Application of Singular Value Decomposition to Collaborative Filtering

Edoardo Fantolino
s286008
Politecnico di Torino
s286008@studenti.polito.it

Abstract

In this essay I exploit the potential of the Singular Value Decomposition. I will apply the results to make predictive recommendations. This document is addressed to Professor Stefano Berrone.

Keywords: predictive recommendations, Singular Value Decomposition, linear algebra.

Introduction

Nowadays, we are more and more involved in a digital world. Everyday, we collect terabytes of data and most of the time we upload them in the web. If there were not some algorithms that allow us to simplify our journey in the digital world, probably we would be lost in this huge amount of data. The strength of companies like Google, Amazon and Netflix lie in the power of their algorithms that are able to understand what a customer is interested in. There are a lot of techniques that improve the user experience like the famous Google PageRank and others like 'Collaborative Filtering' and 'Content Filtering'. All this mechanisms need a deep help of linear algebra. The first step is to understand what kind of data are useful in order to obtain a useful result. For example, Netflix constructs a matrix whose entries are the rank of a movie with respect to a certain user. This kind of information is used in Collaborative Filtering. They also have information about the film itself like: the actors, the director, the genres and the year of release. This data are related to Content Filtering. In this document we will concentrate in the SVD applied to Collaborative Filtering.

1 Structure of the data

Imagine that in our platform we have n users and m items. We will call the score matrix, the matrix that has as entries the rank that a user give to a certain product. So, our score matrix $A \in^{n \times m}$ is as follow:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

The column vectors contains the scores of a certain item while the rows represent the set of scores that a user gives to the items.

We can have $n > m$ that means more users than products or we can have $n < m$ and that mean more products than users. It is really difficult to find a practical case were $n = m$. Anyway our methods will work in every of the cases mentioned above.

In the real world, the score matrix will be a large sparse matrix and the companies will probably even use some techniques like Principal Components Analysis in order to reduce the number of features. Here, we will make some example with full matrices with fewer dimensions.

1.1 Observation over the score matrix

We can start analyzing our score matrix and make some useful observation. For example, given our score matrix, we want to understand how similar are two users. Let's pick user i and user j. I remember you that the score of a user are represented by the row of the matrix. We can use the euclidean distance as follow:

$$distance_{ij} = \sqrt{\sum_{k=1}^{n} (a_{ik} - a_{jk})^2}$$

If two user are similar the distance will be near zero, while if the two user have different taste and they like different items the distance between them will increase.

Another kind of measure that we can use to evaluate two customers is the cosine similarity. It is defined as:

$$similarity_{ij} = \frac{\sum_{k=1}^{n} a_{ik} a_{jk}}{\sqrt{\sum_{k=1}^{n} a_{ik}^2} \sqrt{\sum_{k=1}^{n} a_{jk}^2}}$$

The similarity is a value between -1 and 1. We obtain one when we compare two equal objects while -1 when the object compared are opposite. The values between this two threshold indicate intermediate similarity or dissimilarity.

2 Singular Value Decomposition

The Singular Value Decomposition allow us to factorize the score matrix as follow:

$$A = USV^T$$

$$U \in \mathbb{R}^{n \times n}, \quad S \in \mathbb{R}^{n \times m}, \quad V \in \mathbb{R}^{m \times m}$$

where U and V are called respectively: left singular vector matrix and right singular vector matrix. We know that U and S are orthogonal matrices. The matrix S contains on the main diagonal the singular values.

If we have the decomposition of the original matrix, we can compute a single entry of A as follow:

$$A_{ij} = \sum_{k}^{r} U_{ik} S_{kk} V_{jk}$$

where r is the minimum between n and m, $r = min(n, m)$.

2.1 Interpretation of U, S and V

This decomposition allow us to change prospective. Basically with the score matrix A we are able to see the rank of an item with respect to a user. With the SVD we introduce a new concept, we can call it feature or content. We can think about it in this terms: the item have some various contents and characteristics and the users react to this contents in different ways following their taste.

- U represents the user response to a feature
- S contains the overall importance of a feature
- V holds the amount of a feature in a give item.

