Movie Recommendation using SVD

Brian, Darsh, Maggie, Richa, Shubham

Introduction/Aim of Project

- **Movie Recommendation Lab:** We want to recommend a movie to a user, or know whether a user will like a certain movie, based on how they have rated previous movies.
- (Movie, user) ratings can be represented by points in a matrix where the rows are the users, columns are the movies and ratings are the values.
- However, much of this matrix is sparse, because not all users have rated every movie
  - Need to find way to predict values of missing
- Singular value decomposition is a common method of Matrix factorization for recommendation systems
  - The model finds an association between the users and movies based on past ratings
  - Then it predicts the rating for the item in which the user may be interested

Hypothesis

Performing Singular Value Decomposition(SVD) on a full dataset is beneficial for the accuracy of ratings of users on movies, in comparison to replacing null values with the movie's average rating and then performing SVD on a subset of movies.

Different Approaches to SVD matrix factorization:

- **Method 1:** Replace null values in the matrix with the average of that movie's existing ratings. Then, Utilize Singular Value Decomposition (SVD) on a smaller subset of the ratings.
- **Method 2:** Utilize Stochastic Gradient Descent (SGD) to replace the null values and then SVD on the full matrix.
- To properly evaluate our methods, we will remove 10 points from the matrix, and replace them with null values
- Then we will use each separate method to replace the null values
- Perform SVD on each matrix
- Compare which method predicted these 10 ratings more accurately

Method 1

- We used a matrix including the 100 movies with the most ratings, not necessarily the best ratings
- We removed the following 10 ratings from the matrix, and replaced them with NaN

UserID	4	5	6	15	18	22	33	37	49	52
MovieID	3763	1380	1234	260	1544	3462	2396	480	1097	589
Movie Name	F/X	Grease	The Sting	Star Wars: Episode IV	Lost World: Jurassic Park	Modern Times	Shakespeare in Love	Jurassic Park	E.T. The extra terrestrial	Terminator 2: Judgement Day
Actual Rating	3.0	4.0	4.0	5.0	4.0	5.0	4.0	3.0	4.0	5.0
Average Rating of that Movie	3.649	3.588	4.392	4.526	3.039	4.309	4.115	3.917	4.125	3.985

Reconstruction Method 1

$$A = USV^T$$

Original Matrix → **Matrix with Null Values Replaced** → **Extracted values from Reconstructed Matrix**

```
movieID  1    2    3    4    5    6    7    9    17   19   ... \
userID                                                     ...
4        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
5        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
6        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
15       NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
18       NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
...      ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
6024     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
6025     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
6029     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
6030     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...
6038     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...

movieID  3936 3937 3938 3943 3944 3946 3948 3949 3950 3952
userID
4        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
5        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
6        NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
15       NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
18       NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
...      ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
6024     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
6025     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
6029     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
6030     NaN  NaN  NaN  NaN  NaN  NaN  4.0  NaN  NaN  NaN
6038     NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN
```

```
movieID         1         6    21        45        47
userID
4        4.333333  4.108696   3.8  2.948718  4.081633
5        4.333333  4.108696   3.8  2.948718  4.081633
6        4.333333  4.108696   3.8  2.948718  4.081633
15       4.333333  4.108696   3.8  4.000000  5.000000
18       4.333333  4.108696   3.8  2.948718  4.081633
...           ...       ...   ...       ...       ...
6024     4.333333  4.108696   3.8  2.948718  4.081633
6025     4.333333  4.108696   3.8  2.948718  4.081633
6029     4.333333  4.108696   3.8  2.948718  4.081633
6030     4.333333  4.108696   3.8  2.948718  4.081633
6038     4.333333  4.108696   3.8  2.948718  4.081633
```

```python
#svd matrix values for the 10 removed:
type(svd_matrix)
print(svd_matrix.iloc[4][3763])
print(svd_matrix.iloc[5][1380])
print(svd_matrix.iloc[6][1234])
print(svd_matrix.iloc[15][260])
print(svd_matrix.iloc[18][1544])
print(svd_matrix.iloc[22][3763])
print(svd_matrix.iloc[33][2396])
print(svd_matrix.iloc[37][480])
print(svd_matrix.iloc[49][1097])
print(svd_matrix.iloc[52][589])
```

