Recommendation System using iBeacons

Ekta Vaswani, Nipun Ahuja
Rohil Shah, Sumit Gerela

Computer Engineering Department, Charles W. Davidson College of Engineering,
San Jose State University, San Jose, CA 95112

ekta.b.vaswani@gmail.com, nipunahuja572@gmail.com, rohilshah70@gmail.com, sumitgerela@gmail.com

***Abstract*— Aim of the system is to design a recommendation system for Grocery Outlests using iBeacons to digitize the customer buying experience. This system aims to assist the customers with a simple iPhone application as front end to recommend products they might be interested to buy based on their location within the store which is determined using iBeacon.**

Research work for this system describes various recommendation algorithms and compares them. The research recommends techniques like clustering, association rules, search based models, similarity matrix, apriori algorithm that can be combined together to improve the overall accuracy of the system to avoid irrelevant suggestions. iBeacons minor and major ids are mapped to a department within the store and store location respectively. Based on the users location in the store, a beacon with respective major id, minor id and UUID will be used to trigger the recommendation system using pre-build association rules based on historical data. There are total of approximately 10,000 transactions for __ customers and around 160 product categories. 60% of this data will be used as training set and remaining 40% will be used as testing set. The proposed system will use Apriori algorithm to generate association rules combined with other techniques to improve the overall accuracy of the system.

***Keywords*— apriori algorithm, iBeacons, indoor recommendations, Recommendation System, Association Rule Mining.**

I INTRODUCTION

1. Overview of the Project

We all use smartphones or wearable these days to make our daily lives easy. When to it comes to our daily shopping needs, we do it with our gadgets and expect them to be more personal devices which spell out suggestions while shopping. These devices can become more powerful and personal when combined with your previous buying habits. Use-case at hand, aims to solve this problem for grocery outlet to optimize retail shopping experience for the customers.

This requires perfect integration of two technologies viz. Internet of Things and Data Science. To make this use-case work, we will be using iBeacons, which are small, low powered bluetooth device which are used to determine indoor position of the user. Once the user position is determined near a particular aisle, we run data mining algorithm to make variety of suggestions based on various parameters like customer historical buying habits, trending items in the store and much more

2. Importance

This use case solves various problems, from the consumer's perspective as well as commercial perspective. Below are some of the problem this use-case can address:

1. It eases customer decision making process while shopping from huge array of products within the stores.
2. It helps increase sale by using targeted marketing to recommend items, the customer is more likely to buy thereby increasing the sale of products.
3. It helps other customers as well by recommending products by using historical data of the entire sales within the store.
4. It helps the store owners to keep track of most shopped items and hence make the inventory changes.

These are some of the problems that can be solved by merging IoT and Data Science to optimize retail shopping experience.

3. Problems Solved

With the recent advancements in tech space, it looks like consumer with high end portable devices and wearable are capable of using them in every domain. But from the retailers perspective it is very important to develop an ecosystem which is not only reliable but also works as the consumers might expect from such services.

Recent study found out that 41% of consumers ignored the push notification sent to them because they were irrelevant to location and/or interests. This use-case will ensure that the machine learning algorithm along with iBeacons will avoid this problem by making accurate predictions based on the location and interests.

3. Outline of the Report

The rest of this report is structured as follows. Chapter 1 explains Project Introduction, description and related work. Chapter 2 explains Dataset Specification of the system and the Literature Survey. Chapter 3 explains Proposed System architecture. Chapter 4 deals with Conclusion and Future Work.

II LITERATURE SURVEY & DATASET EXPLANATION

1. Dataset Explanation:

The Dataset for Grocery outlet has around 10,000 customer transactions, 160 product categories and a small dataset for 3 iBeacons.

