A Real-Time Targeted Recommender System for Supermarkets

Panayiotis Christodoulou[1], Klitos Christodoulou[2] and Andreas S. Andreou[1]
[1]*Department of Electrical/Computer Engineering and Informatics, Cyprus University of Technology*
31 Archbishop Kyprianos Street, Limassol, Cyprus
[2]*Department of Information Sciences, Neapolis University Pafos*
2 Danais Avenue, Paphos, Cyprus
paa.christodoulou@edu.cut.ac.cy, klitos.christodoulou@nup.ac.cy, andreas.andreou@cut.ac.cy

Keywords: Context-Aware Recommender Systems; Location-Based systems; Entropy-based Hard *k*-modes Clustering; Bayesian Inference; `iBeacon` indoor positioning system

Abstract: Supermarket customers find it difficult to choose from a large variety of products or be informed for the latest offers that exist in a store based on the items that they need or wish to purchase. This paper presents a framework for a Recommender System deployed in a supermarket setting with the aim of suggesting real-time personalized *offers* to customers. As customers navigate in a store, iBeacons push personalized notifications to their smart-devices informing them about offers that are likely to be of interest. The suggested approach combines an Entropy-based algorithm, a Hard *k*-modes clustering and a Bayesian Inference approach to notify customers about the best offers based on their shopping preferences. The proposed methodology improves the customer's overall shopping experience by suggesting personalized items with accuracy and efficiency. Simultaneously, the properties of the underlying techniques used by the proposed framework tackle the data sparsity, the cold-start problem and other scalability issues that are often met in Recommender Systems. A preliminary setup in a local supermarket confirms the validity of the proposed methodology, in terms of accuracy, outperforming the traditional *Collaborative Filtering* approaches of user-based and item-based.

1 INTRODUCTION

Recommender Systems (RS) are *intelligent engines* that collect information related to what a user has previously seen or bought, with the aim of providing personalized suggestions on unobserved items that are likely to be of interest. RS are often classified into three broad categories: (a) *Content-Based* (CB) systems, which use keywords to suggest items to a user similar to the ones preferred in the past (Konstan and Riedl, 2012); (b) *Collaborative Filtering* (CF) techniques, that recommend items based on the items a user has previously seen or bought. Often such systems use various strategies based on e.g., user-based, item-based, matrix factorization and clustering techniques to find correlations between users/items (Ning et al., 2015); (c) *Hybrid Recommendation* (HR) approaches, which use a combination of CB and CF methods to solve some of the limitations that exist in the aforementioned systems (Adomavicius and Tuzhilin, 2005). A brief description of these limitations follows.

CB systems provide better accuracy results when dealing with items containing textual information. However, such systems lack the ability to distinguish how well a text description is written from a badly one, especially in the case of using similar or related terms (Adomavicius and Tuzhilin, 2005). Additionally, such systems are often limited by the *over-specialization problem*; when a system recommends items with a high similarity compared to a user's profile, the user is likely to be suggested with items similar to the ones that has already seen (Lü et al., 2012). Furthermore, when a new user registers into the system and has few or no ratings at all, it is more likely to get low accuracy recommendations (known as the *cold-start* or *new-user problem*) (Adomavicius and Tuzhilin, 2005). As discussed in (Lü et al., 2012) CB systems require a significant number of ratings before suggesting items with high accuracy to a user.

In contrast to CB systems, CF approaches lead to poor performances due to the *sparsity problem* (Adomavicius and Tuzhilin, 2005). The number of items that exist on *e*-commerce websites is enormous; as a result, the most active users usually rate only a subset of the entire dataset. This means that many of the most popular items have few ratings and thus the likelihood of the system suggesting them is relatively low (Adomavicius and Tuzhilin, 2005) (Pu et al., 2012). Similarly to CB systems, CF systems require a significant amount of existing data on a user profile before being able to make accurate suggestions. In addition, new items need to be rated by a substantial number of users; otherwise the RS would not be able to come up with recommendations for them (Ning et al., 2015). In general, RS are challenged with *scalability issues*; taking into account the vast amount of data that exists on websites and applications, a considerable amount of computational power is needed to compute recommendations (Pu et al., 2012).

Summary of Contributions. This paper describes a framework for using a Recommender System in a supermarket environment that aims to suggest real-time personalized offers to customers using `iBeacons`. The following questions motivated this study. RQ1 – How does a Recommender System deals with data sparsity, the cold-start problem and scalability issues that may exist in a real-world application? RQ2 – How accurate and personalized are the recommendations provided by the system and how do they affect the store's revenue? In seeking answers to the aforementioned questions, this paper contributes the following: (a) the use of an Entropy-based algorithm and a Hard *k*-modes clustering algorithm to overcome the cold-start and data sparsity limitations; (b) the adoption of a rule-based system to discover sub-datasets of products, thus facing scalability issues; (c) the use of a Bayesian Inference model, that is trained on dynamic information provided by each user to predict whether an item is likely to be recommended or not; (d) the employment of a recommendation engine that suggests the top-*n* personalized items to the users through `iBeacons`.