We observe that we have more feature in matrix U with respect to V^T. This is because in matrix U are represented some feature with importance equal to zero. Then, we can assign a name and a meaning to this features, but this task become more difficult as the number of dimensions increase.

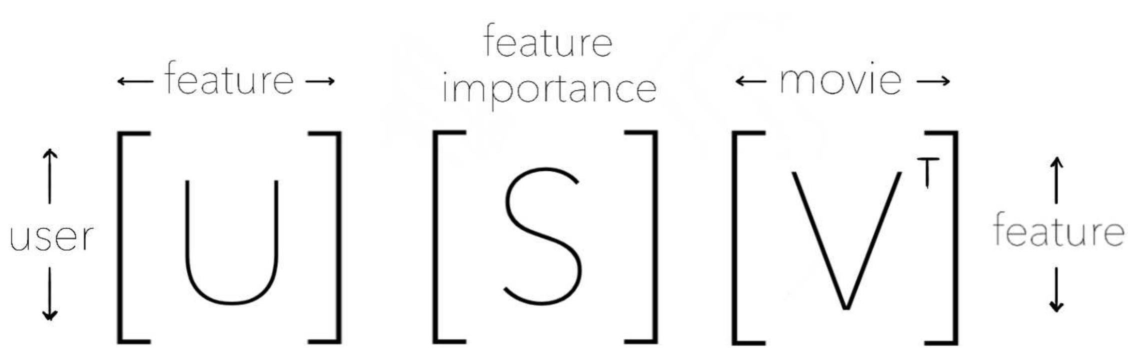

FIGURE 1: Structure and links behind the SVD

3 Practical example

I realized an example to show the potential of the SVD applied to Collaborative Filtering technique. Imagine that we have to manage a movies platform. Our objective is to make movies recommendations to new customers thanks to the information that we have and the information that we collect during the users experience of our service. We need to understand the customer taste in order to suggest to him some movie that he will probably like.

I asked some of my friends and family members to rate some famous movies. The title are: Mission Impossible, Harry Potter, Jack Reacher, The Lord of the Ring, Dunkirk and Saving Private Ryan. I choose this film because they are easy to classify in different groups. For example, in the list written above we have:

- 2 spy movies with weapons and adventure contents, both with the same star actor but with a slightly different character.
- 2 fantasy movies with a lot of magic features, fairy tales and wizard world.
- 2 war movies that deals with some battle of the WWII, so they are inspired by a true story.

Our score matrix is $A \in^{12\times6}$. The SVD will generate $U \in^{12\times12}$, $S \in^{12\times6}$ and $V \in^{6\times6}$. The score matrix obtained is the following:

A	$movie_1$	$movie_2$	$movie_3$	$movie_4$	$movie_5$	$movie_6$
$user_1$	2	9	3	3	3	9
$user_2$	2	8	2	1	3	9
$user_3$	4	9	8	2	7	8
$user_4$	2	9	7	3	8	9
$user_5$	7	2	4	8	4	1
$user_6$	8	1	3	8	4	2
$user_7$	8	2	2	8	1	3
$user_8$	7	2	2	9	3	1
$user_9$	2	1	8	2	9	2
$user_{10}$	1	3	9	2	8	1
$user_{11}$	1	3	8	2	8	2
$user_{12}$	2	3	7	2	9	2

Given that this matrix is not large, we can already start seeing some pattern. So, I ordered the movies by category and I grouped the users by taste. We ended up with 3 movies category and 4 user tastes. In the following pages I start analyzing the 3 matrices obtained with the SVD algorithm.

3.1 The Singular Values: analysis of S

After applying the SVD we can start observing the S matrix. We have more users than movies so $n > m$ and the S matrix will have a structure similar to:

$$S = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \\ 0_{n-m,1} & \cdots & \cdots & 0_{n-m,m} \\ \vdots & \ddots & \ddots & \vdots \\ 0_{n,1} & \cdots & \cdots & 0_{n,m} \end{bmatrix}$$

We can divide S into two parts. The upper part is a diagonal matrix and its entries will be in decreasing order in terms of magnitude.

$$\sigma_1 > \sigma_2 > \cdots > \sigma_m$$

The black spaces in the upper part of the matrix represent zero entries. In the lower part we will have a null matrix, $\varnothing \in \mathbb{R}^{(n-m)\times m}$. I plotted a graph that show the magnitude of the average importance of the feature.

