Improving matrix factorization

Shrainik Jain (1323338)
shrainik@cs.washington.edu

Arunkumar Byravan (1222561)
barun@cs.washington.edu

Summary

- One of the common techniques used content Recommendation systems (like Netflix) is Matrix Factorization. Matrix in such is cases can be assumed to be of low rank.
- It has been shown that we can achieve better results if we relax the low rank assumption to that of a matrix that is 'locally' of low rank. Also, local low rank allows parallelizability.
- Usual techniques ignore contextual information that is available. We explore ways to incorporate contextual information into Matrix Factorization and also use the local low rank assumption to keep our solution theoretically parallelizable.

Background

The idea of Local Low Rank factorization is simple:

- Select a data point in the Matrix being factorized as an anchor point (a,b).
- Approximate the matrix M as $\mathcal{T}'$ for a local region nearby anchor point (a,b). The locality is found by a kernel function $K_h^{(a,b)}$, which removes or gives less weightage to points not similar. Similarity measure could be as simple as a cosine similarity.

$$\hat{\mathcal{T}}(a,b) = \arg\min_{\hat{\mathcal{T}}} \sum_{(a',b')\in A} K_h^{(a,b)}(a',b')\left([UV^T]_{(a',b')} - M_{(a',b')}\right)^2$$

$$s.t.\quad rank(\hat{\mathcal{T}}(a,b)) = rank(UV^T) = r$$

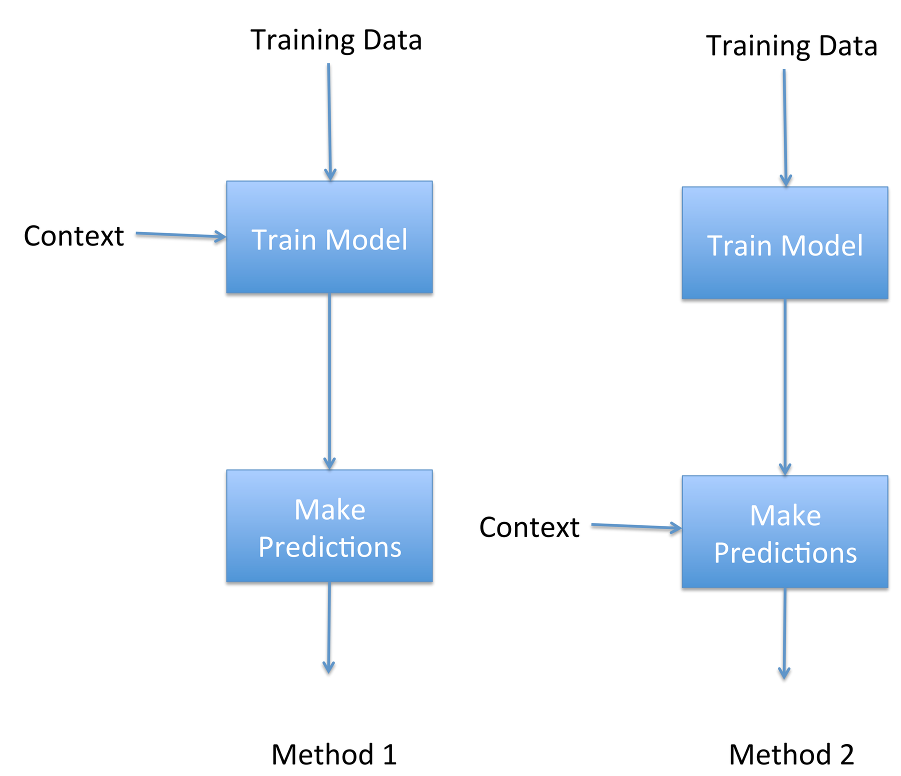
- Select Q such anchor points and give the final rating as:

$$\hat{\hat{\mathcal{T}}}(c,d) = \sum_{(a,b)\in Q} \frac{K_h^{(a,b)}(c,d)\hat{\mathcal{T}}(a,b)}{\sum_{(a',b')\in Q} K_h^{(a',b')}(c,d)}$$

- Computation of $\mathcal{T}'$ for Q local regions can be done in parallel.

