Homework 3: Discovering connections between domains to boost the recommendation quality

Machine Discovery (Fall 2016)

Shou-De Lin

High-quality recommendation on small data: How to handle Cold-Start Problems

- Cold-start is a notorious issue for any recommendation system
- For users or items with fewer ratings, CF cannot do a good job
- However, new users/items normally have fewer ratings
    - New users require the support from an RS, but there is no sufficient data for them.
    - New items are required to be recommended, but the system feels that they are not liked by users.
- You are given an assignment to improve the cold-start situation.

Task 3.1: Same domain/user/item, unknown mapping (1/2)

- You are running a opinion poll company that wants to investigate people's ratings on Legislators.
- You have identified 1000 people's phone number to call, and collect a list of legislators' names to be rated.
- Two employees R1, R2 are hired to do this survey, they don't know each other and make the survey independently.
  - Each of them do not need to dial all 1000 numbers, nor do they need to ask for the opinion on ALL legislators for each call.
  - R2 is lazier so dialed fewer calls than R1.
- Since this survey is sensitive, R1 and R2 are asked to anonymize the callee's names as well as the legislators' name while turning in the results. Eventually they need to turn in their results, each as a matrix.

Task 3.1: Same domain/user/item, unknown mapping (2/2)

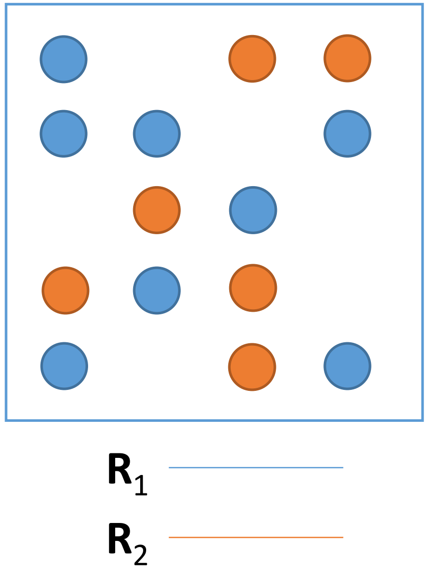
- R1 and R2 performed anonymization independently, meaning they are using different id for the same callee, and different ids for the same legislator.
- After tuning in their matrices, R1 left the company and disappeared. Nobody knows the mapping between real names and id in this matrix.
- In R2 we know the true mapping, but since the rating is too few, we cannot train a good prediction model.
- Your goal is to predict the missing values in the R2 matrix (which is sparser than R1) using not only R2 but R1.

Task 3.1:Problem setting

- Suppose that two rating matrices $\mathbf{R}_1$, $\mathbf{R}_2$ recorded for the user preferences on a set of items, and we also know that
  - The user set in $\mathbf{R}_1$ is exactly the same as $\mathbf{R}_2$'s.
  - The item set in $\mathbf{R}_1$ is exactly the same as $\mathbf{R}_2$'s, too.
  - $\mathbf{R}_1$ is sparse, and $\mathbf{R}_2$ is even sparser.
  - Mappings between IDs in R_1 and R_2 is unknow.
  - However, the set of recorded ratings in $\mathbf{R}_1$ and $\mathbf{R}_2$ are disjoint.
- Can we achieve better predictions on $\mathbf{R}_2$ by leveraging the additional matrix $\mathbf{R}_1$, instead of only considering $\mathbf{R}_2$-train?
  - Baseline: using biased-MF on only R_2-train

Task 3.1: Data

- Auxiliary rating matrix ($\mathbf{R}_1$):
  - #user = 50,000
  - #item = 5,000
  - #ratings = 1,650,387
- Training rating matrix ($\mathbf{R}_2$-train):
  - #user = 50,000
  - #item = 5,000
  - #ratings = 825,193
- Testing rating matrix ($\mathbf{R}_2$-test) (our target):
  - #user = 50,000
  - #item = 5,000
  - #ratings = 825,194