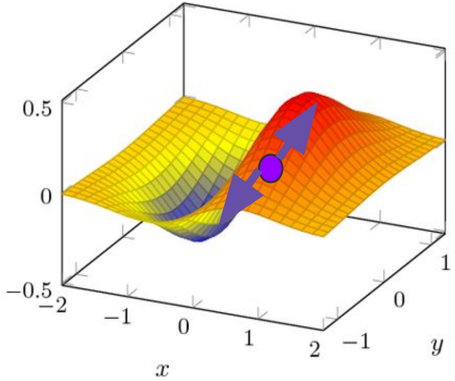
```
3.650084740676425
3.586349603139138
4.400038673825473
4.5239231637887025
3.038399252373335
3.63628203353962
4.13092840305752
3.8999283419067083
4.12283250970689
3.9831878058504286
```

Method 1 Predictions of our (user, movie) combos

UserID	4	5	6	15	18	22	33	37	49	52
MovieID	3763	1380	1234	260	1544	3462	2396	480	1097	589
Movie Name	F/X	Grease	The Sting	Star Wars: Episode IV	Lost World: Jurassic Park	Modern Times	Shakespe are in Love	Jurassic Park	E.T. The extra terrestrial	Terminator 2: Judgemen t Day
Actual Rating	3.0	4.0	4.0	5.0	4.0	5.0	4.0	3.0	4.0	5.0
Average Rating of that Movie	3.649	3.588	4.392	4.526	3.039	4.309	4.115	3.917	4.125	3.985
Method 1 Prediction	3.65	3.586	4.4	4.524	3.038	3.636	4.131	3.899	4.123	3.983

Note: Method 1 Prediction is heavily influenced by the average rating of the movie.

Method 2: Matrix factorisation using SGD



- We use SGD to create the vectors p and q.
- Weights matrix is dot product of p and q vectors, this produces a matrix, W, that connects Movie and user data.
- Formula for SGD: $W_i = W_i - lr(\nabla_W \text{Loss})$, where W is the matrix with null values, $lr = 0.001$, $\nabla_W \text{Loss} = \text{rmse}(\text{input matrix}, W_i)$
- Essentially, every iteration of SGD, i'th iteration is a step towards the minima where the loss is minimum.
- Size of the step that is taken is lr.

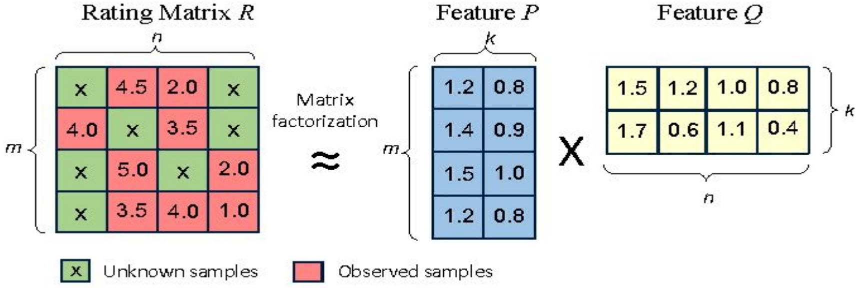
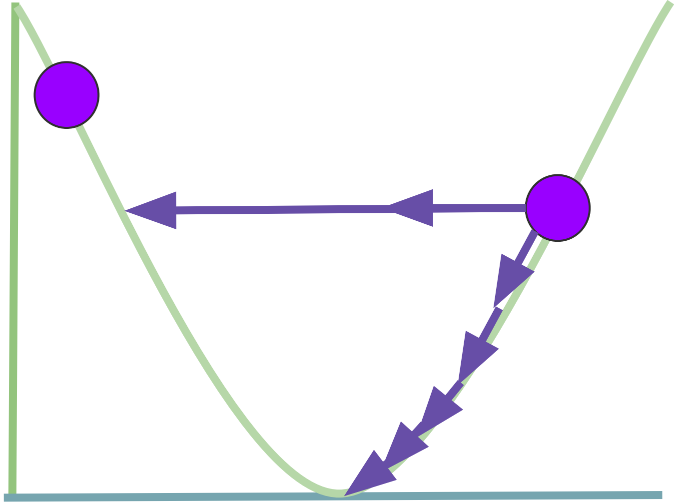


Figure 1: Matrix factorization, m=4, n=4, k=2.



