A relationship between Linear Algebra and Machine Learning

Yong Hoon Do, Sojung Kim

March 26, 2017

1 WHAT IS MACHINE LEARNING

Machine learning is defined as a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. The term, Machine Learning may sound strange, but it is not an exaggeration to say that we have been taking advantage of it for our entire Google searching experience and also the experience purchasing items on Amazon, or selecting movies in Netflix. Machine learning is used at almost every part of the stack at major search engines, and anything that requires some sort of intelligence is often solved using machine learning. Spelling suggestion/correction and also search ranking system are great examples that can be solved by Machine Learning.

A recommender system is one of the successful field that the concept of Machine Learning is deeply engaged with. An algorithm called, **Collaborative Filtering** is one that commonly used nowadays, and it works by building a database of preferences for items by a group of users. Fo example, a new user, Guerrero, is matched against the database to discover neighbors, which are other users who have historically had similar taste to Guerero. In this specific example, the solution may be defined as providing a list of the best items to the user, so that the user, Guerrero could highly consider buying the items that recommended by the system. We all agree that increasing a probability of purchasing items should give a benefit to the company, and we can achieve this goal by using Machine Learning.

Since Machine learning is deeply relating with numerous properties of Linear Algebra, we can also say that recommending the best items can be accomplished by utilizing many concepts of Linear Algebra.

In this paper, we would not address how the algorithm really works in detail, but will discover how the basic concepts of Linear Algebra are applied to one of the hottest software, a recommender system that is derived from Machine Learning.

2 A relationship between Linear Algebra and Movie Recommender System

Now, we want to show that Linear Algebra is highly engaging in Machine Learning by showing many properties of Linear Algebra are really applied to a movie recommender system as a foundation concepts. Before we do jump into the specific properties of Linear Algebra how it is used in a recommender system, let us consider a problem what we really want to achieve from doing it.

2.1 What Does A Movie Recommender System Do For Users?

Netflix completely relies on Machine Learning to build a movie recommender system, and the recommendations that come out from this system as an output drive 60% of Netflix's DVD rentals. We can think of improving the accuracy of predictions how much someone is going to love a movie based on their movie preferences. Now, Netflix uses a well-known algorithm, called **Collaborative Filtering** to discover patterns in observed preference behavior (e.g. purchase history, item ratings and click counts) across community users, and to predict new preferences based on those patterns.

We can concretize this problem using tools that we have learned in class. Before we do analyze how the basic concepts of *Linear Algebra* are used in this application, let us define the problem in terms of mathematical tools that we should have seen in class.

Let us denote

1. X is a set of users
2. S is a set of movies
3. R is a set of ratings, that should be r_{ui} for some user-movie pairs such that (u, i).

4. U is a linear transformation that takes two vectors spaces X and S as an input, and generates a set of ratings R.

Our primary goal is that building a brand new set of predicted ratings for each user such that

$$U : X \times S \rightarrow R$$

As users give more ratings on movies, more accurated ratings will be offered to users by the Collaborative Filtering Algorithm.

3 A Relationship between Linear Algebra and Collaborative Filtering

As we have already introduced an idea that basic concepts of Linear Algebra are indispensable foundation to have Collaborative Filtering Algorithm, we would like to analyze how these concepts are really used in the algorithm. We are going to look at a few key concepts of Linear Algebra to realize how they are deeply connected:

1. Linear Transformation
2. Linear Combination
3. Basis and Dimension
4. Nonsingularity

The other fundamental concepts like a set, a matrix, a vector space and other tools like a matrix multiplication and matrix transpose are also significant to discover the deep engagement with the algorithm, but we find that these concepts are so natrual when it comes to prove the relationship between Linear Algebra and the Collaborative Filtering Algorithm. They will be frequently appearing all over the section.

3.1 Linear Transformation

The first essential idea that makes a relationship between the algorithm and Linear Algebra should be Linear Transformation since it defines a function. To give predicted ratings for community users as an output, we need an input!
As you may have already noticed, it is defined as

$$U : X \times S \to R$$

U is a linear transformation that takes $X \times S$ as an input, which includes two vector spaces; one is a set of users and another one is a set of movies. Multiplying these two vectors allows to find out the third vector space, namely a set of ratings beloing to each users in a set of users.

The type of ratings that we are using in this paper would be divided into two; one is a rating that a user granted in person, and another rating should be a predicted rating that granted by the Collaborative Filtering Algorithm. This brief defintion may bring us to realize that we may have a matrix with unknown ratings, which are eventually known as a natrual number like $[0,5]$, but they are up on the same matrix. We can look this matrix so that we can understand what the algorithm is really trying to accomplish.

$$\mathbf{M} = \begin{bmatrix} ? & ? & 1 & \dots & 4 \\ 3 & ? & ? & \dots & ? \\ ? & 5 & ? & \dots & 5 \end{bmatrix}$$

Each row indicates a user, each column indicates a movie and each entry in the matrix indicates a rating that the user directly granted. Then, the entries filled in a question mark are the ratings that eventually the algorithm would find out, and it will be offered to users such that

$$r_{\mathbf{x}} = \{5, 5, 5, 4, 3, 1, \dots, 1\}$$

where $r_{\mathbf{x}}$ is the vector of the user $\mathbf{x}$'s predicted ratings. This notion defines the problem simpler as all we have to do is that filling numbers in unknown entries of sparse matrix, and emitting the matrix as an output.

3.2 Linear Combination

You might have a question how we come up with a pretty good predicted rating based on the ratings that users granted. Based on the ratings of users, the algorithm predicts each sparse entry in the matrix by firstly finding similar users to the user, and predicting the rating with their feature sets. Lets clarify each step and eventually let us discover how Linear Combination is applied to.