- Customer Transactions: This dataset contains customer id in the first field and other fields contain multiple products bought by customers along with transaction dates.
- Product Categories: This dataset contains product id, product name and product category it belongs to. There are around 160 product categories.
- Beacon dataset: Beacon dataset contains the major id, minor id, UUID and product categories this beacon is near to. This is a very small dataset that contains just 3 entries as the proposed system uses only 3 beacons for demonstration purpose. This dataset can be easily modified so that if the mode beacons need to be added in the store or some of them mapped to different categories.

We use 60% of this dataset for training purpose to verify the model and remaining 40% to for testing purpose to validate the model.

1. Literature Survey

Model of recommendation system

An effective recommendation system is made up of multiple algorithms and takes into consideration diverse forms of data and information available. Another factor to be considered is the scalability of the system as data is growing exponentially. Also real time analysis on this growing data has become the need of the day. This section analyses various recommendation algorithms in data mining. And creates a model for real time personalized recommendation system.

Data Mining is the process of extracting unknown and potentially useful information from a large diverse and distributed data set. It uses machine-learning techniques to predict, classify and recognize patterns in data. In a typical e-commerce application, user interacts with the application and this interaction builds up information that can be used to provide personalized recommendations to the user. A typical workflow of an e-commerce system consists of following steps:

1. Acquire Information: Information can be acquired explicitly by asking user preferences or implicitly by observing users clicks on web page and navigation behavior.
2. Information Preprocessing: This step selects the relevant data and transforms into the format required by the algorithm. This step determines the overall quality of the recommendation system.
3. Develop Recommendations: This is one of the crucial steps in which the data selected is used to form recommendations using multiple recommendation algorithms.
4. Display results to user: Recommendations are displayed to the user by personalizing them and displaying at appropriate time and format.

Different data sources that can be used for data mining are as follows:

1. Server Data: Log data i.e. server logs, error logs, transaction logs, etc.
2. Query Data: Data queries performed by each user.
3. Online market data: Data in relational databases about purchases, products, etc.
4. Web pages: web page data such as HTML, XML, images, audio, video, etc.
5. Hyperlinks: hyperlinks accessed by user on a page
6. Customer registration information: Demographic information and other relevant customer information.

Recommendation methods that can be used are as follows:

1. Non-personalized recommendation methods: These methods come up with same recommendations for all users based on average evaluation of statistical information of the product. They are easy to use and provide quick real time analytics but lack personalization.
2. Based on product attributes: Based on similarity of attributes with respect to user preferences either explicitly provided by the user or implicitly computed an algorithm.
3. Based on product associations: Based on association between commodities. The association rules are formed based on data in users shopping cart or purchase records.
4. Based on user associations: Based on association between users by analyzing user information for similarity factors.

Various data mining algorithms for recommendation systems are as follows:

1. Path Analysis: Used to determine frequently visited links or paths and design the website accordingly.
2. Association Rules: Determine correlations between pages visited together and/or items bought together so that the website can be structured accordingly.
3. Sequential Patterns: Determine chronological patterns of buying products by analyzing timestamps so that the products can be targeted and marketed accordingly.
4. Classification and Prediction: Classify items to specialized groups based on certain attributes which can be later used to predict certain features or perform functions for the classified group.
5. Cluster Analysis: Customers with similar attributes are grouped together to facilitate future marketing strategies and personalized services.
6. Anomaly Detection: Determine unusual behavior or attributes which can be used for fraud detection, network intrusion detection, etc.

Once recommendations have been formed, they are displayed to the user using following techniques:

1. Display the most popular product.

2. When user queries for a specific product, display the result with products that match users interest first.
3. Recommend products based on user interests while user is browsing the website.
4. Provide evaluation of products that the user is visiting.
5. Recommend products based on products already in the shopping cart.
6. Send email regarding products the user might be interested in.

