Notes on Cluster Model

Chengliang Tang

April 4, 2018

This note is a detailed explanation for the cluster model in Section 2.3.1 of Paper 1. And we will use the EachMovie data as an example for implementing the algorithm.

1 Notations

Suppose we have N users, and M movies in the data set. For each user $1 \leq i \leq N$, let $I(i)$ be the set of movies that user i has already scored in the training set. For $\forall j \in I(i)$, we denote the score that user i gave to movie j by $v_j^{(i)}$, where $v_j^{(i)} \in \{0, 1, ..., 5\}$. In order to discriminate, we use $v_j^{(i)}$ for the data, and $V_j^{(i)}$ for the random variable.

Also, in the cluster model, we assume all the users can be categorized into one of C different classes. And for each user i, denote his class by Δ_i.

2 Score Estimation

The goal of collaborative filtering is to estimate $\mathbb{E}[V_b^{(i)}|v_j^{(i)}, j \in I(i)]$ for each i and $b \notin I(i)$. Thus, we can have the following equation

$$\mathbb{E}[V_b^{(i)}|v_j^{(i)}, j \in I(i)] = \sum_{k=1}^{5} k \cdot P\big(V_b^{(i)} = k|v_j^{(i)}, j \in I(i)\big), \tag{1}$$

and for each term in RHS,

$$\begin{aligned}
&P\big(V_b^{(i)} = k|v_j^{(i)}, j \in I(i)\big) \\
=&\frac{P\big(V_b^{(i)} = k; V_j^{(i)} = v_j^{(i)}, j \in I(i)\big)}{P\big(V_j^{(i)} = v_j^{(i)}, j \in I(i)\big)} \\
=&\frac{\sum_{c=1}^{C} P\big(V_b^{(i)} = k; V_j^{(i)} = v_j^{(i)}, j \in I(i); \Delta_i = c\big)}{\sum_{c=1}^{C} P\big(V_j^{(i)} = v_j^{(i)}, j \in I(i); \Delta_i = c\big)} \\
=&\frac{\sum_{c=1}^{C} P(\Delta_i = c) \cdot P(V_b^{(i)} = k|\Delta_i = c) \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)}|\Delta_i = c)}{\sum_{c=1}^{C} P(\Delta_i = c) \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)}|\Delta_i = c)},
\end{aligned} \tag{2}$$

where the last equation is due to the standard Naive Bayes formulation (see in paper 2).

3 Log-likelihood Function

As we can see from above, in order to estimate $V_b^{(i)}$, we need to know the following parameters:

$$\begin{aligned} P(\Delta_i = c), &\quad \text{for} \quad c = 1, ..., C; \\ P(V_j^{(i)} = k|\Delta_i = c), &\quad \forall j \in \{b\} \cup I(i), \forall k \in \{0, ..., 5\}. \end{aligned} \tag{3}$$

Also, for simplicity, we need to assume the users in the same class will have the same conditional distribution of scores. This means for any pair of user i_1 and user i_2, we have

$$\begin{aligned} P(\Delta_{i_1} = c) = P(\Delta_{i_2} = c), &\quad \text{for} \quad c = 1, ..., C; \\ P(V_j^{(i_1)} = k|\Delta_{i_1} = c) = P(V_j^{(i_2)} = k|\Delta_{i_2} = c), &\quad \text{for} \quad \forall c, j, k. \end{aligned} \tag{4}$$

So that in the model, the number of parameters is about $(C + 5CM)$. And we can simplify our notations in the following way:

$$\begin{aligned} \mu_c := P(\Delta_i = c), &\quad \text{for} \quad c = 1, ..., C; \\ \gamma_{c,j}^{(k)} := P(V_j^{(i)} = k|\Delta_i = c), &\quad \text{for} \quad \forall c, j, k. \end{aligned} \tag{5}$$

where we know $\sum_{c=1}^{C} \mu_c = 1, \sum_{k=0}^{5} \gamma_{c,j}^{(k)} = 1$.

In this section, we estimate these parameters by maximum likelihood estimation. For user i, his log-likelihood function can be written as

$$\begin{aligned} l_i(\mu, \gamma|data) &= \log \big[\sum_{c=1}^{C} P(\Delta_i = c) \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)}|\Delta_i = c) \big] \\ &= \log \big[\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)}|\Delta_i = c) \big] \end{aligned} \tag{6}$$

so that the log-likelihood function for all the training data is

$$\begin{aligned} l(\mu, \gamma|data) &= \sum_{i=1}^{N} l_i(\mu, \gamma|data) \\ &= \sum_{i=1}^{N} \log \big[\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} P(V_j^{(i)} = v_j^{(i)}|\Delta_i = c) \big], \end{aligned} \tag{7}$$

which can maximized by the EM algorithm.

4 EM Algorithm

The key of EM algorithm used in this model is to consider Δ_i as unobserved data, and then take (expectation + maximization) iteratively.

For user i, denote his class by δ_i, which is unobserved. Similar with the notation of $V_j^{(i)}$, here δ_i means the data, and Δ_i means the random variable.

As a result, our updated log-likelihood function with "new" observed variable δ_i can be written as

$$\begin{aligned} l(\mu,\gamma|\tilde{data}) &= \sum_{i=1}^{N} \log\left[P(\Delta_i=\delta_i)\cdot \prod_{j\in I(i)} P(V_j^{(i)}=v_j^{(i)}|\Delta_i=\delta_i)\right] \\ &= \sum_{i=1}^{N} \log[P(\Delta_i=\delta_i)] + \sum_{i=1}^{N}\sum_{j\in I(i)} \log[P(V_j^{(i)}=v_j^{(i)}|\Delta_i=\delta_i)] \end{aligned} \tag{8}$$

Then, in the EM algorithm:

- *Step 1:* Take initial guess for all the parameters $\hat{\mu},\hat{\gamma}$.

  To avoid local optima, don't use uniform initial values.

- *Step 2:* Expectation.

  Compute the responsibilities for each user i

  $$\hat{\pi_i^c} = \frac{\hat{\mu_c}\cdot\hat{\phi}_c(D(i))}{\sum_{c=1}^{C}\hat{\mu_c}\cdot\hat{\phi}_c(D(i))} \tag{9}$$

  for $c=1,...,C$ and $i=1,...,N$.

  In the above equation, $\hat{\phi}_c(D(i)) = \prod_{j\in I(i)}\hat{P}(V_j^{(i)}=v_j^{(i)}|\Delta_i=c)$, where $\hat{P}(V_j^{(i)}=k|\Delta_i=c)=\hat{\gamma}_{c,j}^{(k)}$.

- *Step 3:* Maximization.

  Update the parameters

  $$\begin{aligned} \hat{\mu}_c &= \frac{\sum_{i=1}^{N}\hat{\pi_i^c}}{N}, \quad \text{for} \quad c=1,...,C \\ \hat{\gamma}_{c,j}^{(k)} &= \frac{\sum_{i:j\in I(i)}\hat{\pi_i^c}\cdot\mathbb{I}\big(v_j^{(i)}=k\big)}{\sum_{i:j\in I(i)}\hat{\pi_i^c}}, \quad \text{for} \quad \forall c,j,k \end{aligned} \tag{10}$$

  where $\mathbb{I}(\cdot)$ is the indicator function taking values in $\{0,1\}$.

  The idea is this step is that in order to calculate MLE in a weighted multinomial distribution, we only need to take the weighted frequency for each class.

- *Step 4:* Iteration.

  Iterate steps 2 and 3 until convergence.

After we get the converged estimators $\hat{\mu},\hat{\gamma}$, we can come back to Section 2 to make predictions for each user.