This graph contains bars that represent how much is important a feature, while the line represent the cumulative sum of the terms. Everything sum to one because I normalized the data.

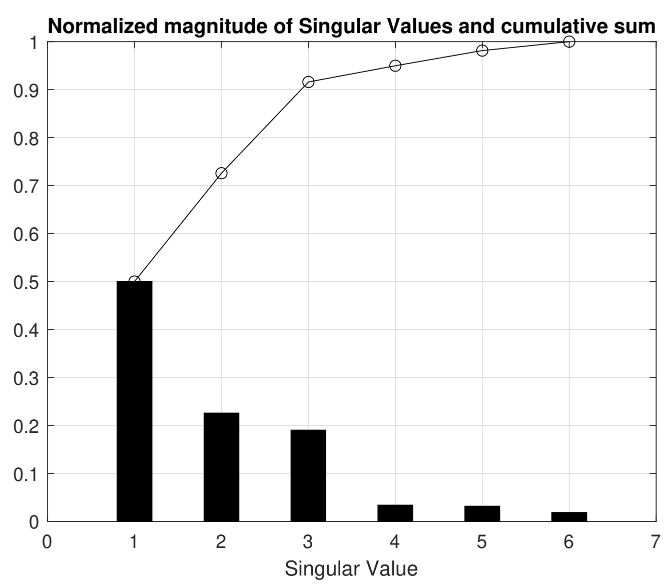

FIGURE 2: The bar are the magnitude of the singular values while the line with the circle is the cumulative sum

3.2 The Right Singular Vectors: analysis of V^T

The singular value decomposition generate the so called:"Right Singular Vectors" matrix. It is the matrix that multiply on the right the matrix of the singular values. In our case this matrix represent which features are present in a given movie and how much they are present.

Our score matrix has 6 films and so our algorithm find at most 6 different contents. We can observe the presence of each feature in a given movie in two perspective.

1. The presence of a feature in $Movie_i$:

$$V^T_{k,i}\ ,\ k = 1, 2, \cdots , m$$

2. The presence of a feature in $Movie_i$ with respect to the overall importance of that feature:

$$\bar{S}V^T_{k,i}\ ,\ k = 1, 2, \cdots , m \ \wedge \ \bar{S} = S^{m\times m}$$

In Fig.3, you can see the results for $Movie1$. We are tempted to say that the most important feature in the first movie is $feature_5$, but instead the most important feature is $feature_1$. $Feature_5$ represent the content most present in that particular movie but it is not the most important. Thanks to the analysis over S that we did before we can say that the most important feature for all the movies are the first three.

