Spatially Aware Recommendation Algorithm

Veer Chandra, Ashis Senapati and Suvodeep Majumder

Department of Computer Sc. & Engineering

Heritage Institute of Technology

Kolkata-700107, West Bengal, India

veer.cg@gmail.com

***Abstract*—Recommender systems help users in making decisions by recommending items of interest like movies, music, books, news, images, web pages, etc. to them. Collaborative filtering is one of the most widely studied and widely used techniques behind recommendation algorithms. It tries to recommend items to users based on user-user or item-item similarities computed from existing data. In this work, we propose a recommendation algorithm that takes the user's location into account. The algorithm uses Voronoi Diagrams which are widely used in Computational Geometry to decompose a metric space into regions based on distances from a specified finite set of points. We have tested the algorithm on the MovieLens dataset.**

Keywords - Collaborative filtering, Voronoi diagrams, Recommendation systems.

I. INTRODUCTION

In our day to day lives, we often need to choose amongst alternatives without sufficient personal experience. Whether it is about choosing a movie to watch on a Sunday afternoon or a mobile phone purchase, we often depend upon recommendations from family, friends and colleagues, newspaper reviews etc. In fact, recommender systems are attempts to partially automate and augment that process. Collaborative filtering is one of the most widely studied and widely used techniques behind recommendation algorithms. It tries to recommend items to users based on user-user or item-item similarities computed from existing data, often in the form of ratings given by users. In this paper, we propose a location-aware variant of collaborative filtering, which partitions the data based on a Voronoi diagram.

II. PRELIMINARIES

A. Collaborative Filtering

Collaborative Filtering [1, 2] is one of the most commonly used techniques for developing recommendation engines. It has been used for years by the researchers for implementing recommender systems. Collaborative Filtering, also known as social information filtering, is based on the principle of finding a subset of users who have similar taste and preferences to that of the active user, and offering recommendations based on that subset of users. The idea is that given an active user, u, compute her n similar users $\{u_1, u_2, \dots u_n\}$ and predict u's preference based on the preferences of $\{u_1, u_2, \dots u_n\}$. Similar users mean users who share the same kind of tastes and preferences over items. The basic idea behind collaborative filtering is that users who agreed on the past tend to agree on the future also. Collaborative Filtering works based on the following assumptions:

- Users with similar interest have common preferences.
- Sufficiently huge number of user preferences is available.

Applications of Collaborative Filtering typically involve very large data sets. If we have a sufficiently large number of customer preferences and users with similar interest share common preferences, collaborative filtering can accurately predict user preferences. In many commercial applications, getting access to large set of user preference data is infeasible and therefore collaborative filtering based applications suffer from sparsity issues.

B. Voronoi Diagrams

In Computational Geometry the Voronoi Diagram [3] is used to decompose a metric space into polygons based on distances from a discrete set of points. This space decomposition technique was discovered by Georgy Voronoi and is also known as Voronoi tessellation. Suppose P be a set of n distinct points (sites) in the plane. The Voronoi Diagram of P is the partitioning of the plane into n cells or polygons one for each site. For any point q, if Euclidean_Distance (q , p_i) < Euclidean_Distance (q , p_j), for each $p_j \in P$, $j \neq i$, then q lies in the cell corresponding to site p_i. For any site s, the Voronoi cell v(s) consists of all the points which are nearer to s than any other site. Voronoi segments are all the points which are equidistant to two closest sites. A Voronoi vertex is the centre of an empty circle touching three or more sites. It has degree ≥ 3. Applications of Voronoi diagram are found in problems where the entire space is to be decomposed into smaller regions to reduce computational complexity. Let us consider an example where we have a map over a city with n cell phone towers. We know that a cell phone always tries to connect to the closest tower, so it would be better to split up the city into zones, where each zone has exactly one cell phone tower and each location inside such a zone is closest to the cell phone tower found in the same zone. Thus, a cell phone always connects to its closest tower. Voronoi Diagrams have linear complexity, and so is a good choice for algorithms decomposing a huge space.

