www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 5 May 2017, Page No. 21360-21365
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i5.29

Location Based Social Network For Rating Procedure Geographical Location Using Extended Collaborative Algorithm

M.Muralikrishnan[1], G.Kalpana[2], M.Karthikraj[3]
[1]PG Student, Dept. of Computer Science and Engineering, SRM University, Ramapuram, Chennai, Tamilnadu, India.
[1]muralimvks84@gmail.com
[2]Assistant Professor, Dept. of Computer Science and Engineering, SRM University, Ramapuram Chennai, Tamilnadu, India,
[2]g.gkalpana@gmail.com
[3]PG Student, Dept. of Computer Science and Engineering, SRM University, Ramapuram, Chennai, Tamilnadu, India.
[3]karshan91@gmail.com

Abstract: ***The Rapid Development of the mobile based social networks is to improve the geo-graphical rating procedure, which makes enormous volume of the raw data. In normal mobile user compared to recent trends of the geo-position enabled technology will produce the volume of data is higher than the normal mobile user. The social networks involving geographical information as location-based social networks, this information bring out the new challenge in recommended system to solve the data sparsity problem of data set and cold start problem, in this paper we make the full use of the deeply exploring the user and check-in user for various categories first, user to user geographical connection distance, then user to item geographical distance and one user similarities. Check-in behaviors of users will be deeply explored by considering the above factor their multi-activity centers and the attribute of POIs.***

Keywords: *Attribute Encryption, Key-policy, Cipher text, Schemes.*

1. INTRODUCTION

The rapid development of the location based social networks using mobile environment will provide enormous amount of information and it's enclosed with the mobile device geographical information. Also produce the volume of the data is higher than the normal mobile user. These data has to create big challenge for the recommended system for calculate the rating procedure. Using the geographical latitude and longitude information will attract huge numbers of users. POI recommended systems have played an important role in LBSNs since they can not only meet users 'personalized preferences for visiting new places, but also help LBSNs for increase revenues by providing users with intelligent location services, such as location-aware advertisements. The empty data-set and new user rating has been major problem while using recommended system. There are few more topics have been discuss about LBSN's user rating procedure using geo-graphical distance. Geographical distance between the two users' to calculate the rating procedure is first categories. Calculate the location distance user to social web site rating prediction is the second. Calculate the location distance between the user similarities rating i.e geo-graphical distance connection between users. In this paper will study and achieve deeply explored by between the user connection distance, user similarities distance connection will help to increase the point of Interest including the urban area and rural area of the LBSN using the Extended Collaborative Filtering approach. Information system will collect and filtering the data i.e. Prior right of order various item called Rating. Collaborative filter otherwise called social Filtering. The filtering information using others for help to recommendation others. Example: A person who wants to share their social networks such like that post reviews and comments, ask something for recommendations from friends. Those who are the similar of interest recommend to others, so this will help to see and share their interest. Generally, official website computes the average rating, and sets it as rating level for each item. Sometime opposite approach for the items those have been rated by a large number of users. But for anew item, we cannot straightforwardly to accept or give the few ratings as the objective evaluation of this item. Users have different patterns of giving ratings to the services may be users' tastes and habits are drifting over time. Users' preferences and ratings confidence are different in different location at different period of times. Additionally, sometimes users gave high ratings but many negative comments in their eviews for some kind of valuable reasons. Thus, it is necessary for us to deep understand Social users and urban services by exploring users' Rating behaviors, social circles.

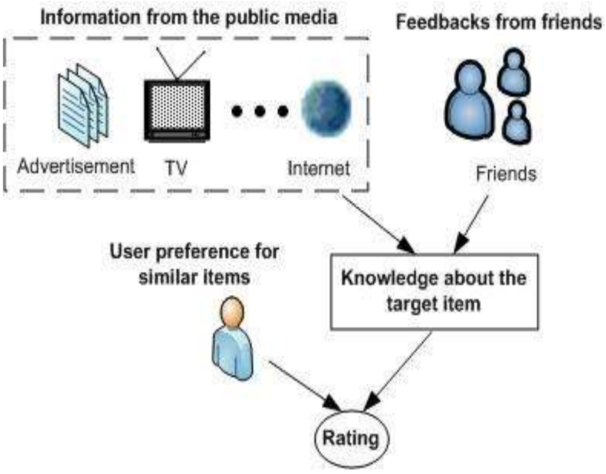


Figure 1: Block diagram Rating prediction using Recommendations System

1.2. PROBLEM STATEMENT AND SOLUTION

SPARSITY: It is common in e-commerce and other domains that people usually purchase or rate relatively few items and compared with the total number of items. That leads to a sparse users-items representation matrix and therefore, inability to locate neighbors or derive common behavior patterns and the final result is low-quality recommendations. This problem is addressed in latent factor models algorithms, which utilizes dimensionality reduction on items and users resulting in finding common behavior patterns in reduced dimensional space, which is not sparse.