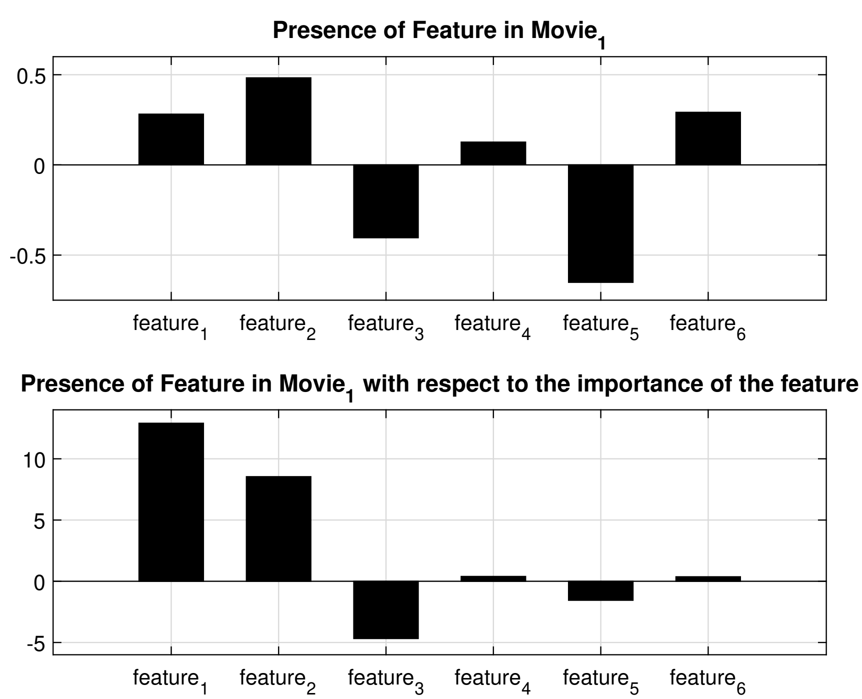

FIGURE 3: Graphical representation of the presence of a feature in $Movie_1$ with two different perspective

3.3 The Left Singular Vectors: analysis of U

The left singular vectors matrix is what we called U. It is the matrix that multiply the singular values matrix S on the left. In our practical case represent the reaction that a given user has with respect to a feature. As in section 3.2 we can observe the same concept with two perspective. The first most intuitive is to simply take the row of the matrix U, but we can even multiply U by S and obtain the response to a feature linked with the importance of that feature. We can do that because:

$$A = USV^T$$
$$V^T V = VV^T = I \Rightarrow V\ is\ orthogonal$$

So, we can multiply both the right hand side and the left hand side of the equation by V and we obtain:

$$AV = US(V^T V) \Rightarrow AV = USI$$
$$AV = US$$

As before, we can observe the same phenomenon. That means that feature 5 seems to be the most relevant characteristic in which a user is interested, but if we transform the matrix by the singular values matrix we will see that the first three features are the most significant.

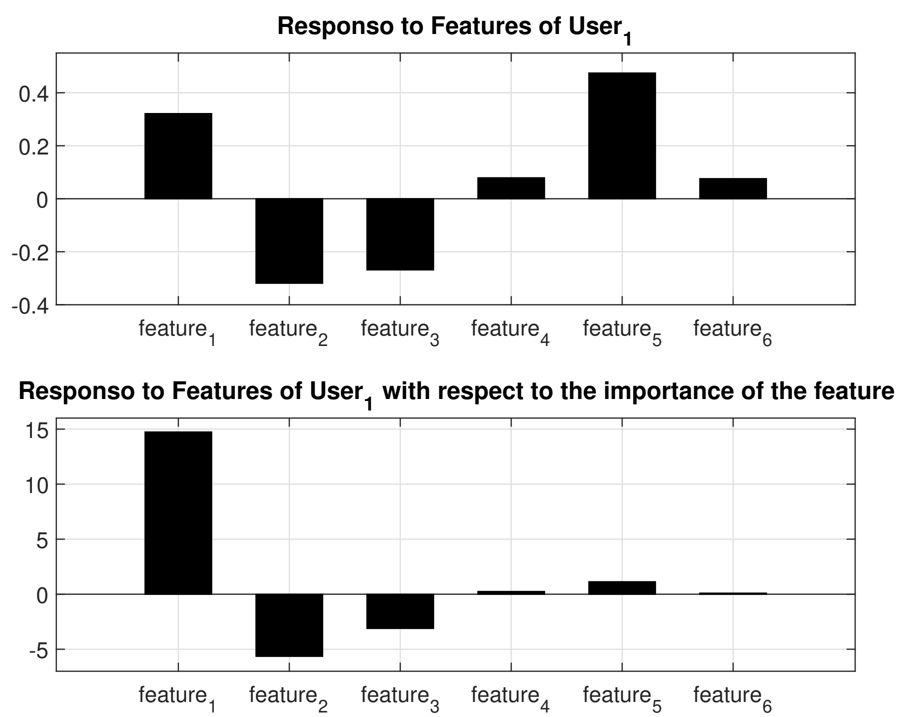

FIGURE 4: Graphical representation of reaction to a feature for $User_1$ with two different perspective

4 Feature interpretations

In this section I give an interpretation to the 3 most important features. So, I will plot the right singular vector for every movie and then we will compare the results.

