Product Recommender System

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by
**Sajad Ansari, Rajeev Ranjan, Naval Kishore,
Vaibhav Badole, Rebba Prashanth**

Under the Supervision of
Mr. Rajesh Tripathi
Associate Professor

to the
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD
April, 2018

UNDERTAKING

I declare that the work presented in this report titled "*Product Recommender System*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

April, 2018
Allahabad

Sajad Ansari 20154175

Rajeev Ranjan 20154167

Naval Kishore 20154171

Vaibhav Badole 20154033

Rebba Prashanth 20154150

CERTIFICATE

Certified that the work contained in the report titled "*Product Recommender System*", by *Sajad Ansari 20154175, Rajeev Ranjan 20154167, Naval Kishore 20154171, Vaibhav Badole 20154033, Rebba Prashanth 20154150* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Mr. Rajesh Tripathi)
Associate Professor
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

April, 2018

Preface

Recommender System deals with identifying and predicting the most relevant products for a user based on his/her previous interaction. We are interacting with the recommender systems in our day-to-day life like a product recommendation in e-commerce sites (Amazon, Flipkart), friend recommendation in social networking sites (Facebook, Instagram), movie and video recommendation in YouTube, Netflix and job recommendation in Linkedin etc.

Acknowledgements

First and foremost, we would like to take this opportunity to express our sincerest gratitude to our mentor, Mr. Rajesh Tripathi for his continuous support, constant guidance and without whose motivation, words of criticism and patience at our failures, this work would not have been as it stands today. This project work comes from his helpful discussions, supervision and meticulous attention to details. We would also like to take this opportunity to thank Dr. Shashank Srivastava, D.U.G.C. Convener for his support, constant guidance and whose tiring inputs have brought about the possibility of mini projects for CS/IT 2017-18. We would like to extend our heartfelt gratitude for support and guidance. Lastly, we would like to thank our colleagues and seniors for supporting us and enabling us to take a different approach to our project. We would also like to express our deepest gratitude to our parents and family whose love and support was always our greatest rock. Lastly, we would like to thank the Almighty God for everything and pray for his continued blessings.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This report deals with different types of Recommender Systems. A Recommender system is a system that performs information filtering to bring information items such as movies, music, books, news, images,web pages, tools to a user. This information is filtered so that it is likely to interest the user. The aim of a recommender system is often to help consumers learn about new products and desirable ones among myriad of choices.

1.1 Motivation

This Project in particular deals with identification and prediction of the most relevant products for a user based on his/her previous interaction.People interact with recommender systems in their day-to-day life like product recommendation in e-commerce sites (Amazon, Flipkart), friend recommendation in social networking sites (Facebook, Instagram), movie and video recommendation in YouTube, Netflix and job recommendation in Linkedin etc.

1.2 Technical Aspects

After ample research work on the existing models and algorithms,inference here depicts application-specific three recommendation systems.

1. User-User Collaborative Filtering using Pearson similarity- In user based recommender system, we look for users who share the same rating patterns with the active user and use the ratings from those like- minded users to calculate a prediction for the active user.

2. Item-Item Collaborative Filtering using K-nearest neighbours- In this system, item-item matrix is build that determines Relationship between pairs of item and infer the tastes of the current user by examining the matrix and matching that user data.

3. Popularity Model- It recommends the most popular products rated by the users.

They can be used to predict the rating for a product that a customer has never reviewed, based on the data of all other users and their ratings in the system. We implement these three algorithms, and then test them on some existing datasets to do comparisons and generate results.

1.2.1 Similar Ideas

Recommender System is used by many organizations like:
Amazon- https://www.amazon.in
Facebook- https://www.facebook.com
YouTube- https://www.youtube.com
Linkedin- https://www.linkedin.com

Chapter 2

Related Work

There has been a lot of work done in this field. For example, one very popular algorithm is Content-based Filtering.Content-based filtering, also referred to as cognitive filtering, recommends items based on a comparison between the content of the items and a user profile. The content of each item is represented as a set of descriptors or terms, typically the words that occur in a document. The user profile is represented with the same terms and built up by analyzing the content of items which have been seen by the user.