COLD START: The problem appears at early stages of a recommender system's lifecycle or when a new item or product is added to the system. When there is a little information available, ontologies are a proven tool for knowledge extension. The cold start problem affects every recommender system: if there is a little information about content, the content-based filtering will behave poorly; the same applies to collaborative filtering. If it happens to have no information recognizable by content-based methods, and there is no people's behavior history in the system, the hybrid approach will produce nearly random recommendations as well.

A RECOMMENDATION QUALITY

Rating prediction are important Role plays, one of the central roles in recommender systems. The user sensible for false negatives (incorrect recommendations, which the user does not like). Assume the user likes genre sci-Fi and highly rated many other sci-Fi movies. If the recommender system will rate "The Matrix" as bad one, but the user likes it, the prediction will be a false negative. In such cases users lose trust in the system and stop using it. Therefore, it is important to keep recommendation quality at the highest possible level.

2. RELATED WORK

This section reviews existing POI recommendation techniques on how they employ the geographical, social, categorical, and popularity information. Recommender systems or recommendation systems (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item. Recommender systems have become increasingly popular in recent years, and are utilized in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general. There are also recommender systems for experts, collaborators, jokes, restaurants, garments, financial services, life insurance, and Twitter pages.

The design of recommender systems that has wide use is collaborative filtering. Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. Many algorithms have been used in measuring user similarity or item similarity in recommender systems. For example, the k-nearest neighbor (k-NN) approach and the pearson correlation as first implemented by Allen collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. When building a model from a user's behavior, a distinction is often made between explicit and implicit forms of data collection The recommender system compares the collected data to similar and dissimilar data collected from others and calculates a list of recommended items for the user. Several commercial and non-commercial examples are listed in the article on collaborative filtering systems. One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by amazon.com's recommender system. Other examples include As previously detailed, last.fm recommends music based on a comparison of the listening habits of similar users. facebook, mySpace, LinkedIn, and other social networks use collaborative filtering to recommend new friends, groups, and other social connections (by examining the network of connections between a user and their friends). Twitter uses many signals and in-memory computations for recommending who to follow to its users. Collaborative filtering approaches often suffer from following problems: cold start, and sparsity. A particular type of collaborative filtering algorithm uses matrix factorization, a low rank matrix approximation technique. Collaborative filtering methods are classified as memory-based and model based collaborative filtering. A well-known example of memory-based approaches is user-based algorithm and that of model-based approaches is Kernel-Mapping Recommender. matrix approximation technique. Collaborative filtering methods are classified as memory-based and model based collaborative filtering. A well-known example of memory-based approaches is user-based algorithm and that of model-based approaches is Kernel-Mapping Recommender.

MOBILE RECOMMENDER SYSTEMS

One growing area of research in the area of recommender systems is mobile recommender system. With the increasing ubiquity of internet-accessing smart phones, it is now possible to offer personalized, context-sensitive recommendations. This is a particularly difficult area of research as mobile data is more complex than data that recommender systems often have to deal with (it is heterogeneous, noisy, requires spatial and temporal auto-correlation, and has validation and generality problems. Additionally, mobile recommender systems suffer from a transplantation problem – recommendations may not apply in all regions (for instance, it would be unwise to recommend a recipe in an area where all of the ingredients may not be available).One example of a mobile recommender system is one that offers potentially profitable driving routes for taxi drivers in a city. This system takes input data in the form of GPS traces of the routes that taxi drivers took while working, which include location (latitude and longitude), time stamps, and operational status (with or without passengers). It uses this data to recommend a list of pickup points along a route, with the goal of optimizing occupancy times and profits.Mobile recommendation systems have also been successfully built using the "Web of Data" as a source for structured information. A good example of such system is SMARTMUSEUM. The system uses semantic modeling, information retrieval, and machine learning techniques in order to recommend content matching user interests, even when presented with sparse or minimal user data.

