Project 3 Algorithms Notes

STAT GU4243

Applied Data Science

Cynthia Rush
Columbia University

March 21, 2018

Second Project Task

Evaluate potential performance enhancements on the baseline momeory-based algorithm (Model 2 above) by considering changes to various components of the algorithm.

- **[All Groups]** Consider different similarity weights: (1) Spearman's correlation, (2) vector similarity, (3) entropy, (4) mean-square difference, and (5) SimRank. Most are discussed in section 5.1 of paper [2]. SimRank is discussed in paper [4].
- **[Groups 1, 2, 3]** Consider significance and variance weighting. Section 5.2 and 5.3 of paper [2].
- **[Groups 4, 5]** Consider selecting neighborhoods. Section 6 of paper [2].
- **[Group 6, 7]** Consider rating normalization. Section 7 of paper [2].

All Groups

Motivation

- Determination of the similarity weights is very important to the way the algorithm makes preditions.

Explore: Will prediction accuracy improve if we consider different similarity weights than that based on Pearson correlation?

Let's look at some examples...

Baseline – Pearson correlation – what we coded today:

Let $r_{\mathrm{u,m}}$ be the rating by user 'u' for movie 'm' and $\bar{r}_{\mathrm{u}}$ be user u's average rating.

$$\hat{r}_{\mathrm{a,m}} = \bar{r}_{\mathrm{a}} + \frac{\sum_{\mathrm{u} \in \{\mathrm{users}\}} (r_{\mathrm{u,m}} - \bar{r}_{\mathrm{u}}) \times w_{\mathrm{u,a}}}{\sum_{\mathrm{u} \in \{\mathrm{users}\}} w_{\mathrm{u,a}}},$$

where $w_{\mathrm{u,a}}$ is the Pearson correlation coefficient

$$w_{\mathrm{u,a}} = \frac{\sum_{\mathrm{m} \in \mathrm{movies}} (r_{\mathrm{u,m}} - \bar{r}_{\mathrm{u}}) \times (r_{\mathrm{a,m}} - \bar{r}_{\mathrm{a}})}{\sqrt{\sum_{\mathrm{m} \in \mathrm{movies}} (r_{\mathrm{u,m}} - \bar{r}_{\mathrm{u}})^2 \sum_{\mathrm{m} \in \mathrm{movies}} (r_{\mathrm{a,m}} - \bar{r}_{\mathrm{a}})^2}}.$$

Motivation

- Pearson correlation measures a linear relationship.
- Spearman's correlation assesses monotonic relationships (whether linear or not).
- Spearman correlation is equal to the Pearson correlation between the rank values of those two variables.

Now $w_{\mathrm{u,a}}$ is calculated as

$$w_{\mathrm{u,a}} = \frac{\sum_{\mathrm{m} \in \mathrm{movies}} (\mathrm{rank}_{\mathrm{u,m}} - \bar{\mathrm{rank}}_{\mathrm{u}}) \times (\mathrm{rank}_{\mathrm{a,m}} - \bar{\mathrm{rank}}_{\mathrm{a}})}{\sqrt{\sum_{\mathrm{m} \in \mathrm{movies}} (\mathrm{rank}_{\mathrm{u,m}} - \bar{\mathrm{rank}}_{\mathrm{u}})^2 \sum_{\mathrm{m} \in \mathrm{movies}} (\mathrm{rank}_{\mathrm{a,m}} - \bar{\mathrm{rank}}_{\mathrm{a}})^2}}.$$

Cosine Similarity

Motivation

- In information retrieval, often compute similarity between two documents by considering vectors of word frequencies and computing cosine f the angle between the two vectors.
- Now users take the role of documents, items are words, and votes are frequencies.

Now $w_{\mathrm{u,a}}$ is calculated as

$$w_{\mathrm{u,a}} = \frac{\sum_{\mathrm{m} \in \mathrm{movies}} r_{\mathrm{u,m}} \times r_{\mathrm{a,m}}}{\sqrt{\sum_{\mathrm{m} \in \mathrm{movies}} (r_{\mathrm{u,m}})^2 \sum_{\mathrm{m} \in \mathrm{movies}} (r_{\mathrm{a,m}})^2}}.$$

Also want to compare entropy-based, mean-square difference, and SimRank similarity measures.

Groups 1-3

Motivation

- How much do we trust the computed correlation values if it's based on very small amount of co-rated items.
- Neighbors based on a small number of co-rated samples not usually good predictors for the active user.
- More data points to compare, more we trust the computed correlation.

Explore: Will prediction accuracy improve if we add a correlation significance factor that devalues similarity weights based on a small amount of co-rated items?

