

Recommending POIs based on the User’s Context and Intentions

Hernani Costa, Barbara Furtado, Durval Pires,
Luis Macedo, and Amilcar Cardoso
hpcosta@dei.uc.pt, {bfurtado,durval}@student.dei.uc.pt,
{macedo,amilcar}@dei.uc.pt

CISUC, University of Coimbra, Portugal

Abstract. This paper describes a Recommender System that implements a Multiagent System for making personalised context and intention-aware recommendations of Points of Interest (POIs). A two-parted agent architecture was used, with an agent responsible for gathering POIs from a location-based service, and a set of Personal Assistant Agents (PAAs) collecting information about the context and intentions of its respective user. In each PAA were embedded four Machine Learning algorithms, with the purpose of ascertaining how well-suited these classifiers are for filtering irrelevant POIs, in a completely automatic fashion. Supervised, incremental learning occurs when the feedback on the true relevance of each recommendation is given by the user to his PAA. To evaluate the recommendations’ accuracy, we performed an experiment considering three types of users, using different contexts and intentions. As a result, all the PAA had high accuracy, revealing in specific situations F_1 scores higher than 87%.

Keywords: information overload, machine learning algorithms, multiagent systems, personal assistant agents, recommender systems, user modelling.

1 Introduction

Nowadays, we are experiencing a huge growth in the quantity of information available. This was mainly caused by the advent of communication technology which humans cannot handle properly, and made critical the need for intelligent assistance when browsing, searching or exploring for interesting information. In order to cope with this superabundance, Recommender Systems (RS) are a promising technique to be used, for instance in location-based domains [19,4]. The majority of RS’ approaches focus on either finding a match between an item’s description and the user’s profile (Content-Based [2,15,14]), or finding users with similar tastes (Collaborative Filtering [11,5,9]). These traditional RS consider only two types of entities, users and items, and do not put them into a context when providing recommendations – context, in ubiquitous and mobile context-aware systems, can be defined as the location of the user, in

order to identify people and objects around him, and the changes in these elements [18]. However, the most relevant information for the user may not only depend on his preferences and context, but also in his current intentions [7]. For example, the very same item can be relevant to a user in a particular context, and completely irrelevant in a different one. This is due to the fact that user's preferences and intentions change over time. For this reason, we believe that it is important to have the user's context and intentions in consideration during the recommendation process [20,1,4,3,10,8].

In this work, our goal is to develop a RS that implements a Multiagent System (MAS). Given the user's context and intentions and the sources at our disposal, it will be imperative to create a system capable of recommending POIs in a selective fashion [12]. Additionally, we intend to ascertain how well-suited different Machine Learning (ML) algorithms are to automatically filter irrelevant POIs. After collecting a small set of POIs from a Web location-based service and updating these manually with extra information, three different users' models via stereotypes are created by using a set of rules. Our assumption is that the system will be able to understand the differences between each user, since each one has unique preferences, intentions and behaviours, resulting in different recommendations for different users, even if their context is the same. In order to accomplish that, a MAS was embedded with the purpose of provide selective information to the users [6]. The recommendations' accuracy will be evaluated by correlating the recommendations outputs given by these algorithms, with nine human judges.

The remaining of the paper starts with a presentation of the system's architecture (section 2). Then, section 3 provides an overview of the experimental set-up, and section 4 describes the results obtained. Finally, section 5 presents the final remarks.

2 System Architecture

In this section, we present the system's architecture and all its components used in this work (see figure 1). This architecture can be seen as a *middleware* between the user's needs and the information available.

More specifically, the **Master Agent** is responsible for starting, not only the Web agents, but also the PAAs, described in figure 1 as $PAA_1 \cdots PAA_n$. The system is capable of retrieving POIs' information from several location-based services. However, for the purpose of this work it is only used the Foursquare service, which explains why we used only one Web agent (**Agent_{foursquare}**).

Agent_{foursquare} implements several methods available through the Foursquare API¹, allowing it to start requesting for POIs in a pre-defined area (see section 3.1). During this process, it filters all the POIs that do not belong to the categories we will use in this experiment, and stores the remaining POIs in our system's database (presented in figure 1 as **POIs Database**). This autonomous

¹ <https://developer.foursquare.com>

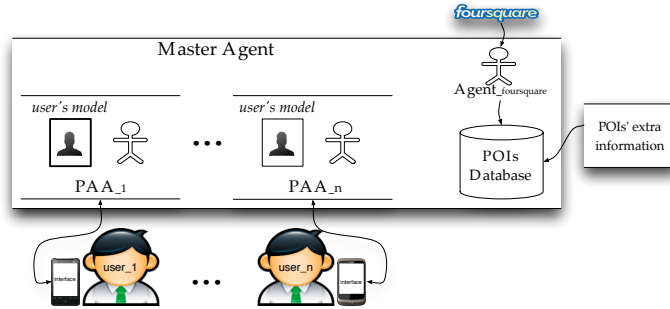


Fig. 1. System's Architecture.

agent is constantly searching for new information, and verifying if the data stored in the database is up-to-date.

