CMSC35300 Final Project: Non-Matrix Factorization For Yelp's Recommendation System

Aya Liu; Ta-Yun Yang; Yuwei Zhang
Department of Computer Science
Harris School of Public Policy
the University of Chicago
Chicago, IL 60637
ayaliu@uchicago.edu; tayuny@uchicago.edu; yuweiz@uchicago.edu

1 Introduction

A recommendation system predicts a user's rating or preference on an item and is widely used in commercial applications like Yelp for targeted marketing. In this project, we explore if decomposing Yelp's business ratings into representative businesses profiles finds the representative businesses that are most predictive of user ratings. In particular, we examine if these most predictive representative businesses for different types (e.g. Chinese, Mexican, Italian restaurants) are distinct from each other. If so, the system can take advantage of those distinctions to make recommendations more tailored to each business type.

From our analysis, we find that Chinese, Mexican, and Italian businesses 1) share the same most important representative business predictor and 2) have different second and third most important representative businesses. However, because the second and third best predictors only marginally improve the models, tailoring recommendations to different business types likely would not improve those recommendations substantially.

2 Literature Review

Truncated singular value decomposition (SVD) is a commonly used algorithm in implementation of recommendation systems. Sawant and Pai (2013) complete the matrix X by filling in the blank space with row means, compute the complete matrix's informative SVD, and further reduce it to its best rank-k approximation $\hat{X}$. This $\hat{X}$ contains the observed and predicted ratings for each user/business pair that can be used to make user recommendations. Non-negative matrix factorization (NMF) completes the matrix in a different way. Using a two-block coordinate descent algorithm (Colyer, 2019), NMF decomposes matrix X into two non-negative matrices W and H such that $X \approx WH$.

Despite the higher accuracy of truncated SVD, this approach has two limitations. First, it assumes that each observation can only be clustered in one cluster. Second, it can only recover the span of eigenvectors rather than the eigenvectors themselves (Arora, Ge et.al, 2012). By contrast, NMF has a fairly mild assumption about the data so that it can deal with more sparse data matrix. Also, it is easier to understand the decomposition of NMF since it extracts information from non-negative vectors.

3 Data and Methodology

3.1 Data

We use Yelp's user, business, and review data from the Yelp Dataset Challenge (2019), all of which can be linked with unique IDs. This data package includes a subset of Yelp's commercial data, and we pair down our training sample further by focusing on businesses with over 20 users' ratings in order to avoid excessive sparsity. Eventually, we obtain a matrix with 1,100 businesses and 75,641 users. Each entry in this matrix is a user's rating for a business.

3.2 Methodology

3.2.1 Non-negative Matrix Factorization (NMF)

NMF is a decomposition approach to obtain intelligible decomposing matrices that can still construct a fairly good approximation to the original matrix. Although truncated Singular Value Decomposition can give the best subspace approximation in terms of Forbenius norm, U, Σ, V^T do not allow for direct interpretations. On the other hand, the two matrices with non-negative entries from NMF, W and H, offer meaningful insights on users and businesses that can be implemented in a recommendation system or other market segmentation practices.

In this project, we begin with a matrix $X \in R^{75,641 \times 1100}$, where rows are users, columns are businesses, and entries are ratings. We use NMF to decompose X into two non-negative matrices, so that $X \approx WH$. First, we get matrix $W \in R^{75,641 \times k}$ where rows are users and columns are k representative business profiles. Entries in W are user ratings for representative businesses, which are linear combinations of user ratings for all actual businesses. Second, we get matrix $H \in R^{k \times 1,100}$, where rows are representative businesses and columns are actual businesses. Since entries of H are all non-negative, they can be interpreted as the weights of actual business in each representative business profile. For the purpose of this project, we focus on interpreting matrix W using the methodology in the next subsection.

To implement NMF, we rely on the Two-Block Coordinate Descent algorithm:

1) We initialize W and H using truncated SVD. $W = U \times \Sigma$ and $H = V^T$. Then, for each k = 0, 1, 2,... we do:

2) Update W while fixing H until $|| \; W^{(k,l+1)} - W^{(k,l)} \; ||_F \leq \epsilon \; || \; W^{(k,1)} - W^{(k,0)} \; ||_F$, where $W^{(k,l)}$ is the iterate after l updates of $W^{(k)}$ (while $H^{(k)}$ is being fixed).

$W^{(k)} = W^{(k-1)} - \tau \odot [WHH^T - XH^T]$
while $\tau = \frac{W}{WHH^T}$, this formula is equivalent to multiplicative update rule.

3) Update H while fixing W until $|| \; H^{(k,l+1)} - H^{(k,l)} \; ||_F \leq \epsilon \; || \; H^{(k,1)} - H^{(k,0)} \; ||_F$.
$H^{(k)} = H^{(k-1)} - \eta \odot [W^T WH - W^T X]$
while $\eta = \frac{H}{WW^T H}$, this formula is equivalent to multiplicative update rule.

We also set up parameters for max iteration number and manual step size.

