Matrix Completion Using Soft-SVD and Comparison With Other Algorithms

FRE9733 Statistical Analysis of Financial Data

Hatim Kagalwala

Problem Background

- There are many instances when we are interested in predicting user preferences, such as online movie/music recommender systems, news personalization, food ordering apps, etc.

- We have a Matrix X (n_1 x n_2) and only a sample of m entries of X are observed. We seek to recover the remaining entries and complete the matrix.

Matrix Completion in Practice

- Scenario 1: Public survey
- Scenario 2: Online shopping websites.
- Scenario 3: Image/Video completion

Modeling Assumptions

Although the matrix completion problem seems simple in statement but it is very hard to solve in practice as the missing entries could be arbitrarily anything. To proceed, some deep assumptions have to made about the data.

Low Rank – We impose a structure on X which assumes that it is low rank.

Additional requirements

- In order to complete the matrix, the number of entries of matrix X given must obey (Candes & Recht [2008])

$$m \geq Cn^{1.2}rlog(n)$$

  m is the number of observed entries
  C is some positive constant
  Matrix X is n x n and rank r.

- Incoherence condition – The coherence of a matrix is the largest (absolute) cross-relation between columns of matrix. An incoherent matrix has a small value for this measure, which corresponds to sufficiently spread out singular values.

Notations

- We denote by $\boldsymbol{\Omega}$ the location of the observed entries m, i.e. (i, j) $\in \boldsymbol{\Omega}$ if $X_{(i,\ j)}$ is observed.
- We also use the set projection operator $P_{\boldsymbol{\Omega}}$, where $P_{\boldsymbol{\Omega}}(X)$ is the projection of X onto the set $\boldsymbol{\Omega}$. This maps the elements m of matrix X to the set $\boldsymbol{\Omega}$.

$$P_{\boldsymbol{\Omega}}(X) = \begin{cases} X_{(i,j)} \ if \ (i,j) \ is \ observed \\ \quad 0 \quad if \ (i,j) \ is \ missing \end{cases}$$

Mathematical Formulation (Optimization)

- Given that all conditions have been met, the following optimization can be solved to recover the completed low-rank matrix Z, given X

$$\min rank(Z), s.t. Z_{(i,j)} = X_{(i,j)}, (i,j) \in \Omega$$

- However, the rank(Z) constraint makes this problem non-convex. Unfortunately, this problem is NP-hard. This means that the algorithm required to solve them would be too expensive to scale to any substantial size. In particular, all known algorithms with exact solution to this problem require doubly exponential in n and hence are not practical for use (especially for large datasets).

Contd…

- Hence, we consider the alternative nuclear norm minimization problem

$$\min ||Z||_* , s.t. Z_{(i,j)} = X_{(i,j)}, (i,j) \in \Omega$$

- This function is convex in Z. $||Z||_*$ is defined as the nuclear norm of matrix Z. $||Z||_* = \sum_{j=1}^{k} \sigma_j(Z)$, or the sum of the *kth* largest singular values of Z.
- The objective function can be rephrased as follows

$$\min ||Z||_*, s.t. P_\Omega(Z) = P_\Omega(X)$$

- However, the above no noise construction seems too rigid and result in overfitting. Cai et al. (2008) propose a singular value thresholding (SVT) algorithm scalable to large matrices.
- We can again modify and solve the following optimization problem,

$$\min ||Z||_* \,, s.t. ||P_\Omega(X) - P_\Omega(Z)||_F \leq \delta$$

- $\delta \geq 0$ is a regularization parameter controlling the tolerance in training error
- In real-life datasets, there is noise and δ incorporates this noise in our model.

Soft SVD

- Consider the SVD of (fully observed) matrix X_{nxm}

$$X = U.\operatorname{diag}[\sigma_1, \dots, \sigma_m].V'$$

- Now consider the convex optimization problem

$$\min_{Z} \frac{1}{2} ||X - Z||_F^2 + \lambda ||Z||_*$$

- Solution is soft-thresholded SVD

$$S_\lambda(X) := U.\operatorname{diag}[(\sigma_1 - \lambda)_+, \dots, (\sigma_m - \lambda)_+].V'$$

- This is like Lasso for SVD with many singular values shrunk to zero. It is a smooth version of best-rank approximation.

Missing Data Problem

- Now coming back to the missing data problem, in Langrangian form we have

$$\min_{Z} \frac{1}{2} ||P_{\Omega}(X) - P_{\Omega}(Z)||_F^2 + \lambda ||Z||_*$$

- λ is a regularization parameter controlling the nuclear norm of the minimizer $\hat{Z}_{\lambda}$.
- This is semi-definite program (SDP) convex in Z.
- This can be solved using iterative soft SVD with cost per soft SVD $O[(m+n).r + |\Omega|]$ where r is the rank of the solution.

Soft Impute Algorithm

- Create a grid Λ with decreasing values of λ. $\lambda_0 > \lambda_1 >\ldots> \lambda_k > 0$ and initialize $Z^{old} = 0$.
- For each value of λ, repeat a-b till convergence.
  - *a)* $Z^{new} \leftarrow S_\lambda\left(P_\Omega(X) + P_\Omega^\perp(Z^{old})\right)$
  - *b)* $Z^{new} \leftarrow Z^{old}$ and go to above step
  - *c)* $\hat{Z}_\lambda \leftarrow Z^{new}$ and go to the next value of lambda.
- Output the sequence of solutions $\hat{Z}_{\lambda_1}, \ldots, \hat{Z}_{\lambda_k}$.
- The convergence criteria is as follows

$$\frac{||P_\Omega(X - Z)||_F}{||P_\Omega(X)||_F} \leq \varepsilon$$

Application to MovieLens Dataset.

- The dataset considered here is the movie ratings dataset collected by GroupLens Research Project at University of Minnesota. There are a total of 671 users and 9066 movies. Each user at an average has rated 20 movies. About 100,004 ratings are observed and 5,983,282 ratings are missing. We wish to complete the matrix by recovering those missing ratings.
- The computations have been done using the **fancyimpute** package in Python. The results of different algorithms are compared in the next slide.

Results

Method	RMSE
SimpleFill based on Column Mean	0.9436
SimpleFill based on Column Median	0.970819
SimpleFill based on Random	1.2734
KNN (k=20)	1.0534
SoftImpute	1.099
MatrixFactorization	2.10262
SimilarityWeightedAveraging	0.967298

$$RMSE = || \frac{1}{|\Omega|} P_{\Omega}(X - Z) ||_F$$

Conclusion

- It can be seen that SoftImpute achieves an RMSE of approx. 1.099. We can achieve lower RMSE using cross validation by tuning the parameters, but the algorithm is too slow and takes time to solve one pass. One more reason could be that we may likely lose some meaningful signals by assuming a low rank matrix.
- The SimpleFill based on mean outperforms SoftImpute because SoftImpute tries to recover the underlying matrix under the assumption of low rank whereas SimpleFill just replaces the missing values by the column means.
- If any of these algorithms were to be put into production, the ideal thing would be to divide the data into training set and test set. Then optimize the parameters to find the lowest RMSE on test set. The bias-variance tradeoff tells us that RMSE may keep on decreasing on the training set as our model becomes more and more complex but may not be able to generalize well.