Bayesian Hierarchical Modeling of Recommendation System

Shailesh Vedula

13 December, 2016

1 Motivation

Recommendation systems are a vital component of modern web. They help users find the new content based on their search or consumption history. They are important because they help users navigate through the plethora of information present on the Web. They are also an important part of modern day entertainment streaming websites such as Netflix, Spotify etc. Popular e-commerce portals such as Amazon also employ this feature. A recommendation system recommends content to the user based on their past behaviour. The goal is to recommend items which are in close relationship to the ones already consumed.

The problem is interesting in that the number of users and the number of items are very large, most often of the order of tens of millions, yet every user only consumes a small subset of total number of items, with no user consuming all the items. The resulting dataset is extremely sparse in nature and thus making reasonably accurate predictions for the unconsumed items for a particular user then becomes challenging. With this motivation, in this project I have attempted to to build a recommendation system which can recommend new movies to a user based on the ones he has already watched.

2 Problem Statement

Given a dataset of N users and a total of M movies where each user has only watched and rated a small subset of N movies, predict the ratings for the movies which have not been watched by the user and based on this recommend top movies to him. Figure 1 represents how the data looks like. It is a MxN matrix where the rows

	1	2	..	4000
1	X	X	X	X
2	X	X	X	X
3	X	X	X	X

Figure 1: Visualization of dataset

represent the number of users and columns represent the number of movies. The blue points represent the movies which have been rated by the users. The objective then is to predict the red data points given blue. The dataset that I have chosen is from the Movielens website. For this dataset $M = 6040$ and $N = 3952$.

3 Probabilistic Model

3.1 Sampling Model

The concept that is important here is that each user has a set number of preferences and each movie has a set number of attributes.

Let R_{ij} be the rating given by user i to movie j.
For each user i let $\mathbf{U_i}$ be a K dimensional vector of user preferences.
For each user j let $\mathbf{V_j}$ be a K dimensional vector of movie attributes.

$$p(R|U,V,\alpha) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[N\left(R_{ij}|U_i^T V_j, \alpha^{-1}\right)\right]^{I_{ij}}$$
$$p(U|\mu_U,\Lambda_U) = \prod_{i=1}^{N} N\left(U_i|\mu_U,\Lambda_U^{-1}\right)$$
$$p(V|\mu_V,\Lambda_V) = \prod_{j=1}^{M} N\left(V_j|\mu_V,\Lambda_V^{-1}\right)$$

We place a Gaussian-Wishart priors on the user and movie hyper parameters. Let $\Theta_U = \mu_U, \Lambda_U$ and $\Theta_V = \mu_V, \Lambda_V$ be the set of user and movie hyper parameters. Then we have

$$\begin{aligned} p(\Theta_U|\Theta_0) &= p(\mu_U|\Lambda_U)p(\Lambda_U) \\ &= N(\mu_U|\mu_0,\beta_0\Lambda_U^{-1})W(\Lambda_U|W_0,\nu_0) \end{aligned}$$

$$\begin{aligned} p(\Theta_V|\Theta_0) &= p(\mu_V|\Lambda_V)p(\Lambda_V) \\ &= N(\mu_V|\mu_0,\beta_0\Lambda_V^{-1})W(\Lambda_V|W_0,\nu_0) \end{aligned}$$

We set $\mu_0 = 0, \nu_0 = D, W_0 = I, \beta_0 = 2$.