Reconstruction Method 2

Original Matrix → Matrix with Null Values Replaced → Extracted values from Reconstructed Matrix

Original Matrix

movieID userID	1	2	3	4	5	6	7	8
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	4.0	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN	5.0	NaN	NaN
17	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
6025	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6027	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6029	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6030	NaN	NaN	3.0	NaN	3.0	NaN	NaN	NaN
6038	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

2353 rows × 1465 columns

Matrix with Null Values Replaced

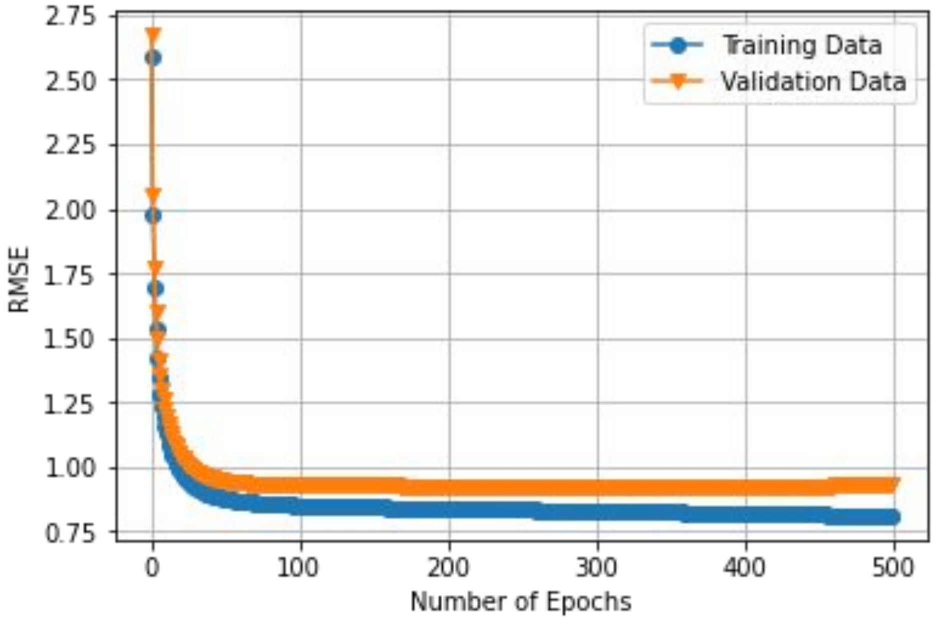
userID	1	2	3	4	5	6	7	8
4	4.117361	3.130594	2.754825	2.481583	2.741904	3.728677	3.298964	3.756266
5	4.713221	3.244251	2.930516	1.761217	3.810450	3.822011	4.784862	2.931872
6	4.567372	3.194773	3.025605	2.090041	3.662915	4.000000	4.216108	3.190689
15	4.705058	3.483709	3.192952	2.692446	3.368536	5.000000	3.860008	4.015518
17	4.816594	3.783636	2.911864	2.721473	2.784882	4.326639	4.189280	4.513391
...	...	...	...	...	...	...	...	...
6025	4.569637	3.005563	3.178447	1.886385	4.171588	3.735788	4.295740	2.685420
6027	4.794953	3.515605	3.314773	2.755127	3.540545	4.266982	3.888914	4.031725
6029	6.889036	5.380136	4.383191	4.139833	4.157152	6.266178	5.662447	6.555022
6030	4.393095	3.223731	3.000000	2.449380	3.000000	3.884294	3.656553	3.650448
6038	3.108311	2.188036	2.084814	1.490606	2.473165	2.644367	2.798672	2.242990

2353 rows × 1465 columns

Extracted values from Reconstructed Matrix

```
print(results.loc[52, '589'])
```