The information source that content-based filtering systems are mostly used with are text documents.A standard approach for term parsing selects single words from documents.The vector space model and latent semantic indexing are two methods that use these terms to represent documents as vectors in a multi-dimensional space.Relevance feedback, genetic algorithms, neural networks, and the Bayesian classifier are among the learning techniques for learning a user profile.

There are also other algorithms that try to exploit graph structures to predict links or ratings.Random walks algorithms could be used in predicting links in complex graphs in a very efficient manner. And also, if we model the user and product graph as a bipartite graph, then it is also feasible to use Bipartite Projection algorithm to calculate the relevance between two customers. So the predicted rating is essentially based on the other relevant customers' ratings.

Chapter 3

Proposed Work

In this project,a product recommender system is built based on the Amazon dataset.Collaborative filtering method is used to predict users product rating by means of which clean recommendation of good product to customers is possible, which they potentially give high ratings according to prediction.The root-mean-square error (RMSE) and Accuracy are calculated to carry out evaluation.

3.1 Dataset

This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014.[7]

This dataset comprises reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). There are various categories in Amazon dataset like Books, Electronics, Home-products, Kindle, Cell-phones, Sports and Outdoors,Health and Personal cares etc.In particular the following data sets are taken into consideration. [5]

- Health and Personal Care
- Clothing, Shoes and Jewelry

Sample Review of Dataset :

{ ”reviewerID”: ”A2SUAM1J3GNN3B”,

```
”asin”: ”0000013714”,
”reviewerName”: ”J. McDonald”,
”helpful”: [2, 3],
”reviewText”: ”I bought this for my husband who plays the piano. He is having a
wonderful time playing these old hymns. The music is at times hard to read because
we think the book was published for singing from more than playing from. Great
purchase though!”,
”overall”: 5.0,
”summary”: ”Heavenly Highway Hymns”,
”unixReviewTime”: 1252800000,
”reviewTime”: ”09 13, 2009”
}
```

Description of features of Dataset :

- ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

Each product has more than 10 ratings from different users. The ratings for each product are from 1 to 5.This dataset is randomly divided into 2 parts: the training set and the test set. For each user,the training set contains 70% of the users ratings. The rest 30% ratings build up the test set. Collaborative filtering is trained based on the training set and algorithm evaluation is carried out based on the test set.

3.2 Data Preprocessing

In this, creation of dataframe in json file is done in prior.This is made out of dataset considered,further conversion of json file into csv file is done for future processing.Proper extraction of total number of unique users and products in our dataset is done.We have extracted helpful-numerator and helpful-denominator from h́elpfulćolumn of the dataset.We have described mathematical features like mean ,count,standard deviation etc. of our dataset to get some insights of our dataset.Then we have counted how many times a product is reviewed by users.

3.3 Implementation

In this part,we have implemented different types of recommender systems in python using different inbuilt libraries and inbuilt functions of machine learning.We have used collaborative filtering as our learning model.The different types of recommender systems are implemented as follows.

3.3.1 Item-Item Based Collaborative Filtering

Item-item collaborative filtering[2], or item-based, or item-to-item, is a form of collaborative filtering for recommender systems based on the similarities between items calculated using people's ratings of those items.Item-item collaborative filtering was invented and used by Amazon.com in 1998.

Item-item models use rating distributions per item, not per user. With more users than items, each item tends to have more ratings than each user, so item's average rating usually doesn't change quickly. This leads to more stable rating distributions in the model, so the model doesn't have to be rebuilt as often. When users consume and then rate an item, that item's similar items are picked from the existing system model and added to the user's recommendations.

▮ *Algorithm Proposed: K nearest neighbours classifier*

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common among its K nearest neighbours measured by a distance function.

The nearest neighbours model computes the distance between every interaction vector in the query set against every interaction vector in the reference set. For each vector in the query set, the k closest reference vectors are returned. Evaluation of dissimilarity metric, Euclidean distance is done. In our approach, we take average of the interaction vectors of the k neighbours to get an interaction score between 0 and 5 for each product.The predicted interaction scores are then used to rank the product.

