Movie Rating Recommender System using PMRF

Dhruvkumar Patel
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
Email: dhruvkumar.patel@utdallas.edu

Bikramjit Mandal
Department of Computer Science
University of Texas at Dallas
Richardson, Texas
Email: bikramjit.mandal@utdallas.edu

***Abstract*— Recommendation algorithms are used by systems that provide users with speedy customized experience. Users give ratings to movies based on the extent they like them on websites like IMDB, MovieLens, Rotten Tomatoes. In this paper, we try to predict ratings for movies user has not given yet, based on the ratings user has given to other movies in past. We want to discuss this movie recommendation problem as an inference problem that can be represented using Pairwise Markov Random Fields (PMRF). Applying PMRF to movie ratings and calculating marginal probabilities for unseen movie ratings yields exponential time complexity. For this reason, we have utilized Belief Propagation algorithm, which only increases time complexity linearly with number of items. This approach is basically a collaborative filtering approach where the items are represented as nodes and the similarities among the users are predicted in their rating behavior based on their past ratings. We can use this approach to only provide real-time update for a single active user without affecting the others, keeping complexity linear for a single user. The prediction of the ratings is not linear, but is exponential in complexity with increasing variables, so Belief Propagation and inference via message passing are preferred among the nodes of the model. We apply a deterministic method to calculate the similarities, as well as show a learning approach for more accuracy in similarities of the active user with others. We also evaluate our results via various error comparing schemes.**

Keywords—MRF; Recommender System; PMRF; Belief Propagation.

I. Introduction

The surge of information in this age of technology has overwhelmed humankind with its variations, patterns and frequency. Information has become key to the businesses which rely on user response and interactivity. Quick and easy access to choices and data is the key for all. Fast data filteration techniques are required for filtering irrevelant entities and selecting the most important ones. The biggest challenge is to perform these tasks in large scale data overflows in large scale systems in real time.

The recommender systems can be based on Collaborative Filtering, Content based filtering and the newest are the Hybrid Filtering. Collaborative filtering methods mostly rely on predicting the unseen values based on the past data or behavior of the entities or users. Content based filtering constitutes methods to identify user's preference and the user's interaction with the recommender system. It uses description of the user's likes and dislikes in their profiles to predict the recommendations. Hybrid Filtering are new techniques combining the former two approaches into one. Netflix is one of the best examples of this kind of a system.

In our project we have used a Collaborative Filtering (CF) technique where in, for each active user, we consider their past ratings on items to predict the unseen item ratings. We have used a memory based approach by using Neighbourhood method to find similarities with other users of similar tastes and recommend ratings. The other approach commonly also used is the model based approach, where we need to learn model from history ratings and predict the future ratings based on these.

The similarity between the entities in this project is a key component of the Neighborhood approach. Similarities between the users and items have been calculated using a relation that takes into account the average of past ratings. The similarities can be calculated also, using a learning algorithm, that uses factor graphs to relate the similarities among users' ratings and loopy BP over the graphs to infer more accurate similarity. We study the algorithm and it's benefits also. In coming sections we unfold the entire mechanism of a fast recommendation algorithm feasible for application in modern recommendation systems.

II. Formulation of Recommendation System on PMRF

We have used PMRF- Pairwise Markov Random Field to formulate movie ratings inference system on a graph. We assume two different sets in the system: i) U is a set of all users in and ii) I is a set of all items, which in this case are movies. Y is a set of all possible rating values. In our case it was $\{1,2,3,4,5\}$. Users provide opinions or ratings about the items. Each user u in U has provided some rating on some items from I, and may not have provided ratings for other items in I. The goal is to provide accurate recommended ratings for each user for items user has not recommended before. In this setting, we take one user, say z, at a time and infer ratings for the user z on unseen items. Suppose there are $|U|=M$ users and $|I|=N$ items in the system. Let's say I_z is the set of items user z has already rated. Thus $R_z = \{ r_{zi} : i \in I \setminus I_z \}$ be the set of variables which needs to be predicted for the active user z. Here $I \setminus I_z$ is the set of items user z has not rated.