3. RISK-AWARE RECOMMENDER SYSTEMS

The risk in recommender systems is the possibility to disturb or to upset the user which leads to a bad answer of the user. However, in many applications, such as recommending personalized content, it is also important to consider the risk of upsetting the user so as not to push recommendations in certain circumstances, for instance, during a professional meeting, early morning, or late at night. Therefore, the performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process.

4. PROBLEM AND OVERVIEW

Recommendation research got a boost with the challenge, which means there are lots of quality ratings on how to predict a 1-5 or increasing rating for items from a dataset of previous ratings. That is one field that seems to be saturated, but there are lots of open problems:

CROSS-DOMAIN-ECOMMENDATION Current systems are really good at learning preferences in one domain (say movies), but the same algorithms do not work as well in other domains. E.g. if you like rock and pop in music, what does it say about your movie tastes? I would really like to see a unified model of preference for an individual that explains how different domains interact and inform our preferences.

GROUP RECOMMENDATION: Here, the basic premise is to recommend an item to a group of people, e.g. going to a movie together. This problem has been explored for some time, but we are still excelling in only a part of the problem. Typical models compute individual recommendations and then use a smart way to combine them. But often there will be disagreements, and different groups may have different dynamics. **SIMILARITY MEASURESCOSINE-BASED SIMILARITY** Also known as vector-based similarity, this formulation views two items and their ratings as **vectors**, and defines the similarity between them as the angle between these vectors: Pearson (correlation)-based similarity This similarity measure is based on how much the ratings by common users for a pair of items deviate from average ratings for those items:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}}$$

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$

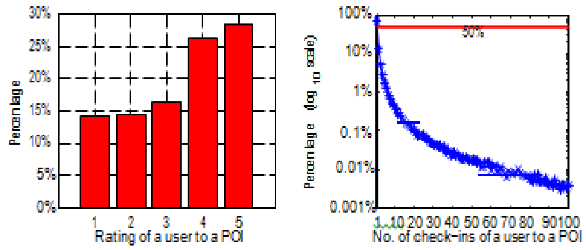
4.1 COMPLEXITY ANALYSIS

A. Many of the methods we will describe, such as those based on expectation maximization, involve iterating a set of update rules until convergence. When the computationalcomplexity of a learning or prediction method depends on iterating a set of operations until the convergence of an objective function is obtained, we introduce the notation Ito indicate this dependence. For most learning and prediction algorithms will be afunction of the number of users N, the number of items M, the number of vote valuesV , and a model size parameter K. We provide average case estimates of the number of Iterations needed to obtain good prediction performance for data sets tested.The space complexity of the representations found by most methods will be a functionof the number of items M, the number of vote values V , and a model size parameter K.For instance based methods and certain degenerate models, the space complexity of the learned representation will also depend on the K-Nearest Neighbor Classifier the K-Nearest Neighbor (KNN) classifier is one of the classical examples of a memory based,or instance-based machine learning method. A KNN classifier learns by simply storing all the training instances that are passed to it. To classify a new query vector xq given the stored training set fxi; cig, a distance dqi = d(xq; xi) is computed for all i. Let xn1 ; :::; xnk be the K nearest neighbors of xq, and cn1 ; :::; cnK be the corresponding outputs. The output for xq is then calculated as an aggregate of the class labels. In the standard case where the input vectors consist of real numbers and the outputs are discrete classes, the distance function d() is taken to be Euclidean distance given by equation 4.1. The predicted output value is taken to be the class of the majority of xq'sK nearest neighbors as seen in equation 4.2. If the outputs are continuous, then the Chapter 4. Classification and Regression 22 predicted output is computed as the mean of the outputs of xq's k nearest neighbors as seen in equation 4.3. This yields K-Nearest Neighbor regression.

$$d(\mathrm{x}_q, \mathrm{x}_i) = \sqrt{\sum_{j=1}^{M} (x_{nj} - x_{ij})^2} \quad (4.1)$$

$$c_q = \arg\max_{c \in C} \sum_{k=1}^{K} \delta(c, c_{n_k}) \quad (4.2)$$

$$c_q = \frac{1}{K} \sum_{k=1}^{K} c_{n_k} \quad (4.3)$$



(a) Distribution of user ratings to POIs in Yelp [24]

(b) Distribution of check-in times in Foursquare [3]

Figure 2: (a) Only about 55% ratings are positive (> 3); (b) Above 50% places have been checked in only once by the same user.

B. KNN- ALGORITHM

C. Makes predictions using the training dataset directly. Predictions are made for a new instance (x) by searching through the entire training set for the K most similar instances (theneighbors) and summarizing the output variable for those K instances. For regression this might bethe mean output variable, in classification this might be themode (or most common) class value.