C. Pearson Correlation Coefficient

In statistics, the Pearson's correlation coefficient is a measure of the correlation (linear dependence) between two variables *X* and *Y*, giving a value between +1 and −1 inclusive. Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations. In our context, we defined Pearson's correlation coefficient as follows:

Consider two users *a* and *b* who rate a set of m items. For example, the ratings of user *a* on item *i* are denoted as $r_{a,i}$. Pearson correlation between the user *a* and user *b* is defined as the ratio of covariance of user *a* and *b* to that of the product of standard deviation of users *a* and *b*. Mathematically,

$$C_{a,b}=\frac{\sum_{i=1}^{m}(r_{a,i}-\hat{r}_a)(r_{b,i}-\hat{r}_b)}{\sqrt{\sum_{i=1}^{m}(r_{a,i}-\hat{r}_a)^2\sum_{i=1}^{m}(r_{b,i}-\hat{r}_b)^2}}$$

Where $r_{i,j}$ is the user I's rating for product j; m is the total number of items or products; and

$$\hat{r}_x=\frac{\sum_{i=1}^{m}r_{x,i}}{m}$$

is the average rating for user x on all the m items.

The correlation coefficient value ranges between -1 and +1. The correlation value of 1(and -1) is treated as positive (and negative) preferences between users. A correlation value of 0 means that the users have no common set of preferences.

III. PAST WORKS

A lot of work has been done in the area of recommendation engines in the past. Several systems (in the movie domain) have been developed which attempted to recommend movies according to user preferences. Examples of online recommendation systems based on the movie domain include Jinni [4], MovieLens [5], and Netflix [6]. They used collaborative filtering approach to recommend movies. Brunato, Battiti, Villani, and Delai [7] combined the user's current location along with the preferences for providing recommendation services. They developed a location dependent recommender system which is based on a standard web browser. The system recommends a specific URL to a user in a given location that considers where and how often it was accessed by the previous users. Brunato and Battiti [8] used user's position as a relevant piece of information while selecting and ranking links of interest to the users. Their work designed a middleware layer, the location broker that maintains a historic database of location and corresponding links used in the past. Google developed Hotpot [9], a recommendation engine for places, to make local recommendations more personal and relevant, by recommending places based on your ratings and ratings of your friends. It allows you to rate places, to make and invite friends to share those ratings and as you rate sites via Hotpot, the service will recommend other similar places that you might also like. Li. Mi. Zhang and Wu [10] integrated GPS into recommender system to create a location-aware recommender system. They explored the increasing demand of mobile commerce and developed a recommender system for tourism mobile commerce. The system can recommend attractions to the customer with the customer's rating of attractions and customer's sensitivity to location. Yang, Cheng, and Dia [11] proposed a location-aware recommender system for mobile shopping. The system identified the customer's shopping needs and suggested vendors WebPages which includes offers and promotions depending on the location of the customer.

IV. OUR CONTRIBUTION

In this work, we have proposed a spatially-aware recommender system using Voronoi diagrams. And, we have tested our ideas on the MovieLens [5] dataset. MovieLens is a collaborative filtering recommender system developed by GroupLens Research Group. The dataset contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000.The proposed work will use this dataset to recommend movies to users that are likely to be preferred by them. While recommending, our primary focus will be on the location of the user. We use zip-code (city) to identify the user's location. We try to explore the concept of Spatial Autocorrelation - similar values cluster together on a map, by using some statistical measures. Voronoi diagram is used to tessellate the user's space with respect to location. Space decomposition partitions the entire user's space into smaller regions (polygons) and we will provide recommendation to users in those polygons. Our work tries to find the presence of correlation in the polygons and then recommend movies with a view that the suggested movies will be liked by the user. Collaborative filtering algorithms will be used for recommendation. The system will offer recommendations to a new user according to his choices and the preferences of other users who share the same location with the current user. Similarity metrics' such as Pearson's correlation coefficient [1] is used to compute the similarly among the users. In this work, we try to recommend items by computing user-user or item-item similarity among the users in the polygons.

