A survey of matrix completion and clustering techniques in the context of Recommender Systems

Niharika Shimona D'Souza
PhD Candidate
Department of Electrical and Computer Engineering
Shimona.Niharika.Dsouza@jhu.edu

Abstract

Matrix completion is the task of filling in the missing entries of a partially observed matrix. A wide range of datasets are naturally organized in matrix form, for eg. the movie-ratings matrix, as is manifested in the popular Netflix problem. In this project, we formulate this as a low rank matrix completion problem, and leverage the completed user ratings matrix for setting up subspace clustering for identification separation of movies by genre information.

1. Introduction

A recommender system or a recommendation system forms a subclass of information filtering systems which seek to predict the rating or preference that a user would give to an item. Since an extremely limited amount of information is available apriori (a given user would typically rate only a subset of films that hes watched), the data matrix consisting of the user ratings for the exhaustive list of movies available is very sparsely populated, thus making the problem of predicting the remaining entries intelligently and finding homogeneous groups of similar genre movies extremely challenging.

The project aims to employ some of the matrix completion introduced in class to employ available user information and combine it with genre information extracted through subspace clustering techniques. In the past, many collaborative filtering algorithms have developed for the same.

1.1. Problem Description

The Movie Lens 100k dataset, made publicly available by GroupLens providing 100,000 ratings (1-5) from 943 users on 1682 movies was used for the purpose of testing the completion algorithms. Here, each user has rated at least 20 movies and information about genre has been provided for each movie listed. These movies are subdivided into 18 different genre: Action , Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western. This matrix is highly sparse in nature. The dataset is also manifested with multiple assignment of genre to a large number of movies. Thus, the choice of datasets for the evaluation of clustering techniques for genre separation needs to be performed wisely.

1.2. Datasets

Based on the genre information provided, two small datasets consisting of two distinctive genres i.e. Comedy and Thriller were extracted, and combined into a medium dataset, along with a large dataset containing all genres. The dataset also provides five cross validation splits into training and testing sets for the purpose of evaluating the performance of matrix completion. For the purpose of clustering, instead of using the entire large matrix, a pseudo-large matrix consisting of 4 genre with minimum overlap are used: Comedy, Thriller, Film-Noir, Documentary. The few overlapping entries are removed from the set.

2. Proposed Solution

2.1. Matrix Completion

Since information about the exact location for the missing entries is also available to us, we have made use of the following three algorithms:

2.1.1 Low-Rank Matrix Completion via Proximal Gradient:

Solves the following optimisation problem, where X is rating matrix, the tunable parameters being τ and the proximal parameter β [5]

$$\arg\min_{A} f_\tau(A) = \tau||A||_\star + \frac{1}{2}||A||_F^2 \qquad (1)$$

$$\text{s.t. } P_\Omega(X) = P_\Omega(A)$$

Algorithm:

1. Initialise $Z = 0$
2. $A \leftarrow \mathcal{D}_\tau(P_\Omega(Z))$
3. $Z \leftarrow Z + \beta(P_\Omega(X) - P_\Omega(A))$
4. Repeat until convergence of Z

A is the completed matrix of entries obtained.

2.1.2 Matrix Completion using Soft-Impute

Solves the optimisation problem where X is the input matrix. The optimisation parameter is a sequence of decreasing λ's. i.e. $\lambda_1 > \lambda_2 ... > \lambda_k$ [3]

$$\arg\min_Z f_\lambda(Z) = \frac{1}{2}||P_\Omega(X) - P_\Omega(Z)||_F^2 + \lambda||Z||_\star \quad (2)$$

Algorithm:

Initialise $Z^{old} = 0$, For $\lambda_1 > \lambda_2 ... > \lambda_k$, repeat:

1. Compute $Z^{new} \leftarrow S_{_k}(P_\Omega(X) + P_\Omega^\perp(Z^{old}))$
2. if $\frac{||Z^{old} - Z^{new}||_F^2}{||Z^{old}||_F^2} < \epsilon$ exit
3. $Z^{old} \leftarrow Z^{new}$
4. $\hat{Z}_{\lambda_k} = Z^{old}$