Let us denote

1. $r_{\mathbf{x}}$ is the vector of user **x**'s ratings

2. N is the set of **k** users most similar to **x** who have reated movie **i**

Then we can define r_{xi} as

$$r_{xi} = \frac{1}{k}\sum_{y\in N} r_{yi}$$
$$r_{xi} = \frac{\sum_{y\in N} S_{xy} r_{yi}}{\sum_{u\in N} S_{xy}}$$

Also, the similarity can be shown as

$$S_{xy} = sim(x, y)$$

Based on this function, we can find the user group that has a similar preference to the target user and we can predict some ratings.

$$\begin{pmatrix} 3 & 5 & 3 & ? & 4 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 & 5 \\ 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 3 & 2 \end{pmatrix} \begin{array}{c} sim(1, m) = 1 \\ sim(1, m) = -0.18 \\ sim(1, m) = \mathbf{0.41} \\ sim(1, m) = -0.31 \\ sim(1, m) = \mathbf{0.59} \end{array}$$

If we would like to know the rating, r_{14} =?, and we know the similarities. Then, we can say that r_{14} can be found by a linear combination of two vectors such that

$$r_{14} = \alpha_1 r_{34} + \alpha_2 r_{54}$$
$$r_{14} = \frac{(0.41)(2) + (0.59)(3)}{(0.41 + 0.59)}$$
$$r_{14} = 2.6$$

We can predict the rating by taking weighted average, that is really a linear combination of other ratings that granted by another gorup of users.

3.3 Basis and Dimension

Let us say there are n number of movies and n number of users. Each user, who has already seen the movie, can rate the movie once, ranging from 0 to 5. This can be shown in a table like below:

$$
\begin{array}{c}
\begin{array}{ccccc} u_1 & u_2 & u_3 & \dots & u_n \end{array} \\
\begin{pmatrix} 1 & ? & ? & 2 & ? \\ 3 & ? & ? & ? & 1 \\ ? & 4 & 4 & 3 & 3 \\ 5 & ? & ? & ? & 2 \end{pmatrix}
\end{array}
\begin{array}{c} \\ m_1 \\ m_2 \\ \dots \\ m_n \end{array}
$$

Not all users have watched the existing movies, so the table is incomplete. This table can be also written in a set of n x m vectors. Let V be a set containing vectors of users, movies, and ratings. Then,

$$V = \left\{ \begin{bmatrix} m_1 \\ u_1 \\ r_1 \end{bmatrix}, \dots, \begin{bmatrix} m_n \\ u_n \\ r_n \end{bmatrix} \right\}$$

Or,

$$V = \{v_1, v_2, v_3, \dots, v_n\}$$

Since the matrix is incomplete, this is a sparse matrix. A sparse matrix contains empty values, in this case, ratings. Note that empty values —the question marks on the matrix —cannot be replaced with zeroes as zeroes will be taken as 0 out of 5 ratings.

In order to fill these empty values, there exists several ways: Jaccard similarity measure, cosine similarity measure, and Pearson correlation coefficient, most commonly. Further explanations about these procedures are omitted in this section for simplicity. One of the procedures are used to predict ratings and looped to fill all the empty values. Once they are all filled, they are no longer a sparse matrix and can be graphed in a 3-dimensional space with x, y, and z-axis (movies, users, and ratings). Then the set V can be said to be a vector space (**Definition VS**) that is 2) linearly independent and 3) a spanning set of its dimension because:

1. Each user can only leave one rating per movie - each vectors are unique and cannot be represented by a linear combination of other vectors

2. Each vector in the set V can be represented in the 3-dimensional space $\mathbb{R}^3$.

A basis is the minimum set of vectors that spans the subspace, which requires the set to be linearly independent and a spanning set. Since the set V met the requirements, V is a basis for $\mathbb{R}^3$. By **Definition D**, the dimension of V is m, or $\dim(V) = m$.

3.4 NONSINGULARITY

In Basis and Dimension section, there were n number of users and n number of movies. However, most of the times, this cannot be true —one value outnumbers the other. For example, Netflix has about 480,000 users and 17,770 movies. According to **Definition NM**, a matrix has to be a square matrix even before checking checking for its trivial solution. Then is it possible to make the matrix even close to a square? No, it is impossible! Yet there are ways to reduce values, making the matrix closer to a square one. This involves complex multiple steps and algorithms such as singular value decomposition (SVD) and regularization of data. Even without deeper understandings of them, it seems obvious why it would be easier for the program to have a square, nonsingular matrix. As the vector starts out sparse, the main goal is to fill the missing entries based on the given information, in this case, ratings submitted by other users. However, if, as the case of Netflix, the number of users far outnumbers the the umber of movies, the set of vectors will always have linearly dependent vectors. This means that the prediction, made by one of the procedures mentioned in the earlier section, will more likely to have errors which will result in falsely recommending a movie that an user does not want to watch. Thus, keeping the nonsigularity of the system is idealy the key to lower the possible errors.

4 CONCLUSION

We have discovered that there is a strong relationship between the basic concepts of Linear Algebra and Collaborative Filtering as the CF algorithm uses them at every part of the stack from defining a problem to offering a set of predicted ratings to the community users. We notice that the discovery of basis is the key part of the algorithm as it gives a possible solution to the problem: giving predicted ratings. As the entry filling in an unknown rating of the sparse matrix could be found by using a linear combination, it clearly shows that a recommender system is really an application of Linear Algebra since it is a linear problem! Based on this careful observation, we can conclude that Machine Learning is an application of Linear Algebra as they are strongly connected.