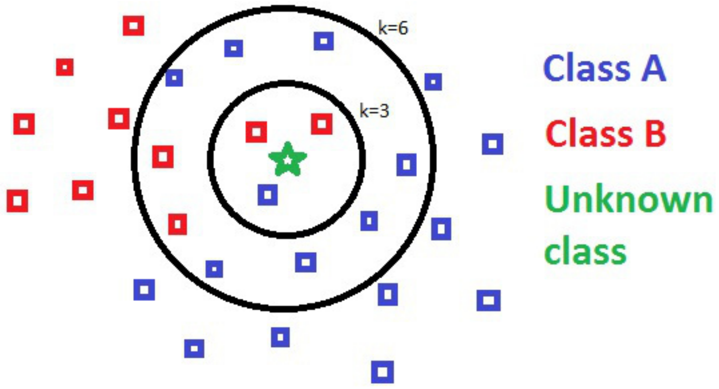

Figure 1: KNN classifier algorithm[9]

The above figure represents the KNN classifier algorithm. If we select k=3 then the classifier predicts the unknown class, class B but for k=6 it predicts the unknown class,class A. If we increase the value of k then the accuracy of the algorithm will increase but the complexity of the algorithm also increases.

Here our item-item collaborative filtering works on **summary review** column of our dataset. First we have removed all the stop words and selected 400 text features using count vectorizer.Then we have split our dataset into training and testing in ratio of 70:30. We have applied KNN algorithm to train our dataset. Then we have predicted three most similar products for each item in testing dataset.

We have evaluated performance of the system by calculating accuracy score and root mean square error.

3.3.2 User-User Based Collaborative Filtering

The basic idea of a user-user collaborative filtering recommender[4] is to select a neighbourhood of users who are similar to us and best represent our taste. Similarity is computed through various correlation and then scoring various items by those neighbourhood of users to generate a recommendation for us. A variation of the first step is to select people whose taste we know and trust. Eg: Amazon builds our user profile through the ratings and purchases we have made and then uses this to compute similarity with other users and depending on the ratings that these users have given on other items it sends us recommendations in an email of items that we may like.

We have used three different similarities in our user-user based collaborative filtering.

- *Cosine similarity*

The cosine similarity between two vectors (or two users on the Vector Space) is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not magnitude, it can be seen as a comparison between users on a normalized space.Cosine Similarity[8] generates a metric that says how related are two users by looking at their behaviour in rating the products.

$$sim(A,B) = cos(\vec{A},\vec{B}) = \frac{\vec{A}.\vec{B}}{\parallel \vec{A} \parallel * \parallel \vec{B} \parallel} \tag{1}$$

$$similarity = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{2}$$

Euclidean distance

Euclidean distance[3] is also known as simply distance. When data is dense or continuous, this is the best proximity measure. The Euclidean distance between two points is the length of the path connecting them.The Pythagorean theorem gives this distance between two points.

$$distance(i,j) = \sqrt{\sum_{i \in item} (s_i - s_j)^2} \tag{3}$$

$$DistanceBasedSimilarity = \frac{1}{1 + distance(i,j)} \tag{4}$$

Pearson correlation

Correlation between sets of data is a measure of how well they are related. The most common measure of correlation in stats is the Pearson Correlation. It shows the linear relationship between two sets of data. Pearson's correlation coefficient [1] when applied to a population is commonly represented by the Greek letter (rho) and may be referred to as the population correlation coefficient or the population Pearson correlation coefficient. The formula for ρ is:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{5}$$

where:

- cov is covariance
- σ_x is standard deviation of X
- σ_y is standard deviation of Y

The formula for ρ can be expressed in terms of mean and expectation. Since

$$cov(X,Y) = E[(X-\mu_X)(Y-\mu_Y)] \tag{6}$$

so

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{7}$$

- $\sigma_X^2 = E[(X-E[X])^2] = E[X^2] - [E[X]]^2$
- $\sigma_Y^2 = E[(Y-E[Y])^2] = E[Y^2] - [E[Y]]^2$

Formula for ρ can also be written as :

$$\rho_{X,Y} = \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - [E[X]]^2}\,\sqrt{E[Y^2] - [E[Y]]^2}} \tag{8}$$

Example : we have one dataset $x1, ..., xn$ containing n values and another dataset $y1, ..., yn$containing n values then that formula for correlation cofficient r is :

