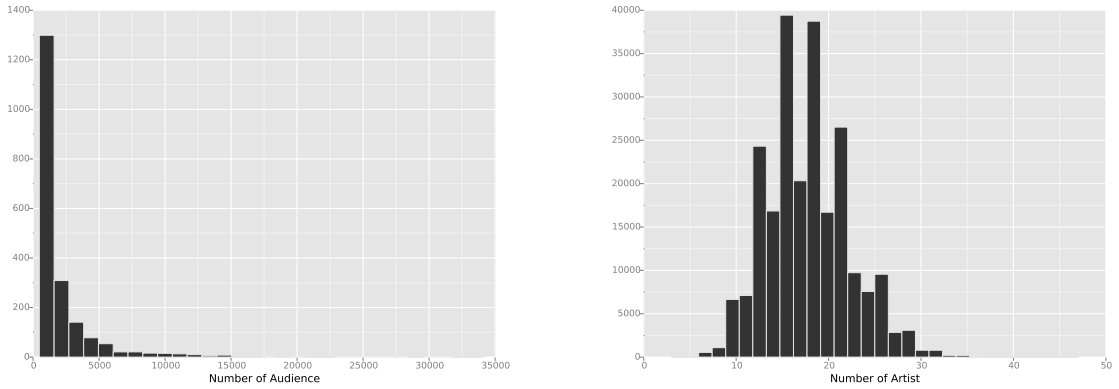
CS 181 Machine Learning
Practical 3 Report, Team *Deep Ellum*

(Vivian) Wenwan Yang[1], Jing Wen[2], and (Jeremiah) Zhe Liu[2]

[1]Department of Computer Science, Harvard School of Engineering and Applied Sciences
[2]Department of Biostatistics, Harvard School of Public Health

April 22, 2015

1 Exploratory Analysis



(a) Number of Audience per Artist (b) Number of Artist per Audience

Figure 1: Distribution of the number of audience per artist (**Left**) and the number of artist per audience (**Right**)

2 A Collaboration-Filtering-based Approach

2.1 Background

The number of plays r_{ui} measures the preference by user u of artist i, where high values indicate stronger preference. Baseline predictors capture the systematic tendencies for some artists to receive more numbers of plays than others, and for some users to play more often than others. We denote the overall number of plays by μ.The parameters b_u and b_i respectively represent the observed bias of user u and artist i from the average. So a baseline prediction for the number of plays r_{ui} is denoted by b_{ui}:

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

We then minimize the regularized square error to obtain the model parameters:

$$\min_{\mathbf{b}*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i) + \lambda_1 \Big(\sum_u b_u^2 + \sum_i b_i^2 \Big)$$

The first term intends to find b_u's and b_i's that fit the given numbers of plays. The second regularizing term penalizes the magnitude of the parameters to avoid overfitting.

2.2 SVD Model

Matrix factorization techniques model user-item interactions as inner products in a joint latent factor space of f dimensions. Accordingly, for a given artist i, the elements of q_i measure item characteristics. For a given user u, the elements of p_u measure user preferences. Hence the dot product, $q_i^T p_u$ represents the interaction between artist i and user u, where high values mean stronger overall interest of the user in the artist's characteristics. Thus, the number of plays is predicted by the formula:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

In order to estimate (b_u, b_i, p_u and q_i) one can solve the least squares problem:

$$\min_{\mathbf{b}*,\mathbf{q}*,\mathbf{p}*} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \mu - b_u - b_i - q_i^T p_u)^2 + \lambda_4 \Big(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2 \Big)$$

The constant λ_4 is usually determined by cross validation. A straightforward stochastic gradient descent optimization involves the algorithm which loops through all numbers of plays in our training data. Denote by $e_{ui} = r_{ui} - \hat{r}_{ui}$ the associated prediction error. We move in the opposite direction of the gradient and modify the parameters:

1. $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_u)$
2. $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_4 \cdot b_i)$
3. $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda_4 * q_i)$
4. $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_4 * p_u)$

2.3 SVD ++ Model

"SVD++ model" integrates other implicit feedback in order to increase prediction accuracy. It models a user factor by the identity of the artists, and offers accuracy superior to SVD. A new set of item factors is added to characterize users according to the set of artists. The new item factors relate each artist$i$ to a factor vector $y_i \in \mathbb{R}^f$. The model is specified as below, where set $R(u)$ contains the artists correspond to user u:

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \Big(p_u + |R(u)|^{-\frac{1}{2}} \sum_{j\in R(u)} y_j \Big)$$

Model parameters are determined by solving the related least squares problem. We loop over all known plays in $\mathcal{K}$:

1. $b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_u)$
2. $b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda_5 \cdot b_i)$
3. $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j\in R(u)} y_j) - \lambda_6 * q_i)$
4. $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda_6 * p_u)$
5. $\forall j \in R(u)$: $y_j \leftarrow y_j + \gamma \cdot (e_{ui} \cdot |R(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_6 \cdot y_j)$

2.4 Fused item-item user-user model

A user-user neighborhood model predicts rating by considering how users with similar tastes rated the same items. Each user u is associated with two vectors $p_u, z_u \in \mathbb{R}^f$. Item-item relationships could be factored by connecting each item i with three vectors: $q_i, x_i, y_i \in \mathbb{R}^f$, which map items into a latent factor space. Combining predictions of both item-item and user-user models gives improved overall accuracy. The below model sums the user-user model and item-item model, which optimizes the two models simultaneously:

$x_i, y_i, q_i \in \mathbb{R}^f$

$$\hat{r}_{ui} = \left\{\mu + b_i + b_u\right\} + q_i^T\left\{\frac{\sum_{j\in R(u)}(r_{ui} - b_{uj})x_j + y_j}{|R(u)|^{-\frac{1}{2}}}\right\} + p_u^T\left\{\frac{\sum_{v\in R(u)}\left[(r_{vi} - b_{vi})z_v\right]}{|R(i)|^{-\frac{1}{2}}}\right\}$$

In order to estimate $(x_i, y_i, z_v, p_u, q_i)$, one can solve the below least square problem through Stochastic Gradient Descent:

$$\min_{\mathbf{q},\mathbf{p},\mathbf{x},\mathbf{y},\mathbf{z}} \sum_{(u,i)\in\mathcal{K}} (r_{ui} - \hat{r}_{ui}) + \lambda\Big(\|b_i\|^2 + \|b_u\|^2 + \|q_i\|^2 + \|p_u\|^2 + \sum_{j\in R(u)}(\|x_j\|^2 + \|y_j\|^2) + \sum_{v\in R(i)}(\|z_v\|^2)\Big)$$

LearnFactorizedModel(r_{ui}, f)

for iteration = 1,..., **do**

  for user = 1,..., **do**

2.5 Parameter selection

3 Results

4 Discussion

The collaborative filtering algorithms described above may not be successful to model the number of plays, as the existing algorithms have trouble making accurate predictions for users with very few plays. Alternatively, probabilistic algorithms that scale linearly with the number of observations have been proved to perform well on very sparse and imbalanced datasets, such as our Streaming Music dataset.

4.1 Probabilistic Matrix Factorization (PMF)

Denote the number of plays of user u for artist i by R_{ui}. Suppose we have M artists and N users. Let $U \in R^{D*N}$ and $V \in R^{D*M}$ be latent user and artist feature matrices, with column vectors U_u and V_i representing user-specific and artist-specific latent feature vectors respectively. We can define the conditional distribution over the observed numbers of plays as

$$p(R|U, V, \sigma^2) = \prod_{u=1}^{N}\prod_{i=1}^{M}\left[\mathcal{N}(R_{ui}|U_u^T V_i, \sigma^2)\right]^{I_{ui}}$$

,

where $\mathcal{N}(x|\mu, \sigma^2)$ represents the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I_{ui} is the indicator function that is equal to 1 if user u listened to the track from

artist i and equal to 0 otherwise. We also place zero-mean spherical Gaussian priors on user and artist feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_u|0, \sigma_U^2 I)$$

$$p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_i|0, \sigma_V^2 I)$$

The log of the posterior distribution over the use and the artist feature is:

$$\begin{aligned} \ln p(U, V|R, \sigma^2, \sigma_V^2, \sigma_U^2) = &- \frac{1}{2\sigma^2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{ui}(R_{ui} - U_u^T V_i)^2 - \frac{1}{2\sigma_U^2} \sum_{u=1}^{N} U_u^T U_u - \frac{1}{2\sigma_V^2} \sum_{i=1}^{M} V_i^T V_i \\ &- \frac{1}{2}\Big(\Big(\sum_{u=1}^{N} \sum_{i=1}^{M} I_{ui} \Big) \ln \sigma^2 + ND \ln \sigma_U^2 + MD \ln \sigma_V^2 \Big) + C \end{aligned}$$

Maximizing the above log-posterior over artist and user features is equivalent to minimizing the below sum-of-squared-errors objective function:

$$E = \frac{1}{2} \sum_{u=1}^{N} \sum_{i=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^{N} \|U_u\|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{i=1}^{M} \|V_j\|_{Fro}^2$$

where $\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$, and $\|\cdot\|_{Fro}^2$ represents the Frobenius norm. This model is a probabilistic extension of the SVD model, since the objective function reduces to the SVD objective in the limit of prior variances going to infinity.

4.2 Automatic Complexity Control for PMF

Adaptive priors can be included in the PMF model over the artist and user feature vectors to control model complexity automatically. Specifically, the model complexity is controlled by the hyperparameters: the noise variance σ^2 and the parameters of the priors σ_U^2 and σ_V^2. Using spherical priors for user and artist feature vectors leads to automatically chosen λ_U and λ_V. We maximize the below log-posterior to find a point estimate of parameters and hyperparameters:

$$\ln p(U, V, \sigma^2, \Theta_U, \Theta_v|R) = \ln p(R|U, V, \sigma^2) + \ln p(U|\Theta_U) + \ln p(V|\Theta_V) + \ln p(\Theta_U) + \ln p(\Theta_V) + C$$

where Θ_U and Θ_V are the hyperparameters for the priors over user and artist feature vectors respectively. We can simplify learning by alternating between optimizing the hyperparameters and updating the feature vectors using steepest asent with the values of hyperparameters fixed.

4.3 Constrained PMF

Based on the assumption that users who play tracks from similar artists have similar preferences, we can also implement a constrained version of the PMF model. We constrain user-specific feature vectors that have a strong effect on infrequent users. Let $W \in R^{D*M}$ be a similarity constraint matrix. The column vectors of the W matrix capture the effect of a user having played music from a particular artist has on the prior mean of the user's feature vector. Therefore, users that have played music from the similar artists will have similar prior distributions for their feature vectors.

As a result, the training time for the constrained PMF model scales linearly with the number of observations, providing a fast and simple implementation.

5 Reference