MovieLens Ratings Predictor

Arthur Nghiem

December 2024

1 Problem Description

Digital services use recommendation systems to suggest items that a particular user is likely to appreciate. Such systems require data regarding the user's past behavior. This data might include how many times they purchased a certain product, how long they spent watching a video, whether or not they put a song into their playlist, and so on. Based on this information, the recommendation system makes inferences on how the user will likely react to other items.

The problem of low rank matrix completion is relevant to recommendation systems. We have a matrix $M \in \mathbb{R}^{m\times n}$ representing the interactions between users and items. Row i represents user i, column j represents item j, and M_{ij} is some numerical representation of the user's response to the product. Since each user only interacts with some of the items, only a sparse subset of the entries of M can be observed. The goal of a recommendation system is to accurately estimate the unobserved entries. This would not be possible if all entries were independent. However, we theorize that each user and item can be described by $d << \min(m, n)$ latent factors. M then equals U^TV, where $U \in \mathbb{R}^{d\times m}$ is the user matrix and $V \in \mathbb{R}^{d\times n}$ is the item matrix. $rank(M) \leq d$ in this case, which is the motivation for completing M with low rank.

This project is a computational study of low rank matrix completion applied to the ml-32m dataset [1] collected by MovieLens. This dataset contains 32 million movie ratings. Each rating consists of a user ID, a movie ID, and the user's review score of the movie as an integer from 1 to 5. To reduce the matrix to a manageable size, we exclude data outside of a small subset of movies and users. This filtered data is then split into a training set containing 80% of the data and a test set containing 20% of the data. The matrix M will be initialized from the training set, then completed using various optimization techniques. Each completed matrix will then be compared against the test data to evaluate the accuracy of the algorithms.

2 Optimization Models

The basic formulation of the low rank matrix completion problem is:

$$\min_X \operatorname{rank}(X)$$

$$\text{s.t. } X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$

M and Ω are obtained from M_train.csv and df_train.csv respectively. The following subsections will address some matrix completion techniques which can be used to find X.

2.1 Baseline Method

To establish a standard for comparison, we first evaluate the performance of two baseline models. The first possibility is to assign the global average of observed ratings (a score of 3.78) to every unobserved entry. Alternatively, for any unobserved entry X_{ij}, we could assign the average rating given by user i. We let $X_{ij} = M_{ij}$ for all $(i,j) \in \Omega$ in both cases. This method doesn't explicitly attempt to minimize the rank of X, but our understanding of the problem assures us that these are reasonable ways to generate rough estimates. Using the global average results in a test error of $\|P_{\Omega_{test}}(X - M_{test})\|_F^2 \approx 906.694$, whereas using the user averages results in a slightly lower test error of 769.480.

2.2 Matrix Factorization

One way to reformulate the matrix completion problem is:

$$\min_{U \in \mathbb{R}^{d \times m}, V \in \mathbb{R}^{d \times n}} \left\| P_\Omega(U^T V - M) \right\|_F^2$$

M and Ω are still determined through data collection. We also have $d \geq \text{rank}(U^T V)$, which acts as a hyper-parameter in this optimization model. Since the problem is convex and the gradient is computable, we can apply a gradient descent method. The partial derivatives of the objective are $\frac{\partial}{\partial U} = 2VP_\Omega(U^T V - M)$ and $\frac{\partial}{\partial V} = 2UP_\Omega(U^T V - M)$. It is now possible to derive an update rule to implement stochastic gradient descent. To update U and V in response to M containing a single randomly sampled observation M_{ij}, we necessarily have $(i, j) \in \Omega$. Therefore, for some fixed step size α, we have the following update rules:

$$u_i \leftarrow u_i - \alpha(u_i^T v_j - M_{ij}) v_j$$

$$v_j \leftarrow v_j - \alpha(u_i^T v_j - M_{ij}) u_i$$

One may notice that u_i is needed to update v_j and vice versa. We address this by first using v_j from the previous iteration to update u_i, then using the newly updated u_i to update v_j. This is known as an alternating minimization approach. During each iteration of stochastic gradient descent, we use each observation once in a randomized order. We repeat this process for T iterations. After tuning the hyper-parameters, it appears that $d = 1$, $\alpha = 0.01$, and $T = 100$ yields the best results. This combination achieved a test error of 728.489 in one trial - this test error does vary between trials, but not significantly.

2.3 Convex Relaxation

Another way to complete the matrix is to solve the convex relaxation of rank minimization. Since we know that there is significant noise in our data, we express this as follows:

$$\min_X \|X\|_*$$

$$\text{s.t. } \|P_\Omega(X - M)\|_F^2 \leq \delta^2$$

We can reformulate this problem to:

$$\min_{\|X\|_* \leq \vartheta} \|P_\Omega(X - M)\|_F^2$$

M and Ω are once again pre-determined, whereas ϑ is a hyper-parameter. We now recognize from a previous assignment that this problem can be solved by the Frank-Wolfe method, in which we can use the singular value decomposition of $\nabla f(X)$ to solve the minimization problem posed at each iteration. We repeat this process for T iterations. Implementing this method for this problem and optimizing the hyper-parameters, we achieve a test error of 829.223 with step size $\tau = \frac{2}{2+t}$, $\vartheta = 600$, and $T = 2000$.

3 Conclusions

We have demonstrated that a low rank matrix completion methods could be used to more accurately predict a user's response to items in the context of movie reviews. The matrix factorization solution achieves lower test error than both baseline methods. This technique could be used to provide users with movie recommendations that they would likely enjoy, as indicated by their ratings. This is a task of great commercial importance to companies such as Netflix. Furthermore, since all of our methods terminate within a few seconds on a 198×100 matrix, there is reason to believe that they could be applied to datasets on a much larger scale.

The performance of the convex relaxation method is disappointing. It fails to achieve a test error below the one of the baseline methods, so this optimization formulation might be inappropriate for this application. We are also somewhat suspicious of the matrix completion method because the best solution supposedly occurs at $d = 1$, even though training error is strictly decreasing as d increases. Perhaps there is a significant over-fitting issue for $d > 1$, but this warrants further investigation.

References

[1] F. Maxwell Harper and Joseph A. Konstan. “The MovieLens Datasets: History and Context”. In: *ACM Trans. Interact. Intell. Syst.* 5.4 (Dec. 2015). ISSN: 2160-6455. DOI: `10.1145/2827872`. URL: `https://doi.org/10.1145/2827872`.