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{9}$$

▮ *Prediction model for user-user collaborative filtering*

weighted correlation coefficient is used to predict product for users. Suppose observations to be correlated ,have differing degrees of importance that can be expressed with a weight vector w. To calculate the correlation between vectors x and y with the weight vector w (all of length n)

Weighted mean:

$$m(x;w) = \frac{\sum_i w_i x_i}{\sum_i w_i}. \tag{10}$$

Weighted co variance:

$$cov(x,y;w) = \frac{\sum_i w_i (x_i - m(x;w))(y_i - m(y;w))}{\sum_i w_i}. \tag{11}$$

Weighted correlation:

$$corr(x, y; w) = \frac{cov(x, y; w)}{\sqrt{cov(x, x; w)cov(y, y; w)}}. \tag{12}$$

3.3.3 Popularity Model

Popularity model recommends the most popular products rated by the users. It is the most trivial and simple form of recommender system. It doesn't consider any behaviour of user.

Firstly, we have calculated how many times a product is rated. Then we sorted the product according to count in descending order. Then we have recommended top 10 most popular products to every user. This algorithm also works for a new user who have not rated any product. This model is very less flexible and intrusive to the users.

Chapter 4

Experimental Setup and Results Analysis

We make use of the Amazon Product co-purchasing network metadata, a dataset available on the Stanford Large Network Dataset Collection [1]. This dataset contains 0.5 million products with 1.8 million co-purchasing data and 7.8 million user reviews. It provides sufficient information for us to experiment and get meaningful results and conclusion for the project.
We also build a testing framework before implementing all these algorithms. Each data in the test dataset is a triplet of product, customer, and rating. The product and customer are sent to our recommendation system for predicting, and the rating is our ground truth for testing.
Then we test our systems with the test dataset. The score is measured with reference to mean squared error (MSE),precision and recall values. After picking the test dataset, our test system returns an average MSE for the dataset. This average MSE serves as our main metric on how the recommendation system is performing. A lower score means the system is predicting ratings that are closer to what we have in the original dataset.

- Mean Squared Error: Mean squared error (MSE) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of squares of the errors or deviations ,i.e the difference between the estimator and what is estimated.

The above figure shows Precision and Recall in pictorial format. Precision is to measure the quality of our predictions only based on what our predictor claims to be positive. Recall is to measure such quality with respect to the mistakes resulted.

- Precision and Recall: Precision is to measure the quality of our predictions only based on what our predictor claims to be positive (regardless of all it might miss).
  Recall is to measure such quality with respect to the mistakes resulted. (what should have been predicted as positive but we flagged as negative).
- Accuracy: The degree to which the result of a measurement, calculation, or specification conforms to the correct value or a standard.

Precision, recall and accuracy are then defined as:

$$Precision = \frac{tp}{tp + fp} \tag{13}$$

$$Recall = \frac{tp}{tp + fn} \tag{14}$$

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{15}$$

where true positive (tp), true negative (tn), false positive (fp), false negative (fn).

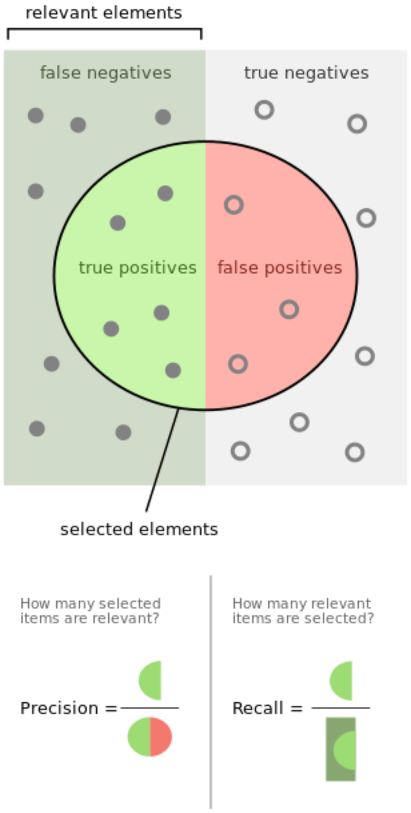

Figure 2: Precision and Recall[6]

4.1 Result Analysis

- Item-Item Collaborative filtering: We have calculated similarities among items based on summary review of the product using k-nearest neighbour algorithm. Then we have calculated accuracy and mean squared error(MSE).