Due to the fact that Foursquare service did not have all the information needed for the experiment, we decided to gather more information about the POIs on the field (e.g., POIs' price, timetable, dayOff, as well as some of the attributes missing). This allowed us to have more details about each POI in order to fulfil the set-up requirements of the experiment (see section 3). Thus, this extra information was used to update the POIs' attributes in the database.

As we can see in figure 1, each user has a PAA assigned to him. This agent expects the user to make a request, and, based on his context and intentions (see section 3.2), recommends a list of nearby POIs. In order to improve its recommendations, the PAA continuously learns from the user's experiences. Concretely, each PAA implements a probabilistic classifier to assign a probability value to the relevance of each POI, given the current user's context and intention. Therefore, when the feedback of the true relevance of each recommendation is given by the user to his PAA, the PAA updates its model (described in figure 1 as *user's model*). As a result, the agent learns every time the user decides to make a request and give his feedback.

3 Experiment Set-up

Our main objective is to demonstrate how we can face the information overload problem in the location-based service domain by using a MAS architecture. More precisely, it is our intention to take advantage of the multiple independent, autonomous, and goal-oriented units, so called PAAs. In addition, we intend to verify how accurate different ML algorithms perform the task of predicting the user's preferences, while taking his context and intentions into account.

To achieve this goal, an effectiveness evaluation of our system will be performed in section 4. But firstly, we start by explaining the experiment set-up. In detail, the area where we will perform the experimentation is presented in section 3.1. Then, the main attributes used to define the user's context and

intentions are presented in section 3.2. Finally, section 3.3 presents the user stereotypes considered in this work, as well as how their models were created.

3.1 Area of Work

The experimentation was performed in Coimbra (Portugal). The number of POIs existent in the city made it impossible to manually update all the POIs retrieved with the extra information needed to this experiment. Thus, a smaller part of the city that had more POIs density and diversity (Coimbra's Downtown) was used. Furthermore, the type of POIs used were restricted to {Food, Shopping, Nightlife} (the categories that contain more POIs in this area). The number of sub-categories for Food are 44, Shopping 8 and Nightlife 11, with 271, 10 and 84 different POIs, respectively. The extra information manually gathered from these 365 places was the POI's price, the day off and the timetable (the possible values for each attribute is explained in the next topic).

3.2 Defining Context and Intentions

Context is the key to personalise the automatic recommendations made by the PAAs for their users. Thus, a set of attributes need to be defined in order to characterise the POI's context, as well as the user's context and intentions. Since these attributes need to be combined, an **interface** was used to visualise current user location, i.e., his context and intention. The main attributes used to define the user, the POI and the information available in the interface are shown in figure 2.

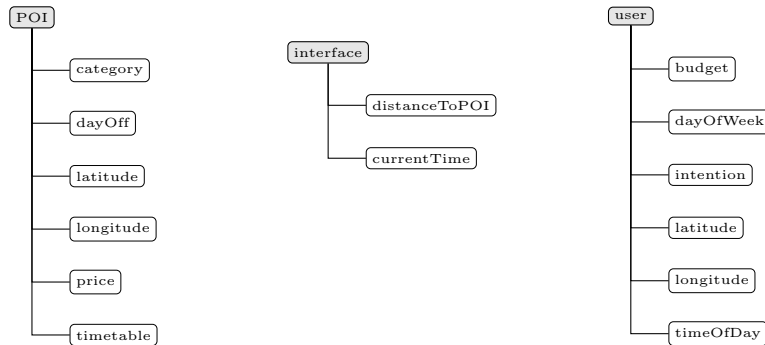


Fig. 2. Main attributes used to define the context of the user, POI and the interface.

Possible values for each attribute of the POI's context are:

- ◊ category = {food, shopping, nightlife}, actually we use the sub-category, e.g., food = ⟨sandwichShop, vegetarian, etc.⟩, shopping = ⟨men'sApparel, women'sApparel, etc.⟩ and nightlife = ⟨wineBar, disco, etc.⟩

- ◊ `dayOff` = {a day of the week or combinations}
- ◊ `price` = {cheap, average, expensive}, e.g., for lunch {cheap \leq 5€; 5€>average \leq 7€; expensive $>$ 7€}
- ◊ `timetable` = {morning, afternoon, night, or combinations}

Possible values provided by the interface are:

- ◊ `distanceToPOI` = {near \leq 200m; 200m >average \leq 300m; far $>$ 300m}
- ◊ `currentTime` = {current day of the week and period of the day (morning, afternoon or night)}

Possible values for each attribute of the user's context are:

- ◊ `budget` = {low, medium, high}, e.g., for lunch {low \leq 5€; 5€>medium \leq 7€; high $>$ 7€}
- ◊ `dayOfWeek` = {current day of the week}
- ◊ `goal` = {coffee, lunch, dinner, party}, e.g., drink coffee in a {bakery, coffeeShop, etc.}, have lunch and dinner in {burgers, BBQ, etc.} and party in a {bar, disco, etc.}
- ◊ `timeOfDay` = {morning, afternoon or night}

After defining the main attributes considered in this work, we are now able to explain how these were used to create the users' models.