Incorporating Context in predictions

- Incorporating context into predictions has been researched separately, but most matrix factorization techniques do not make use of context.
- Ideally, increasing the rank r for incomplete SVD as shown above takes care of missing context as the latent features increase with r. But increasing the rank increases computational complexity and this becomes increasingly difficult. Using context information, we should be able achieve similar results at lower ranks.
- Context information like genre (for movies/music), age/gender/location of user can be either used to train the model or appropriately transform the train model into a similarity based weighted model, refer figure below.

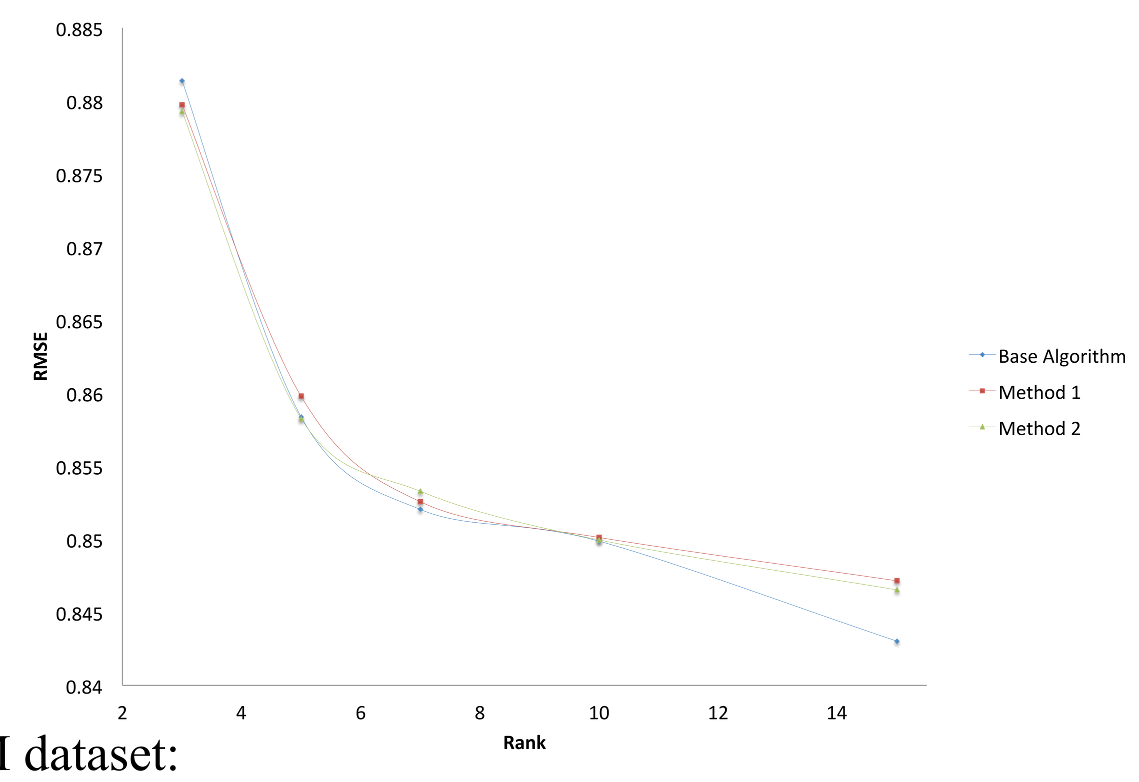


- We explore both these methods by using the genre information available for movies (and gender/age/location information for users).
  - For method one, we change the kernel K_h before training the model to use genre based similarity measure instead of (or along with) the cosine similarity of the rows of V^T .
  - For method two, we train the model as is, but at the time computing a weighted prediction, we change the kernel K_h model to use genre based similarity measure.