Neighbors	**Accuracy**	**M.S.E.**	**Precision**	**Recall**
1	53.84	68.67	.59	.54
5	61.89	50.19	.59	.62
9	64.77	44.64	.59	.65
13	67.23	40.74	.61	.67
17	68.38	39.19	.62	.68
21	68.81	38.43	.62	.69
25	68.63	37.63	.59	.69
29	69.43	36.73	.61	.69

Table 1: Performance of Item based recommender for different value of neighbours

The above table gives the result of algorithm for different values of neighbours in the KNN algorithm.

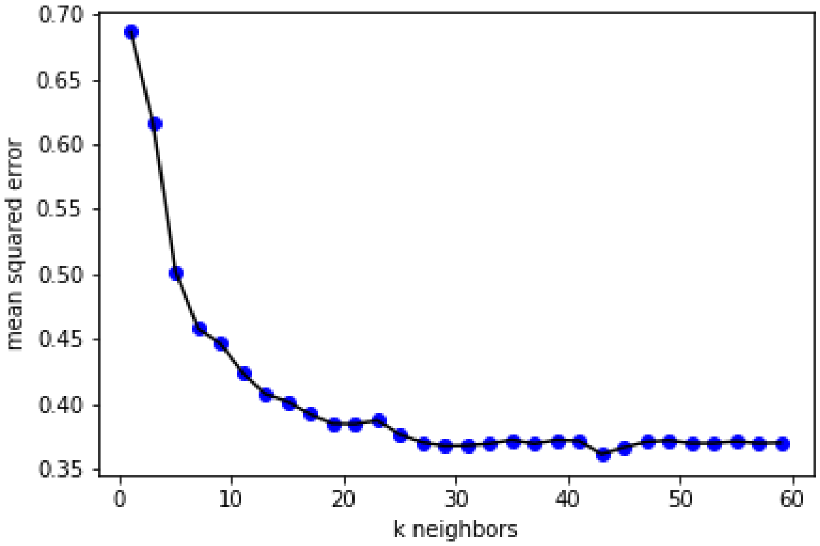

Figure 3: MSE vs K neighbours

Figure 3, represents relation between Mean Squared Error and the number of neighbours. When we increase the value of k, MSE decreases and after a certain point the value of MSE becomes nearly constant, This particular value of k is called knee point.

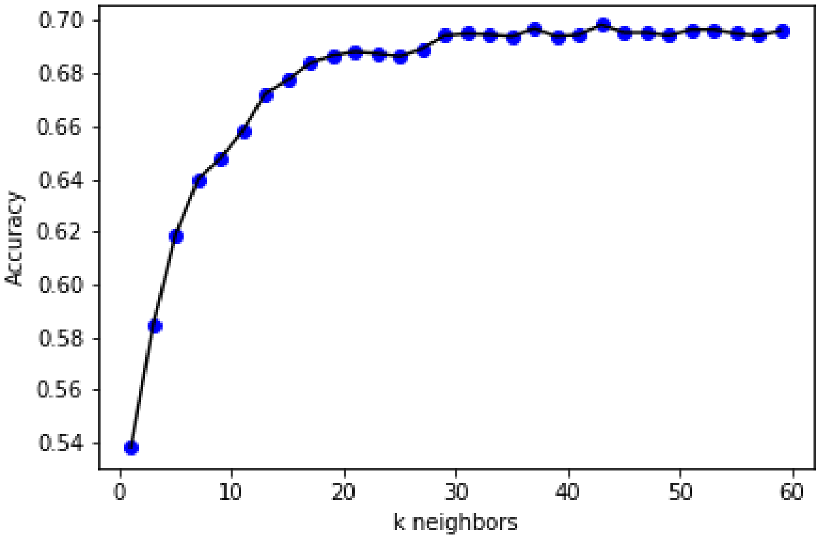

Figure 4: Accuracy vs K neighbours

Figure 4, represents relation between Accuracy and the number of neighbours. When we increase the value of k, Accuracy increases and after a certain point value of accuracy becomes nearly constant, This particular value of k is called knee point. Accuracy of the algorithm increases with the value of k but the complexity of the algorithm increases with k.