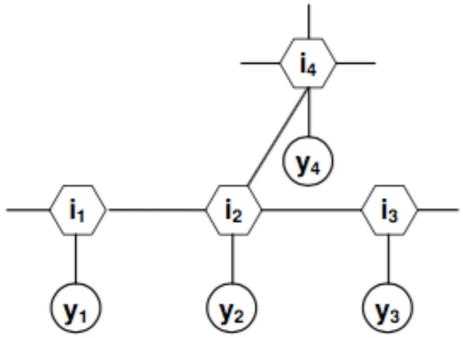

Fig. 1: Representation of item ratings on PMRF.

Figure above is taken from [1].

For active user z, we formulate the recommender system as given in following figure(1), where each unknown item in $I \setminus I_z$ has item node in the graph. Every item node i in graph is also connected with local evidence node y_i in the graph. This local evidence node y_i value is computed by following formula.

$$y_i = \bar{r}_z + \frac{\sum_{v \in U_i} s_{zv}(r_{vi} - r_v)}{\sum_{v \in U_i} |s_{zv}|} \qquad (1)$$

In the equation above, $\bar{r}_z$ is average rating on all items by user z. U_i is a set of all users who have rated item i. s_{zv} is similarity between users z and v. r_{vi} is rating given by user v to item I, and r_v is average rating of user v. To compute similarity s_{zv} between two users, we user Pearson Correlation Coefficient (PCC) which is given by following formula.

$$s_{uv} = \frac{\sum_{i \in I_{uv}}(r_{ui} - \bar{r_u})(r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in I_{uv}}(r_{ui} - \bar{r_u})^2} \sqrt{\sum_{i \in I_{uv}}(r_{vi} - \bar{r_v})^2}} \qquad (2)$$

Also, we connect two items i and j in the PMRF graph only if following conditions are met:

- The similarity s_{ij} between items i and j is above some threshold s_{th}.
- At least one of i and j is among the top K similar items of the other.

To compute similarity s_{ij} between item i and j, we use similar PCC equation as given above.

$$s_{ij} = \frac{\sum_{v \in U_{ij}}(r_{vi} - \bar{r_v})(r_{vj} - \bar{r_v})}{\sqrt{\sum_{v \in U_{ij}}(r_{vi} - \bar{r_v})^2} \sqrt{\sum_{v \in U_{ij}}\left(r_{vj} - \bar{r_v}\right)^2}} \qquad (3)$$

We define initial evidence function $\emptyset_i(r_{zi})$ which captures the dependence between the local evidence y_i and item node i. $\emptyset_i(r_{zi})$ is probability function over possible rating values from Y.

- If $1 \le y_i \le 5$ then,
  $\lfloor y_i \rfloor \emptyset_i(r_{zi} = \lfloor y_i \rfloor) + \lceil y_i \rceil \emptyset_i(r_{zi} = \lceil y_i \rceil) = y_i$
- If $y_i < 1, \emptyset_i(r_{zi} = 1) = 1.$
- If $y_i > 5, \emptyset_i(r_{zi} = 5) = 1.$

We define compatibility function from item node i_a to i_b as follows

$$\varphi_{ab}(r_{zi_a}, r_{zi_b}) = \frac{1}{Z_{ab}} \exp\left\{-\frac{\left(r_{zi_a} - r_{zi_b}\right)^2}{\sigma_{ab}^2}\right\} \qquad (5)$$

where Z_{ab} is a normalization constant. σ_{ab} is adjusted according to item similarity computed in equation (3) according to following formula.

$$\sigma_{ab} = \frac{1}{\sqrt{2}}\left(\frac{1 - s_{i_a i_b}}{1 - s_{th}} + 1\right) \qquad (6)$$

where s_{th} is similarity threshold as described above.

The probability joint distribution over all unseen item ratings $r_{zi} : i \in I \setminus I_z$ for active user z, is given by following genral formula of Markov Random Field:

$$P(\{\text{rzi} :\ \text{i} \in \text{I} \setminus \text{Iz}\}) = \frac{1}{Z} \prod_i \emptyset_i(r_{zi}) \prod_{i_a i_b} \varphi_{ab}\left(r_{zi_a}, r_{zi_b}\right) \quad (7)$$

where Z is a normalization constant.