3.2.2 Ridge Regression

Given W, we are interested in whether certain combinations of ratings for representative businesses can successfully predict a user's average rating for a particular type of business, such as Chinese businesses, Mexican businesses and Italian businesses.

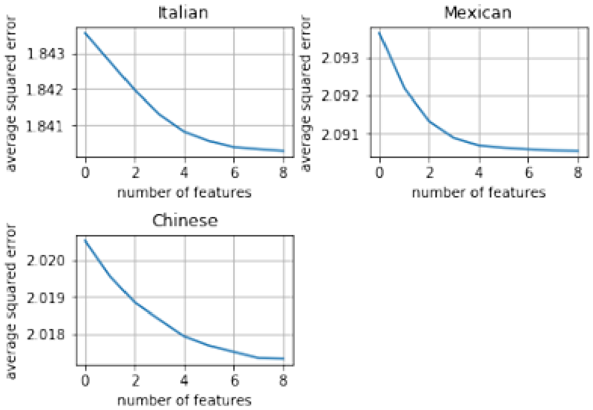
After obtaining W from NMF decomposition, we search through all combinations of columns of W to train $2^k - 1$ models using ridge regression for each business type ($\lambda = 0.01$). We identify the optimal model with the lowest squared errors for each business type. Then, we compare optimal models for the three business types to see whether the combinations of representative businesses and coefficients that best predict Chinese, Mexican, and Italian businesses are different. If they are, we may be able to make tailored recommendations for a specific business type using the most relevant information.

Each regression model has a feature matrix $\widetilde{W} \in R^{75,641 \times p}$ and a weight vector $z \in R^p$, where p is the number of representative businesses used as features. The label $y \in R^{75,641}$ is a vector of a user's average rating for a particular type of business. If a user has no rating for a certain type, then this user is dropped.

$$z = \widetilde{V}(\widetilde{\Sigma}^T\widetilde{\Sigma} + \lambda I)^{-1}\widetilde{\Sigma}^T\widetilde{U}^T y, \quad \text{where } \widetilde{W} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$$

4 Results and Discussion

Using SVD, we observe that the first singular value of X is much bigger than the others, and the following non-zero singular values are close to each other. Given that, we conservatively choose 10 as the reduced dimension number. After obtaining W, we run 1,023 ($2^{10} - 1 = 1023$) regression models for each business type, and then select the optimal model for each type using fewer than or equal to 3 features. We limit feature number to 3 because when k = 3, the average squared error has an significant drop when plotting the average squared error for k= 1, 2, ...,10. Therefore, adding more features into regression model would not improve the performance.



Here are the results of optimal models:

Business Type	Feature Index (k =1)	Avg. Squared Error	Feature Index (k <= 3)	Avg. Squared Error
Chinese Business	0	2.0205	0, 4, 8	2.0188
Mexican Business	0	2.0936	0, 7, 8	2.0913
Italian Business	0	1.8436	0, 5, 7	1.8420

From the table above, we can learn that:

- The first representative business can be a good predictor for all business types. It corresponds to the biggest singular value, despite the fact that it has been adjusted by NMF.
- Adding more features to the model does not reduce the average squared error significantly.
- Besides the first representative business, optimal models for Chinese businesses and Italian businesses share one other common feature (the ninth representative business). Optimal models for Mexican businesses and Italian businesses share one other common feature (the seventh representative business).

- Each type has a unique representative business feature in their optimal model. This indicates some distinction between the different business types, but the marginal effect of those distinctions on reducing errors are not substantial.

Acknowledgments

We would like to express our sincere gratitude to Dr. Rebecca Willett, Ruoxi Jiang, Lang Yu, Zhuokai Zhao, Yuhao Zhou, Takintayo Akinbiyi, Bumeng Zhuo for your teaching, guidance, and support throughout the quarter.

References

[1]Adrian Colyer(2019, Feb 18). *The why and how of nonnegative matrix factorization*. Retrived from https://blog.acolyer.org/2019/02/18/the-why-and-how-of-nonnegative-matrix-factorization/
[2]Arora, S., Ge, R., Moitra, A. (2012, October). *tLearning topic models–going beyond SVD*. In 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (pp. 1-10). IEEE.
[3] Asghar, N. (2016). *Yelp dataset challenge: Review rating prediction*. arXiv preprint arXiv:1605.05362.
[4] Gillis, N., & Glineur, F. (2012). *Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization*. Neural computation, 24(4), 1085-1105.
[5]Keita Kurita (2017, Dec 28). *A Practical Introduction to NMF*. Retrieved from https://mlexplained.com/2017/12/28/a-practical-introduction-to-nmf-nonnegative-matrix-factorization/
[6]Piotr Gabrys(2018, Nov 13). *Non-negative matrix factorization for recommendation systems*. Retrieved from https://medium.com/logicai/non-negative-matrix-factorization-for-recommendation-systems-985ca8d5c16c
[7] Suykens, J. A., Signoretto, M., & Argyriou, A. (Eds.). (2014). *Regularization, optition, kernels, and support vector machines*. CRC Press.
[8] Sawant, S., Pai, G. (2013). *Yelp food recommendation system*.

Appendix