V. DECOMPOSITION ALGORITHM

In this work, we use a Voronoi diagram based approach for space decomposition. Space partitioning is done on the basis of zip-codes (cities). User data file in the MovieLens dataset has information about the user and their locations. Our work tries to decompose this user space. Location is represented by zip-codes. We construct the Voronoi diagram using these zip-codes. Our algorithm finds some zip codes that have a minimum number of users (threshold). These zip-codes become the Voronoi sites. Each site S has a Voronoi cell V(S) consisting of all the points closer to S than any other site. To construct the Voronoi diagram, zip-codes should be represented by some coordinates. Our work considers the longitude-latitude of the centroids of the polygonal regions representing the zip-codes as the coordinates. The algorithm considers the distance of a point (zip-code) from each of the site points and the point is allocated to the region represented by the site that has the minimum distance to the point. Continuing this way we map each point onto some Voronoi cell. The output of the algorithm is the set of Voronoi polygons of user zip-codes. Partitioning algorithm decomposes the entire user space into some smaller polygons.

We next try to find out the correlation coefficient values for different pairs of users lying in each of the polygons, using Pearson's correlation coefficient formula (described above). The algorithm uses the rating file of dataset, to find out the different ratings given by the users lying in each of these polygons to the different movies. And based on the common set of movies rated by any pair of users, this algorithm finds out correlation coefficient for that pair of users.

A. Algorithm Voronoi_Decomposition

Step1: Represent zip-code centroids as coordinates (longitude-latitude). Let this set be denoted by P.

Step2: Select those zip-codes (sites) from user space that satisfies the threshold value. Let this subset of P be denoted by T.

Step3: Create the Voronoi diagram with the sites from T.

Step4: Locate each zip-code (centroid) c in P-T in the Voronoi diagram constructed in Step 3.

Step5: Allocate the zip-code centroid c to the appropriate Voronoi polygon (represented by the site in P closest to c).

B. Algorithm Find_Correlation

Step1: Choose one polygon from the Voronoi polygons.

Step 2: Find the ratings on different movies, given by all the users within this polygon.

Step 3: For a pair of user, first find common set of rated movies and apply Pearson's correlation coefficient formula, to calculate correlation coefficient value for that pair of user.

Step 4: Repeat step 3 for the rest of the users.

Step 5: Repeat step 2-3 for other polygons also.

The output of the Voronoi_Decomposition algorithm is the set of Voronoi polygons represented by their centroids (sites) and a list of centroids located in each such polygon. Each and every city (zip-code) within a polygon is closer to its centroid than the centroid of any other polygon. The output of the Find_Correlation algorithm is the set of the correlation coefficient values for different pairs of users within those polygons, which determine the presence or absence of correlation among the users within those polygons.

VI. RECOMMENDATION ALGORITHM

Once the Voronoi diagram has been constructed, polygons defined, and correlation among users measured, we can start our recommendation algorithm. Recommendations can be provided to a naïve user or an existing user. For a new user we identify the location (zip-code) and accordingly map the user to its destined Voronoi polygon. Collaborative filtering technique is used to recommend items of interest to the user. As per some standard experimental calculations (like Geary's index) and results, it is well established that spatial correlation exists in the polygons, and hence it seems very promising that if we recommend items based on the preferences of the users who share the same neighbourhood with the current active user, quality of recommendation will increase. Similarity in the user's preferences in a polygon is measured by Pearson's correlation coefficient [1]. During pre-processing, we calculate Pearson's coefficient value for every pair of users in the polygon. The algorithm is briefly described below:

A. Algorithm Recommend_Movies

Step 1: Select a user (*say active user*) for the recommendation, and identify the location (zip-code) of that user.

Step 2: Map the user in the Voronoi polygon according to his/her location.

Step 3: Find a set U of users from the same voronoi polygon (found in step-2), those are highly correlated with the active user, means having correlation coefficient values greater than a threshold value (say 0.6).

Step 4: Find the set M_a of highly rated movies of active user (say of having rating 4 or 5 out of 5 scale), and find out Top-2 categories of genres in which most of the movies of this set lie.

Step 5: Also find out the set M_u of movies highly rated by rest of the users of set U.

Step 6: Filter out the subset of movies M_{rec} from set M_u, which are falling in both the Top-2 genres (calculated in step 4) as well as not yet seen by the active user.