Personalized recommendation system is made up of two modules- online and offline. The offline module creates recommendation mode rules and thus provides support to online module. Online module consists of recommendation engine, which forms real time recommendations based on rules. These recommendations are again fed to the rules in offline module. The offline module is made up of data preprocessing unit (data cleaning, user, session, transaction identification), characteristics mining unit (recommendation algorithm, rule type library, mode analysis, evaluation) and recommendation rule library. The offline module preprocesses data, chooses an algorithm from the recommendation algorithm library based on rule type and produces mode rules which are saved in mode rule library. Online module uses recommendation engine based on customer information, calls mode rule from library and uses data to form recommendation set. It calculates weighted average of the set based on user information and forms a personalized recommendation set and displays it. Different modules in this model can be developed and updated independently. Also it supports efficient real time use of different algorithms based on input data.

Analysis of iBeacons implementation techniques:

Apple brought a key feature to generate location-based marketing through "iBeacon" with the launch of iOS 7. iBeacon is actually Apple's name for Bluetooth Low Energy emitters. Apple's integration of iBeacon with iOS and other devices has given a great start to the wide use of iBeacon. This section discusses five implementation of iBeacon that are currently successfully used worldwide.

1. Connection between iBeacon and web landing page: This is the simplest implementation of iBeacon which connects an iBeacon with a URL. A small innovative startup from Florida called Kinwa has managed to do this through their app called "Bubble". Bubble helps to make this link and connect to the iBeacon. This implementation is perfect in cases like indoor or outdoor customer information, small organizations, local museums etc. Kinwa has already started implementing this in their hometown Gainseville, Florida from their airports to car agencies
2. Generic app that can be customized: mApp developed by a Dutch company LabWerk is such an app that can be customized depending to the customer requirements. This app was developed for museums and is used by the Tulpen Museum in The Netherlands. It provides a rich user interface and simplifies the deployment for a business by providing strong tools
3. Integrate with existing app: Another way of using iBeacon is to implement it with an existing app. These apps can be shopping center app, brands or grocery outlets. Brands can also use iBeacons that are already deployed and develop their app from already existing iBeacon network. One problem of this type of implementation is that if many brands use the same app, then too many messages are sent to the user which lets to the customer leaving the app.
4. Add location based feature to an app: One way is to build upon an already existing app to add value to the app. Downside of this method is that additional development is required on the app and iBeacon used will be brand specific. This method can be costly as iBeacon deployment will be managed by each brand and number of beacons used per location will also be more
5. Passive tracking of customer behavior: So far all the methods discussed above have some or the interaction with the customer. But there can also be some interaction between mobile app and iBeacon in the background without interacting with the customer. This type of implementation can either be for market analytics purpose or for a reward program. For example, ANKAMall in Istanbul, Turkey has a reward program wherein when a customer passes by a shop in the mall, it simply increases the credit score of that customer without sending any notification to the customer.

Market Basket Analysis using Frequent Itemset Mining:

Market basket analysis using Frequent Itemset Mining technique has been used widely to mine frequent patterns to recommend most frequent items from the available dataset D. Association rules are derived based on two important parameters viz. support and confidence. Frequent itemset mining can be done using algorithms like Apriori, K-Apriori and FP-Growth Algorithm.

Apriori algorithm uses an iterative approach known as level-wise search where k-itemsets are used to explore (k+1)-itemsets. Initially, the algorithm starts finding the 1-itemset which is basically the all the items which qualify to be considered for next set that satisfies the minimum support. All the items that qualify are inserted into a dataset L1. Now, L1 is used to determine L2 and so on. The algorithm finds all the datasets until there are no more items available to scan from the available datasets. So to form individual datasets, it requires one complete scan on the D. This approach becomes expensive as there are continuous scans on D. To improvise this level-wise approach, the algorithm uses an Apriori Property to reduce the number of scans required every-time. This property is based on the observation that All non-empty subsets of a frequent itemset must also be frequent.

Apriori algorithm can be described as follows:

- Input: Database D and Minimum support count threshold min_support.