The output is a sequence of solutions $\hat{Z}_{\lambda_k}$s

2.1.3 Matrix Factorisation:

Solves the optimisation problem where X is the input matrix, for a low rank solution to factors U and V of rank r using the regularisation parameters λ_u and λ_v.[4] provides a closed form solution for the fully observed matrix case.

$$\arg\min_{U,V,r} f(X) = ||P_\Omega(X - UV^T)||_F^2 + \lambda_u||U||_F^2 + \lambda_v||V||_F^2 \quad (3)$$

Maximum margin matrix factorisation , as mentioned in [4] when implemented setting $\lambda_u = \lambda_v = \lambda$ in the objective function mentioned above, solution of the form:

$$\hat{U}_{M\text{x}r} = U_X S_\lambda(D_r)^{\frac{1}{2}}; \hat{V}_{N\text{x}r} = V_X S_\lambda(D_r)^{\frac{1}{2}} \quad (4)$$

Where U_x and V_x are the left and right singular vectors of the low rank matrix fully observed matrix X.

When X is not fully observed as in our case, the problem is formulated as a biconvex optimisation problem in U and V which is solved by multiple ridge regressions using alternating least squares method and iteratively solving for U and V as alluded to in [2]

Algorithm: Repeat until convergence:

1. $X^* = P_\Omega(X - UV^T) + UV^T$
2. $U = X^*V(V^TV + \lambda I)^{-1}$
3. $X^* = P_\Omega(X - UV^T) + UV^T$
4. $V = X^{*T}U(U^TU + \lambda I)^{-1}$
5. $k \rightarrow k + 1$

This algorithm is known as Soft-Impute ALS and outputs the factors U and V.

2.2. Clustering Techniques

The following clustering algorithms were used:

2.2.1 K-Means:

K- means formulates the following optimisation problem

$$\arg\min_{\mu_i, w_{ij}} \sum_{i=1}^{n}\sum_{j=1}^{N} w_{ij}||x_j - \mu_i||^2 \quad (5)$$

and $w_{ij} \in \{0, 1\}$ and $\sum_{i=1}^{N} w_{ij} = 1$ where the means are given by μ_i's and w_{ij}'s are the cluster assignment.

Algorithm:

1. Initialise the cluster centers μ_i's
2. Calculate the assignments $w_{ij} = \min ||x_j - \mu_i||^2$
3. Recalculate the cluster centers

$$\mu_i = \frac{\sum_{j=1}^{N} w_{ij}x_j}{\sum_{j=1}^{N} w_{ij}} \quad (6)$$

Repeated until convergence, the outputs are the set of cluster centers and assignments for each example.

2.2.2 Spectral Clustering:

The clustering is determined by the eigenvectors of the Graph Laplacian. The Laplacian is constructed by using an affinity graph while the clustering of the eigenvectors is performed using k means. For this application, two different affinities are tested out:

- k-NN
- rbf

Algorithm:

1. Construct an affinity graph $\mathcal{G}$ with weight matrix W
2. Compute the degree matrix $D = \text{diag}(W\mathbf{1})$ and the Laplacian $\mathcal{L} = D - W$

3. Compute n eigenvectors of $\mathcal{L}$ associated with the n smallest eigenvalues, n being the number of clusters.

4. Normalise the eigenvectors y_i's which are the columns of the matrix $Y \in \mathbf{R}^{nxN} = [u_1, ..., u_N]^T$ of eigenvectors of the Laplacian

5. Perform k means on the set of normalised eigenvectors to assign into n cluster centers.

3. Evaluation Metrics:

3.1. Matrix Completion

Since the completed matrix of all ratings isn't available to us, we resort to other methods to evaluate the matrix completion performance.

RMSE can defined over a cross validation set Ω_S which is a randomly selected set (10 percent) of observed values , held back at each trial for the purpose of robustness and reliability of experiments. An average over 10 different trials gives a rough estimate of the goodness of performance.

$$\mathbf{RMSE}(X, \hat{A}) = \sqrt{\frac{1}{\Omega_S} \sum_{ij \in \Omega_S} (X_{ij} - \hat{A}_{ij})}$$

3.2. Clustering Techniques

The cluster assignment being random at each iteration, a Hungarian matching is done to determine the correspondence between cluster assignments. Following this, the error per label is reported after performing a run of the algorithm.