As we can observe from Fig.5, we can group the movies into 3 main category:

- fantasy movies
- spy movies
- WWII movies

If we observe the first bar of each picture, we can see that the value is positive for every movie. So, I decided to call this characteristic: action. Action means movement, interesting story, nice scene, etc. It is not to interpret like action movie but more in a general way. The second bar is positive only for the fantasy movies so I think that this represent the presence of something like the magic contents. The last one that is negative for every movies with exception for the WWII movies, I decide to call it the "true content".

One of the power of this SVD applied to the score matrix is that we can understand in what the customer are interested in and in what they react most and how. So, that our platform can create new movies in function of the general customer taste and interest.

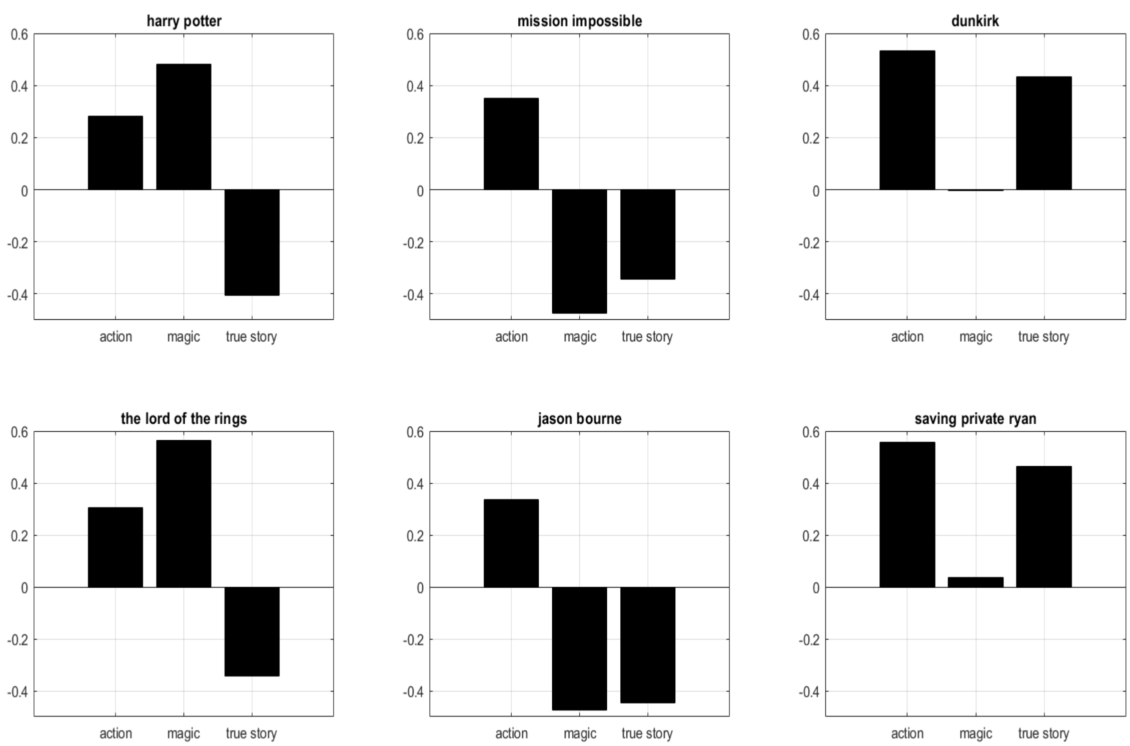

FIGURE 5: In this figure there are the 3 most important feature for every film- In the horizontal axis you can see the name given to the feature.

5 Application of SVD: recommendations and forecasts

Now, I will present some strategies in order to make predictions over the users.

5.1 Practical example

5.1.1 Day 1

A new customer arrive in our platform. In the evening she see $Movie1$ and she give 2 as score ($\mu_1 = 2$). Make some prediction and recommendation for the new customer.