- Output: L frequent itemsets in D
- Approach:

```
L1 = find_frequent_1-itemsets(D)
k=2
while (Lk-1 is not empty){
Form Candidate set Ck from Lk-1;
for each transaction t in D
{
Get the subset Ct for all t's qualifying as candidates;
for each Candidate c in Ct
        increment the count;
}
Lk = set of all candidates such that its count is greater than min_support.
}
return L
```

The most efficient approach which is K-Apriori uses the above defined apriori property to to enhance the performance.

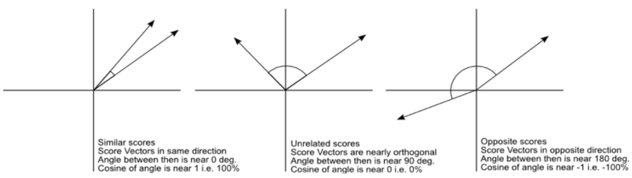
Traditional Collaborative Filtering:

A customer is represented by an N dimensional vector where N is the distinct catalog items. In this algorithm the component of vector is positive for every purchased or positively rated item. In order to compensate for the best selling values the vector component is multiplied by the inverse of number of customer who have purchased or have rated the item.

It is to be noted that this algorithm generated recommendation based on a very few customers(customers who are most similar to the user). Similarity between two customers can measured in various ways. One measure can be the measure of cosine of angle between two vectors.

$$similarity(\vec{A},\vec{B}) = \cos(\vec{A},\vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| \bullet \|\vec{B}\|}$$

The Cosine Similarity values for different vectors, 1 (same direction), 0 (90 deg.), -1 (opposite directions).



Generating recommendations using collaborative filtering is computationally expensive. It is O(MN) in the worst case, where M is Number of customers and N is the product catalog. Generally speaking the average case complexity of this algorithm is O(M+N) as the average customer vector is extremely sparse.

There are some performance and scaling issues that are related to this algorithm when the datasets are large. Although these performance issues can be addressed by using dimensionality reduction techniques like PCAs and Clustering, using these methods also leads to poor recommendation quality.

Collaborative Filtering using Real-time User Interest Model:

Recommendation by Collaborative Filtering Recommendation Algorithm based on Real-time User Interest Model is described as follows:

1. Use users' labeling information to form mappings for standard labels through the process of standardization; displaying users' interest in the real-time users' interest model, user interest model is in short; standard labels are fewer and more stable than items.
2. The measurement of users' interest degree is objective, different which needs to consider the rating differentiation of different users.
3. Use users' interest model vectors to build linear regression equation to estimate users' interest degree of every single item using the formula given below, then make recommendations to users by Top-N method.

User u_i is interest in the project p_j, calculated as:

$$I_{ij} = \sum_{t_k \in T} w_k \times p_k^j$$

$$\text{Where, } p_k^j = \begin{cases} 0 & t_k \notin \text{text feature of item} \\ 1 & t_k \in \text{text feature of item} \end{cases}$$

Item to Item Collaborative Filtering:

Item-to-Item collaborative filtering matches each of the user's purchased and rated items to similar items, then it combines those similar items into a recommendation list.

To find the most similar match for an item this algorithm builds a similar items table by finding items that are bought together. Similarity in this algorithm is computed by the cosine measure. Although some other similarity measure can also be incorporated for finding the similarity. The similarity between a single product and all related items is calculated by the following approach.

