ intera}\ldots}$$

$$S_{\mathbf{x},\mathbf{z}} = \ln\left(Int_{\mathbf{x},\mathbf{z}}\right)$$

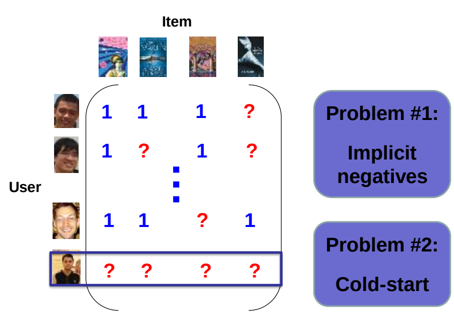
PMF + Social Regularization

$$\min_U \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends_{\mathbf{x}}} \frac{1}{2}\left(S_{\mathbf{x},\mathbf{z}} - \langle U_{\mathbf{x}}, U_{\mathbf{z}}\rangle\right)^2$$

PMF + Social Spectral Reg.

$$\min_U \sum_{\mathbf{x}} \sum_{\mathbf{z}\in friends_{\mathbf{x}}} \frac{1}{2} S^{+}_{\mathbf{x},\mathbf{z}} \|U_{\mathbf{x}} - U_{\mathbf{z}}\|_2^2$$

Cold-start Recommendation with Implicit Feedback (RecSys-14)



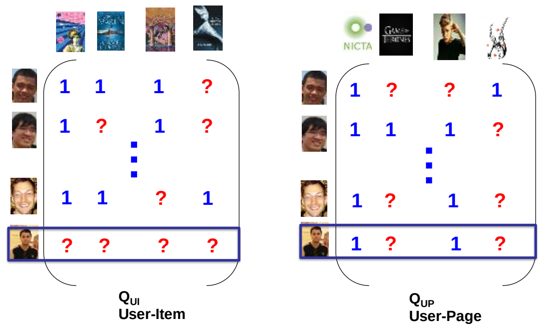
Problem #1: Implicit negatives

Problem #2: Cold-start

Implicit Negatives

- Also called “one-class” collaborative filtering
- Only occurs for problems with binary feedback
  - Assume “true” class is observed (liked, purchased, …)
  - Why doesn’t it occur in case of 1-5 rating feedback?
- What if we impute missing values = “false”
  - It throws probabilistic calibration
  - But it is **OK for ranking** under certain conditions
    - C. Elkan and K. Noto. Learning Classifiers from Only Positive and Unlabeled Data. KDD 2008.
- Often Jaccard works as better metric in one-class case

Cold-start: Leverage Side Information (e.g., Social Content)



What other sources of side information could be helpful?

How would you integrate it into the recommender system?

Additional Lecture Material

(Not tested)

Note on “Implicit”

- Used in many contexts (not only missing negatives)
  - Cases where have additional information on items
    - Whether a user rated a movie, book, etc.
    - Whether a user clicked on a movie, book, etc.
    - How much of a movie a user watched, or a book read
    - Which book pages they read
  - A form of (user,item) side information
    - Same item space (unlike user, or social side information)
    - Not clear what time on a page means vs. book purchase
    - Information such as click feedback may be very weak

Additional CF Tricks I

- User row normalization
  - Subtract user average from each user
  - Add back in before prediction
- Use of Pearson correlation similarity
  - Reported to work better for Netflix
- Computer weighted sum
  - I.e., Not weighted average so remove normalizer
  - OK for ranking (but not bounded for RMSE)
  - Prevents low similarity items from being divided by a small weight (=large rating)

Additional CF Tricks II

- Binary view of rating feedback?
  - Not only is a user rating important
    - But the fact that they rated (watched) it is as well
    - Consider users who’ve seen the same movies?
  - Convert 1-5 ratings to a single value 1 (rated)
  - Use Jaccard to measure user overlap
  - Use to augment cosine/Pearson similarity

Additional CF Tricks III

- Time sensitive recommendation
  - Item popularity changes over time
  - User preferences change over time
    - Each handled differently
  - CF ranking approach to handle user drift
    - $R(u,i) = \sum_{j \neq i} Sim(i,j) * decay(u,i,j) * R(u,j)$ / (optional normalizer)
    - $Decay(u,i,j) = e^{-\lambda \text{ (time\_now\_or\_i\_rated – time\_when\_u\_rated\_j)}}$
  - Assume i rated after j
    - Or we would not be trying to recommend it!
    - Weights user’s more recent ratings more highly