We know that:

$$A_{ij} = \sum_{k}^{r} U_{ik} S_{kk} V_{jk}, \quad r = min(n, m)$$

Our unknown in this case is the respose of the new user to the different features. We can make an approximation of her reaction to $feature_1$ as:

$$u = \frac{\mu_1}{S_{11} V_{11}^T}$$

If we want to make a prediction over the score that she will give to the other movies we can calculate the following vector:

$$prediction = u S_{11} v_1, \quad v_1 \; = \; row\ vector\ of\ V^T$$

The obtained vector in our case is:

$$prediction_1 = [2.0000 \quad 2.6963 \quad 3.1054 \quad 2.1443 \quad 3.2814 \quad 2.5337]$$

As we can see the prediction don't seem to be reliable. This is due to the fact that we have few information about the new user.

5.1.2 Day 2

The next day, the new user watch $Movie_2$ and give a score of 9 ($\mu_2 = 9$). Adjust the prediction that you made before.

This time we have to solve a linear system.

$$Mu = b$$

Where M and b are:

$$M = \begin{bmatrix} S_{11} V_{11} & S_{22} V_{12} \\ S_{11} V_{21} & S_{22} V_{22} \end{bmatrix}, \quad b = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

If we solve the linear system we will obtain the vector u that represent an approximation of the user response to the features. Now we can adjust our previsions:

$$prediction = u_1 S_{11} v_1 + u_2 S_{22} v_2$$

the prediction is:

$$prediction_2 = [2.0000 \quad 9.0000 \quad 9.1140 \quad 1.9608 \quad 9.5578 \quad 8.4549]$$

5.1.3 Day 3

The third day, the new user watch $Movie_3$ and give a score of 8 ($\mu_3 = 8$). Recalculate the prevision.

We need to use the same approach, but we have to update matrix M and vector b as follow:

$$M = \begin{bmatrix} S_{11}V_{11} & S_{22}V_{12} & S_{33}V_{13} \\ S_{11}V_{21} & S_{22}V_{22} & S_{33}V_{23} \\ S_{11}V_{31} & S_{22}V_{32} & S_{33}V_{33} \end{bmatrix}, \quad b = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$$

With this parameters we obtain the following result:

$$prediction_3 = [2.0000 \quad 9.0000 \quad 8.0000 \quad 1.8991 \quad 8.3890 \quad 8.6069]$$

5.1.4 Consideration and error analysis

Let's take into consideration the evolution of the prevision of $Movie_3$. We will compute the error as follow:

$$relative\ error = \frac{|true - prediction|}{|true|}$$

The first prediction that we made for $Movie_3$ was equal to 3.1054, while the second prediction for the same movie is 9.1140. So the history of the relative errors is:

$$relative\ error_1 = \frac{|8 - 3.1054|}{|8|} = 0.6118, \quad relative\ error_2 = \frac{|8 - 9.1140|}{|8|} = 0.1392$$

We can see that the value of the relative error decrease in function of the amount of information that we have over the given user.

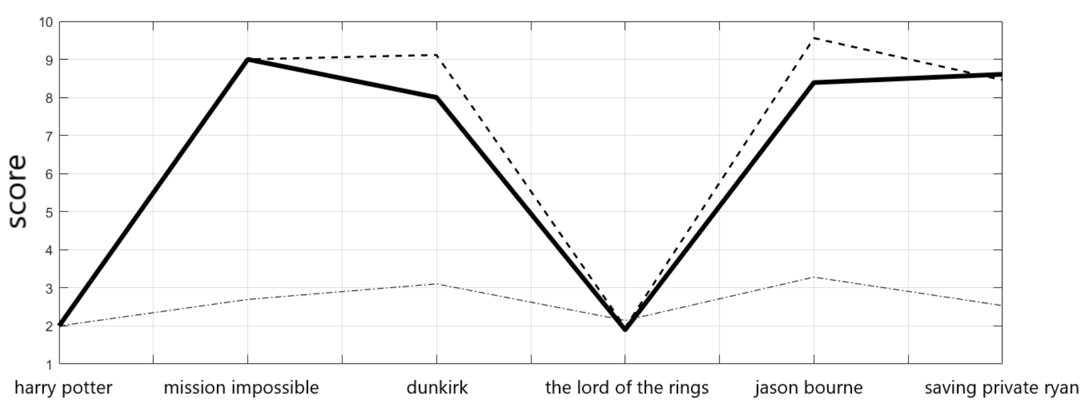

FIGURE 6: Here you can see the evolution of the score prediction. The line become more thick as the information about the user grow.