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by
       customer C
      Record that a customer purchased I1
       and I2
  For each item I2
    Compute the similarity between I1 and I2
```

Worst case complexity of this algorithm is $O(N^2M)$ but in practice complexity is O(NM) as most of the users have few purchases.

One area in which this algorithm shines is scalability and performance as similar-items table is created offline. Also, finding similar items to the user's purchases and ratings scales-up independently of the catalog size. It also performs well with a limited amount of user-data. One problem with this algorithm is that it requires feedback loop.

Cluster Models:

To find customers who are similar to user this algorithm assigns a user to a cluster which has the max no. of similar customers. It then uses the purchases and ratings to provide recommendations.

In the real world, greedy approach for cluster generation is used as the datasets are large and generating optimal clusters on a large dataset is somewhat impractical. In the greedy approach, one starts with an initial set of segments which often contain randomly selected customers each. These algorithms then match customers to existing clusters, by either creating new clusters or by merging existing clusters.

After the clusters have been generated, the algorithm then computes user's similarity to vectors of each cluster. It then chooses the cluster with highest similarity and classifies the user accordingly.

Cluster models compares the user to a number of segments rather than the entire customer base which leads to better online scalability and performance than collaborative filtering.

The computationally expensive clustering runs offline but recommendation quality is still low. Recommendation quality can be improved a bit by increasing the number of segments/ clusters, but this makes the online user-segment classification more expensive.

Search Based Model:

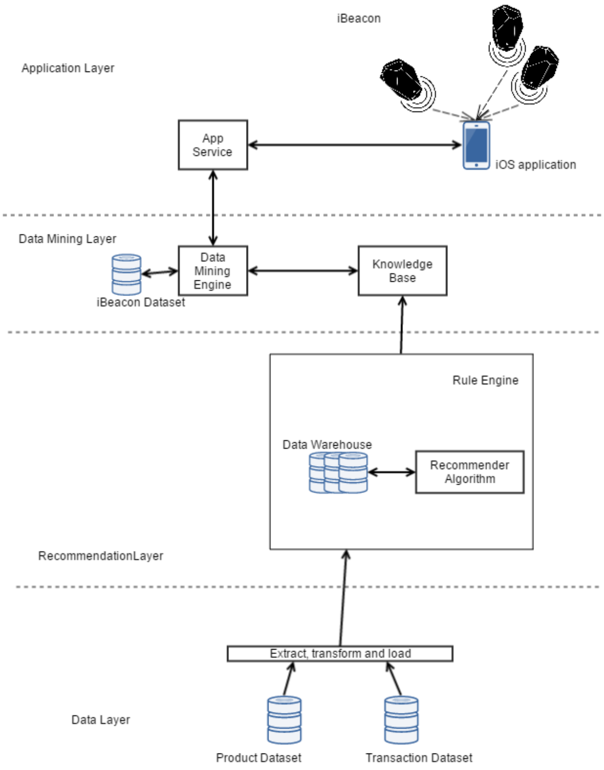
These models treats recommendation problems as a search for related items. Once the user has bought or has rated a particular item the algorithm then constructs a search query to find other popular products with the similar keyword, or by the same author, or by the same director, or by the same artist.

This algorithm scales and performs well when the user has few purchases or ratings and fails for users with thousand of purchases as it is somewhat impractical to base a query on all the items.

The recommendation quality is poor in all the cases as the recommendation are either too generic or are too narrow.

III SYSTEM ARCHITECTURE

The architecture for proposed recommendation system using iBeacon is as follows:



The overall system is divided into following layers:

1. Application Layer: At this layer, we have range of iBeacon that continuously broadcast information in a specific format. To be specific, every iBeacon has an UUID, Major ID and Minor ID. UUID stands for Universally Unique ID and is completely random string. This can be considered as the top-level information about your system. As developer, When you generate any specific UUID and assign in to the beacons, an application that wants to communicate with the beacons does so by identifying its UUID. To add more granular information, iBeacon have Major ID and Minor ID. Since the proposed system is targeted for Grocery outlets, every store will have major id and every department in the store will have minor id. So an iBeacon in San Jose store for Vegetable department will have different set of major and minor id compared to an iBeacon for Santa Clara store for Vegetable department. Hence an app can communicate and know, which beacons it is talking to. Once the app, receives this information, necessary actions can be taken to trigger the recommendation system and display the notification on the customer iPhone.
2. Data Mining Layer: This layer has three most important components of the system viz. Data Mining Engine, Knowledge Base and iBeacon Dataset. iBeacon data set contains the mapping of iBeacon information to the category of products they refer to within the store. Our system will have 3 iBeacon hence this data set would

