Journal of Engineering Science and Technology
Special Issue on ACSAT, October (2021) 57 - 65

DYNAMIC RADIUS FOR CONTEXT-AWARE RECOMMENDER SYSTEM

NAYEF ALAWADHI[1,*], IMAD ALSHAIKHLI[1], ABDULRAHMAN ALKANDARI[2]

[1]Department of Computer Science, KICT, IIUM, 50728, Kuala Lumpur, Malaysia
[2]Computer Department, CBE, PAAET, Kuwait
*Corresponding Author: alawadhi.nayef@live.iium.edu.my

Abstract

In the field of transportation and smart cities, context-aware recommendation is a trending subject that targets users' satisfaction and comfort. The subject has developed from e-commerce to cover different domains including path planning at urban areas. Researchers were competed in delivering the level of contribution in many areas. Search radius is considered a major pillar at any context-aware recommender framework, however, in many studies the search radius is not having a major contribution in the delivered systems. In this paper, a dynamic search radius algorithm is introduced as part of a context-aware recommender framework. Agglomerations and competition effects are used to enhance search radius results. Deep neural network is a major artificial intelligence method used in this research to tackle cold-start problem and to improve recommendation outcomes.

Keywords: Agglomeration effect, Context-aware recommender, Competition effect, Deep neural network, Proximity, Radius.

1. Introduction

Improving data integration and context-awareness are considered some of major identified challenges for the purpose of recommender systems (RS) in smart city experience [1]. As defined by Falk [2], a RS is a system that provides relevant content to the user based on knowledge of the content, user, and interactions between the user and the item. Context-aware RS is a subfield of RS where recommendations are delivered to the user based on information about the user such as location, time and probably social data. Moreover, contextual information can, to a high level of extent. improve the recommendation process efficiency [3].

Recent researches in the field of context-aware recommendation were introduced, especially in transportation domain. Major works were identified in smart cities environment such as smart parking systems, tourist guides and travel planning. The travel planning aka path planning for a certain destination is of a huge concern in smart cities where a user/driver must not waste his time and must reach his destination with a high level of satisfaction. One of the major challenges in this regard is how the user making sure that he is going to a right and a nearby location/area that includes the driver/user needs. To answer such a question, the driver must have enough information about the thing that he is going to get from the location that he is going to. An optimized path planning method should not only take you to the nearby location but also to let you have your needs. Nobody wants to waste time jumping between different stores in a crowded place to search for pair of shoes. In other scenarios, you are in a location where you have no idea what is surrounding you, what is closest, which one is open or not. The closest areas are identified by a destination from the location of the driver. Usually, the driver is located in a centre of a circle defined by a radius which covers the needed search space. Researches in the context-aware recommendation systems and frameworks had less focus on the radius search and most of the time the radius search methods are based on either a constant incremented value or user defined value.

In this paper, we are proposing an optimized radius searching method by utilizing different components for the purpose of creating a high level of satisfaction for the user and delivering an acceptable level of competition between stores/areas. The remaining parts of the paper introduce related work in the filed followed by details of the proposed system. Discussion and analysis are then presented followed by conclusion and future work.

2. Related Work

In this sub section, recent literatures were reviewed to investigate researches that have contributed on the radius level associated with a travel/path planning for the use of recommender systems in the transportation domain.

Narman et al. [4] produced a ride sharing recommender model to overcome problems such as pollution, fuel consumption and congestion. The model is based on rider's characteristics (such as safety, punctuality, chattiness, comfortability and friendliness) and user waiting threshold time. At the end of trips, feedback is collected from users according to the mentioned characteristics. The characteristics are supplied to a machine-learning classification module. For every rider there are two major characteristics identified among the others. Similar characteristics (especially these two major characteristics,) between riders, are recommended to each other for future journeys. From radius perspective, there was not much focus and not much details about the radius, even the size of the radius is not mentioned.

Bahramian et al. [5] a hybrid interactive context-aware tourism recommender system. The system considered users' feedbacks in addition to contextual information and preferences. The system utilized artificial intelligence by combining a case-based reasoning framework and an artificial neural network. Nearest- neighbour methods in terms of user effort, accuracy, and user satisfaction were implemented. The focus on the distance identified by the search radius was out of concern in the study.