III. Belief Propagation

To infer ratings for all unseen items in figure (1), we compute marginal probabilities for all possible ratings for unrated items for active user z. To compute marginal probabilities, we use Belief Propagation algorithm with following standard message form:

$$m_{ab}^n(r_{zi_b}) = \frac{1}{Z_{ab}} \sum_{r_{zi_a}} \varphi_{ab}(r_{zi_a}, r_{zi_b}) \emptyset_{i_a}(r_{zi_a}) \prod_{c \in N_a \setminus b} m_{ca}^{n-1}(r_{zi_a}) \qquad (8)$$

A. Initialization

We initialize messages value to $\frac{1}{|Y|}$.

B. Iterations

- At each iteration n, we pass messages from every tpair of nodes a to b using equation 8.
- We repeat this process until values of messages converge.

C. Marginals calculation

After we have converged value of messages, we calculate marginal probabilities for each possible ratings for each node as follows:

$$P(r_{zi}) = \frac{1}{Z_i} \emptyset_i(r_{zi}) \prod_{j \in N_i} m_{j,i}(r_{zi}) \qquad (9)$$

Once we have calculated values of marginals, we can compute expected value of rating using following formula:

$$\bar{r}_{zi} = \sum_{r \in \Upsilon} rP(r_{zi} = r) \tag{10}$$

If there are no loops then beliefs are exact. Otherwise if they do not converge, we break the iterations after maximum 100 iterations.

IV. ANOTHER APPROACH FOR SIMILARITY COMPUTATION

We have discussed so far the use of *Pearson Correlation Coefficient (PCC)* for calculating the similarity for a active user, which is a deterministic method. PCC though is easier to compute and use, does not provide accuracy in large-scale and complex modern systems.

A better approach would be to consider using a Neighborhood method approach, based on Belief Propagation, to predict more accurate similarities for each user. In this approach we try to improve the accuracy for Neighborhood method and with a complexity similar to PCC.

$$P(S_u|R)$$

We construct a bipartite factor graph to represent the factorization structure for the function. We derive the factorization for $P(S_u|R)$ as

$$P(S_u|R) = \frac{1}{Z}\prod_{i \in I_u} f_i(S_{ui})$$

where Z is a normalization constant.

We assume that the set of items is I_u and for each item in I_u we can deduce a set of similarities among the users who have rated the item ***i***.

$$S_{ui} = \{s_{uv}: v \in U \backslash u\}$$

That is the set of all similarities of user *u* with all the users who rated the item *i*. Here, we see that the similarities of the given user with the other users depend on the ratings of the other users on the same item. The dependencies among the similarities of

For constructing the bipartite factor graph we assume all the factors to be the function $f_i(S_{ui})$ and the variable nodes denote the similarities for the item *i*. The factor node function $f_i(S_{ui})$ is defined as

$$f_i(S_{ui}) = \frac{1}{Z_i} exp\left\{-\frac{1}{\sigma^2}(\widehat{r_{ui}}(S_{ui}) - r_{ui})^2\right\}$$

Where the $\widehat{r_{ui}}(S_{ui})$ is the rating produced by

$$\widehat{r_{ui}}(S_{ui}) = \bar{r_u} + \frac{\sum_{v \in V_i} s_{uv}(r_{ui} - \bar{r_v})}{\sum_{v \in V_i}|s_{uv}|}$$

Here, Z_i is normalization constant. Sigma Square is a design parameter. The function for the factor node can be seen to be a constraint between the true rating r_{ui} and the predicted rating $\bar{r}_{ui}$. The design parameter sigma-square helps to control the sensitivity of the factor node function to the discrepancy between the true and predicted rating.

Iterative Message Passing methodology

We intend to calculate the marginal probabilities for each s_{uv} in S_u, that is the similarities for the user u. The factor graph can contain loops, so we will use the Loopy Belief Propagation algorithm to perform iterative message passing among the variable nodes and the factor nodes, until the messages converge.

We define two types of messages:

- λ message -$\lambda_{i,v}(S_{uv})$ sent from factor node i to variable node v
- μ message- $\mu_{v,i}(S_{uv})$ sent from variable node v to factor node i.