The remainder of this paper is structured as follows. Section 2 presents an overview of the proposed methodology and discusses the technical background of the described Recommender System. Section 3 demonstrates the results from our preliminary experimentation. Section 4 reviews related work. Finally, Section 5 describes future work and concludes the paper.

2 OVERVIEW OF APPROACH

Modern supermarkets use loyalty schemes (e.g., loyalty cards) to reward repeated customers on purchases. In this work, we suggest an interactive rewarding scheme guided by a dynamic, real-time recommendation engine. More specifically, we propose the replacement of loyalty cards with an interactive smart-device application, which acts both as a bonus card and as a recommendation engine. This section presents an overview of the proposed methodology (as depicted in Fig. 1).

2.1 Bootstrapping

A simple application allows users to register or link their existing loyalty account to the system. A mobile application guides the registration process where each user is asked to develop its *personal profile* by providing information, such as: Full Name, Telephone Number, Address, Phone Number, any Nutrition/Diet Information, any Fasting or Vegetarian preferences, any Allergies, Health status, Nationality and Religion. An alternative option to registration is to link an existing loyalty card to the system. In this case, the customer is asked to provide only the information that is missing to build a personal profile. This process is important since profiles are used during the recommendation process.

From now on, we will refer to the information provided by each user as the user's *static information*. We should note here that static information is updated automatically when users manually make changes on their profiles. Apart from user's static information the system stores the user's dynamic information; which are the most recent transactions, historical transactions, location of the most visited departments and products to buy. We refer to such information as the user's *dynamic information*. Both *static* and *dynamic* information comprise a user's profile which is used to guide a personalized recommendation process according to his/her preferences.

2.2 Propagation of Recommendations

A novelty of our approach is that personalized recommendations are propagated in real-time to each user using `iBeacons` that reside within a certain range. The notification process takes place while users navigate into the store. In essence, `iBeacon` is a Bluetooth low-energy, wireless technology developed by Apple[1], which allows mobile applications (iOS or Android) to listen for signals from `iBeacons` that are in certain range. Broadly speaking, `iBeacons` consist of two processes: (i) the device that broadcasts the data (i.e., `iBeacon`), and (ii) an application installed on a

[1] `https://developer.apple.com/ibeacon/`

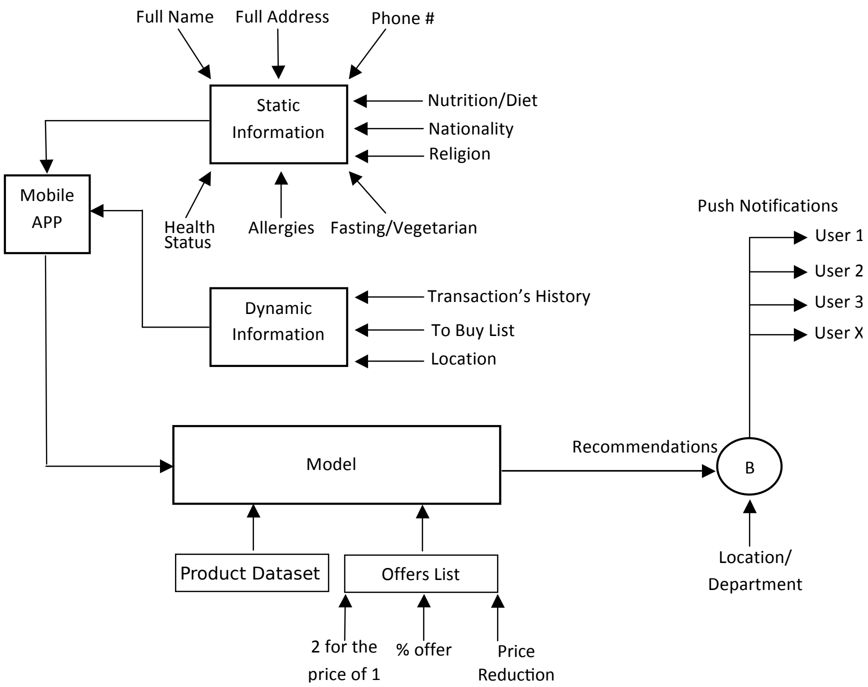

Figure 1: Overview of the Recommender System.

smart device, which acts as the recipient (Yang et al., 2015).