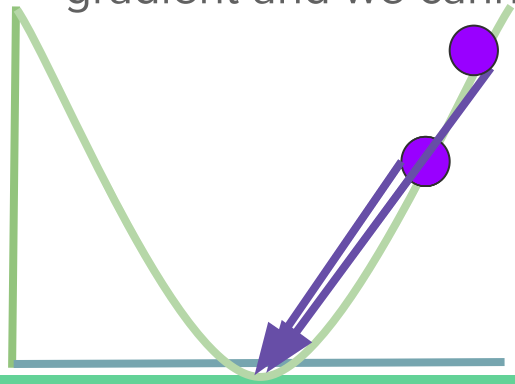
```
4.759251610268691
3.791587478839326
3.269501626453873
4.912319669022091
3.581086877594935
3.781532485585248
3.0268310543453536
2.51840460581563
3.4134458244366144
3.264102372884025
```

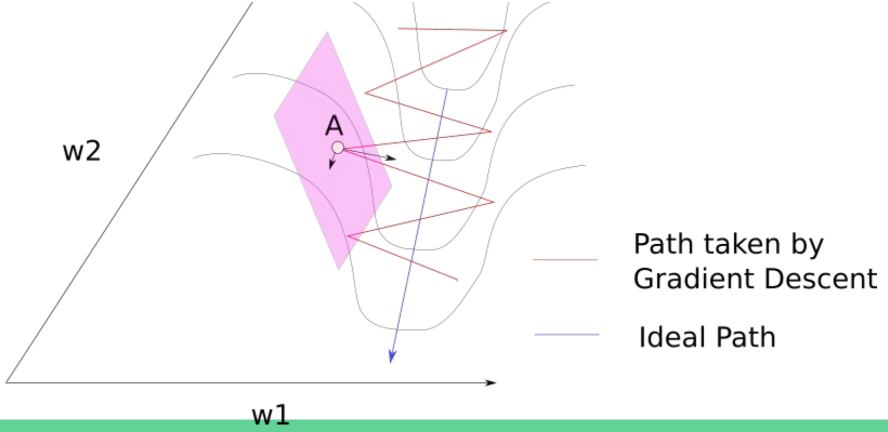


2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
RMSE
0
100
200
300
400
500
Number of Epochs
Training Data
Validation Data

SGD and Overshooting: Problems/How to make it better

- The downside of this is that it can continuously overshooting if one does not reduce the learning rate properly.
- In actual use cases, SGD has to be coupled with a decaying learning rate function
- A random initialization of vectors p and q affects how far the algorithm starts from the actual minima which affects the time and space complexity.
- Loss function needs to be differentiable, if it not then we cannot find the gradient and we cannot perform SGD.





Comparing predictions from Method 1 and Method 2

Movie Name	F/X	Grease	The Sting	Star Wars: Episode IV	Lost World: Jurassic Park	Modern Times	Shakespeare are in Love	Jurassic Park	E.T. The extra terrestrial	Terminator 2: Judgement Day
Actual Rating	3.0	4.0	4.0	5.0	4.0	5.0	4.0	3.0	4.0	5.0
Method 1 Prediction	3.65	3.586	4.4	4.524	3.038	3.636	4.131	3.899	4.123	3.983
Method 2 Prediction	3.225	2.344	4.023	4.912	3.805	4.762	3.054	2.518	3.413	3.264

Method 1: SSE = 5.641192
RMSE: 0.75108

Method 2: SSE = 7.3814
RMSE = 0.85915

Conclusion

- Method 2 (using SGD to replace nulls) is closer more of the time, but when it is wrong, it is very wrong
- When method 1 (replacing nulls with average rating) was farther from the actual rating, it was not too far off
    - Important to note that method 1 only used 100 values, whereas method 2 looked at entire dataset
- The added value of entire dataset and the intricacy of SGD is minimal and leads to poorer generalization
    - Is this because of size of dataset or method of replacing null values?
- Do we really need massive data analysis methods or is it better to just pick a set of movies?

References

- https://towardsdatascience.com/recommender-system-singular-value-decomposition-svd-truncated-svd-97096338f361#:~:text=Singular%20value%20decomposition%20(SVD)%20is%20a%20collaborative%20filtering%20method%20for,factor%20model%20for%20matrix%20factorization.
- https://www.kdd.org/exploration_files/simon-funk-explorations.pdf