An intelligent context-aware management framework for cold chain logistics distribution was proposed Li et al. [6]. The system consists of acquisition, risk management, recommender systems, tracing back and user portrait frameworks. The main purpose of the system is to enhance recommendation systems in the logistic domain by focusing on cold start problem and data sparsity. For radius search matter, the system uses a fixed radius (60 kilometers) from the driver's location to deliver recommendation about cargo information. The identified radius amount could be in this scenario acceptable, however, there was no justification for Fig. 1 and why not for example having 70 or 80 kilometers.

Design and implementation of audio augmented reality systems in urban tourism were investigated [7]. Purpose of the system was to deliver speech-based information about surrounding tourist sights. The system utilized users' location to support tourists' exploration of open urban environments. The system supported two major functionalities: exploration and route planning/navigation. A mobile application was developed to host the system for easy use by tourists. The system utilized a radius-based approach to identify activation zones near the tourist sights. The radius around the sights was strategically and manually placed for each sight. The radius values were set between 25 and 65 meters.

Dadouchi and Agard [8] introduced a recommender system that can accommodate delivery constraints in product recommendations. A methodology to adjust product recommendations was proposed in order to move customers' interests towards specific products with consideration for remaining unit loads of scheduled deliveries. Quasireal-time information about the supply chain was considered to enhance the number of shippable products in a recommendation list. Accordingly, a possible improvement in truck-load utilization, lower operation costs and reduced lead-times for delivery can be achieved. A list of active trucks was determined based on selected parameters. Authors determined a perimeter with a radius of 350 km around a user. There is no justification for the 350 km and no dynamic behavior was sketched.

There is various other research in different sub domains such as smart parking recommender systems [9-12], healthcare recommender systems [13, 14], and location-based recommender systems [15], the use of radius as part of the search space was either a fixed distance or unclear determination of the radius value.

It can be seen that from the above discussed literature, there is a lack of explicitly defining the value of the radius search. Furthermore, these was also a lack introducing a level of autonomicity that is so important in the design of context-aware recommender systems.

3. Dynamic Radius Search Method

In this section, the radius method is introduced. First, mathematical representation is introduced to determine how agglomeration and competition zones can facilitate the search outcomes followed by proposed search radius algorithm.

3.1. Search radius

Major literatures apply a fixed radius to do the search to find nearby locations. Other literatures ask the user to input manually the needed search radius. To complement the autonomous and contextual-aware performance which characterizes the proposed framework, an algorithm is needed to control the search radius in a smart way such that no intervention from the user/driver is required.

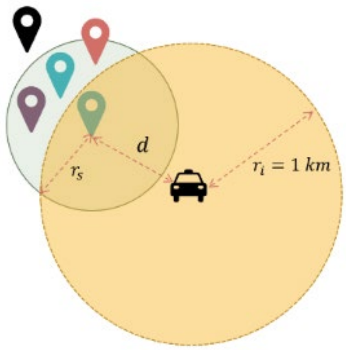

Fig. 1. Dynamic search radius layout [2].

Mathematical relation between agglomeration area D_g and competition area D_c denoted by $D_g \leq D_c$ is utilized to deliver needed cluster for the search radius [2], and to control the agglomeration effect [3]. Agglomeration area is the radius area associated with a store location placed close to its competitors for the purpose of getting more consumers, whereas competition area is the radius area associated with a store location placed far from its adversaries to control price.

Initially (the first search attempt by the user since he subscribed to the service) the D_c search area with radius r_i is setup by default to 1 km from the user's location. If the search returns results, then the distance between closest location and driver's location will be the search area D_g with radius r_s. The parameter r_s is a supportive radius that will help giving a fair chance to other nearby locations that could possibly have better $g(x)$ score than the closest location. The algorithm will use the supportive radius to do a second search starting from the closest location as depicted in Fig. 2. To always satisfy the inequality $D_g \leq D_c$ the supportive search radius will be $r_s = \frac{d}{r_i}$. If the initial search returned no results then r_i will exponentially incremented such that the next values for r_i are 2, 4, 8, …, n. The reason for having such a dynamic search radius and an exponential increment is to eliminate number of iterations for each radius r_i which will reduce computation and to limit the results for the user as it is not convenient to proceed with further outputs while the user purpose is to reach the nearby one. The results of the initial radius allocation algorithm will be fed into a deep neural network (DNN). The DNN purpose is to predict radius value for future request either from same driver or other one. Five values will be fed into the DNN that comprises item/service/info, driver's location (determined by GPS), radius, time and day. Output of the DNN is the result of the search (either 1 or 0) where 1 and 0 represent 'RESULT FOUND' and 'NO RESULTS' respectively. After having the DNN trained (at least one time) in real time, then any upcoming new search will be processed as per the following algorithm:

i. User apply a search request (using for example a mobile) for item/service/info from the intended database.
ii. The input request will proceed in the DNN (including the five values) to decide the output (0 or 1).
iii. If DNN output is 1 ➔ there is a result and accordingly the intended area defined by radius r_i will be considered.
iv. Output will be sorted in descending order based on score within the area identified by radius r_i.
v. All results within the nominated radius will be displayed to the user.
vi. If output is 0 ➔ there is no result ➔ increase radius exponentially.
vii. If results found, then display them to the user and then train DNN in the background.
viii. If results are not found, then go to step 6 again.

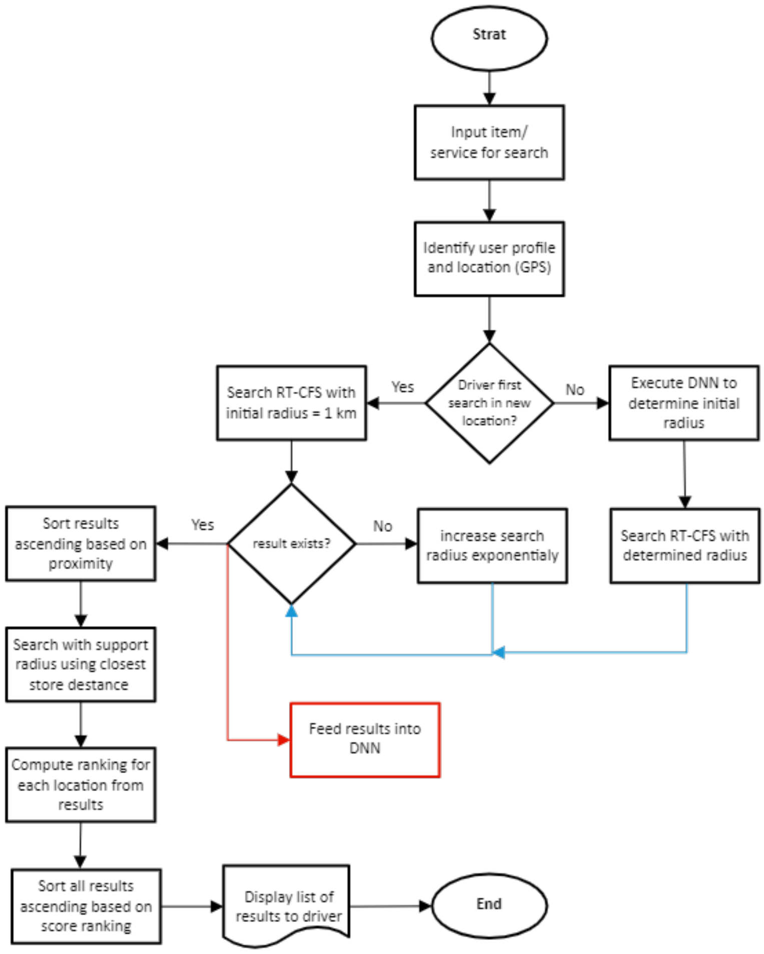

Fig. 2. Proposed algorithm for search radius determination using DNN.

3.2. Worst case scenario

There is a possibility of having a scenario that may lead to add further iterations, see Fig. 3. In case a store at location l_1 is found (caused by a request triggered by driver A) at the edge of a competition area with radius $r_i = d$. Since it is vital to include an agglomeration area with radius $r_s = \frac{d}{r_i}$, then $r_s = 1$ km.

Similarly, another store at location l_2 is found at the edge of the agglomeration area with radius r_s that has better $g(x)$ score than l_1. In this case the store at l_2 will be on the top of the result list for driver-A.

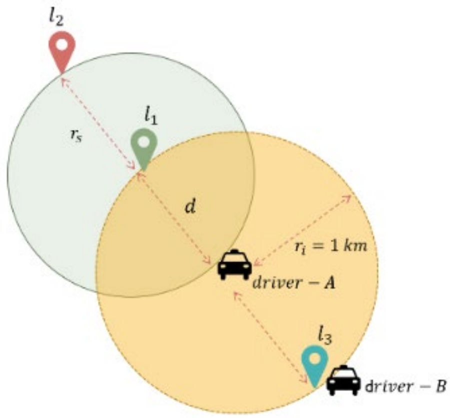

Fig. 3. Worst case scenario for two drivers in the same competition area requesting the same service/item [3].

