A Comparative Study of Recommender Systems

Abhinav Singh (1801004)
Aryan Sharma (1801028)
Ashmik Harinkhede (1801068)
Dheeraj Kumar (1801052)
Dhruv Narayan Gupta (1801054)
Diwakar Bharti (1801059)

December 1st, 2020

1 Introduction

The popularity of smart phones accelerates the growth of mobile applications, or apps for short. As Google reported, there have been over 2,714,499 apps available in Google Play Store. Moreover, currently there have been 250 million app downloads per day [11]. With so many apps providing different functions (e.g., games, utility, books, etc.), it is difficult for users to find right apps they are looking for via keywords search [13]. Therefore, there is an urgent need to provide an effective app recommendation service.

Recommendation services have been deployed for many types of items such as books [9], music [3], movies [7] and even point-of-interests [14, 15]. Compared with those items, mobile apps have a unique characteristics, i.e., a downloaded app may affect the adoption of alternative apps. Based on her experience of the app, a user may review and rate the app. This review greatly helps in assessing the performance of an app as compared to its competitors.

If a user likes an app and gives it a positive review and a good rating, we may recommend her some other apps similar to it from the same genre and age group. In case the user gives a negative review and a poor rating, then we can recommend them better apps from the same genre that have a higher rating. It is assumed that when a user is browsing for apps, she particularly focuses on a genre and generally, does not migrate from that.

Unlike most other works [16, 17, 2], in this work we utilise an app specific dataset that contains information about an app, rather than the user information on an app. Apps are clustered using an intelligent clustering algorithm. If a user likes an app, she is recommended similar apps from the same cluster. In case, she doesn't like an app, she is recommended apps from the same genre that have a better rating.

In this work, we present a study of various clustering algorithms that can be used for recommender systems wherein the database contains both numerical and categorical attributes. We have designed our recommender using two different algorithms and have compared their performances based on popular evaluation metrics [10, 8].

2 Literature Survey

Owing to the growing popularity of smartphones and the large number of apps made available in the market, a number of research works on mobile app recommendation have appeared in the literature. However, all of them focus on user specific database, i.e., the dataset that they have used contains

information about a user that has its own id and collection of apps. This data is generally collected from smartphones using an application or a crawler is designed to crawl the app stores.

In [16], Yin et al have utilised a user-specific dataset and have formulated their problem as a tuple containing user id, app id, user's current app collection and the action that user takes on the said app. In their work, they have used an Actual-Tempting model that captures such factors that invoke a user to replace an old app with a new one. They use an actual satisfactory value for each owned app and a tempting value for each new (candidate) app and argue that the process of app adoption therefore is a contest between the owned apps' actual values and the candidate app's tempting value. They have used the data logged from an iPhone app, namely, Limited-time Free8, which collects information of iPhone apps that become available for free for a limited time in the Apple App Store.

In [1], Aher et al have built a course recommendation system using simple k-means algorithm and association rule algorithm – Apriori. They have considered 13 course categories and 82 courses that were related to two branches i.e. Computer Science & Engineering (CSE) and Information Technology (IT) to collect data from students.

In [17], Zhu et al. have recommended apps by considering the popularity of mobile Apps, personal preferences of mobile and mobile device constraints . They first carry out a methodology to formulate the popularity of mobile Apps by considering mobile App usage time duration, frequency, the amount of usage, and the number of users .Then They formulate a user's personal preference by Bayesian probability model . And the device constraints they took care of are Power constraints , Network Constraints . It is calculated with the help of Cumulative distributive Function of power and network consumption.

In [2], Chamorro-Vela et al propose a technique for recommending mobile applications based on social and context information . It is called Vanilla approach . The mobile device is the primary source of information that gathers the necessary information for the consumption analysis. The data collection and data preparation components receive the consumption information and generate an overall app rating (V-FPR), a context-aware app rating (V-FPR-C) and a set of categories according to the current user's context.