5. K-NN CLASSIFICATION

the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.*K-NN* is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.

EUCLIDEAN DISTANCE.

Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (xi) across all input attributes j.

EuclideanDistance(x, xi)

= sqrt(sum((xj – xij)^2))

The *k*-nearest neighbour classifier can be viewed as assigning the*k* nearest neighbour a weight and all others *0* weight. This can be generalized to weight nearest neighbour classifiers. An analogous result on the strong consistency of weighted nearest neighbour classifiers.

K-NN REGRESSION

In *k*-NN regression, the *k*-NN algorithm is used for estimating continuous variables. One such algorithm uses a weighted average of the *k* nearest neighbors, weighted by the inverse

$$p(X_i) = \frac{\sum_{Y} Y_i \cdot r(X, Y)}{n}$$

of their distance. This algorithm works as follows:Compute the Euclidean or Mahalanobis distance from the query example to the labeled examples.Order the labeled examples by increasing distance.Find a heuristically optimal number *k* of nearest neighbors, based on RMSE. This is done using cross validation.Calculate an inverse distance weighted average with the *k*-nearest multivariate neighbors.

Figure3: User Distance relationship using KNN

KNN for Classification

This is the simplest scenario. Let x be the point to be labeled . Find the point closest to x . Let it be y. Now nearest neighbor rule asks to assign the label of y to x. This seems too simplistic and sometimes even counter intuitive. If you feel that this procedure will result a huge error, you are right – but there is a catch. This reasoning holds only when the number of data points is not very large.Instead of just relying on the most similar person, a prediction is normally based on the weighted average of the recommendations of several people. The weight given to a person's ratings is determined by the correlation between that person and the person for whom to make a prediction. As a measure of correlation the Pearson correlation coefficient can be used. In this example a positive rating has the value 1 while a negative rating has the value –1, but in other cases a rating could also be a continuous number. The ratings of person *X* and *Y* of the item *k* are written as X_k and Y_k, while $\bar{X}$ and $\bar{Y}$ are the mean values of their ratings. The correlation between *X* and *Y* is then given by:

$$r(X,Y) = \frac{\sum_k (X_k - \bar{X})(Y_k - \bar{Y})}{\sqrt{\sum_k (X_k - \bar{X})^2 \sum_k (Y_k - \bar{Y})^2}}$$

In this formula *k* is an element of all the items that both *X* and *Y* have rated. A prediction for the rating of person *X* of the item *i* based on the ratings of people who have rated item *i* is computed as follows:If the number of data points is very large, then there is a very high chance that label of x and y are same. An example might help – Let's say you have a (potentially) biased coin. You toss it for 1 million time and you have got head 900,000 times. Then most likely your next call will be head. We can use a similar argument here. Let me try an informal argument here - Assume all points are in a D dimensional plane . The number of points is reasonably large. This means that the density of the plane at any point is fairly high. In other words, within any subspace there is adequate number of points. Consider a point x in the subspace which also has a lot of neighbors. Now let y be the nearest neighbor. If x and y are sufficiently close, then we can assume that probability that x and y belong to same class is fairly same – Then by decision theory, x and y have the same class. The book "Pattern Classification" by Duda and Hart has an excellent discussion about this Nearest Neighbor rule. One of their striking results is to obtain a fairly tight error bound to the Nearest Neighbor rule. The bound is

$$P^* \leq P \leq P^*(2 - \tfrac{c}{c-1}P^*)$$

for constants B_1 and B_2 where $s_n^2 = \sum_{i=1}^{n} w_{ni}^2$ and $t_n = n^{-2/d} \sum_{i=1}^{n} w_{ni}\{i^{1+2/d} - (i-1)^{1+2/d}\}$

$$R_{\mathcal{R}}(C_n^{wnn}) - R_{\mathcal{R}}(C^{Bayes}) = (B_1 s_n^2 + B_2 t_n^2)\{1 + o(1)\},$$

R scores for the experiments with k-NN in standard formulation on the user-document data for different values of k, ranging from 10 to 60 with an interval of5. 8We modify Breese et al.'s formula slightly for the case ofObserved accesses rather than ratings. The maximum R value achieved in these experiments was 1.87 for k = 25. R scores have local maxima, suggesting theirsensitivity to the sparsity of the user-document data