Now let suppose a driver-B at location l_3 started a search for surrounded same service, then what would be a possible radius value that allows the driver to find results from the first iteration search/check? As we can see driver-B is in the area that was already been searched by driver-A. This can give driver-B a chance to get same results received by driver-A if we consider the right radius value. To achieve a radius value that can optimize the search while giving driver-B an opportunity closer to the one achieved by driver-A (considered nearby driver), the following equation is proposed for the initial radius.

$$r_i = 2d + r_s \quad (1)$$

In this matter, driver-B has a search radius that can cover up to location l_2 which can increase the probability for driver-B to have results searched by a closer driver in the history of the search while at the same time minimizes radius iteration increment.

Stage one of the processes, the driver searches for a place or a service, the database sends information to the driver, this information are stored in a knowledge base. Stage two of the process, the driver starts the journey from the source to the destination. Stage three, the feedback of the driver about the location, availability, services is received (via direct feedback or comments). Stage four; based on the feedback from the driver, we will consider adding the safety category corresponding to the feedback.

4. Discussion and Analysis

In the previous section, dynamic radius method is introduced with some major scenarios illustrated. In this section, other scenarios are investigated.

Recalling that, a driver is either in a new or in existed/visited radius. As depicted in Fig. 4, a historic location was visited by a user with positive feedback fed into a database. Accordingly, the following sub-cases are expected to occur when another driver is in the same radius:

Sub-Case-1; within a radius of one km there exist at least one service location that was recommended & visited by at least one driver as depicted in the Fig. 4 above. The historic user could be from within or outside the radius.

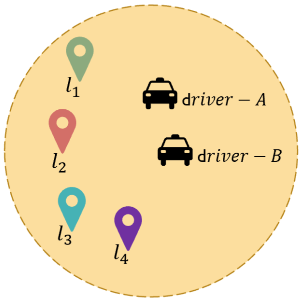

Fig. 4. Illustration of two drivers in a same radius.

Sub-Case-2; within a radius of one km there are no existed locations that were recommended and visited by any user. This means at least there is one location in the radius that was not recommended nor visited.

Sub-Case-3; within one kilometre radius there are no existed locations visited by any users, but at least there existed one location that was at least recommended. The recommendation could be from within or outside the radius.

Sub-Case-4; within one km radius there is at least one location visited but was not recommended. This situation is not ideal (a special discussion for this case is introduced at the end of this section).

Sub-Case-5; There is at least one service/item location within one km radius exists but not recommended nor visited.

We assume the two drivers are in very close locations such that their radiuses are almost identical.

Let us examine for sub-case-1 by assuming L3 was recommended and visited by driver/user A. In this case, driver B will search for 1 km radius to find a set of locations {L1, L2, L3, L4}. Since L3 was recommended to user A (this could be provided by DNN) driver B would check two things: 1) the results of the radius search set S1= {L1, L2, L3, L4}; the second thing to check is the information from the DNN if any of the founded locations were recommended or visited by another user. This way the DNN would provide the set S2={L3}.

By applying union operation on the two sets yields $S1 \cup S2 = \{L1, L2, L3, L4\}$. This is what we consider as the search space of the results for the intended radius optimized.

For the equation to be generalized:

$$DNN_{Results} \cap Radius_{Results} \quad (2)$$

What if historic driver/user is outside the radius as depicted in Fig. 5.? By applying equation (2) the algorithm will provide same results.

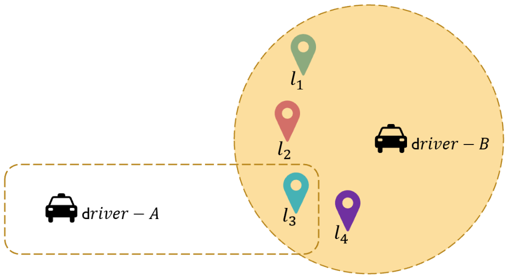

Fig. 5. A case scenario where driver A is located outside the radius and historically L3 was recommended and visited.

Another possible case is when the two drivers are far from each other such that we have two different radiuses as shown in Fig.6.