In [6], Jisha et al used a dataset of 100 android applications along with their user ratings which was obtained by web crawling and web scraping.They have considered user popularity of the app and permissions of the app as recommendation parameters.They have proposed recommendation system that ensures security. K-means clustering algorithm is used and based on the popularity and app permission values the applications are clustered into either high or low and is then recommended to an user. In this way user can decide whether to install or discard the application based on the recommendation.

In [12], Xia et al first mention the methods employed by RS's used by AppJoy, AppBrain and Appaware which all focus on the single traditionally dominating objective of app recommendation i.e the similarity. The authors here propose a multi-objective strategy which captures the other aspects such as diversity or dissimilarity, robustness and the utility as well. The paper proposes a new algorithm based on the Set-based Particle Swarm Optimization(SPSO) called the Cylinder Filling algorithm. To utilise the multi-objective framework the Google Play Market, AppBrain and AppAware have been chosen as the source recommendation systems. They all give different results for the same app as they all have different concerns. In evaluations as they are focusing on multiple objectives they have set the similarity, diversity and utility in the objective function to be equally weighted.

3 Problem Formulation

We have a google play store app dataset (Click here to view) [4] that contains information about an app such as its category, rating, number of reviews, size, installs, type, price, content rating, genres and current version as shown in Figure 1. We also have a user review dataset that contains user reviews of

an app along with the sentiment polarity and sentiment subjectivity of the review given by the user as shown in Figure 2.

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5,000,000+	Free	0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50,000,000+	Free	0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

10841 rows × 13 columns

Figure 1: App Dataset

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	NaN	NaN	NaN	NaN
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000
...	...	...	...	...	...
64290	Houzz Interior Design Ideas	NaN	NaN	NaN	NaN
64291	Houzz Interior Design Ideas	NaN	NaN	NaN	NaN
64292	Houzz Interior Design Ideas	NaN	NaN	NaN	NaN
64293	Houzz Interior Design Ideas	NaN	NaN	NaN	NaN
64294	Houzz Interior Design Ideas	NaN	NaN	NaN	NaN

64295 rows × 5 columns

Figure 2: User review Dataset

We define our problem for recommending apps by dividing it into two parts:

1. First we form clusters of the apps given in the google play store app dataset using appropriate algorithms which are suitable for the given dataset.

2. Then, with the help of certain attributes, we utilise our clusters formed in part one to recommend apps to the user.

INPUT to part one the problem is an app dataset that contains information about the app such as category, genre, ratings, installs, reviews, price, etc.

OUTPUT is a set of clusters of the app.

The above is fed into an *app recommender system* that suggests apps to a user based on her review on an app. *INPUT* to the recommender system is a user's review sentiment polarity and sentiment

subjectivity $OUTPUT$ is a set of apps recommended to the user.

The *objective* of this problem is to study which clustering algorithms that will perform better for this particular modelling of a recommender system.

4 Methodology

In this section we dive into the details of how we implement our solution to the above stated problem definition. The first step is to clean the dataset. In the data cleaning process we first start by dropping the attributes which are not suitable for our clustering like Size, Genre, Current Ver, Android Version. We dropped the Android version and Current Version because we are not taking those details into account for determining our clusters. Then we drop the Genre as that data under this attribute can be encapsulated by the 'Category' column. We drop the rows with missing values, null values and with anomalies. To deal with the duplicates which are present in our dataset due to the same apps with different versions we sort the apps according to the date when it was last updated (obtained from the 'Last Updated' attribute of the dataset). This ensures that we only take into account the latest version of each app in the dataset for the clustering purpose while simultaneously eliminating redundancy. After completing the data cleaning we are left with the following columns :-

- **Application Name**: Name of the application
- **Category**: Category to which the App belongs to
- **Rating**: Overall user rating of the app
- **Reviews**: Number of user reviews for the app
- **Installs**: Number of users who installed the app
- **Type**: Mentions whether our app is paid or free
- **Price**: Price of the app
- **Content Rating**: Target age group of the app

After performing the data cleaning, we scale our numerical columns using a min-max scaler function.