contain only three entries. Knowledge Base represents the set of association rules that are generated by the Recommendation Layer. These association rules along with the iBeacon dataset is used by Data Mining Engine to come out with the notifications that can be sent to the user. Data Mining Engine is the main core that is responsible for sending the response back to the user. Data Mining engine can scale up with the growing data set.

3. Recommendation Layer: This layer is responsible for running the Apriori Algorithm explained in literature section. Apriori Algorithm is an Association Rule Mining technique which is an important data mining model for Market Basket Analysis. It finds items frequently purchased items and relation between them. This is achieved by calculating support and confidence variables to ensure the correctness of the Association Rule to be used for recommendation. Consider X —> Y in a transaction V. Support s is the probability that a transaction contains X U Y , where X,Y are items in a Transaction v which belongs to a the set of Transaction

$$Support = \frac{frq(X,Y)}{N}$$

Confidence c is the conditional probability that a transaction having an item X also contains the item Z.

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

The rule engine is combination of Data warehouse and recommender algorithm and pre-calculates the rules as soon as the data is added to data warehouse. So the system provides, real time data analytics to the users as soon as they change the location in the store.

4. Data Layer: This layer consist of the two data sets viz. Product Dataset and Transaction Dataset. Product dataset consist of all the products available in the store along with the category. Transaction dataset consists of all the transactions for particular set of customers for certain period of time. This data is extracted at regular intervals from the mentioned datasets and transformed into the format that is suitable for Data warehouse and loaded to it. This ETL process can be configured to run weekly, daily or so on.

Data Flow:

1. When a customer enters a store, there are multiple iBeacon across multiple aisles. Each beacon broadcasts the information in a specified format. Beacon dataset consists of major id, minor id, UUID and product categories.
2. Customers have to initially sign in to the iPhone application on the device so that system knows about the customer id when they enter the store and hence trigger personalized location based recommendations. When customer comes near a particular beacon, it receives the beacons information and app determines the best beacons information to use for the recommendations. Suppose beacon major id is 12 and minor id is 20 and UUID is say B558CBDA-4472-4211-A350-FF1196FFE8C8. When app receives this information, the app transmits this from application layer to data mining layer.
3. At this layer, the beacons data and customer data is used along with the knowledge base by the data mining engine to determine the personal recommendations for the customer. The UUID is mapped from the iBeacon dataset to determine the categories of items located within the specified iBeacon range. For example, iBeacon with UUID B558CBDA-4472-4211-A350-FF1196FFE8C8 has category C101 mapped to it. This category is made up of items I1, I2, I3, I4, I5, I6, I7.
4. Data Layer is made up of product dataset and transaction dataset. Product dataset consists of Item ID, Item names.Transaction dataset is made up of transaction date, items bought and customer ID. This data is extracted, transformed and loaded in the data warehouse. Recommender algorithm which here is Apriori algorithm executed on the data in data warehouse renders various association rules. For example

   Users who buy I1 also buy I13 (support = 30%, Confidence = 80%)

   Users who but I12 also buy I4 (support = 50%, Confidence = 88%)

   Users who buy I5 also buy I1 (support = 33%, Confidence = 60%)

   Users who but I22 also buy I6 (support = 20%, Confidence = 78%)