In a real-world setup `iBeacons` are distributed in the different departments of a store. In this work the `iBeacon` technology is used to push personalized recommendations to customers based on information (i.e., static and/or dynamic) from their profiles.

2.3 A Recommendation Model

In subsequent sections we describe in detail the algorithms that compose the proposed recommendation model. Firstly, we discuss how the *static information* is utilized by the system.

User's static information is used in a twofold purpose: (i) to identify the number of clusters k that exist in the dataset of registered users with the system, and (ii) to group each user in a cluster based on user's preferences. In cases where a user's profile is updated/altered, then the cluster where the user belongs is updated as well.

2.3.1 Determining the Number of Clusters

An *Entropy-based* approach is used as an external cluster evaluation measure (Stylianou and Andreou, 2007). It groups data objects with similar characteristics into clusters based on the entropy values of the objects using cosine similarity measure (Christodoulou et al., 2014). For the purposes of our framework the Entropy-based approach is used to compute the number of clusters that exist within the dataset of registered users (Christodoulou et al., 2014). In doing so the algorithm determines the *centroids* of each cluster utilizing users' unique static information.

The entropy value H_{ij} of two data objects X_i and X_j is defined as follows:

$$H_{ij} = E_{ij}\log_2(E_{ij}) - (1-E_{ij})\log_2(1-E_{ij}) \text{ for } i \neq j \quad (1)$$

where, E_{ij} is the similarity measure between the objects X_i and X_j that is computed using Equation 2.

$$E_{ij} = e^{-aD_{ij}} \quad (2)$$

D_{ij} is the distance between the objects X_i and X_j and a is calculated by $a = \dfrac{-ln(0.5)}{\bar{D}}$, where, $\bar{D}$ is the mean distance among all data objects.

The total entropy value of X_i with respect to all other data objects is computed as:

$$H_i = -\sum_{j=1, i\neq k}^{n} \left[E_{ij}\log_2(E_{ij}) - (1-E_{ij})\log_2(1-E_{ij})\right] \quad (3)$$

In more detail the algorithm proceeds with applying the following steps:

(i) Select a threshold of similarity β and set the initial number of clusters to $k = 0$.

(ii) Determine the total entropy values H for each data object X.

(iii) Set $k = k + 1$

(iv) Select the data object X_{min} with the least entropy H_{min} and set $Z_k = X_{min}$ as the k_{th} cluster center.

(v) Remove X_{min} and all data objects having similarity β.

(vi) If X is empty then terminate, otherwise go to Step 3.

2.3.2 Hard *k*-modes clustering

The Hard *k*-modes clustering algorithm (Stylianou and Andreou, 2007) groups categorical data by removing the *numeric-only* limitation (Huang and Ng, 1999) imposed by other clustering techniques (e.g., *k*-means), using a matching dissimilarity measure. This feature of the *k*-modes algorithm enables it to be used efficiently for clustering large categorical datasets. For the purposes of our framework and to deal with potentially a large dataset of users we consider Hard *k*-modes as a suitable clustering technique for grouping users in specific clusters in relation to the centroids. Due the nature of the clustering algorithm, each user belongs to a distinct cluster. The centroids for each cluster are discovered by applying the methodology as described in Section 2.3.1.

Given a pair of data objects $X_1, X_2 \in X$ defined by m attributes, where X is the set of all data objects, the dissimilarity between the pair of objects is defined as:

$$d(X_1, X_2) = \sum_{j=1}^{m} \delta(x_{1j}, x_{2j}) \text{ where,} \tag{4}$$

$$\delta(x_{1j}, x_{2j}) = \begin{cases} 0, & \text{if } x_{1j} = x_{2j} \\ 1, & \text{if } x_{1j} \neq x_{2j} \end{cases}$$

In the case of Hard *k*-modes clustering, if an object X_i in a given iteration has the shortest distance from a cluster center Z_i, it is represented by setting the value of the nearest cluster equal to 1 and the values of the other clusters to 0.

$$F(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li}^{a} d(Z_i, X_i) \text{ where,} \tag{5}$$

$W = [w_{li}]$ is a $k - \text{by} - n$ real matrix, $Z = [Z_1, ..., Z_k] \in \mathbb{R}^{mk}$, α is the fuzziness exponent, w_{li} is the weight degree of an object belonging to a cluster and k is the number of clusters.

For hard clustering $a = 1$, the weight degree w_{li} of an object belonging to a cluster is given as follows:

$$w_{li} = \begin{cases} 1, & \text{if } d(Z_i, X_i) \leq d(Z_h, X_i), 1 \leq i, l, h \leq k \\ 0, & otherwise \end{cases}$$

The Hard *k*-modes algorithm proceeds by applying the following steps:

(i) Select k initial modes, one for each cluster.