4.1 KPrototype Clustering

As we are having "`mixed`" attributes in our dataset we choose the *KPrototype Clustering Algorithm* [5] which deals with numerical as well as categorical data. K-Prototype is an combination of K-means algorithm (for numerical-type data) and K-modes algorithm (for categorical data) that performs clustering on a mixed-type dataset. After clustering is done, we move on to the recommendation part of our problem. Herein, we make two assumptions based on the dataset :-

- If a user gives a positive review about an app, she must also prefer similar apps that are equally popular. For this, we define a metric to calculate the popularity of an app based on its rating, number of reviews and number of downloads, :-

$$popularity = rt + rv + ins \tag{1}$$

where rt is the relative rating of the app, rv is the relative number of review of the app, and ins is the relative number of downloads of the app. Having done this, we will recommend the user

better apps (with respect to the popularity metric) from the same cluster as the app that was reviewed by her.

- If she gives a negative review about the app, then she must require apps from the same category which should perform better than the rated app. For this, we recommend her better apps (with respect to the overall rating of the app) that belong to the same category and the same cluster as the reviewed app.

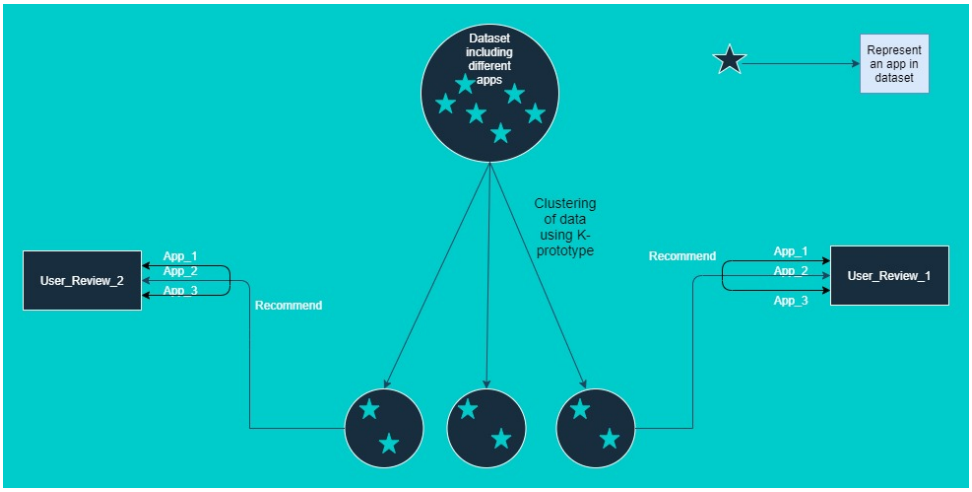

Figure 3: K-Prototype flowchart

4.2 K-Means and the Genre clusters

In this method, we utilise the K-means clustering method to cluster the numerical attributes of the dataset, while leaving out the categorical attributes. The clusters, thus formed, are known as **K-Means Clusters**.

Besides, to entirely justify the dataset, we use the *Genre* attribute to help with the recommendation of each app. Towards this end, we create a dissimilarity matrix with respect to the Genre attribute and utilise a k-mode like approach to cluster the dataset only on the Genre attribute. We take the number of clusters to be equal to the number of distinct categories of the apps. As Genre is a more diverse attribute, all genres falling under one category will be clustered as one. We call these clusters as **Genre clusters**

The above will help us in recommending apps to the users. To proceed further with the recommendation, we will make the same assumptions as in the previous method, but will proceed differently with the recommendation in each case :-

- If a user gives a positive review about an app, she must also prefer similar apps that are equally popular. We need not define any popularity metric here because the k-means clustering was done solely on the attributes that make up the popularity metric. Hence, the idea of popularity is already included in the clustering. In this case, we will recommend apps from the same k-means cluster and same genre cluster.

