Chp 2: Neighborhood Based Collaborative Filtering

§ 2.1 Introduction

- Types of collaborative filtering

  (a) User based: ratings provided by similar users to user A are used to recommend to the A

  (b) Item based: To make recommendation for target item B, you determine a set of similar items to B, rated by user A, and predict the rating of item B

- Rmk: user based – uses neighbors ratings, similarity among rows of R

  item based – uses your own ratings, similarity among columns of R.

- $R \in \mathbb{R}^{m \times n}$ m – users n – items

- Problems: (a) predicting missing values in R

  (b) Determing the top k-items/users of unknown ratings

§ 2.2 Properties of R

- Entries in R can be supported on really any subset of $\mathbb{R}$
- Typically the # ratings is a skewed dist. so if $M \in \mathbb{R}^{m \times n}$ s.t. $m_{ij} = \begin{cases} 1 & R_{ij} \neq \text{missing} \\ 0 & \text{o.w} \end{cases}$

  then $\text{colsums}(M) = I^T M$ is a skewed dist.

- Implications: over recommendation of popular items, few often rated products make neighborhood methods unstable.

§ 2.3 Predicting Ratings with Neighborhood – Based Methods

- Basic idea: similar users rate items similarly

  similar items are rated similarly by the same user.

- Think about this like a N.N. classification problem

§ 2.3.1 User-Based Neighborhood Models

- Create neighboords for each user

- Let $I_u = \{$ indicies from which ratings have been given by user $u\}$

- Measures for similarity $Sim(u,v)$

  - Pearson Corr. for ratings in $I_u \cap I_v$

$$Sim(u,v) = \frac{\sum_{k \in I_u \cap I_v} (R_{uk} - \mu_u)(R_{vk} - \mu_u)}{\sqrt{\sum_{k \in I_u \cap I_v} (R_{uk} - \mu_u)^2} \sqrt{\sum_{k \in I_u \cap I_v} (R_{vk} - \mu_v)^2}}$$

  where $\mu_u = |I_u|^{-1} \sum_{k \in I_u} R_{uk}$.

  Rmk: Not quite Pearson because μ_u, μ_v is computed over I_u, I_v, respectively, not $I_u \cap I_v$. But you get some computational speed up this way.

  - Define a users neighborhood on an item-by-item basis so that your peer group all have ratings for the item in question

  - The estimated rating is then provided by the weighted average

$$\hat{R}_{ui} = \frac{\sum_{v \in P_u(i)} sim(u,v) \cdot R_{vi}}{\sum_{v \in P_u(i)} |sim(u,v)|} \quad : P_u(i) = \{\text{peers most similar to } u \text{ who rated } i\}$$

- Issues: Ratings on different scales. (general neg./pos. raters)

- Sol: Rowise centering - before defining estimate.

- Let $s_{uj} = R_{uj} - \mu_u$ be the user-centered rating

- So the estimated centered rating is then added to the user's rating to provide the estimate

$$\hat{R}_{uj} = \mu_u + \frac{\sum_{v \in P_u(i)} \text{sim}(u,v) \cdot s_{vj}}{\sum_{v \in P_u(i)} |\text{sim}(u,v)|}$$

- Extensions are abound with modifications to $\text{Sim}(u,v)$ and $P_u(j)$.

- Other similarity functions may include

  - Raw Cosine: $\text{RawCosine}(u,v) = \frac{\sum_{k \in I_u \cap I_v} R_{uk} \cdot R_{vk}}{\sqrt{\sum_{k \in I_u} R_{uk}^2} \sqrt{\sum_{k \in I_v} R_{vk}^2}}$

  - issues: no way to control for user bias (in general high/low rater)

  - Discounted Similarity $DS(u,v) = \text{Sim}(u,v) \cdot \frac{\min\{|I_u \cap I_v|, \beta\}}{\beta}$

    - weights the pearson coefficient by a factor in $[0,1]$ to discount a user who rates differently from u.

- Other Prediction Functions

  - Z-Score prediction $\hat{R}_{uj} = \mu_u + \sigma_u \frac{\sum_{v \in P_u(i)} \text{sim}(u,v) \cdot z_{vi}}{\sum_{v \in P_u(i)} |\text{sim}(u,v)|}$

  where $z_{vi} = \frac{R_{vi} - \mu_v}{\sigma_v}$.

  Rmk: Not clear if this is better/worse in practice.

- One may choose to amplify a similarity by $[\text{sim}(u,v)]^\alpha$: $\alpha \in \mathbb{R}_+$

  Rmk: $\alpha > 1$, amplifying the importance of similarity.

- all of this can be done for classification as well by mapping the peer groups rating to the support of R (e.g. voting).