3.2 Graphical Model

Figure 2 represents the complete graphical model of the system

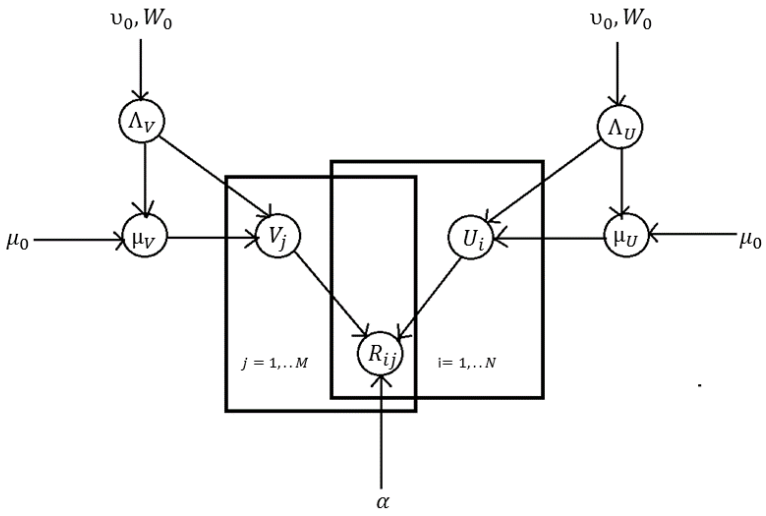

Figure 2: Graphical model

3.3 Posterior inference and full conditionals

After we have set up the sampling model, in order to perform inference we need the full conditionals of U and V. If we have this we can do predictions using Gibbs sampling. The full conditionals can be derived as follows.

$$p(U|R, V, \Theta_U) \propto \prod_{i=1}^{N} p(U_i|R, V, \Theta_U) \tag{1}$$

$$\begin{aligned} p(U_i|R, V, \Theta_U) &\propto p(R|U_i, V, \Theta_U)p(U_i|\Lambda_U) \\ &\propto \prod_{j=1}^{M} [N(R_{ij}|U_i^T V_j, \alpha^{-1})]^{I_{ij}} p(U_i|\mu_U, \Lambda_U) \\ &\sim N(U_i|\mu_i^*, [\Lambda_i^*]^{-1}) \\ \Lambda_i^* &= \Lambda_U + \alpha \Sigma_{j=1}^{M} [V_j V_j^T]^{I_{ij}} \\ \mu_i^* &= [\Lambda_i^*]^{-1} \left(\alpha \Sigma_{j=1}^{M} [V_j V_j^T]^{I_{ij}} +_U \mu_U\right) \end{aligned}$$

$$p(\mu_U, \Lambda_U|U, \Theta_0) \propto N(\mu_U|\mu_0^*, (\beta_0^* \Lambda_U)^{-1}) W(\Lambda_U|W_0^*, \nu_0^*) \tag{2}$$

$$
\begin{aligned}
\mu_0^* &= \frac{\beta_0\mu_0 + N\bar{U}}{\beta_0 + N} \\
\beta_0^* &= \beta_0 + N \\
\nu_0^* &= \nu_0 + N \\
[W_0^*]^{-1} &= [W_0]^{-1} + N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}(\mu_0 - \bar{U})(\mu_0 - \bar{U}^T) \\
\bar{U} &= \frac{1}{N}\Sigma_{i=1}^{N} \\
\bar{S} &= \frac{1}{N}\Sigma_{i=1}^{N} U_i U_i^T
\end{aligned}
$$

The full conditionals for V i.e. $p(V|R, U, \Theta_V)$ and $p(V_j|R, U, \Theta_V)$ will be the same as above because of the symmetricity of the model. The only changes will be $U \to V, V \to U, i \to j$ and $N \to M$. Therefore we have

$$p(V|R, U, \Theta_V) \propto \prod_{j=1}^{M} p(V_j|R, U, \Theta_V) \tag{3}$$

$$p(V_j|R, U, \Theta_V) \propto p(R|V_j, U, \Theta_V)p(V_j|\Lambda_V) \tag{4}$$

4 Training and Prediction

4.1 Gibbs Sampling

Once we have the full conditionals from above we do a Gibbs sampling to fit the the user and movie hyper parameters to the training data. In each iteration we check the $rmse$ on the validation dataset and stop the algorithm once the $rmse < \epsilon$. The algorithm is mentioned below.