- If she gives a negative review about the app, then she must require apps from the same category which should perform better than the rated app. For this, we recommend her better apps (with respect to the overall rating of the app) that belong to the same genre cluster and the same k-means cluster as the reviewed app.

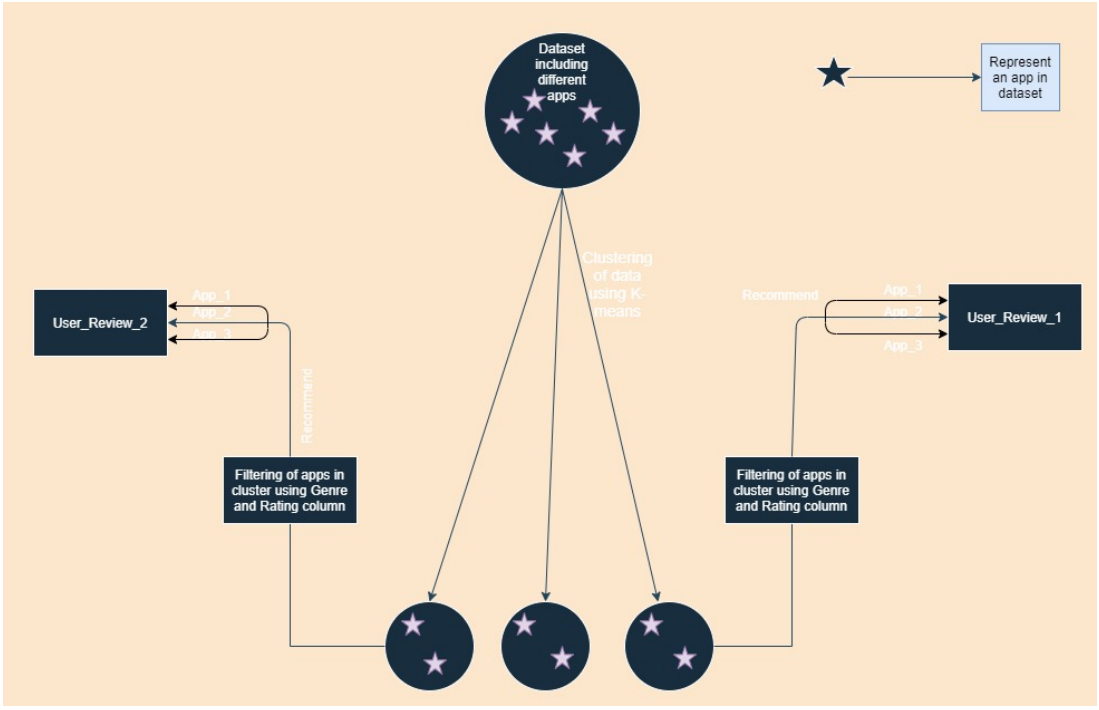

Figure 4: K-Means flowchart

5 Evaluation

The code for the implementation is provided here and here. The performance of both the methodologies are compared based on the recommendations they make to each user.

The following metrics are taken into account :-

- **Coverage:** It is the percent of items in the training data the model is able to recommend on a test set.
- **Intra-List Similarity:** It is the average cosine similarity of all items in a list of recommendations. This calculation uses features of the recommended items to calculate the similarity.

Table 1 displays the metrics for the two methods.

Method	Coverage	ILS
K-Prototype	92.9%	14%
K-Means and Genre Cluster	93.2%	16%

Table 1: Evaluation metrics

6 Conclusion

As can be seen, both the methods perform almost the same on this dataset. This is quite an untraditional way of designing a recommender system as the literature has not seen any work that has targeted any dataset like this. This recommender system might be useful when the recommendation has to be made but no information on the user is available. This might be the case when one has just logged in to an app store or any online store for that matter.

References

[1] Sunita B. Aher and L.M.R.J. Lobo. "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data". In: *Knowledge-Based Systems* 51 (2013), pp. 1–14. ISSN: 0950-7051. https://doi.org/https://doi.org/10.1016/j.knosys.2013.04.015. http://www.sciencedirect.com/science/article/pii/S0950705113001275.