Step 7: Recommend this set of movies M_{rec} to the active user.

VII. EXPERIMENTS AND RESULTS

To validate our scheme, we tested our algorithm on the MovieLens dataset. The dataset has a User database with demographic information of the users. It has a Movie database with detailed information about movies. The dataset also includes a Ratings database having the ratings given to different movies by the users. We use this dataset as it has enough information that we need to implement our algorithm. The user location information in the database is represented by their corresponding zip-codes (city). This location data becomes the basis of the Voronoi_Decomposition algorithm. We have tested the algorithm with three threshold values, (5, 10, 15), which indicates the minimum number of users that a zip-code must have to be considered as a site in Voronoi diagram. Remaining cities will be mapped to its corresponding site by the algorithm. Table I shows a sample result of the experiment.

TABLE I. RESULT OF SPACE DECOMPOSITION

Threshold Value	**Number of Polygons**
15	10
10	35
5	175

As the threshold value increases, number of polygon decreases. We next test our Find_Correlation algorithm for all the polygons found by the decomposition algorithm. We defined a set of ranges for the correlation values and counted the number of pairs of users that's fall in a particular range. Table II depict a sample result of the correlation algorithm.

TABLE II. CORRELATION RESULT FOR POLYGONS WITH THRESHOLD = 15

(Correlation Range)⟶ Polygon no	**[-1–0)**	**[0–.25)**	**[.25–.6)**	**[.6–1]**	**% above 0.6**
1	1040	895	926	359	11.15
2	61776	69033	64189	23605	10.79
3	127542	141086	135821	50832	11.16
4	107155	117974	110970	40190	10.68
5	57872	67631	63365	20608	09.83
6	8877	9060	8385	2900	09.92
7	64962	65866	63088	24980	11.41
8	19528	19663	17262	7194	11.30
9	2283	2305	1912	746	10.29
10	26469	28818	26900	9500	10.36

At last, we tested our proposed recommendation algorithm 'Recommend_Movies', to find out how fine, our system is actually recommending. The idea behind this testing procedure is as follows:

A: Testing Algorithm

Step 1: Choose a particular user u_i (i.e. Active user), and find out the set of movies M_u (*refer Algorithm Recommend_Movies)* for this user.

Step 2: Filter out the subset $M_{(R,a)}$ of movies from this set, which the active user has already seen and rated. And Calculate average of the ratings, given to all the movies of this subset. [Say Avg_R].

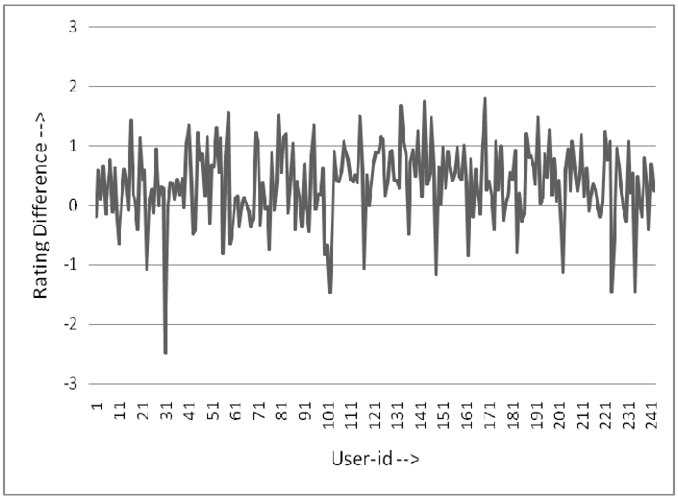
Step 3: Also Calculate average of the ratings given by active user to all the movies,he has seen and rated so far.[Say Avg_{all}].

Step 4: Find out the difference between these two above average ratings. [Say diff(i), where diff(i)= Avg_R - Avg_{all}].

Step 5: Repeat Step 2-3 for N number of users and store all these differences in a table.