- Typical to further refine $P_u(i)$ s.t. weakly/negatively correlated users are removed

- To deal with the fact that item ratings have long tail behavior, it is typical to weight $Sim(u,v)$ by item popularity

  - Let m_j = # of times item $j \in [m]$ has been rated.

  - Typical to use the weighted pearson similarity

$$Sim(u,v) = \frac{\sum_{k \in I_u \cap I_v} w_k \cdot (R_{uk} - \mu_u)(R_{vk} - \mu_u)}{\sqrt{\sum_{k \in I_u \cap I_v} w_k \cdot (R_{uk} - \mu_u)^2} \sqrt{\sum_{k \in I_u \cap I_v} w_k \cdot (R_{vk} - \mu_v)^2}}$$

  where $w_k = \log\left(\frac{m}{m_j}\right) \geq 0$

§ 2.3.2 Item-Based Neighborhood Models

- Suppose R is row-wise mean centered so that $S = R - \frac{1_m^T R}{m}$ and let

  $U_i = \{$ user ids who have rated item $i \}$

- Then we measure similarity among items by their adjusted cosine-similarity

$$Adj\,Cosine(i,j) = \frac{\sum_{u \in U_i \cap U_j} S_{ui} \cdot S_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} S_{ui}^2} \sqrt{\sum_{u \in U_i \cap U_j} S_{uj}^2}}$$

  - Rmk: We could use pearson, but cosine typically has better results.

- Consider estimating for item $t \in [n]$. Let $Q_t(u) = \{$ indices for most similar items to t which have been rated by $u \}$

- Then the estimated rating is given by

$$\hat{R}_{ut} = \frac{\sum_{j \in Q_t(u)} Adj\,Cosine(j,t) \cdot R_{uj}}{\sum_{j \in Q_t(u)} |Adj\,Cosine(j,t)|}$$

- The similar extensions as discussed above are also available.

§ 2.3.3 Efficient Implementation / Computational Complexity

- Split the computation into an online and offline phase
- Offline: similarity / peer group calculations Online: computation of ratings
- Let $n' << n$ be the max number of ratings of a user

  $m' << m$ be the max number of ratings of an item
- Complexity:

	Offline	Online
User-Based	$O(m^2 \cdot n')$	$O(m \cdot k)$
Item-Based	$O(n^2 \cdot m')$	$O(n \cdot k)$

§ 2.3.4 Comparing Methods

- Item based is typically more accurate b.c. using a users own rating
- User-Based enjoys novelty because users in Peer Groups may have non-overlapping interests
- Interpretable w.r.t. nearest neighbors
- Stable w.r.t to the addition of new items / users
- Central disadvantage is that the offline phase may be impractical in large settings
- Sparsity is also a concern

§ 2.3.6 Unified User / Item Based Methods

Q: User / Item based methods look for the most similar rows/columns

Can we look for the most similar "entries" instead?

A: Yes. 1. Determine most similar entries (e.g. by summing row/column cosine similarity. 2. Predict using weighted comb. of these entries

§2.4 Clustering & Neighborhood Based Methods

- Sometimes when the dataset is particularly large $O(m^2 n')$ is just too costly.
- Idea: instead of creating a peer group for every user, instead partition the set of all users and use those clusters
- Let $\bar{Y}_1, \ldots, \bar{Y}_k$ define the cluster centers / representatives $\in \mathbb{R}^n$
- A K-means sketch is given by

  a. Determine clusters $C_1, \ldots, C_k$ by assigning rows of R to the closest representative $\bar{Y}_1, \ldots, \bar{Y}_k$

  b. Reset $\bar{Y}_i$ to the centroid of the current set of points
- Since missing data is an issue, using ℓ_1 distance is preferable.

§ 2.5 Dimensionality Reduction & Neighborhood Methods

- Due to the amount of missing data, latent factor models are often used.
- For user based methods the goal is to map $R \to R' \in \mathbb{R}^{m \times d}$ for $d << n$
- Applying basic collaborative filtering methods to this will be much more efficient
- The reduction occurs by the following SVD
  1. Impute missing values by rowmean/colmean to yield $R_f \in \mathbb{R}^{m \times n}$
  2. Compute the item-similarity $S = R_f^T R_f = P \Delta P^T$
  3. Then $R' = R_f P = P \Delta^{1/2}$
- PCA approaches are also used. The most robust may be mean centering raw-column then taking the svd.

