Lecture 16

- Collaborative Filtering
  (Model-Based Approaches)

IT492: Recommendation Systems (AY 2023/24) — **Dr. Arpit Rana**

Baseline Estimates

Typical *CF* data exhibit systematic tendencies -

- for some users to give higher ratings than others, and
- for some items to receive higher ratings than others.

We account for these systematic tendencies within the ***baseline estimates***.

Baseline Estimates

A ***baseline estimate*** for an unknown rating r_{ui} is denoted by b_{ui} and accounts for the user and item effects:

$$b_{ui} = \mu + b_u + b_i$$

Here,

μ is the overall average rating,

b_u is the observed deviation in user u's ratings, and

b_i is the observed deviation in item i's raings.

Baseline Estimates

In order to estimate b_u and b_i one can solve the following least squares problem:

$$\min_{b_*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 \left(\sum_u b_u^2 + \sum_i b_i^2\right)$$

Here, $K = \{(u, i) \mid r_{ui} \text{ is observed}\}$

Baseline Estimates

An easier, yet somewhat less accurate way to estimate the parameters is by decoupling the calculation of the b_i 's from the calculation of the b_u 's.

First, for each item *i* we set,

$$b_i = \frac{\sum_{u \in \mathrm{R}(i)} (r_{u,i} - \mu)}{\lambda_2 + |\mathrm{R}(i)|} .$$

Then, for each user *u* we set,

$$b_u = \frac{\sum_{i \in \mathrm{R}(u)} (r_{u,i} - \mu - b_i)}{\lambda_3 + |\mathrm{R}(u)|} .$$

Averages are shrunk towards zero by using the regularization parameters, λ_2 , λ_3 , which are determined by cross validation.

Typical values on the Netflix dataset are: λ_2 = 25, λ_3 = 10.

MF with Baseline Estimates

A typical SVD model associates each user u with a user factor $p^u \in R^k$, and each item i with an item-factor $q^i \in R^k$.

The prediction is done by taking an inner product, i.e.,

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

To learn the factors (p_u and q_i), the system minimizes the regularized squared error on the set of known ratings:

$$\min_{p_*, q_*, b_*} \sum_{(u,i) \in \mathcal{K}} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda_3 (\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2)$$

Neighborhood-based Models: Similarity-based Methods

In item-item CF, the predicted value of r_{ui} is taken as a weighted average of the ratings of neighboring items:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N_i} w_{ij} \cdot (r_{uj} - b_{uj})}{\sum_{j \in N_i} w_{ij}}$$

We can see ratings are adjusted for user and item effects through the baseline estimates; and w_{ij} is a ***shrunk correlation coefficients***.

$$w_{ij} = \frac{n_{ij}}{n_{ij} + \beta} \cdot s_{ij}$$

Neighborhood-based Models: Similarity-based Methods

Note the dual use of the similarities in the previous formulation -

- identification of nearest neighbors, and
- as the interpolation weights

Similarity-based methods became very popular because they are intuitive and relatively simple to implement.

- **Explainability**: Identifying which of the past user actions are most influential on the computed prediction.
- **New ratings**: Item-item neighborhood models can provide updated recommendations immediately after users enter new ratings without needing to re-train the model and estimate new parameters.

  This assumes that relationships between items (the s_{ij} values) are stable and barely change on a daily basis.

Neighborhood-based Models: Similarity-based Methods

They also raise some concerns -

- The similarity function (s_{ij}), which directly defines the interpolation weights, is arbitrary.

  Suppose that a particular item is predicted perfectly by a subset of the neighbors. In that case, we would want the predictive subset to receive all the weight, but that is impossible for bounded similarity scores like the Pearson correlation coefficient.

- SImilarity weights do not take into account the relationship among neighbours.

  Suppose that our items are movies, and the neighbors set contains three movies that are highly correlated with each other (e.g., sequels such as “Lord of the Rings 1–3”). By ignoring the similarities among the three movies, we may end up triple counting the information provided by the group.

Neighborhood-based Models: Similarity-based Methods

They also raise some concerns -

- By definition, the interpolation weights sum to one, which may cause overfitting.

  Suppose that an item has no useful neighbors rated by a particular user. In that case, it would be best to ignore the neighborhood information, staying with the more robust baseline predictors, whereas, the standard neighborhood formula uses a weighted average of ratings for the uninformative neighbors.

- Neighborhood methods may not work well if variability of ratings differs substantially among neighbors.