(ii) Allocate data object to a cluster whose mode is nearest to the selected one.

(iii) Compute the new modes for all clusters.

(iv) Repeat steps 2 and 3 until no data object has changed.

2.3.3 Pre-processing using a Rule-based System

Personalization techniques are deployed for various purposes in RS (Adomavicius and Tuzhilin, 2005). For the proposed methodology a rule-based personalization technique is used for pre-processing and filtering the overall dataset of products. This procedure takes place after clustering each user in a cluster containing users that share the same or similar characteristics. A set of rules is then applied to create sub-datasets of the overall dataset of products that are more suitable for each cluster.

To reduce the time needed for computing the clusters and hence the computation power necessary to derive the recommendations, this procedure is performed off-line.

The rules are associated with specific types of information inserted by the user, more specifically:

(i) Country (List of Countries) – The system takes into account the different demographic characteristics and food habits of each country.

(ii) Nationality (List of Nationalities) – Each nationality has its own characteristics, habits and preferences. The system takes into account such particularities.

(iii) Religion (List of Religions) – All religions have their own unique characteristics that are also considered by the proposed system.

(iv) Fasting (Enable/Disable) – Some religions have a fasting period, for example during Christmas and Easter for Christians. If a user enables this option the system determines any fasting habits, as well as, the starting and ending date of each period.

(v) Vegetarian (Enable/Disable) – When a user enables this option the system is suggesting products suitable for vegetarians only.

(vi) Diet (Enable/Disable) – If a user enables this option, the system immediately understands that a user is likely to be on a diet or wishes to start a new diet. The system also requires users to specify the type of diet: Low-calories, Low-carbohydrate, High-protein, Low-fat, etc. This information is taken into account during the recommendation stage.

(vii) Allergies (Enable/Disable) – If this option is enabled it means that a user is suffering from allergies on specific products. The user must choose the specific type of allergy is suffering from a list that is presented. Each allergy type is linked with products that must be avoided by the user. (a) Food Allergies, (b) Latex Allergy (c) Drug Allergy (d) Skin Allergy (e) Allergic rhinitis (f) Other (g) None

(viii) Health status (Enable/Disable) – If this option is enabled the user is asked to provide more information for a list of health indicators (a) Cholesterol (b) High Blood Pressure (c) Sugar (d) Depression (e) Other (f) None

The system considers a set of pre-defined rules and initializes them based on a certain order or priority. For example, if a user belongs to a religion group that forbids the consumption of beef then this is taken into account. In addition, any change on the rules updates the list of products of each cluster.

2.3.4 A Probabilistic Model

After clustering the users and finding the appropriate sub-dataset of products for each cluster of users, the system assesses the shopping preferences of each user to determine whether a product is suitable for purchase or not using a Bayesian Inference approach (Christodoulou et al., 2015). A training procedure takes place to estimate the likelihoods by taking into consideration characteristics from the user's transaction history (e.g., frequently bought products of a certain category and average price spent on products from a certain category).

Given a pair of products from the same category/sub-category, a Bayesian Inference approach similarly to (Christodoulou et al., 2015) is utilized to characterize with a probability, whether a product is suitable for purchase or not. To reason over such a hypothesis the model takes into consideration a set of evidence from the customer's transaction history.

To formalize the probabilistic model, we denote with H a Boolean hypothesis on whether a product is suitable for purchase or not. Let $E = e_1, e_2, \ldots, e_n$ denote a sequence of independent evidence inferred from the transaction history. To reason over our hypothesis, we model it as a conditional probability $P(H|E)$ and apply the formula,

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}. \tag{6}$$

We assume that the prior probability $P(H)$ is the degree of belief in judging the hypothesis in the absence of any previous evidence, therefore, we assume a uniform distribution similarly to (Christodoulou et al., 2015). $P(E)$ is the probability of the evidence that is used as a normalization factor. We derive this probability using the law of total probability (Papoulis, 1991).

Training Data. To apply Equation 6 we need to estimate the likelihood of observing the evidence (denoted by $P(E|H)$), when the hypothesis is true or false. To obtain suitable data to train our model the system performs as follows:

Algorithm 1 returns a personalized Ranked List (denoted by *RK*) of Products from each category. Our aim is to recommend products of the same category from the overall dataset of products that are on offer (referred to as the "Offer-List"), according to the list of ranked products from *RK* (a running example is shown in Section 3).

