Jacob Penrod Final Project – Result Milestone

For the first part of my final project implementation of Hidden Markov Model in LibRec, I cleaned a dataset consisting of LastFM Playlists for users. To have a place to develop my dataset and to have a point of comparison I decided to tune LibRec’s implementation of TimeSVD++. The dataset that I used for TimeSVD++ consisted of a [User Id], [Artist Id], [Rating], [Hour] to fit the LibRec text reader UIRT format.

A few notes on my dataset.

* User ID and Artist ID are both just iterating keys based on the original data. I loaded the full dataset into Microsoft SQL, 19 million rows and created foreign keys to tables of distinct users, artists and songs for easy reference and aggregations.
* Ratings are all 5, I am using an implicit rating, basically assuming if a person listened to a song they liked it, the original dataset does not contain user ratings.
* Hours are the average hour that they listened to that artist. If I listened to Metallica at 1 3 5 it would average out to a (9/3) 3 for that user.

I will play with different aggregations, such as Min and Max or Median for the Hour column to see if that gets better results in my final paper.

Presumably I could do better by going out to genre, but I don’t have that information in my dataset.

I started with songs and the hour they were listened to, but I would always get loss that was too big for TimeSVD++ and then it would not converge. I needed to use artists instead, and I was able to get results after that.

I trimmed my dataset down to users who had 12 distinct hours of listening to artists. There are 501 users and 3057 distinct artists and 35,650 total records, still sparse but workable.

For TimeSVD++ in LibRec I had to make changes to the base algorithms to use the difference in hours as opposed to number of days ( the default implementation ).

**private static int** hours(**long** t1, **long** t2) {  
 **return** *hours*(Math.*abs*(t1 - t2));  
}

My metrics are all with ranking set to true and with my Top-N being set to 5.

I worked with trying other bins other than 6, but no other values seemed to converge.

My results metrics are AUC, Precision and Recall. By emphasizing artists in the model and deemphasizing users I observed better results. If I go much outside the range in the chart below, I will get no precision, recall and AUC. Even increasing the User Regularization by .001 will remove all predictive power. Item Regularization of .02 and a User regularization of .001 gives the best results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item Regularization | User Regularization | Precision | Recall | AUC |
| .001 | .001 | 0.0071 | 0.0039 | 0.5177 |
| .001 | .002 | 0 | 0 | .5 |
| .002 | .001 | 0.0071 | 0.0039 | 0.5177 |
| .005 | .001 | 0.0071 | 0.0039 | 0.5177 |
| .009 | .001 | 0.0071 | 0.0039 | 0.5177 |
| .01 | .001 | 0.0071 | 0.0039 | 0.5177 |
| .02 | .001 | 0.0143 | 0.0397 | 0.5355 |
| .03 | .001 | 0.0071 | 0.0357 | 0.5178 |
| .04 | .001 | 0.0071 | 0.0357 | 0.5178 |