§ 2.5.1 Handling Problems with Bias

- Due to low samples, mean imputation can cause very high level of bias
- Some suggest using EM to estimate a covariance matrix between items after proposing a probabilistic model
- The Basis matrix P on the top d-eigenvectors of $\hat{\Sigma}_{EM}$ to stabilize variance
- Another idea is projecting R itself onto P by taking weighted sum of the entries of P.
- There are still issues in estimation of $\vec{\Sigma}$ when R is very sparse
- Suppose $R = Q \Sigma P^T$ (assuming no missing) with truncation $Q_d \Sigma_d P_d^T$
- The challenge occurs in doing this over missing data
- Chp 3 covers some of the non-convex solutions.

§ 2.6 Regression Perspectives

- Since All methods here are just predictions using weighted sums, they can be interpreted as linear models
- By reformulating the heuristic weight choices we can see these methods as optimization based methods

§ 2.6.1 User-Based N.N Regression

- Suppose we adopt the model $\hat{R}_{uj} = \mu_u + \sum_{v \in P_u(j)} w_{uv}^{(user)} (R_{vj} - \mu_v)$
- This inspires the minimization problem

$$\underset{w^{(user)}}{\text{minimize}} \quad J_u = \sum_{j \in I_u} (R_{uj} - \hat{R}_{uj})^2 = \sum_{j \in I_u} \left(R_{uj} - \left[\mu_u + \sum_{v \in P_u(j)} w_{uv}^{(user)} (R_{vj} - \mu_v) \right] \right)^2$$

- Hence the full problem is stated by $\bar{W} = \underset{W}{\arg\min} \sum_{u=1}^{m} J_u(W)$

- Nice because we can use regularization ideas in addition

- There are a set of further refinements to unbias estimates in sparse settings

§26.2 Item-Based N.N. Regression

- Define the predictions by $\hat{R}_{ut} = \sum_{j \in Q_t(u)} w_{jt}^{(item)} R_{uj}$ $\quad Q_t(u) = \{$items similar to t rated by $u\}$

- Inspires the item-wise OLS problem

$$\min J_t = \sum_{u \in U_t} \left(R_{ut} - \sum_{j \in Q_t(u)} w_{jt}^{(item)} R_{uj} \right)^2$$

$$U_t = \{\text{users who have rated } t\}$$

- And the full OLS problem is $\min \sum_{t=1}^{m} J_t$ w.r.t. W

- Similar de-biasing techniques can be used to fine tune the model.

- Fusing together both the item/user based problems generally tend to lead to better estimation of R

§ 26.4 Joint Interpolation w/ Similarity Weighting

- Let $S = \{(u,t) : R_{ut} \text{ is observed}\}$

- Idea: Penalize predictions $\hat{R}_{us}$ for being too far from R_{ut} when $\text{Sim}(u,t)$ is high

$$\min \sum_{s:(s,u) \in S} \sum_{j \neq s} \text{Adj}\cos(j,s) \cdot (R_{us} - \hat{R}_{uj})^2$$

§26.5 Sparse Linear Models (SLIM)

- Suppose we want to (possibly) use all users who rated an item in prediction

$$\hat{R}_{ut} = \sum_{j=1}^{n} w_{jt}^{(item)} R_{uj} \qquad \forall\, u \in [m],\ t \in [n]$$

- Then assuming $w_{tt}^{(item)} = 0$ we hope to model $\hat{R} = RW^{(item)}$ s.t. $diag(W^{(item)}) = 0$

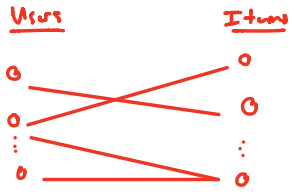
- This suggests the minimization $\min_W \|\hat{R} - RW^{(item)}\|_F^2 + \text{Regularization}$ which written component-wise gives

$$\min J_t^s = \sum_{u=1}^{m} (R_{ut} - \hat{R}_{ut})^2 + \lambda_2 \sum_{j=1}^{n} (w_{jt}^{(item)})^2 + \lambda_1 \sum_{j=1}^{n} |w_{jt}^{(item)}|$$

- Elastic-net approach to regularization — not just N.N. fitting

§2.7 Graph Models for Neighborhood-Based Methods

- Represent the rating matrix as a Bi-Partite Graph where edges represent ratings



- We can define neighborhoods (both User/Item) based on Graph properties
  - Random Walks: Use PageRank similarities for neighborhood definition
  - Katz Measure:
    - Let $m_{ij}^{(t)}$ = # of walks of length t between $i \leftrightarrow j$.
    - Then for parameter $\beta < 1$, $Katz(i,j) = \sum_{t=1}^{\infty} \beta^t m_{ij}^{(t)}$
    - Let $K \in \mathbb{R}^{m \times m}$ be the symm. matrix of Katz metrics then

$$K = \sum_{t=1}^{\infty} (\beta A)^t = (I - \beta A)^{-1} - I$$

could also be W

    - Spectral clustering of K can then give you groups