However, the pureness of each cluster needs to be quantified to correctly identify cases where all the points get assigned to clusters randomly.The Silhouette Coefficient s_i, is one such measure, which, for the i^{th} object evaluates the quality of fit. The average distance to all objects in it's cluster is given by a_i, while the minimum value of the distance to every other cluster is given by b_i:

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)}$$

4. Experiments:

Initial experiments on matrix completion were carried out using the proximal gradient algorithm. The convergence of the algorithm on this dataset is highly sensitive to the β i.e. the proximal parameter for the gradient ascent step, which was set to 1.9 for experiments on the small, medium and large datasets, (the algorithm is provably convergent for $0 < \beta < 2$) and the τ parameter was set to $5\sqrt{nd}$ as mentioned in [1]. The tolerance limit for the algorithm is set at 10^{-2} (early stopping).

Soft-Impute as introduced in [3] was implemented with initial nuclear norm rank penalty was set by searching across an exponential grid of values (10^{-4} to 10) and fixing it as the one giving the lowest testing error. The tolerance value for convergence was set to 10^{-4}.

The Soft-Impute ALS algorithm [2] for matrix factorisation was implemented with the λ varied over a grid, the rank estimate from LRMC was used as an initial estimate and the r parameter was varied within a small range of this value; the set of parameters giving the lowest test set error is reported.

For the clustering part of the project ,k means was implemented first to get an estimate of the performance over the medium and pseudo-large dataset. Since the algorithm is extremely sensitive to initialisation, the k means++ algorithm was employed.

For spectral clustering, both the rbf and k nearest neighbour affinities were tried. For the medium and pseudo- large datasets, a range from 5-30 was tried over step sizes of 5 to select the optimal k for the affinity graph and the error and silhouette coefficient plots were compared for qualitative assesments. The kNN affinity seemed to outperform the rbf and hence the results from kNN are reported.

After separation into clusters , the matrix completion step is performed individually on the medium as well as pseudo large datasets, and the RMSE values reported based on a weighted average of the cluster size.

The packages scipy, cvxopt, sklearn, pandas over Python 2.7.6 are used for implementing and evaluating the aforementioned algorithms which are publicly available without license.

Some freely accessible online code repositories were used as a reference for these implementations:

1. MMMF

2. Soft-Impute ALS

5. Results

Quantitative results from matrix completion are tabulated for comparison. In each of the cases, the RMSE error is reported. The rank of the final solution obtained for each of the cases is also reported. It was observed that the smaller of the two datasets. i.e. Thriller gave a larger RMSE upon completion. This can be attributed to the fact that the percentage of user ratings in this dataset

For clustering, the misclassification error from each subset is reported after matching. The qualitative evaluation using the silhouette plots illustrate the pureness of cluster separation for each of the algorithms.

After separation into different genre by clustering, results of matrix completion on the clusters is also performed and weighted average RMSE errors have been reported over the medium and the pseudo large dataset.

Method	RMSE	Solution Rank
LRMC Proximal Gradient	1.8651	91
Soft Impute	1.632	93
Matrix Factorisation	1.211	91

Table 1. Matrix Completion results on Dataset 1: Comedy

Method	RMSE	Solution Rank
LRMC Proximal Gradient	1.93	66
Soft Impute	1.915	63
Matrix Factorisation	1.244	64

Table 2. Matrix Completion results on Dataset 2: Thriller

Method	RMSE	Solution Rank
LRMC Proximal Gradient	2.479	153
Soft Impute	2.315	151
Matrix Factorisation	1.479	153

Table 3. Matrix Completion results on the Medium Dataset

Method	RMSE	Solution Rank
LRMC Proximal Gradient	2.905	202
Soft Impute	2.78	204
Matrix Factorisation	1.89	200

Table 4. Matrix Completion results on the Large Dataset

Dataset	K Means	Spectral Clustering (kNN)
Comedy	24.44	14.47
Thriller	43.15	46.23

