

**MAT248 - Applied Linear Algebra**

Section-2

Instructed by: Professor Eshita Mazumdar

| Group-17 |
| --- |

| Bhavya Khakhar | AU2140043 |
| --- | --- |
| Harsh Choksi | AU2140061 |
| Krutarth Vora | AU2140162 |
| Harsh Pandya | AU2140171 |
| Dhruvi Rajput | AU2140224 |

**Problem Statement**

Creating a collaborative recommendation system utilising matrix factorization that is scalable in the presence of missing data, and that can accurately predict user preferences and offer the right products to users.

**Movie Recommendation System Using Linear Algebra**

**Introduction :**

A recommendation system using matrix factorization is a powerful tool used to make personalised recommendations to users of an online system.In order to make personalised recommendations for each user, it uses matrix factorization to gather information about their preferences from previous encounters with the system. Matrix factorization is an effective technique for uncovering latent relationships between users and items, and can be used to identify popular items and predict user preferences. Recommendation systems improve user engagement, and produce more meaningful user experiences.

**Collaborative based filtering recommendation system :**

It is based on the notion that people who share an interest in one sort of thing will also share that interest in a similar type of thing. Contrast this with content-based methods, which mostly rely on information while addressing actual behaviour. We will be able to reduce data sparsity with the help of this kind of filtering with matrix factorization.

**Matrix Factorization :**

The latent features are produced by multiplying two distinct forms of entities. Matrix factorization is used in collaborative filtering to determine the connection between the entities of items and users. We would like to forecast user ratings of store items using the input of user ratings so that users can receive recommendations based on the prediction.

**Use of Linear Algebra (Approach) :**

Define a collection of Users (U), movie ratings (M), and the R size of |U| and |M|. The matrix |U|\*|M| contains all of the user ratings. The aim is to find K hidden features. Given the input matrices P (|U|\*k) and Q (|M|\*k), it would yield the product result R.

R ≈ P **=**

The dot product of the two vectors corresponding to u\_i and m\_j yields the prediction of an item's rating.

ij **=** qj **=** pik qkj

The difference is then minimised through iterations. The approach is known as gradient descent, and it aims to identify a local minimum of the difference.

= (- )2 = (-pik qkj)2

The gradient has the ability to minimise error, thus we differentiate the above equation with regard to these two variables individually.

( ) = -2(ij)()=-2 ( )

( ) = -2(ij)()=-2 ( )

Updated and will be:-

= + = + 2

=+ ( ) = +2

The value of the constant alpha() controls how quickly the minimum is approached.

Let D be a collection of Tuples, each in the form (u_i, d_j, r_{ij}),such that it contains all pairs of user-items and their corresponding ratings.

E=(u_i, d_j, r_{ij})D =(u_i, d_j, r_{ij})D ( - )2

Using the above equation,we can calculate the overall error.

**Truncated Singular Value Decomposition:**

SVD factorised data matrix that has been truncated, with the number of columns being the same as the truncation. For mathematically shorting the value of float digits, it eliminates the digits following the decimal point. For instance, 2.498 can be cut off at 2.5.

While a standard SVD technique would result in matrices with m columns, a given m x n matrix truncated SVD would yield matrices with the specified number of columns. It implies that all characteristics other than the ones already granted to it will be dropped.

A fast randomised SVD solver and a "naive" approach that uses ARPACK as an eigensolver on X \* X.T or X.T \* X, whichever is most effective, are the two algorithms that this estimator supports.

**Regularisation :**

Regularisation terms are often added to the cost function to penalize the complexity of the model.

=( - )2 + ((||P||2 + ||Q||2))

The user-feature and item-feature vectors' magnitudes are controlled by the new parameter beta in such a way that P and Q can approximate R well without needing to contain big values.

Then the new updated and will be:-

= + = + -)

=+ ( ) = +2 -)

**Result :**

As a result, utilising this idea, we may build a useful system called a recommendation system that mostly involves linear algebra principles like matrix factorization and singular value decomposition (SVD). This project is built on a system for recommending movies based on user ratings, but it can also be applied to any similar categories that require algorithms for user choices or suggestions.

**Google Colab Link :**

<https://colab.research.google.com/drive/1gIl41KwcgKCz7KvM3YtLJsUVhhPZibF5#scrollTo=AWjv8Oxax77I>

**References**

Au, A. (2010, September 16). *Matrix Factorization: A Simple Tutorial and Implementation in Python @ quuxlabs*. Quuxlabs. Retrieved December 7, 2022, from <http://www.quuxlabs.com/blog/2010/09/matrix-factorization-a-simple-tutorial-and-implementation-in-python/>

Verma, Y. (2021, July 10). *Beginners Guide To Truncated SVD For Dimensionality Reduction*. Analytics India Magazine. Retrieved December 7, 2022, from <https://analyticsindiamag.com/beginners-guide-to-truncated-svd-for-dimensionality-reduction/>