Component: Variance Weighting

Groups 1-3

Motivation

- All similarity measures treat each item evenly in a user-to-user correlation calculation.
- In fact, a user's rating on certain items is more valueable than others.
- E.g. lots of people rank Titanic highly, so if two users both rank Titanic highly doesn't tell us much about shared interests.

Explore: Will prediction accuracy improve if we give distinguishing movies more influence in calculating correlation?

Groups 4-5

Motivation

- After calculating similarity weights, we select which other users' data are used in computing the predictions (currently, all of them).
- There is evidence that selecting a subset of users improves accuracy.
- Moreover, when there are millions of users, using them all is infeasible.

Explore: Will prediction accuracy improve if we select the best neighbors of the active user to use in calculating predictions?

Groups 6-7

Motivation

- Once a neighborhood is selected, ratings are combined to make a prediction.
- We've been computing a weighted average of the deviation of a neighbor's rating from her mean weighting.
- Could a simple weighted average be better, or maybe a weighted average of z-scores as opposed to simply deviations?

Explore: Will prediction accuracy improve if normalize the weighted average differently?

First Project Task

Implement and **evaluate** the performance of two collaborative filtering algorithms – one model-based and the other memory-based – on both datasets.

- **[Model 1]** Model-based algorithm: clustering discussed in paper [1] section 2.3.
- **[Model 2]** Memory-based algorithm: user-based neighborhood model using Pearson's correlation for similarity weight. This is introduced in equations (1) and equations (2) in paper [2].

Model-based Algorithm

We implemented Model 2 earlier, now let's look at the details of Model 1.

Notation

- N users and M movies.
- Let $I(i)$ be the set of movies users i has scored in the training set.
- Let $r_{i,m}$ be the score user i gave to movie m where $m \in I(i)$ and $r_{i,m} \in \{0, 1, 2, 3, 4, 5\}$.
- Let $R_{i,m}$ be the random variable representing the score of user i on movie m (as opposed to the realization $r_{i,m}$.)

Assume: Each user belongs to one of C different classes or groups. Denote the class of user i by G_i

Probablistic Perspective

CF task is estimating expected value of a vote or rating, given what we know about the user.

Main Idea

Let a be the active user and movie $m \notin I(a)$.

$$\hat{r}_{a,m} = \mathbb{E}(R_{a,m} \,|\, r_{a,j}, j \in I(a)) = \sum_{k=1}^{5} k \cdot P(R_{a,m} = k \,|\, r_{a,j}, j \in I(a)).$$

where

$$P(R_{a,m} = k \,|\, r_{a,j}, j \in I(a))$$

is the probability that user a gives movie m a k rating, given user a's other movie ratings.

Now we need to estimate $P(R_{a,m} = k \,|\, r_{a,j}, j \in I(a))$ for each k.

Can simplify...

$$
\begin{aligned}
&P(R_{a,m} = k \,|\, r_{a,j}, j \in I(a)) \\
&\overset{(a)}{=} \frac{P(R_{a,m} = k;\; R_{a,j} = r_{a,j}, j \in I(a))}{P(R_{a,j} = r_{a,j}, j \in I(a))} \\
&\overset{(b)}{=} \frac{\sum_{c=1}^{C} P(R_{a,m} = k; R_{a,j} = r_{a,j}, j \in I(a);\; G_a = c)}{\sum_{c=1}^{C} P(R_{a,j} = r_{a,j}, j \in I(a);\; G_a = c)} \\
&\overset{(c)}{=} \frac{\sum_{c=1}^{C} P(R_{a,m} = k; R_{a,j} = r_{a,j}, j \in I(a); \,|\, G_a = c) \cdot P(G_a = c)}{\sum_{c=1}^{C} P(R_{a,j} = r_{a,j}, j \in I(a) \,|\, G_a = c) \cdot P(G_a = c)} \\
&\overset{(d)}{=} \frac{\sum_{c=1}^{C} P(R_{a,m} = k \,|\, G_a = c) \cdot \prod_{j \in I(a)} P(R_{a,j} = r_{a,j} \,|\, G_a = c) \cdot P(G_a = c)}{\sum_{c=1}^{C} \prod_{j \in I(a)} P(R_{a,j} = r_{a,j} \,|\, G_a = c) \cdot P(G_a = c)}.
\end{aligned}
$$

- Step (a) follows by Bayes' rule.
- Step (b) includes the users' groups.
- Step (c) uses Bayes' rule again.
- Step (d) uses a standard Naive Bayes formulation.