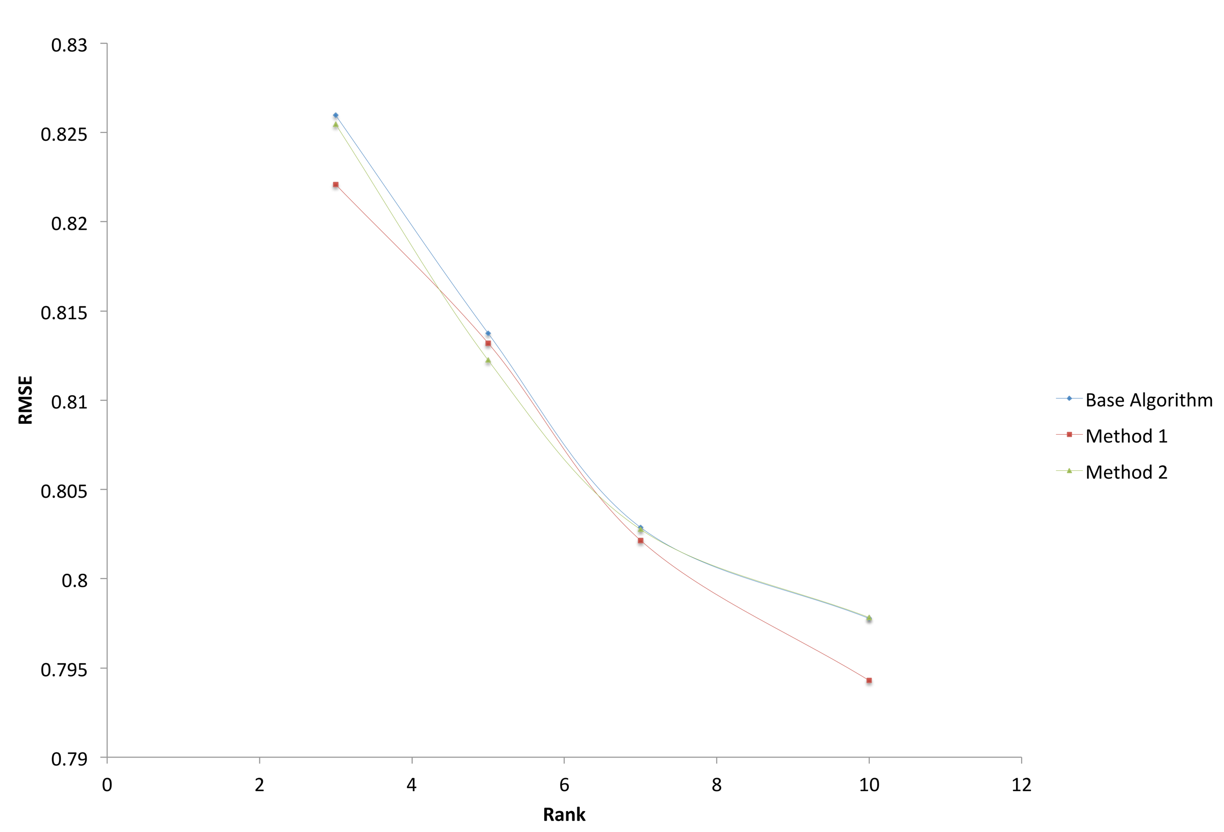
Experiments

We tried multiple experiments with various context available, user::age/gender/location and movies::Genre. User context is not available for MovieLens – 10M dataset, so for comparison we present the results using movies::Genre as additional context information for our experiments.

MovieLens-1M dataset:



MovieLens-10M dataset:



Conclusions:

From our experiments we note that:

- For locally low rank matrix factorization, both methods work better than the base algorithm until a certain rank.
- Context information as used, doesnt help much high ranks. But, for very large datasets (Netflix, MovieLens 10M), when it might be costlier to use higher rank computations, we can use context to make computations faster at a small cost to RMSE.

Challenges

- Not all datasets provide context information (eg netflix), so using context is either not possible or requires merging data sets from multiple sources.
- For smaller datasets, Local Low Rank Matrix Factorization algorithm works better without context information as we can increase r to much higher values without incurring the computational costs.
- Choosing anchor points at random can cause RMSE to go to higher values if we are unlucky and choose an outlier. To avoid this, we had to run each experiment multiple time and use an average RMSE value.

References

[1] Joonseok Lee et al., Local Low-Rank Matrix Approximation. Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.
[2] Yehuda Koren et al., Matrix Factorization Techniques for Recommender Systems. Cover Feature, IEEE Computing Society, 2009.
[3] Adomavicius, G., et al., Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach.
[4] Adomavicius, G., et al., Context-Aware Recommender Systems.