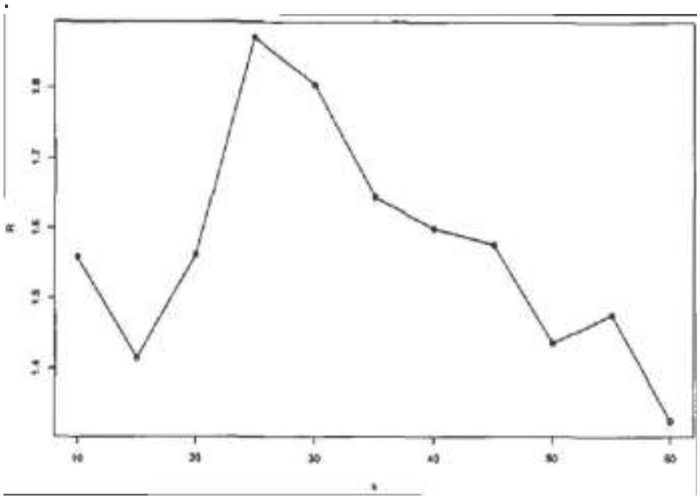


Figure 4: Total utility of the ranked lists over all users produced by collaborative approach

6. CONCLUSION AND FUTURE WORK

mobile based social networks for improve the Geo-graph rating procedure making enormous volume of the raw Data .In normal mobile user Compared to recent trends of the Geo-position enabled technology will produce the volume of data is higher than the normal mobile user. The social networks involving geographical information as location-based social networks, this information bring out the new challenge in recommended system to solve the Data sparsity problem of data set and cold start problem, in this paper we make the full use of the deeply exploring the user and check-in user for the three various categories first user to user geographical connection distance and user to item geographical distance last one user similarities. Check-in behaviors of users will be deeply explored by considering the above factor their multi-activity centers and the attribute of POIs.

7.REFERENCES

[1].G. Zhao, X. Qian, and H. Feng, "Personalized Recommendation by Exploring Social Users' Behaviours," In Proc. MMM, 2014.

[2]. X.-W. Yang, H. Steck, and Y. Liu, "Circle-based recommendationin online social networks," KDD'12, pp. 1267-1275, 2012.

[3]. J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data," ACM SIGSPATIAL GIS'12, Nov. 2012.

[4].C. Zhang, L. D. Shou, K. Chen, G. Chen, and Y. J. Bei, "EvaluatingGeo-Social Influence in Location-Based Social Networks,"CIKM'12, Oct. 2012.

[5].Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Y. Ma, "Recommending friends and locations based on individual location history," ACM Transactions on the Web, 2011.

[6]. J. Zhang, C. Chow, and Y. Zheng, “ORec: An Opinion-Based Point-of-Interest Recommendation Framework,” ACM CIKM’15,2015.
[7]. J. Bao, Y. Zheng, and M. F. Mokbel, “Location-based and Preference- Aware Recommendation Using Sparse Geo-Social Networking Data,” ACM SIGSPATIAL GIS’12, Nov. 2012.
[8]. L. Hu, A. Sun, Y. Liu, “Your Neighbors Affect Your Ratings: OnGeographical Neighbourhood Influence to Rating Prediction,” ACM SIGIR’14, 2014.
[9] R. Salakhutdinov, and A.Mnihdepartment, “Probabilistic matrixfactorization,” in NIPS, 2008.

[10] W. Zhang, G. Ding, L. Chen, C. Li, and C. Zhang, “Generating virtualratings from Chinese reviews to augment online recommendations,”ACM Transactions on intelligent systems and technology (TIST), vol.4, no.1, 2013, pp.1-17.
[11] T. N. Phan, Y. Myungsik, “The lexicon-based sentiment analysis forfan page ranking in Facebook,” IEEE International Conference onInformation Networking, 2014, pp.444-448.
[12] H. Feng, and X. Qian, “Mining user-contributed photos forpersonalized product recommendation,” Neurocomputing, vol.129,2014, pp.409-420.
[13] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, X. Zhu, “Incorporatingreviewer and product information for review rating prediction,” in Proceedings of the Twenty-Second international joint conference onArtificial Intelligence, 2011, pp.1820-1825.
[14] Annamalai, R., J. Srikanth, and M. Prakash. "Integrity and Privacy Sustenance of Shared Large Scale Images in the Cloud by Ring Signature." International Journal of Computer Applications 114.12 (2015).
[15] R Annamalai, J Srikanth, M Prakash, “Prediction Search Of Topics Based On Timestamp And Interest Of Domain With Advertising”, International Journal of Engineering and Computer Science, 4(12), 2015, pp 15387-15391.