Neighborhood-based Models Revisited

To resolve the aforementioned issues,

- as above, we use the similarity measure to define neighbors for each prediction.
- we search for optimum interpolation weights without regard to values of the similarity measure.

Given a set of neighbors $S^k(i; u)$ we need to compute interpolation weights $\{\theta_{ij}^u | j \in \mathbf{S}^k(i; u)\}$ that enable the best prediction rule of the form -

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in N_i} \theta_{ij}^u \cdot (r_{uj} - b_{uj})$$

As explained earlier, it is important to derive all interpolation weights simultaneously to account for interdependencies among the neighbors.

A Global Neighborhood Model

Previous models were centered around user-specific interpolation weights— θ_{ij}^{u} relating item i to the items in a user-specific neighborhood $S^k(i; u)$.

In order to facilitate global optimization, we would like to abandon such user-specific weights in favor of global item-item weights independent of a specific user.

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in \mathrm{R}(u)} (r_{uj} - b_{uj}) w_{ij} \, .$$

More refinements on this formulation have been suggested in the literature which we are not covering in this course.

Learning-based Methods

Learning-based methods that use neighborhood or similarity information can be divided in two categories:

- Factorization methods (e.g. MF), and
- **Neighborhood learning methods (e.g. SLIM).**

Neighborhood Learning Methods

- Standard neighborhood-based recommendation algorithms determine the neighborhood of users or items directly from the data, using some pre-defined similarity measure like PC.

- Recent developments have shown the advantage of learning the neighborhood automatically from the data, instead of using a pre-defined similarity measure.

Neighborhood Learning Methods

A representative neighborhood-learning recommendation method is the Sparse Linear Model (SLIM) algorithm, developed by Ning et al. 2011.

- In SLIM, a new rating is predicted as a sparse aggregation of existing ratings in a user's profile,

$$\hat{r}_{ui} = \mathbf{r}_u \mathbf{w}_i^\top,$$

- Here r_u is the u^{th} row of the rating matrix R and w_i is a sparse row vector containing $|\,\mathrm{I}\,|$ aggregation coefficients.

- Essentially, the non-zero entries in w_i correspond to the neighbor items of an item i.

Neighborhood Learning Methods

- The neighborhood parameters are learned by minimizing the squared prediction error.

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \frac{1}{2}\|R - RW\|_F^2 + \frac{\beta}{2}\|W\|_F^2 + \lambda\|W\|_1 \\ \text{subject to} \quad & W \geq 0 \\ & \operatorname{diag}(W) = 0. \end{aligned}$$

- The non-negativity constraint on W imposes the relations between neighbor items to be positive.
- The constraint diag(W) = 0 is also added to the model to avoid trivial solutions (e.g., W corresponding to the identity matrix) and ensure that r_{ui} is not used to compute the predicted rating.

Revision of Topics Covered So Far…

Content-based Methods:

- Feature Extraction from User Reviews,
- User Preference Modeling, Evaluation.
- Extending it to Conversational (GUI-based multi-round) RS,
- Use Sentiment Analysis to Add More Features from User Reviews
- Evaluation

Collaborative Methods:

- Neighborhood-based Collaborative Filtering (User-User, Item-Item)
- Model-based Collaborative Filtering (Latent Factor Models: MF and its variants)
- Extending it to Conversational (GUI-based multi-round) RS,

Limitations of Collaborative Methods

Collaborative Methods have the following disadvantages:

- ***Sparsity***: The number of observed ratings is usually very small compared to the number of user-item pairs. Therefore, it is challenging to find similar users, similar items, or other patterns that are non-spurious.

- ***Cold-start items and users***: These systems would not be able to recommend the new item (not substantially rated) or recommend to the new user (who has not rated substantial number of items).

- **Popularity bias**: These methods recommend items based on ratings and hence they tend not to recommend products with limited historical data.

- **Shilling attacks/ Data Poisoning:** In collaborative settings, malicious users and/or competing vendors may insert fake profiles in an effort to affect the rating predictions for their own advantages.

Limitations of Content-based Methods

The main advantage of content-based methods is that they are ***easy to explain at feature-level***. Their most significant challenges include the following:

- ***Degree of content analysis:*** Their ability to discriminate between items depends on the granularity of the item representations. If two different items are represented by the same set of features, they are indistinguishable and equally likely to be recommended.

- ***Over-specialization:*** These methods tend to recommend items that are similar to items the user has liked in the past. Thus, they often provide the least serendipitous recommendations.

- ***Cold-start user:*** A new user, with an immature profile, is less likely to get accurate recommendations

Next Lecture

- Hybrid Methods