The λ messages and μ messages are updated iteratively and passed along the edges of the factor graph. The λ message at the n-th iteration will be as follows

$$\lambda_{i,v}^{n}(s_{u,v}) \propto \sum_{S_{ui} \backslash s_{uv}} f_i(S_{ui}) \prod_{w \epsilon V_i \backslash v} \mu_{w,i}^{(n-1)}(s_{uw})$$

$\lambda_{i,v}^{n}(s_{u,v})$ tells how similar user ***v*** is to the user ***u*** in the item ***I's*** point of view. The algorithm computes and updates all the λ messages on the graph and then for the current iteration, we start calculating the μ messages using the ***λ*** messages calculated. This done via this relation

$$\mu_{v,i}^{(n)}(s_{uv}) \propto \prod_{j \in F_v \backslash i} \lambda_{j,v}^{(n)}(s_{uv})$$

F_v Denotes the set of factor nodes that are connected to variable node v. $\mu_{v,i}^{(n)}(s_{uv})$ Tells the similarity of u's rating to the user v's rating by telling the probabilities for the $s_{uv} = s, \forall s \in S$ where S is a definite set of values for the similarities to start with. (*In our project we have assumes S={1,2}.)*

For each iteration, we get the values for all the λ messages and μ messages for all the factor nodes.

We check the convergence of the algorithm after each iteration by calculating the marginal distribution.

$$P(s_{uv}|R) = \frac{1}{Z_v}\prod_{i \in F_v} \lambda_{i,v}^{(n)}(s_{uv})$$

Z_v is a normalization constant. The algorithm converges the $P(s_{uv}|R)$ and we can calculate the s_{uv} as

$$s_{uv} = \sum_{s \in S} sP(s_{uv} = s|R)$$

V. Implementation and Evaluation

We implemented the algorithm described above on the dataset of 100k movie reviews from MovieLens collected by grouplens.org. It has 100,000 reviews from 943 users for 1682 movies. We took 20,000 reviews in evaluation set and 80,000 reviews for training set. We implemented this system on Matlab on mac with 1.4 GHz processor and 4GB RAM.

We assumed following design parameters required in some of the equations above:

- In formulation PMRF graph, we took similarity threshold s_{th}=0.35 and K=5, which is used to decide whether to create edge between items.

To evaluate our predictions, we computed Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which we computed using following formulas for the same.

$$MAE = \frac{1}{N} \sum_{r_{ui} \in R_t} |r_{ui} - \bar{r}_{ui}|$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{r_{ui} \in R_t} (r_{ui} - \bar{r}_{ui})^2}$$

We got following results with our experiments:

- Users- 943
- Reviews- 100,000
- Train set- 80,000
- Test set-20,000
- Number of users for whom belief propagation algorithm was performed- 252
- Mean absolute error- 0.8001
- Root mean square error- 1.0345

Another set of results with lesser number of users and reviews is following:

- Users- 200
- Reviews- 28000
- Train set- 22,000
- Test set-6000
- Mean absolute error- 0.7969
- Root mean square error- 1.0327

VI. Conclusion

In this project, we implemented recommender system using probabilistic graphical models, particularly PMRF. PMRF can be used to accurately predict the movie ratings and it can be done in less time complexity if Belief Propagation is used. In Belief Propagation, time complexity only increases linearly with number of items in the graph. We used PCC to compute similarity of users and items, but same Belief Propagation based Factor Graph method can be also used to learn these similarities, which is expected to yield even better results. In conclusion, Belief Propagation based recommender system can be fast and can be incorporated in real-time systems for better recommendations.

Acknowledgment

We have borrowed whole approach to use PMRF for recommendation system and equations for $\emptyset_i(r_{zi})$ and $\varphi_{ab}(r_{zi_a}, r_{zi_b})$, as well as values of design parameters from [1]. Similarity computation using Factor graphs is borrowed from [2], which we implemented, but time complexity was too high to get results. So results given are obtained using PCC for learning, and PMRF for inference.

References

[1] J. Zou, A. Einolghozati, E. Ayday, and F. Fekri, "A Recommender System Based on Belief Propagation over Pairwise Markov Random Fields ".

[2] J. Zou, A. Einolghozati, E. Ayday, and F. Fekri, "Iterative Similarity inference via factor graphs for colaborative filtering".

[3] http://grouplens.org/datasets/movielens/