Table 5. Clustering Results on the Medium Dataset (Errors)

Dataset	K Means	Spectral Clustering (kNN)
Comedy	36.44	23.14
Thriller	31.75	24.34
Film-Noir	30.45	21.89
Documentary	44.78	31.03

Table 6. Clustering Results on the Pseudo Large Dataset (Errors)

Dataset	Before clustering	After clustering
Medium Dataset	1.479	1.413
Pseudo Large Dataset	1.861	1.872

Table 7. Matrix Completion Result Comparison before and after genre based separation (weighted avg. RMSE)

6. Conclusions

From the run of the low rank matrix completion, the RMSE values from LRMC were relatively higher over both of the small, medium and large datasets. Accounting for

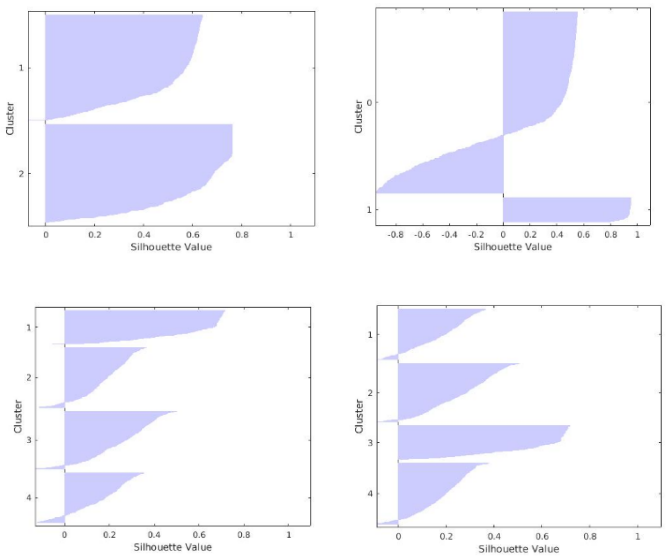

Figure 1. Top left :Results from k means for medium dataset, Top Right: Result of Spectral Clustering for medium dataset, Bottom left: Results from k means on pseudo large dataset, Bottom right: Results from spectral clustering on pseudo large dataset

corruptions using a robust version (with L_1 penalty for sparsity of the corrupted entries) [5] of the algorithm seems to improve the performance of LRMC slightly, but the results weren't comparable to the performance of other methods like matrix factorisation.

The Soft-Impute algorithm performs better with lower RMSE values compared to LRMC and is easier to implement because of lesser number of tunable parameters. It converges faster and fits rank 93 and 63 solutions to the small datasets, rank 151 and a rank 204 solution to the large dataset.

Matrix factorisation seems to outperform both of these in terms of RMSE, because it exploits the structure (Movie information/ User information) in the matrix better.The values of RMSE is significantly lower for the Soft-Impute ALS case.

In terms of clustering, Spectral clustering outperforms k means in terms of the error from each classes. This is expected because k means is only based upon the euclidian distances over the large set of features which isn't a good measure for comparison given the complexity of the problem. However, the overall clustering performance is not very good,since many of entries are misclassified in these cases. An inspection from the silhouette plots shows that though the overall error of separation may be lower, cases where many points get assigned to the same cluster are clearly visible. The genre don't seem to be very well separated given the results on the pseudo large dataset.

However, experiments using more sophisticated tech-

niques eg. Sparse Subspace Clustering, Local Subspace Affinity need to be carried out, whereby, the data can be represented better in a lower dimensional manifold as is the case with the distribution of genre data. Given the time constraints, these weren't implemented for the purposes of this evaluation project.

The matrix completion errors were computed before and after clustering (and subsequent separation of genre). No significant improvement was observed in both the cases of the medium and the pseudo large dataset. An important point to note is that the success of completion algorithm is dependent on the rank estimate of the solution, in our case, this estimate will not be very good , because the clusters aren't well separated.

References

[1] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[2] T. Hastie, R. Mazumder, J. D. Lee, and R. Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *J. Mach. Learn. Res*, 16(1):3367–3402, 2015.

[3] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

[4] N. Srebro, J. D. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, volume 17, pages 1329–1336, 2004.

[5] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.