Algorithm 1 Rank frequency bought products

Require: User c, TransactionHistory th
for all product_categories **do**
Rank each product using:

$$frequency(product) = \frac{quantity}{quantityOfcategory}. \tag{7}$$

end for
return *RK*

Iteration 1: Evidence is Product Description. At the first iteration the model considers, as a piece of evidence, the string similarity computed by using the cosine-similarity measure (Ning et al., 2015) between a pair of products based on the products descriptions.

$$P(H|E_1 = simScore) = \frac{P(E_1 = simScore|H)P(H)}{P(E_1 = simScore)}. \tag{8}$$

For the computation of the likelihoods a bootstrapping procedure similar to (Christodoulou et al., 2015) is followed to deriving probability distributions from similarity scores.

Iteration 2: Evidence is the average Price of a given category. Once a posterior probability is computed for some evidence $e_1 \in E$ a new piece of evidence $e_2 \in E$ leads to compute the impact of e_2 by considering the previously calculated posterior as the new prior.

During this iteration we assume as evidence the average price a customer spends on products from a specific category. Assuming products from the same category our model computes the degree of belief on whether a product from the "Offers-List" is suitable for purchase given its price. This is carried out by,

$$P(H|E_2 = avgPrice) = \frac{P(E_2 = avgPrice|H)P(H|E_1)}{P(E_2 = avgPrice)}. \quad (9)$$

2.4 Recommendation Engine

Given a List of products on offer (denoted by *OD*), the recommendation engine is configured to make personalized recommendations based on the unique shopping preferences of each user. The offers in the *OD* are divided into various categories: 2 products for the price of 1, products that are on sale, or products that had a reduction in price.

Assuming the construction of sub-datasets for each user belonging in a cluster (Sections 2.3.1 to 2.3.3) along with *RK*, the system applies the Bayesian approach (Section 2.3.4) given a pair of products from the *RK* with products from the *OD* dataset in order to calculate the posterior probability.

When there are no any offers that suit the customer's unique shopping preferences the system recommends offers that are suitable for the whole group of users (i.e, cluster).

2.5 Point-of-Sale (POS)

To complement the shopping experience of customers in a supermarket, `iBeacons` that are installed close to the POS systems push notifications to customers regarding products that have not been purchased but appear in the personalized "To-Buy-List" of each customer.

After a successful payment, the recent transaction history is uploaded on a server that is hosted over the cloud. In addition, the system updates the user's static and dynamic information that is used for future purposes. Finally, the system checks if a customer has bought products from the recommendation list. In such a case, it uses this information to derive statistical evidence of any increase on store's revenue as a result of purchases made directly from the recommendation list.

3 PRELIMINARY EXPERIMENTATION

This section describes a preliminary experimental case-study that has a twofold purpose: (i) to observe an optimal value for the parameter β used to determine the number of clusters, and (ii) to get a confirmation of the validity of the techniques used.

3.1 Case-Study: Local Supermarket

For the purposes of our experimentation we deployed our system in a local supermarket with 200 customers (i.e., users).

- By analysing data generated from the deployment we constructed a set of datasets as follows: (a) *users-static* dataset, defined as *SU*; that contains the user *static information*, such as: Customer ID, Full Name, Health Status, Allergies, Nutrition characteristics etc.; (b) *users-dynamic* dataset, defined as *DU*; that includes real-time information such as transaction history, to-buy-list and location; (c) *products dataset*, defined as *DP*; which contains information regarding items; (d) *offers* dataset, defined as *OD*; that includes special-offer products; (e) "to-buy-list" products, defined as *TBD*; that contains products that are essential to buy.
- We asked 200 users to provide us with their static information that was not available from the supermarket's proprietary system.
- `iBeacons` were installed in 2 different departments of the supermarket (i.e., Health care and Grocery). The *DP* dataset contains products from these departments where the missing features for each product were filled in manually.
- Finally, for evaluation purposes, we asked the users to participate in a short survey where they have been asked to rate their recommendations (top-5 items).

The outcome of the above process resulted in the following datasets: the *SU* dataset consists of 200 users with 8 features, the *DU* dataset consisting of 3000 users' historical data monitored from 15 visits for each user between January to June 2016 with 6 features, the *DP* dataset which contains 1000 products with 14 features, the *OD* dataset consisting of 45 products that are on offer with 5 features. Finally, the *TBD* dataset contains 50 products from two departments with 2 features.

The experiments were carried out on a Pentium (R) Dual-Core 2.70 GHz machine with 4GB of main memory running Windows 7 (64 bit).

3.2 The Parameter β

To determine an appropriate set of clusters for the recommendation process the entropy threshold similarity value β, needs to be tuned on the size of the dataset. The higher the β value the lower the number of the clusters discovered (Table 1). The execution time increases proportionally as the number of clusters increases.