[2] Dario Chamorro-Vela et al. "Recommendation of Mobile Applications based on social and contextual user information". In: *Procedia Computer Science* 110 (Dec. 2017), pp. 236–241. https://doi.org/10.1016/j.procs.2017.06.090.

[3] Hung-Chen Chen and Arbee L. P. Chen. "A Music Recommendation System Based on Music Data Grouping and User Interests". In: *Proceedings of the Tenth International Conference on Information and Knowledge Management*. CIKM '01. Atlanta, Georgia, USA: Association for Computing Machinery, 2001, pp. 231–238. ISBN: 1581134363. https://doi.org/10.1145/502585.502625. https://doi.org/10.1145/502585.502625.

[4] Lavanya Gupta. "Google Play Store Apps". In: (2018). https://www.kaggle.com/lava18/google-play-store-apps.

[5] Zhexue Huang. "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values". In: (1998). https://doi.org/10.1023/A:1009769707641.

[6] R. C. Jisha, R. Krishnan, and V. Vikraman. "Mobile Applications Recommendation Based on User Ratings and Permissions". In: *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2018, pp. 1000–1005. https://doi.org/10.1109/ICACCI.2018.8554691.

[7] George Lekakos and Petros Caravelas. "A Hybrid approach for movie recommendation". In: *Multimedia Tools Appl.* 36 (Jan. 2008), pp. 55–70. https://doi.org/10.1007/s11042-006-0082-7.

[8] Claire Longo. "Evaluation Metrics for Recommender Systems". In: (2018). https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093.

[9] Raymond J. Mooney and Loriene Roy. "Content-Based Book Recommending Using Learning for Text Categorization". In: *Proceedings of the Fifth ACM Conference on Digital Libraries*. DL '00. San Antonio, Texas, USA: Association for Computing Machinery, 2000, pp. 195–204. ISBN: 158113231X. https://doi.org/10.1145/336597.336662. https://doi.org/10.1145/336597.336662.

[10] Julio Palacio Niño. "Evaluation Metrics for Unsupervised Learning Algorithms". In: May 2019.

[11] Avinash Sharma. *Top Google Play Store Statistics 2019-2020 You Must Know*. Sept. 2020. https://appinventiv.com/blog/google-play-store-statistics/.

[12] Xiao Xia et al. "Multi-objective mobile app recommendation: A system-level collaboration approach". In: *Computers & Electrical Engineering* 40.1 (2014). 40th-year commemorative issue, pp. 203–215. ISSN: 0045-7906. https://doi.org/https://doi.org/10.1016/j.compeleceng.2013.11.012. http://www.sciencedirect.com/science/article/pii/S0045790613002942.

[13] Bo Yan and Guanling Chen. "AppJoy: personalized mobile application discovery". In: Jan. 2011, pp. 113–126. https://doi.org/10.1145/1999995.2000007.

[14] Mao Ye, Peifeng Yin, and Wang-Chien Lee. "Location Recommendation for Location-Based Social Networks". In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '10. San Jose, California: Association for Computing Machinery, 2010, pp. 458–461. ISBN: 9781450304283. https://doi.org/10.1145/1869790.1869861. https://doi.org/10.1145/1869790.1869861.

[15] Mao Ye et al. "Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation". In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '11. Beijing, China: Association for Computing Machinery, 2011, pp. 325–334. ISBN: 9781450307574. https://doi.org/10.1145/2009916.2009962. https://doi.org/10.1145/2009916.2009962.

[16] Peifeng Yin et al. "App recommendation: A contest between satisfaction and temptation". In: Feb. 2013, pp. 395–404. https://doi.org/10.1145/2433396.2433446.

[17] Konglin Zhu et al. "A Mobile Application Recommendation Framework by Exploiting Personal Preference with Constraints". In: *Mobile Information Systems* 2017 (June 2017), pp. 1–9. https://doi.org/10.1155/2017/4542326.