Task 3.1: format

- Format:
  - Each line of the .txt files contains three numbers.
  - <user-id> <item-id> <rating>
  - The numbers are separated by a single blank character.
  - However, for ‘task1_test.txt’, no rating is provided.
- Rating values are real numbers in [0, 1].
- Notice that the user-id’s do NOT follow an identical mapping, as well as the item-id’s.
  - For example, user_id *x* in $\mathbf{R}_1$ is likely NOT the user_id *x* in $\mathbf{R}_2$, but user *x* in $\mathbf{R}_2$-train will be identical to user *x* in $\mathbf{R}_2$-test.

Task 3.2: Same domain, different user/item

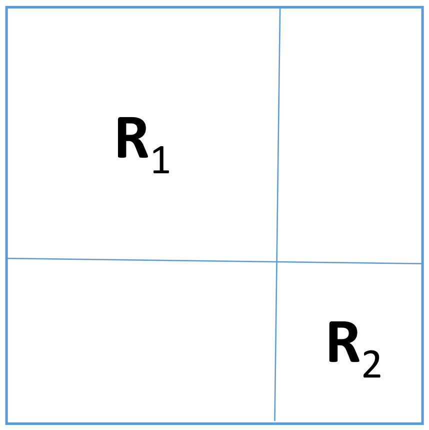
- You are the CEO of an online textbook store Ambzon which used to sell only English books in USA.
- You want to expand the business across the world to sell some Chinese textbooks in Taiwan. Therefore, you need a recommender system for such purpose.
- You only have few user-item rating records for the Chinese textbooks rated by Taiwanese.
- However, you have much more ratings on the English textbooks rated by the American people.
- Can you leverage the ratings from the English textbooks to enhance the recommendation on Chinese textbooks?

Task 3.2 Problem Setting

- Similar to Task 1, we still have two rating matrices $\mathbf{R}_1$, $\mathbf{R}_2$ recorded for the user preferences on a set of items, but now
  - The user sets in $\mathbf{R}_1$ and $\mathbf{R}_2$ are mostly disjoint, and so are the item sets.
  - $\mathbf{R}_1$ and $\mathbf{R}_2$ are very sparse.
- Can we achieve better predictions on $\mathbf{R}_2$ by leveraging the additional matrix $\mathbf{R}_1$, instead of only considering $\mathbf{R}_2$-train?
  - Baseline: using biased-MF on R_2-train

Task 3.2: Data

- Auxiliary rating matrix ($\mathbf{R}_1$):
  - #user = 30,000
  - #item = 3,000
  - #ratings = 1,153,800
- Training rating matrix ($\mathbf{R}_2$-train):
  - #user = 20,000
  - #item = 2,000
  - #ratings = 285,518
- Testing rating matrix ($\mathbf{R}_2$-test) (our target):
  - #user = 20,000
  - #item = 2,000
  - #ratings = 285,518