Table 1: Impact of β on the SU dataset.

β	# clusters	execution time (sec)
1	3	0.37
0.75	4	0.39
0.5	7	0.44
0.25	11	0.61
0.1	23	0.76

By observing the sensitivity of the β parameter over a set of empirical trial-and-error experiments we set $\beta = 0.5$ as the optimal value to obtain a number of clusters in a relatively short execution time. The outcome of the entropy-based and the hard *k*-modes clustering was 7 clusters with users sharing similar characteristics.

Having determined the set of clusters for the users, a set of rules as described in Section 2.3.3 is applied to create the sub-datasets of products suitable for each cluster.

3.3 A Demonstration Example

Cluster c_1 has the following centroid $< 1,2,1,1,0,0,0,1 >$ where each element of the feature vector describes the category depicted in Table 2.

Table 2: Example of features described by the centroid vector.

No.	value	description	feature
1st	1	Cyprus	Country
2nd	2	Greek	Nationality
3rd	1	Christian	Religion
4th	1	True	Fasting
5th	0	False	Vegetarian
6th	0	False	Diet
7th	0	False	Allergies
8th	1	Good	Health Status

The feature vector of the centroid is used for applying the priority rules as described in Section 2.3.3. The output of this filtering process is a sub-dataset of products, $p_j \in PD$, where j denotes the cluster number. Hence, $p_1 \in PD$ is the sub-dataset of products for cluster c_1.

Following our demonstration and using `Algorithm 1` on the *DU* dataset we derived for $user_{12}$ in cluster c_1 the products that the specific customer buys more frequently. The system selects the top frequently bought products for user $user_{12}$ on the *Shampoos* category, that exists in the Health care department, to be compared with products of the same category belonging in the *OD* dataset. This is depicted in Table 3.

Table 3: Ranked frequency bought items

user no.	cluster #	cat.:*Shampoos*	frequency score
$user_{12}$	c_1	Product1	0.36
		Product232	0.27
		Product31	0.20
		Product4	0.17

Using the Bayesian Inference approach the system calculates the posterior probability to guide the recommendation process as:

Iteration 1. Let us assume that the likelihood $P(E_1 = simScore|H)$ is given by a probability density function as an integral over a finite region $[a,b]$, $P(a \leq X \leq b) = \int_a^b f(x)\,dx$

Following our experiment, with $user_{12}$, the system computes the posterior degree of belief between the similarity score, derived using cosine-similarity measure (Ning et al., 2015), returned by comparing the text descriptions of Product 1 from the `Shampoo` category with Products $\in$ *OD*. We also assume that $P(H)$ is a uniform prior using the principle of indifference.

Iteration 2. Let us assume that the likelihood $P(E_2 = avgPrice|H)$ is given by a probability density function as an integral over a finite region $[a,b]$, $P(a \leq X \leq b) = \int_a^b f(x)\,dx$

The previously calculated posterior, for $user_{12}$, is updated using a new evidence i.e., the *average price* that $user_{12}$ spends on products from the `Shampoo` category. In this iteration, the system computes a new posterior degree of belief given the average price spent on products with products that belong to *OD*.

Note that in order for a product to be recommended it must exist in the dataset of the cluster to which a user belongs to. Following our example with $user_{12}$ if a product does not exist in p_1 it cannot be recommended.

Recommendations. Finally, the proposed system recommends to $user_{12}$ a list of products that are on *offer* similar to Product 1 (shown in Table 4).

Table 4: Recommended products for $user_{12}$.

product id	probability of purchase
Product 321	0.76
Product 88	0.65
Product 454	0.59
Product 125	0.46
Product 779	0.39

The procedure returns a personalized list of top-5 items that are on offer, as suggested by the methodology described in Section 2. The recommendations list is broadcast to $user_{12}$ using `iBeacons` that are located in the Health care department.

3.4 Measuring Accuracy

To measure how well the proposed methodology recommends products that are on *offer* we measure precision as follows:

- From the set of 200 users that visited the store 15 times, the system suggests the top-5 products that each user is likely to purchase from the offers list.
- At checkout the system monitors which of the recommended products were actually purchased.
- Along with the above data, users were asked to provide us with explicit feedback by annotating (Suitable/Not Suitable) which of the suggested top-5 products might be of an interest to them (as shown in Table 5).