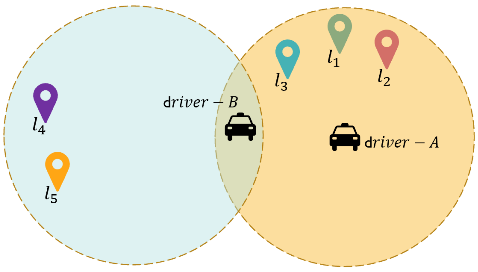

Fig. 6. Two drivers located within two intersected radiances.

In this scenario, result is S1={L4, L5}, while DNN results is s2={L1, L2}(considering L1 and L2 were recommended by drivers belong to the radius). Consequently. S1 union S2 yields {L1, L2, L4, L5}.

5. Conclusions

In this paper, we were able to propose a system that can enhance path planning recommendation by focusing on radius search as a major building block. The radius search method was optimized from traditional static incremental or user defined value to a dynamic mechanism where user has zero intervention. It is not easy to introduce a high level of satisfaction for stores/shops such that each one can have a fair opportunity to be included in the search results when a driver search for a nearby store. However, delivering a standardized approach for business owners on how to participate in such smart environment and its subsystems can improve their level of satisfaction. Context-aware recommender systems in the field of transportation is a promising domain that can contribute to level of autonomicity in smart cities. As a future work, a simulation for the presented system can be done to validate the expected results and further identify weaknesses.

References

1. Ahlers, D. (2020). Making sense of the urban future: Recommendation systems in smart cities. *CEUR workshop proceedings*.
2. Falk, K. (2019). *Practical recommender systems*. Simon and Schuster.
3. Aggarwal, C.C. (2016). *Recommender systems.* (Vol.1). Cham: Springer international publishing.
4. Narman, H.S.; Malik, H.; and Yatnalkar, G. (2021). An enhanced ride sharing model based on human characteristics, machine learning recommender system, and user threshold time. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 13-26.
5. Bahramian, Z.; Ali Abbaspour, R.; and Claramunt, C. (2017). A cold start context-aware recommender system for tour planning using artificial neural network and case based reasoning. *Mobile Information Systems*, Article ID 9364903, 1-18.
6. Li, X.; Wang, Z.; Gao, S.; Hu, R.; Zhu, Q.; and Wang, L. (2019). An intelligent context-aware management framework for cold chain logistics distribution. *IEEE Transactions on Intelligent Transportation Systems*, 20(12), 4553-4566.
7. Boletsis, C.; and Chasanidou, D. (2018). Smart tourism in cities: Exploring urban destinations with audio augmented reality. *In Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference,* 515-521.
8. Dadouchi, C.; and Agard, B. (2021). Recommender systems as an agility enabler in supply chain management. *Journal of Intelligent Manufacturing*, 32(5), 1229-1248.
9. Saleem, Y.; Rehmani, M.H.; Crespi, N.; and Minerva, R. (2021). Parking recommender system privacy preservation through anonymization and differential privacy. *Engineering Reports,* 3(2), e12297.
10. Rizvi, S.R.; Zehra, S.; and Olariu, S. (2018). Aspire: An agent-oriented smart parking recommendation system for smart cities. *IEEE Intelligent Transportation Systems Magazine,* 11(4), 48-61.
11. Guzmán, G.; Torres-Ruiz, M.; Tambonero, V.; Lytras, M.D.; López-Ramírez, B.; Quintero, R.; Moreno-Ibarra, M.; and Alhalabi, W. (2018). A collaborative framework for sensing abnormal heart rate based on a recommender system: Semantic recommender system for healthcare. *Journal of Medical and Biological Engineering,* 38(6), 1026-1045.
12. Dharia, S.; Eirinaki, M.; Jain, V.; Patel, J.; Varlamis, I.; Vora, J.; and Yamauchi, R. (2018). Social recommendations for personalized fitness assistance. *Personal and Ubiquitous Computing,* 22(2), 245-257.
13. Alotaibi, R.; Alnazzawi, T.; and Hamza, N. (2021). A new location-based privacy protection algorithm with deep learning. *Security and Privacy*, 4(1), e139.
14. Li, Y.; and Liu, L. (2012). Assessing the impact of retail location on store performance: A comparison of Wal-Mart and Kmart stores in Cincinnati. *Applied Geography,* 32(2), 591-600.
15. Wang, J.; Tan, R.; Zhang, R. P.; and You, F. (2016). A recommender system research based on location-based social networks. *In International Conference on Social Computing and Social Media.* 81-90.