Bayesian Cluster Model

Now, in order to calculate

$$\hat{r}_{a,m} = \mathbb{E}(R_{a,m} \mid r_{a,j}, j \in I(a)) = \sum_{k=1}^{5} k \cdot P(R_{a,m} = k \mid r_{a,j}, j \in I(a)),$$

we need to estimate the following values:

$$P(G_a = c) \text{ for all } c \in \{1, 2, \dots C\},$$
$$P(R_{a,j} = k \mid G_a = c) \text{ for all } j \in I(a) \cup \{m\}, k \in \{0, 1, \dots, 5\}, c \in \{1, 2, \dots C\}.$$

Cluster Assumption

We assume all users in the same class, c, have the same rating probabilities. Namely, for any two users, i_1 and i_2, we will assume:

1. $P(G_{i_1} = c) = P(G_{i_2} = c)$ for all $c \in \{1, 2, \dots, C\}$.
2. $P(R_{i_1,j} = k \mid G_{i_1} = c) = P(R_{i_2,j} = k \mid G_{i_2} = c)$ for all c, k, and j.

Therefore, approximately $C + 6CM$ parameters to estimate!

Parameters to Estimate

Because of the cluster assumption, we can simplify our notation:

$$\mu_c := P(G_a = c) \text{ for all } c \in \{1, 2, \ldots C\},$$
$$\gamma_{j,c}^{k} := P(R_{a,j} = k \mid G_a = c) \text{ for all } j, k, c.$$

where

$$\sum_{c=1}^{C} \mu_c = 1,$$
$$\sum_{k=0}^{5} \gamma_{j,c}^{k} = 1.$$

Parametric Models, Generally

Models

A **model** $\mathcal{P}$ is a set of probability distributions. We index each distribution by a parameter value $\theta \in \mathcal{T}$; we can then write the model as

$$\mathcal{P} = \{p(x|\theta)|\theta \in \mathcal{T}\} .$$

The set $\mathcal{T} \subset \mathbb{R}^d$, for some fixed dimension d, is called the **parameter space** of the model.

Parametric model

The model is called **parametric** if the number of parameters (i.e. the dimension of the vector θ) is (1) finite and (2) independent of the number of data points. Intuitively, the complexity of a parametric model does not increase with sample size.

Our Problem

$$\begin{aligned} \theta &= (\mu_c, \gamma_{j,c}^k; \text{ for all } j, k, c), \\ \mathcal{T} &= \text{ space of all possible values of } \theta = [0,1]^{C+6CM}. \end{aligned}$$

Maximum Likelihood Estimation

Setting

- Given: Data $x_1, \ldots, x_n$, parametric model $\mathcal{P} = \{p(x|\theta) \,|\, \theta \in \mathcal{T}\}$.
- Objective: Find the distribution in $\mathcal{P}$ which best explains the data. That means we have to choose a "best" parameter value $\hat{\theta}$.

Maximum Likelihood approach

Maximum Likelihood assumes that the data is best explained by the distribution in $\mathcal{P}$ under which it has the highest probability (or highest density value).

Hence, the **maximum likelihood estimator** is defined as

$$\hat{\theta}_{ML} := \arg\max_{\theta \in \mathcal{T}} p(x_1, \ldots, x_n|\theta)$$

the parameter which maximizes the joint density of the data.

Maximum Likelihood

How do we estimate our parameters using ML?

Likelihood Function

For user i, write the **likelihood function** as:

$$L_i(\mu, \gamma \,|\, \text{data}) = \sum_{c=1}^{C} P(G_i = c) \cdot \prod_{j \in I(i)} P(R_{i,j} = r_{i,j} \,|\, G_i = c)$$
$$= \sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} \gamma_{j,c}^{r_{i,j}}.$$

Then the likelihood of the full data is

$$L(\mu, \gamma \,|\, \text{data}) = \prod_{i=1}^{N} L_i(\mu, \gamma \,|\, \text{data}) = \prod_{i=1}^{N} \Big(\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} \gamma_{j,c}^{r_{i,j}} \Big).$$

Likelihood Function

Often easier to work with the **log-likelihood function**:

$$\begin{aligned}\ell(\mu, \gamma \,|\, \text{data}) &= \log \Big[\prod_{i=1}^{N} L_i(\mu, \gamma \,|\, \text{data}) \Big] \\ &= \sum_{i=1}^{N} \log L_i(\mu, \gamma \,|\, \text{data}) \\ &= \sum_{i=1}^{N} \log \Big(\sum_{c=1}^{C} \mu_c \cdot \prod_{j \in I(i)} \gamma_{j,c}^{r_{i,j}} \Big).\end{aligned}$$

Next time, we'll look at how to use the EM algorithm to maximize the likelihood.