1. Initialize the model parameters U^1, V^1
2. while $\Delta rmse >= \epsilon$
   - sample the hyper parameters Θ_U^t from Equation 2 and Θ_V^t from Equation 4.
   - For each $i = 1 \ldots, N$ sample user features $U_i^{t+1} \sim p(U_i|R, V, \Theta_U)$
   - For each $j = 1 \ldots, M$ sample $V_j^{t+1} \sim p(V_j|R, U, \Theta_v)$
   - for each (i, j) pair in the validation set sample $R_{ij} \sim p(R_{ij}|U_i^{t+1}, V_j^{t+1}, \alpha)$
   - $\Delta rmse = \sqrt{\frac{\Sigma_{i,j}(\text{predicted-true})^2_{\text{validation set}}}{\text{length(validation set)}}}$

In the above algorithm a clever way to initialize U^1, V^1 is to set it to the solution found by maximum likelihood estimation. Doing so results in fast convergence of Gibbs sampler. The burn in period is 800.

4.2 Prediction

Once the Gibbs sampler converges, we will have the final values of U, V. Then the ratings of individual (N_i, M_j) pair can be calculated by sampling from $p(R_{ij}|U_i, V_j, \alpha)$ a large number of time and then taking the average of all the values. Therefore a prediction for unseen (i, j) pair

$$R^*_{ij} = \frac{1}{K}\Sigma_{k=1}^{K} p(R_{ij}|U_i, V_j, \alpha)$$

5 Results

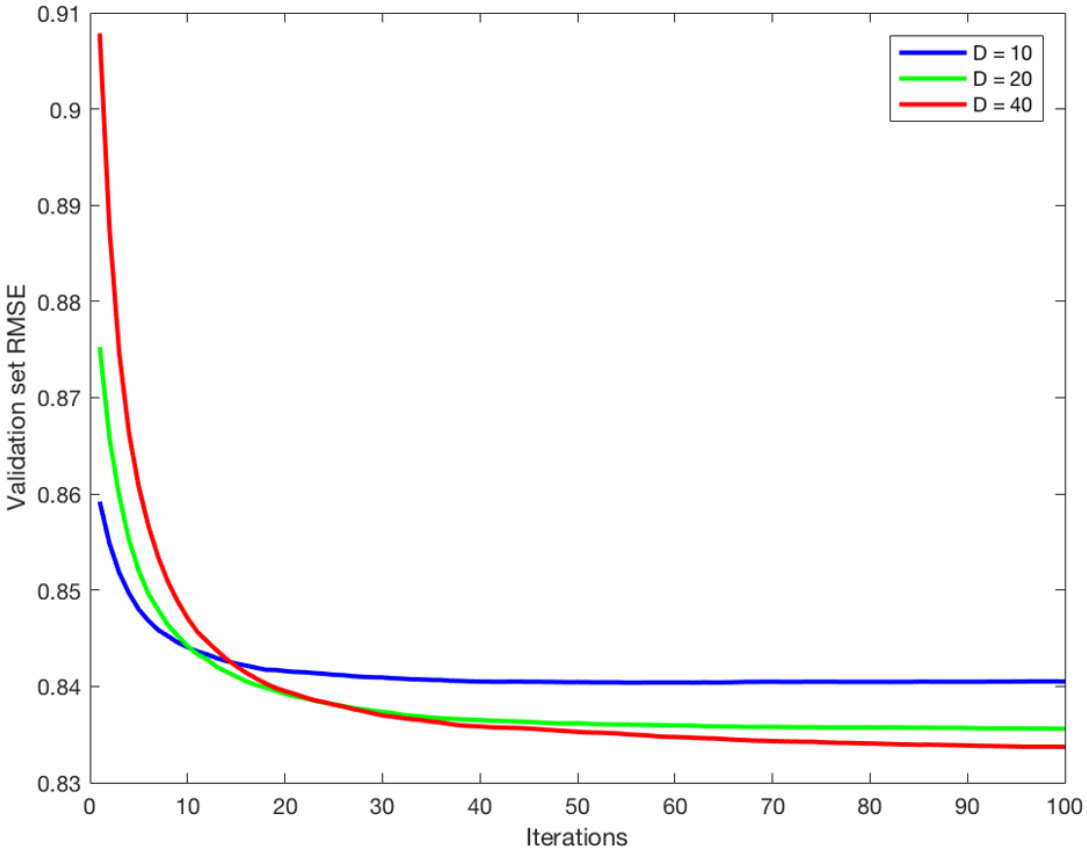

Figure 3: RMSE vs Number of Gibbs iterations

From Figure 3 we see that the Gibbs sampler converges really fast. This was expected since we initialized it to the MLE solution. Further we observed that

