Chapter 3

Method

In the previous chapters, we investigated what are the recommendations and the different techniques applied in the recommender system and also their application in the e-learning domain. This chapter will describe the process in which the item-based collaborative filtering algorithm was applied to the Scratch Dataset to build a recommendation system. This recommendation engine will generate recommendations of projects a user might like based on the previous activity of the user.

**Summary of the Collaborative Filtering process**

The objective of the collaborative filtering algorithm is to recommend new items or to estimate the usability of a particular item for a certain user, based on the user's past activities or linkings. In an archetypal CF scheme there is a list of m users say U = { u1,u2,u3,...,um } and a list of items I = { i1,i2…,in } . Every user ui has a list associated with the them consisting of items IUi , which the user has expressed interest in or has impression about. (In our case of Scratch we look at the projects the user has liked and the projects user has created/published in the past.) . This impressions or likings can either be collected explicitly or implicitly. Notice that the Item list for user IUi I and it is probable for the the list IUi to be a null set. Also there exists an active user ua U , for whom the task has been carried out. The goal for collaborative filtering algorithm is to discover an item (in our case projects) likeliness i.e. how likely the user is to view or love this item this can take two forms.

* **Recommendation** R is a list of items such that Ir which contains the items the user will like the most. It is to be noted that the R should not contain items that the user is already associated with i.e., for example, the projects that the user already has on his page.
* **Predictions is** a number Pa,j representing the predicted likeliness of the item for the user ua. The predicted value is same as that of the normalised scale of the user ratings. In the case of Scratch this is the count for views, loves and likes for the user.

Figure 1 shows a diagrammatic representation of the collaborative filtering process. The CF algorithm shows the whole m x n matrix A as a user vs item matrix of the user ratings. Each cell ai,j in the matrix holds value for every item the user has rated i.e. either liked, loved or remixed. Every rating is on the numerical scale and can be 0 representing the user has not yet rated ( i.e. liked or loved or remixed the project) the project.

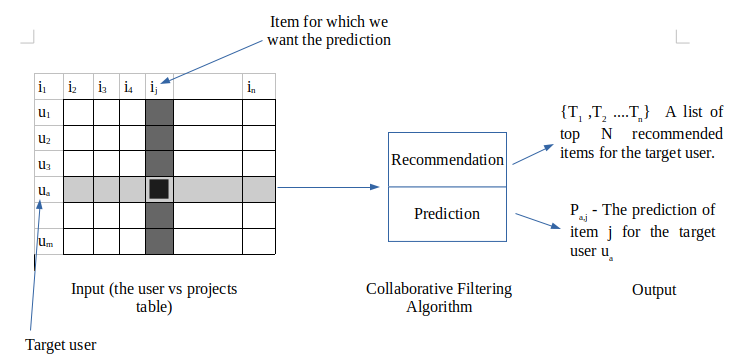


Figure 1. The process of a Collaborative Filtering Algorithm

**The item-based collaborative filtering algorithm**

Unlike the user-based collaborative filtering algorithm shown above, the item-based collaborative algorithm considers the semantics of the items a user has interacted with ( given a rating, liked, created etc.) to calculate the similarity between two items i.e. how closely related items { i1,i2…,in } are. Then it selects the k most closely related items { i1,i2…,ik}. Once the task of computing similarities is complete the predictions is computed by taking the weighted sum average of the target users ratings( in our case liking) on these similar items. This process of similarity calculation and prediction generation is detailed below.

**Similarity computation**

One of the most critical steps in the item-based collaborative filtering algorithm is to calculate the similarity matrix between the items and then to select the most similar once.

The essential idea in similarity calculation between two items i and j are to initially separate the users who have evaluated both of these things and after that to apply a similarity calculation strategy to decide the comparability si,j Figure 2 represents this procedure. Here the matrix rows denote the user and the columns denote the items.

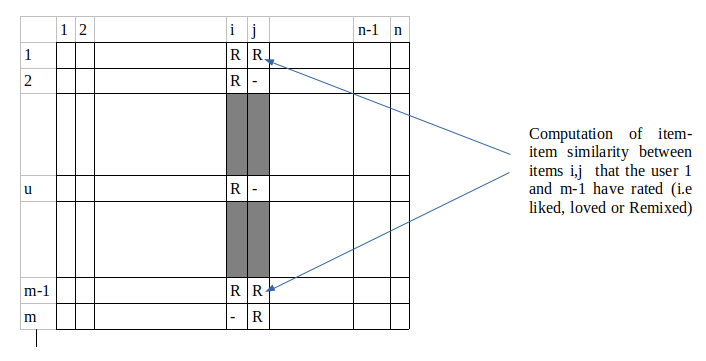


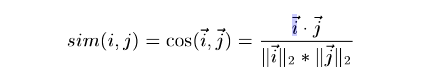
Figure 2 : Computation of item-item similarity

Various techniques have been proposed to compute the similarity between any two items in statics. Some of the most common and well known once are *Correlation-based similarity*, *Cosine-based closeness* and *Adjusted-cosine based closeness.* We will examine each of this technique and apply on our dataset to check which of them suits best for our dataset.