5.2 The US strategy

Let's take again the equation:

$$AV = US$$

we define the matrix N as follow:

$$N = AV \;\Rightarrow\; N = US$$

We are multiplying the score matrix times the movie-feature matrix and is the same as multiplying the user-feature times the overall importance of a feature.

If we plot different combination of dimensions of the resulting matrix N, as we can see from the figure below, we start seeing some clusters. In order to obtain meaningful result I plotted the 3D graph were there are 4 well distinct groups while in this images we can get tricked by the dimension that we are seeing because they don't have enough information to well divide the points. Anyway, some projections are better then other. For example, in Fig.7 you can see the dimension 1 of N and against the dimension 2. If we didn't have different colors and different shapes we will probably tell that there where 3 clusters. While if we observe the second picture with dimension 2 and 3 we can distinguish 4 cluster. With the scatterplot in 3D is even better to see the four cluster because they are even better separated.

5.2.1 The advantages of the US strategy

The interesting and powerful fact about this analysis is that if we want to make a forecast over a new user on our platform and somehow we already know her taste, we can immediately say what movies she is going to love and so we can make prediction without having any new values in our score matrix. Let's imagine that we have information about this customer thanks to the data collected by a different platform, for example an e-commerce platform where she can buy items that she likes, we can understand the values of her U rows and simply multiply it by the S matrix that we know already.

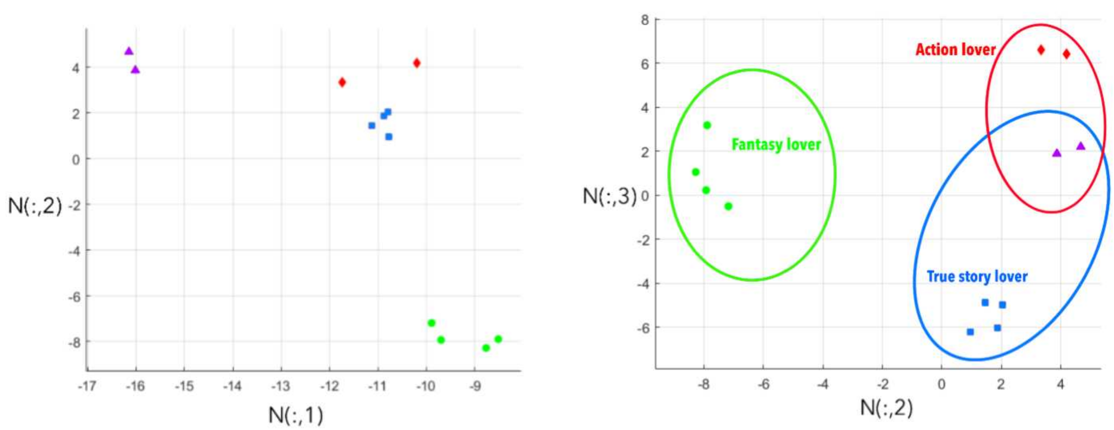

FIGURE 7: Different projection of the 3D space with respect to the matrix N defined above.

6 Conclusion

The application of the Singular Value Decomposition seems to produce useful result for the problem of Collaborative Filtering. This technique allow us to see the same topic in different perspectives and create and advantage in order to make useful prevision over the customers.

This days, more and more companies develop algorithm of this kind because in that way they are able to understand the interest of the customers and so increase their revenue.

In the practical example shown in section 5 we obtain useful information and if we combined them with information obtained in section 6, we can understand the global taste of the customers. So, we could create a new movie that embed all the features that all the customer like in order to make the movie likable to all.