- User-User Collaborative Filtering: We have calculated similarities among users based on Pearson Correlation coefficient, Euclidean Correlation and Cosine Similarity. Then we have predicted the rating based on user's similarity.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Recommendation systems help users discover items they might not have found by themselves and promote sales to potential customers, which provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. Lots of companies have such kind of systems, especially for e-commerce companies like Amazon.com, an effective product recommendation system is very essential to their businesses. In this paper, based on ample research on existing models and algorithms, we design three recommendation systems, Item-Item Collaborative filtering, User-User Collaborative filtering and Popularity model. To examine and compare their effectiveness, we implement these three algorithms and test them on some existing datasets.

In our experiments, we found that, in terms of effectiveness measured with mean squared error (MSE), for all users, Item-Item Similarity has the best result, then followed by User-User Similarity, and Popularity Model is the worst.

5.2 Future Work

In near future, we plan to improve the effectiveness and performance by exploring a hybrid system which will apply different algorithms on different user segments. we also need to experiment on different criteria to decide whether a user is a new user or an old user, and then choose the criterion that provides best result.

We also would like to study how we could control or tweak the outputs of recommendation systems based on application-specific requirements. For example, the company might want to avoid recommending some very popular items to distribute the traffic to other products, or the company would like to promote some newly listed products. In general, it is a promising direction to build recommendation systems that can adapt to more granular and flexible application-specific requirements.

Appendix A

Some Complex Proofs and simple Results

Pearson's correlation coefficient is bounded between -1 and 1, not 0 and one. It's absolute value is bounded between 0 and 1.

Pearson's correlation coefficient is simply this ratio:

$$\rho = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}} \tag{16}$$

Both of the variances are non-negative by definition, so the denominator is ≥ 0. The only way a singularity can occur is if one of the variables has 0 variance.

If two random variables are perfectly uncorrelated, (i.e. independent) then their covariance is 0. So 0 is a valid lower bound.

This can be shown like so:

$$Cov(X, Y) = E[(X - \bar{X})(Y - \bar{Y})] = E[XY] - E[X]E[Y] \tag{17}$$

if two random variables are independent, then $E[XY] = E[X]E[Y]$, and

$$Cov(X, Y) = E[XY] - E[X]E[Y] = E[X]E[Y] - E[X]E[Y] = 0 \tag{18}$$

.

Now for the upper bound. Here we apply the Cauchy-Schwartz inequality.

$$|Cov(X,Y)|^2 \leq Var(X)Var(Y) \tag{19}$$

$$|Cov(X,Y)| \leq \sqrt{Var(X)Var(Y)} \tag{20}$$

plug this result from the Cauchy-Schwartz inequality into the formula for , and we get:

$$|\rho| = |\frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}| \leq \frac{\sqrt{Var(X)Var(Y)}}{\sqrt{Var(X)Var(Y)}} = 1 \tag{21}$$

Thus we have the absolute value of the correlation is bounded below by 0 and above by 1.

References

[1] Real statistics using excel: Correlation: Basic concepts. [Online; accessed 1-April-2018].

[2] Badrul Sarwar, George Karypis, J. K.-J. R. Item-based collaborative filtering recommendation algorithms. *ACM* (2001), 285–295.

[3] Elena Deza, M. D. *Encyclopedia of Distances.* Springer, 2009.

[4] Guanwen Yao, L. C. User-based and item-based collaborative filtering recommendation algorithms design. *ACM* (2001).

[5] Julian McAuley. Amazon product data. [Online; accessed 15-March-2018].

[6] Olson, D. L., and Dursun, D. *Advanced Data Mining Techniques.* Springer, 2008.

[7] Ruining He, J. M. Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Ups and Downs 4*, 2 (2016), 507–512.

[8] Singhal, A. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24*, 4, 35–43.

[9] Yun-lei Cai, Duo Ji, D.-f. C. A knn research paper classification method based on shared nearest neighbor. 3–4.