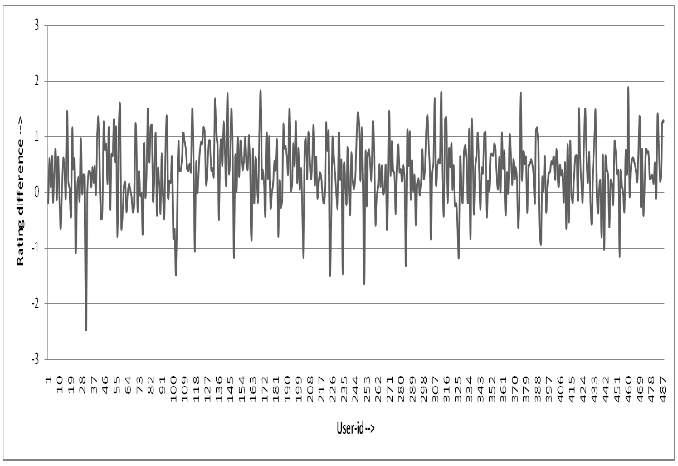
Step 6: From this table of rating differences for different users, calculate the (i) average of absolute of all the differences [Avg_u] (ii) the standard deviation [SD_u] for all these difference values (iii) Number of users having positive value of diff(i) [Pos_count_u] (iv) Number of users having negative value of diff(i) [Neg_count_u].

GRAPH I. RATING DIFFERENCE [diff (i)] VALUES FOR 242 (≈ **250**) USERS.



RESULTS: [A] Pos_count_u = 181, Neg_count_u = 61 [B] Avg_u = 0.35702798. [C] SD_u = 0.63110024.

GRAPH II. RATING DIFFERENCE [diff (i)] VALUES FOR 488 (≈ **500**) USERS.



RESULTS: [A] Pos_count_u = 373, Neg_count_u = 115 [B] Avg_u = 0.3604241. [C] SD_u = 0.60164124.

From these results, we can infer that:

(i) (Pos_count_u) /(Neg_count_u) ≈ 3 : 1, so out of every four users, three users are being recommended relatively better movies by our algorithm, than they have already seen and rated.

(ii) Since Avg_u ≈ 0.3 and SD_u ≈ 0.6, so although the one user out of four, which are not being recommended better movies, Still the average rating of those recommended set of movies(which are not better) differ from the average rating on all the movies he has seen so far, just by [0.3 ± 0.6].

VIII. CONCLUSION

We have proposed a variant of Collaborative Filtering which is location-aware by using Voronoi polygons to partition the space effectively. Exploring other clustering techniques to spatially partition the data and comparing with the Voronoi approach will be the focus of our future research.

REFERENCES

[1]. Bhasker, B. and Srikumar, K., 2010. *Recommender Systems in e-Commerce*, Tata McGraw Hill.

[2]. Herlocker, J. L., Konstan, J.A., Riedl, J., 2000. Explaining Collaborative Filtering Recommendations. *CSCW '00 Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, USA, December 02-06, 2000).

[3]. Van Kreveld, M., de Berg, M., Cheong, O., 2008. *Computational Geometry: Algorithms and Applications*,Springer.

[4]. Jinni, URL : http://www.jinni.com/

[5]. MovieLens, URL : http://www.movielens.org/

[6]. Netflix, URL : http://www.netflix.com/

[7]. Brunato, M., Battiti, R., Villani, A. and Delai, A., 2002. A Location-Dependent Recommender System for the web. *Technical report DIT-02-0093, Universita` di Trento*.

[8]. Brunato, M. and Battiti, R., 2003. PILGRIM: A Location Broker and Mobility-Aware Recommendation System. *In proceedings of IEEE PerComp2003* (March 23-26, 2003), 265-272.

[9]. HotPot, URL : http://www.google.com/hotpot

[10]. Li, X., Mi, Z., Zhang, Z. and Wu, J., 2010. A Location-Aware Recommender System for Tourism Mobile Commerce.*In proceedings of 2nd International Conference Information Science and Engineering (ICISE)* (Hangzhou, China, December 04-06, 2010), 1709-1711.

[11]. Yang, W., Cheng, H. Dia, J., 2008. A Location-Aware Recommender System for Mobile Shopping Environments. *Expert Systems with Applications: An Intenational Journal* (Volume 34, Issue 1, January, 2008), 437-445.