5. Rule engine feeds this information to knowledge base in data mining layer. Also, knowledge base consists of transaction history for each customer. Transaction history is made up all the items the user has bought till date and the count of each item. Suppose transaction history of customer E13 who has sent UUID B558CBDA-4472-4211-A350-FF1196FFE8C8 is I1, I5, I15, I12. When this customer enters aisle with category C101, data mining engine executes following steps to recommend an item:

   Receive UUID and extract category i.e C101

   Use category to extract items in that category i.e I1, I2, I3, I4, I5, I6, I7

   Use customer ID i.e E13 to extract transaction history i.e I1, I5, I15, I12

   Extract association rules from knowledge base

   Use transaction history to consider only those rules that match LHS of the rules i.e.

   Users who buy I1 also buy I13 (support = 30%, Confidence = 80%)

Users who buy I5 also buy I1 (support = 33%, Confidence = 60%)

Use items in category C101 to filter above rules such that only those are considered which match RHS of the rule i.e

Users who buy I5 also buy I1 (support = 33%, Confidence = 60%)

6. Thus, the data mining engine will suggest item I1 when the customer enters aisle with iBeacon UUID B558CBDA-4472-4211-A350-FF1196FFE8C8 based on customer E13's transaction history.

CONCLUSION

Association rule mining is a very useful technique to discover customer's buying habits by using the store's database to generate rules. This helps store to strategically place their goods on the shelf based on the customer's buying habits and most frequently bought items. Related products can be placed in the same section which can increase customer satisfaction and also increase store's profit. We'll be developing an iOS app for the project which would be integrated with iBeacons to notify customers about products based on the rules generated. After comparing different techniques, we came to a conclusion that Apriori algorithm would be a perfect fit for this project.

In the future, this application can be integrated with other devices such as Apple Watch or Android device.

ACKNOWLEDGEMENT

We are deeply indebted to Professor Chandrasekar Vuppalapati for his invaluable comments and assistance in the preparation for this study.

REFERENCES

[1] C. Matheus, P. Chan and G. Piatetsky-Shapiro, 'Systems for knowledge discovery in databases', IEEE Transactions on Knowledge and Data Engineering, vol. 5, no. 6, pp. 903-913, 1993.

[2] G. Linden, B. Smith and J. York, 'Amazon.com recommendations: item-to-item collaborative filtering', IEEE Internet Comput., vol. 7, no. 1, pp. 76-80, 2003.

[3] J. Han, M. Kamber and J. Pei, Data mining: Concepts and techniques. San Francisco: Morgan Kaufmann Publishers In, 2011.

[4] N. Newman, 'Apple iBeacon technology briefing', J Direct Data Digit Mark Pract, vol. 15, no. 3, pp. 222-225, 2014.

[5] R. Thomadsen, 'Product Positioning and Competition: The Role of Location in the Fast Food Industry', Marketing Science, vol. 26, no. 6, pp. 792-804, 2007.

[6] Quora.com, 'How does Amazon's collaborative-filtering recommendation engine work? - Quora', 2015. [Online]. Available: http://www.quora.com/How-does-Amazons-collaborative-filtering-recommendation-engine-work. [Accessed: 08- Apr- 2015].

[7] S. Hilmi Ozguc, 'iBeacons and the future of retail shopping: Consumers are ready, but are retailers? | VentureBeat | Mobile | by Meghan Kelly', Venturebeat.com, 2014. [Online]. Available: http://venturebeat.com/2014/01/15/ibeacon-retailers/. [Accessed: 08- Apr- 2015].

[8] S. Gong, 'Personalized Recommendation System Based on Association Rules Mining and Collaborative Filtering', AMM, vol. 39, pp. 540-544, 2010.

[9] Y. Chen, K. Tang, R. Shen and Y. Hu, 'Market basket analysis in a multiple store environment', Decision Support Systems, vol. 40, no. 2, pp. 339-354, 2005.

[10] Y. Fan, Y. Shen and J. Mai, 'Study of the Model of E-commerce Personalized Recommendation System Based on Data Mining', 2008 International Symposium on Electronic Commerce and Security, 2008.

[11] J. Woo, 'Market Basket Analysis algorithms with MapReduce', WIREs Data Mining Knowl Discov, vol. 3, no. 6, pp. 445-452, 2013.

[12] H. Li, N. Zhang, J. Zhu, H. Cao and Y. Wang, 'Efficient frequent itemset mining methods over time-sensitive streams', Knowledge-Based Systems, vol. 56, pp. 281-298, 2014.