Bayesian approach yields less RMSE when compared to the MLE solution. In general Bayesian solution does not suffer from over fitting and therefore complex models can be estimated with high degree of accuracy. On the other hand the MLE approach suffers from over fitting and also doing a grid search to find the optimum value of hyper parameters is very costly and in model like ours is almost infeasible. Also we see that with increase in D, RMSE decreases. This is expected because large D is able to capture the user preferences and movie attributes more accurately. In literature models with $D = 300$ have been shown to give a very accurate solutions.

User 1

Watched	Recommended
Toy Story	Whatever it takes
Pocahontas	Hand that rocks the cradle
Apollo 13	The Hurricane
Star Wars-Episode IV	Fatal Attraction
Schindler's List	Few good men
Secret Garden	Without Limits
Snow White and the 7 dwarfs	Firelight
Fargo	Titanic
James and the giant peach	Some Mother's son
Wallace and Gromit	Jaws
A close shave	2 or 3 things I know about her
Hunchback of Notre Dame	Top Gun
My fair lady	Swiss Family Robinson

User 12

Watched	Recommended
Silence of the lambs	Laurel and Hardy
Larger than life	Hungarian Fairy tale
The Godfather	My sweet murder
Wizard of Oz	Window to Paris
Citizen Kane	Airplane
Father of the bride	The Exorcist
2 days in the valley	Face in the crowd
One flew over the cuckoos nest	Naked gun
Raiders of the lost arc	Young Frankenstein
GodFather Part2	Bad Lieutenant
Das Boot	Aparajito
The graduate	Once upon a time in the west
Chinatown	Prince of central Park

User 157

Watched	Recommended
Toy Story	Lamerica
Golden Eye	Some mother's son
Mallrats	Salut cousin
Clerks	I am Cuba
Kids in the hall	Some folks call it sling blade
Crow: city of angels	Godfather part 2
Monty Python	6 ways to Sunday
The Princesses Brid	2 or 3 things I know about her
Stand by me	schizopolis
Cool Hand Luke	The Godfather
Young Frankenstein	Chambermaid on titanic
Grasse Point blank	Just the ticket
Titanic	Mille bole blu
Half Baked	Cotton Mary

Figure 4: Illustration of recommendations made by the system

Figure **??** illustrates the working of our system.

6 Observations

- The method presented above is not scalable to very large datasets like the Yahoo music dataset or the Netflix dataset.
- For very large datasets, Bayesian variational inference algorithms are used.
- On the Netflix dataset Naive MCMC methods have been reported to take 16 days for convergence.
- On Netflix dataset with D=300, variational inference models take 13 hours to converge.

- To get discrete ratings so as to evaluate the accuracy softmax function can be used.

7 Conclusion

A fully Bayesian treatment for modeling recommendation systems was presented. A normal sampling model with Gausian-Wishart priors for hyper parameters is presented. The predictions were performed using Gibbs sampling which gives good accuracy. A brief comparison of Bayesian and Maximum Likelihood approaches is also presented.

8 References

1. Gopalan, Prem, et al. "Bayesian Nonparametric Poisson Factorization for Recommendation Systems." AISTATS. 2014.

2. Gopalan, Prem, Jake M. Hofman, and David M. Blei. "Scalable recommendation with poisson factorization." arXiv preprint arXiv:1311.1704(2013).

3. Salakhutdinov, Ruslan, and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo." Proceedings of the 25th international conference on Machine learning. ACM, 2008.

4. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." Machine learning 37.2 (1999): 183-233.

5. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

6. F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872