Task 3.2: Format

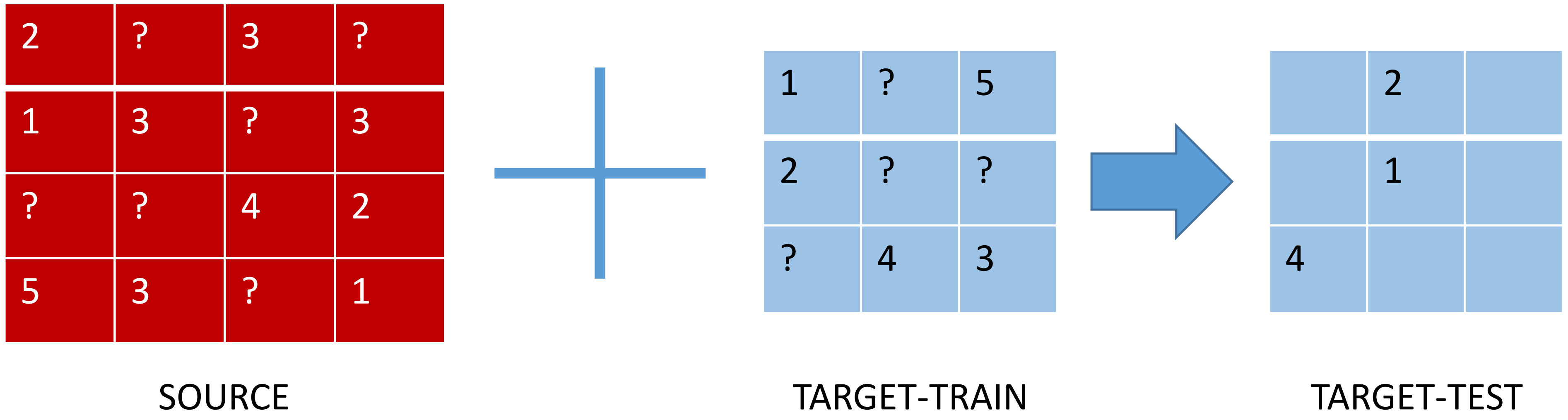
- Format:
  - Each line of the .txt files contains three numbers.
  - <user-id> <item-id> <rating>
  - The numbers are separated by a single blank character.
  - However, for ‘task2_test.txt’, no rating is provided.
- Rating values are real numbers in [0, 1].
- Notice that the user-id’s do NOT follow an identical mapping, and so are the item-id’s

Task 3.3: Different domain, different user/items

- You become a CEO of a new online store selling shoes
  - A recommendation system that recommends suitable shoes to users can boost your selling
  - However, since this is a new business, you don’t have a lot of data about customers’ purchasing record
  - You cannot find online some people sharing rating data about shoes
  - However, you can find that there are many freely available ratings about books, music, movie, etc.
  - Can those ratings be used to boost the quality of your system?

Task 3.3: Problem setting

- Generally, you will have
  - A source matrix
  - A target-training matrix (a sub-matrix of the target matrix, usually sparse)
- Prediction: A target-testing matrix (a sub-matrix of the target matrix)



2	?	3	?
1	3	?	3
?	?	4	2
5	3	?	1

SOURCE

1	?	5
2	?	?
?	4	3

TARGET-TRAIN

	2	
	1	
4		

TARGET-TEST

Task 3.3: Data

- You are given a source matrix $\mathbf{R}_1$ as an open recommendation data, and you have to build a model on your sparse target matrix $\mathbf{R}_2$
- $\mathbf{R}_1$: All rating records for 500 users and 500 items in a domain
- $\mathbf{R}_2$:
  - Only a few (100) users purchased lots of items
  - The rest of the users (400) purchased few (i.e. 5 items)
    - The target users to make recommendation
- Note that
  - User or item ids in $\mathbf{R}_1$ and $\mathbf{R}_2$ are independent (i.e. they are not matched)
  - The source matrix $\mathbf{R}_1$ and the target matrix $\mathbf{R}_2$ are not from the same domain
  - The rating values of $\mathbf{R}_1$ and $\mathbf{R}_2$ are in $\{1,2,3,4,5\}$, but it is ok to predict any real value $\in \mathbb{R}$

Solutions

- Any ***free-to-use*** third-party tools are permitted to use
  - However, you **CANNOT** use your classmates' codes.
  - *It's your obligation to ensure the feasibility and scalability of your program.*
- If you have no idea how to start, feel free to use solutions in the TA course:
  - Multi-Domain Collaborative Filtering
  - Cross-Domain Collaborative Filtering with Factorization Machines
  - Personalized Recommendation via Cross-domain Triadic Factorization.
  - Transfer Learning for Collaborative Filtering via a Rating-matrix Generative Model
  - Cross-domain recommendations without overlapping data: myth or reality?
  - Transfer Learning in Collaborative Filtering for Sparsity Reduction.
  - Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.