Table 5: Obtaining feedback from $user_{12}$.

product id	probability of purchase	purchased at POS?	suitable?
Product 321	0.76	✓	N/A
Product 88	0.65	✗	✗
Product 454	0.59	✗	✓
Product 125	0.46	✗	✓
Product 779	0.39	✗	✗

The feedback phase seeks to obtain additional information from users that guides our evaluation. The feedback is collected in the form of *true positive*; product recommended and purchased or suitable for purchase and *false positive*; product recommended but not purchased and not suitable. From these annotations, the $precision = \frac{tp}{tp+fp}$, for $user_{12}$ during a single visit is 0.60.

To monitor the behavior of the system we repeated the experiment for the entire set of 200 users that participated in our feedback experiment. Fig. 2 shows a frequency plot with the precision (y-axis) obtained for each user grouped into bins. The average precision for this case is 0.7190.

Similarly, to study how the number of recommended items affects the system's precision we repeated the experiment for top-3 items (as shown in Fig. 3). The average precision for this case is 0.8028.

3.5 Comparison with benchmark CF algorithms

To compare the accuracy of the proposed methodology with common CF approaches (item-based, user-based) we additionally asked customers to evaluate the lists of recommended products resulting from both item-based and user-based methods. Table 6 shows the precision between the proposed approach compared with the average precision computed from the aforementioned CF approaches. We observed that our system performs better than the typical CF approaches at various cases of recommended products, with an average improvement of 24.7%.

Table 6: Precision comparison with the benchmark CF algorithm.

	Proposed methodology	Average benchmark-CF
top-3	0.8028	0.6243
top-5	0.7190	0.5951

4 RELATED WORK

Lawrence et al., (2001) contributes a personalized Recommender System that suggests new products to supermarket buyers based on association rules. Association rules are applied to define the relationships among the products. A clustering technique is then utilized to cluster shoppers with similar spending histories. An increase in the supermarket's revenue is observed when shoppers choose to purchase products from the recommendation list.

Modeling temporal dynamics is another parameter that influences present RS due to the fact that users tend to change their preferences over time. Users' behavior can be defined by short-term and long-term preferences (Xiang et al., 2010). According to (Koren, 2009), product attention and popularity are regularly changing making customers reconsider their tastes, interests and feelings. Time changing behavior on the data is therefore an important factor to be taken into consideration when designing a RS.

Authors in (Christodoulou et al., 2014) present a model that utilizes an Entropy-based algorithm and variations of k-modes clustering techniques. The aim

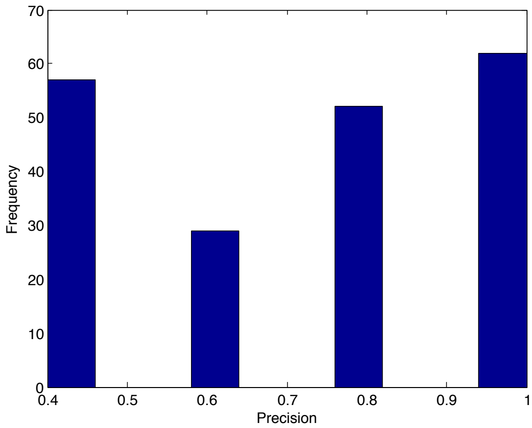

Figure 2: Users' precision for the top-5 case.

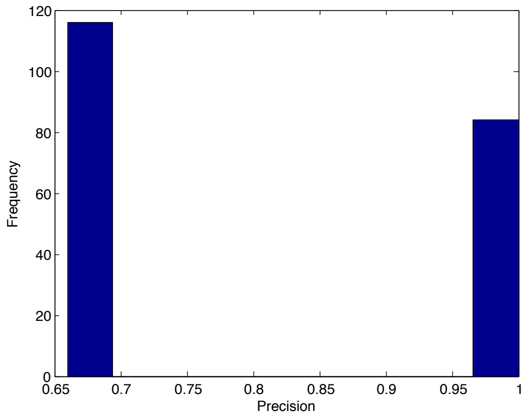

Figure 3: Users' precision for the top-3 case.

is to cluster objects according to certain data features and then use these features to make recommendations to users. The system records the users' behavior that is changing dynamically over time to support the recommendation process. Experimental results show an increase in recommendation accuracy and performance compared to the state-of-the-art algorithms when applied to the same context.

Suksom et al. (Suksom et al., 2010) proposed a personalised food RS based on a rule-based approach. The system aims to offer personalized recommendations on different kinds of meals to users based on their nutrition requirements or other health care characteristics.

The work in (Nikoletic, 2013) implements a RS in physical stores targeting to make shopping more interactive to buyers. The proposed methodology utilizes user-based and item-based CF methods, and a Restricted Boltzmann machine algorithm. The system computes recommendations based on the customers' purchase patterns and uses iBeacons to locate a mobile device in a real-world implementation to push recommendations.