**Cosine based similarity**

Cosine similarity is defined as the measure of similarity between any two non-zero vectors. These vectors reside in as inner product space. Cosine similarity is defined as the cosine of the angle between them. The cosine for a vector with the angle at 0 degrees is 1 and the for any angle between 0 and is less than 1. Therefore it gives us information about the orientation of the vectors and not their magnitudes. Two vectors with orientation in the same direction have a cosine 1 and the two at an angle of 90 degrees with respect to each other is 0. Similarly, two vectors within an exact opposite orientation will have a cosine similarity as -1 irrespective of their magnitude. Cosine similarity mostly used in positive spaces where we require the output to be neatly bounded between [0,1]. Particularly data mining, information retrieval and text mining which have high dimensionality problem extensively use this technique. Each of the terms is assigned a new dimension to calculate the cosine similarity between them. The cosine similarity is useful because even if two vectors are very far apart in terms of Euclidean distance (due to their sizes), there is a chance that they may be similarly oriented closer to each other. The smaller the angle between the two, the higher is their cosine similarity.

In our case, we consider two items as the two vectors in the m dimensional user-space. The similarity measure is calculated by computing the cosine of the angle between these two vectors. In Figure 2 we see the similarity between items i and j in an m x n matrix for user projects. The similarity is shown by sim(i,j) such that



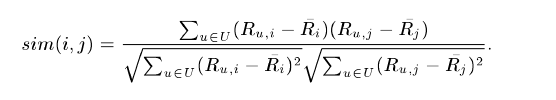
Where the dot product is denoted by ‘.’

**Correlation-based similarity**

In statics, correlation is a statical association between two variables, showing the degree to which the two are linearly related to one another.There are many correlation coefficients defined which show the degree of closeness between variables. Pearson correlation, Interclass correlation, Spearman's rank correlation are some of the examples of correlation coefficients used in statics. Pearson correlation is one of the most widely known and used matrics to show the similarity.

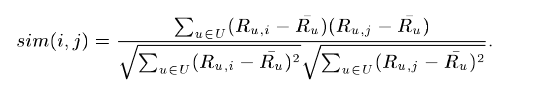
Also, know as the person-r correlation coefficient, is the gives us the linear correlation between any to variables X and Y. It can take a value between +1 and -1, where 1 denotes a total positive correlation between the variables, 0 denotes absolutely no correlation between the variables and -1 denotes a total negative correlation [41]. Mathematically it is defined as the covariance of the two vectors divided by the product of their standard deviation.

In a correlation-based similarity scenario, the likeness amongst two items i and j is computed by calculating the Person - r correlation corr(i,j). For a correlation to be accurate the cases where the users have rated the projects same will need to be separated.



**Adjusted-cosine based closeness**

One common thing between the closeness calculation in user-based CF and item-based CF is that in user-based CF the similarity is processed along the rows of the matrix, however, in the case of the item-based CF the similarity is processed along the segments(columns) i.e., each pair in the co-appraised set relates to a different user(Figure2). Computing similarity utilizing fundamental cosine measure in the item-based case has one significant downside - the distinctions in the rating scale between various users are not taken into consideration. The adjusted-cosine based similarity counterbalances this disadvantage by subtracting the average score from every co-evaluated pair. The formula for this is given as below



Where Ru is the normalized score for uth user.

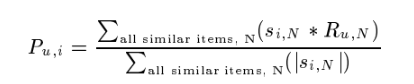
**Prediction Calculation**

One of the most important steps in a CF-based system is to give the output in terms of the prediction. Once we have with us the list of most similar items as given by the similarity measures above, the next step is to examine the user scores and apply a predictive method to obtain the prediction. Various ways have been proposed to compute the prediction. Regression method and Weighted Sum method are the two such most common techniques.

**Weighted sum method:**

In mathematics, a weighted function is the method used when performing average, sum or integral to assign some components more “weight” i.e. importance than others to increase or decrease their impact on the result. Weighted sums are most commonly used in statics to decrease or compensate for the bias present in the data. For example, if a quantity F is measured multiple times independently with variance , we can find the best approximation of this measure by taking the average of all the measurements with the weight . giving us a smaller variance than each individual measurement.

This method computes prediction for project u for user i, by calculating the weighted sum of rating attributes( views, loves, likes etc.) for a similar item. Formally this can be shown as the prediction P u,i as below



Basically, this method attempts to capture how a target user u, is likely to see ( either give love, of like it or remix it) based on the similar items that he has seen (liked, loved, or remixed in past).

**Regression model:**

In statistics, regression modelling is a collection of procedures to estimate the dependency relationships among the variables. Regression analysis typically helps us understand how the value of the dependent variable changes when any one or multiple independent variables are changed.number of regression techniques have been invented to support various statical problems. Linear regression, polynomial regression, principal component analysis etc are some of the popular techniques.

Regression model for calculating the prediction is mostly similar to the weighted sum method discussed above. The only difference is that instead of using ratings of similar items, this method approximates the ratings of items based on the regression models.

When similarity is calculated using either cosine or correlation measurement, it may lead to improper conclusions in practice because the two vectors may be at distance in terms of euclidean distance, but may be highly similar to each other in terms of orientation. In this case, using the ratings from similar items can result in low-quality predictions. The general idea is the same as that of the weighted sum method, but instead of using raw ratings from the N similar items, this method uses their approximate values which are calculated using a linear regression model.

General formulation of a CF-algorithms and the necessary methods required for performing item-item collaborative filtering have been discussed till now. In the next chapter, all these algorithms will be applied to the actual Scratch dataset.