Evaluation

- Evaluation criterion:
  - Root Mean Squared Error (RMSE)
  - If your prediction is $\overline{R_{ui}}$ given testing data R_{ui}
    - $RMSE = \sqrt{\frac{1}{|R|}\sum_{(u,i)\in R}(R_{ui} - \overline{R_{ui}})^2}$

Datasets – data format

- We provide you 3 datasets for testing (`test1/, test2/, test3/`);
  - each line in the ***source*** and ***training*** matrix is a triple "`u_id item_id rating`";
  - for the ***testing*** matrix, you would have "`u_id item_id ?`"

- For each of the three datasets, we have
  - ***Source*** matrix (`source.txt`); the auxiliary ***source dataset*** $\mathbf{R}_1$ for your to discovery knowledge
  - ***Training*** matrix (`train.txt`); the training portion of the ***target dataset*** $\mathbf{R}_2$ to conduct your transfer learning and training
  - ***Testing*** matrix (`test.txt`); the testing portion of the ***target dataset*** $\mathbf{R}_2$ to do the prediction

What you should do - prediction file

- For dataset = {`test1, test2, test3`}, give a testing data
  - `hw3/dataset/test.txt`
  - each line contains `uid item_id ?`(separated by ‘ ’ (space character))
- You should ***make one prediction file*** in your submission for each task
  - `[student_id]/dataset/pred.txt`
- You should make the prediction file by replacing the `?` in each line with your ***prediction*** for rating
  - Each line contains `uid item_id rating` by replacing `?` in `test.txt` with your prediction `rating` ∈ ℝ (rather than ∈{1,2,3,4,5})
  - For example, in the id file: `3 7 ?`
  - You may predict: `3 7 4.98`

What you should do - report

- A single PDF file: `[student_id]/report.pdf`
- ***Explain your ***model*** and ***experiment results*** (under different experimental settings) ***
  - ***model:*** model structure, learning algorithm, feature engineering method (if you made some feature only) and how you conduct discovery and transfer
  - ***experiment results:*** is your discovery/transfer system sensitive to parameter?
  - ***validation:*** how did you make your validation data?
- All the settings/assumptions/references you used.
- The contribution of each individual member in this homework.

Submission Guidelines

- All the contents listed below should be included in a single folder named by the concatenation of the student IDs in your team and then **zip** it before uploading to CEIBA.
  - If the team consists of R04922001, B02922002, D05922012,
    - `[student_ids]= R04922001-B02922002-D05922012` (The order of of IDs doesn't matter)
    - Please submit the file using only one of your team member's CEIBA account.
  - source codes placed in path: `[student_ids]/src/`
  - a list of the third-party tools you used: `[student_ids]/used-tools.txt`
  - a report explaining your method: `[student_ids]/report.pdf`
  - Your predictions for `test1` and `test2` and `test3`:
    - `[student_ids]/test1/pred.txt`
    - `[student_ids]/test2/pred.txt`
    - `[student_ids]/test3/pred.txt`
  - a README file including the instructions to execute your codes and reproduce your results:
    - `[student_ids]/README.txt`
  - You **don't need to upload data**. You can assume they are available in `test1/`, `test2/` and `test3/` in your code.
- Submission deadline: 2016/12/12 09:00am
  - Late penalty: your score will obtain 50% deduction if within 24 hours delay. You will not receive any credit if delayed for more than 24 hours.
- Feel free to contact TAs via mails or TA hours.

Scoring

- Prediction improvement over baseline (50%)
  - Baseline RMSE (MF on R2-train) minus your model's RMSE
- Report (50%): novelty, efforts, and clearness
- Any type of plagiarism is strictly inhibited.
  - Please *cite* any references or materials you used.
  - Discussions between classmates are encouraged, but ALL the code and final design of model must complete by yourself.
  - Please give credit to your classmates in the report if you used any ideas from them.
  - You will simply receive -*X* point for any violations.
- Violation on the submission guideline: (-20%)