Lacic et al. (Lacic et al., 2015) present a RS based on CF that makes use of a user's location captured by indoor position systems. The RS relies on user-based CF to suggest items with no data available. The authors argue that the proposed method outperforms Matrix Factorization approaches when dealing with cold-start users.

Discussion. The framework described in this paper makes use of an Entropy-based approach to determine the number of clusters and a Hard K-modes Clustering algorithm to group together users with similar characteristics. We propose this technique as a solution to the data Sparsity limitation, the cold-start problem and scalability issues often met in RS. Additionally, the proposed system uses a dataset which contains not only the traditional 2D Users x Items space, but also multiple other criteria such as: countries, nationalities, religions, fasting periods, vegetarian habits, diets, allergies and health statuses.

To reduce the search space in the set of overall products, used during the recommendation process, a rule-based system is applied over the products dataset with the aim of creating personalized sub-datasets of products for each cluster of users. Our probabilistic model utilizes user's transaction history to learn frequent shopping habits of each user. This is then used as input to a Bayesian Inference approach to reason over a hypothesis given certain pieces of evidence. Overall our work presents a recommendation engine that makes personalized suggestions on products that are on offer to users, in real-time, using the `iBeacon` protocol.

5 CONCLUSIONS AND FUTURE WORK

In a supermarket setting, where users are constantly changing their preferences or shopping habits, and products potentially change their characteristics or lose their popularity, there is a need for a system that captures the dynamic environment of a supermarket aiming to recommend products that are on offer. Throughout this paper we discussed how the cold-start problem, the data sparsity and other scalability issues often met in RSs are minimized by utilizing an Entropy-based Hard *k*-modes clustering methodology. Using a Bayesian inference approach we showed how the system can suggest personalized recommendations to users considering different pieces of evidence.

To explore whether our methodology meets our expectations we deployed `iBeacons` in 2 departments of a supermarket where customers installed the prototype mobile application on their smart devices. Feedback was obtained on the recommendations made for each customer and used for evaluating the system's accuracy. Our findings showed an average precision of 0.7190 for the top-5 case and 0.8028 for the top-3 case outperforming traditional CF approaches (item-based and user-based).

Although our preliminary experimentation suggested the validity of our methodology, additional experimentation needs to be undertaken to validate the accuracy and performance of the proposed system in supermarkets of a larger scale; in terms of products and customers. In addition, the parameter n that controls the number of recommended products needs to be automatically tuned to further increase the system's accuracy. Furthermore, an investigation needs to be undertaken on how the number of products influences the number of clusters discovered.

Finally, experiments will be carried out to study how efficient the system propagates notifications through `iBeacons` and how does group recommendations affect buyers into purchasing products.

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749.

Christodoulou, K., Fernandes, A. A. A., and Paton, N. W. (2015). Combining syntactic and semantic evidence for improving matching over linked data sources. In *Web Information Systems Engineering - WISE 2015 - 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I*, pages 200–215.

Christodoulou, P., Lestas, M., and Andreou, A. S. (2014). Applying hard and fuzzy k-modes clustering for dynamic web recommendations. *International Journal on Engineering Intelligent Systems*, 22(3-4).

Huang, Z. and Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Trans. Fuzzy Systems*, 7(4):446–452.

Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Model. User-Adapt. Interact.*, 22(1-2):101–123.

Koren, Y. (2009). Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 447–456.

Lacic, E., Kowald, D., Traub, M., Luzhnica, G., Simon, J., and Lex, E. (2015). Tackling cold-start users in recommender systems with indoor positioning systems. In *Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16, 2015.*

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1–49.

Nikoletic, M. (2013). Recommender systems-a smarter shopping solution. Bachelor's thesis. http://milnik.net/.

Ning, X., Desrosiers, C., and Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 37–76.

Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Companies, 3rd edition.

Pu, P., Chen, L., and Hu, R. (2012). Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Model. User-Adapt. Interact.*, 22(4-5):317–355.

Stylianou, C. and Andreou, A. S. (2007). A hybrid software component clustering and retrieval scheme using an entropy-based fuzzy k-modes algorithm. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), October 29-31, 2007, Patras, Greece, Volume 1*, pages 202–209.

Suksom, N., Buranarach, M., Thein, Y. M., Supnithi, T., and Netisopakul, P. (2010). A knowledge-based framework for development of personalized food recommender system. In *Proc. of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems*.

Xiang, L., Yuan, Q., Zhao, S., Chen, L., Zhang, X., Yang, Q., and Sun, J. (2010). Temporal recommendation on graphs via long- and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 723–732.

Yang, J., Wang, Z., and Zhang, X. (2015). An ibeacon-based indoor positioning systems for hospitals. *International Journal of Smart Home*, 9(7):161–168.