3.3 Datasets and Users Model Description

As one of the goals in this work is the study of different user's profiles, we have created three different types of users, which we believe to be very common in our society:

- u_1 – a user which prefers POIs that are **cheap** and **near**
- u_2 – a user which prefers POIs that are **near**
- u_3 – a user which prefers POIs that are **expensive**

Another objective for this work is the analysis of different probabilistic classifiers suitable for the task of filtering irrelevant POIs, in a completely automatic fashion. For this purpose, two different Bayes classifiers (Naïve Bayes and BayesNet) and two versions of the C4.5 decision tree learner (J48 pruned and unpruned), available through the Weka API², were implemented.

However, the cold-start problem [17] needs to be resolved. In order to overcome this problem, a set of rules were used to create three datasets for the three user stereotypes. In detail, the rules used (described in equation 1) consider not only the type of user, but also his goals (i.e., intentions) when given automatic feedback to the POIs in a specific situation. The values given by these rules are binary, 1 if satisfies the user's goal or \emptyset if not. To create the datasets we simulated 420 *runs*, for each type of user. For the sake of clarity, a run (r) represents a combination of the user's context and goals with the POIs' context (all the POIs retrieved by the interface in the radius of 400m), i.e., a situation. The resulted datasets, represented as d_1 , d_2 and d_3 , contain:

- d_1 = 5844 instances, 1371 classified as 1 and 4473 as \emptyset
- d_2 = 6014 instances, 1774 classified as 1 and 4240 as \emptyset

² <http://weka.sourceforge.net/doc>

· $d_3 = 6259$ instances, 2590 classified as 1 and 3669 as \emptyset
The dataset number (d_n) correspond to the user type (u_n).

$$R(\text{Goal}(u_n)) = \begin{cases} \forall \text{Goal}(u_1) \text{ if } (\text{distance} \leq 200m \ \&\& \\ \qquad \qquad \qquad \text{price} = \text{cheap}), \quad R(u_1) = 1 \\ \forall \text{Goal}(u_2) \text{ if } (\text{distance} \leq 200m), \quad R(u_2) = 1 \\ \forall \text{Goal}(u_3) \text{ if } (\text{price} = \text{expensive}), \quad R(u_3) = 1 \\ \qquad \qquad \qquad \text{otherwise}, \quad R(u_n) = \emptyset \end{cases} \quad (1)$$

4 Results

Our experiment can be divided into three different evaluations. Firstly, we made a manual evaluation and calculated the exact agreement between the human judges (section 4.1). Then, using the feedback of the judges about the true correctness of the recommendations, a 10 times tenfold cross-validation was performed with purpose of evaluating the ML algorithms' performance (section 4.2). Finally, well-know metrics were used to compare and analyse the recommendations given by the PAAs with manual evaluation (section 4.3).

4.1 Manual Evaluation

To test our approach, it was used a set of scenarios from real situations. More precisely, in this experiment it were used 3 locations (the ones that had more POI density) in 5 different situations (i.e., considering different user's contexts and intentions). These combinations were named runs (r). Next, it is presented the first 5 runs (location, time of day, day of the week and goal).

$r_1 = [40.208934, -8.429067, \text{Morning}, \text{Sunday}, \text{Coffee}]$
 $r_2 = [40.208934, -8.429067, \text{Morning}, \text{Monday}, \text{Coffee}]$
 $r_3 = [40.208934, -8.429067, \text{Afternoon}, \text{Wednesday}, \text{Lunch}]$
 $r_4 = [40.208934, -8.429067, \text{Night}, \text{Thursday}, \text{Dinner}]$
 $r_5 = [40.208934, -8.429067, \text{Night}, \text{Saturday}, \text{Party}]$

Then, these runs were manually evaluated by a set of human judges, whose purpose was to analyse the exact agreement (EA) between them, as well as to compare the PAAs' recommendations with their evaluation.

In this experiment it was used 9 human judges (H), divided into three group (G) of 3 people, to evaluate one of the 3 stereotypes (u), i.e., $G_1 = \langle u_1 \rightarrow H_1, H_2, H_3 \rangle$, $G_2 = \langle u_2 \rightarrow H_4, H_5, H_6 \rangle$ and $G_3 = \langle u_3 \rightarrow H_7, H_8, H_9 \rangle$. They were asked to give their personal opinion for a list of scenarios (15 runs), but never contradicting the user's profile they were evaluating.

To perform this evaluation, we have created a user interface using Google Maps³, see figure 3. The blue icon represents the current user's location, and the other icons represent all the POIs retrieved by the recommender system.

³ <http://code.google.com/apis/maps/index.html>

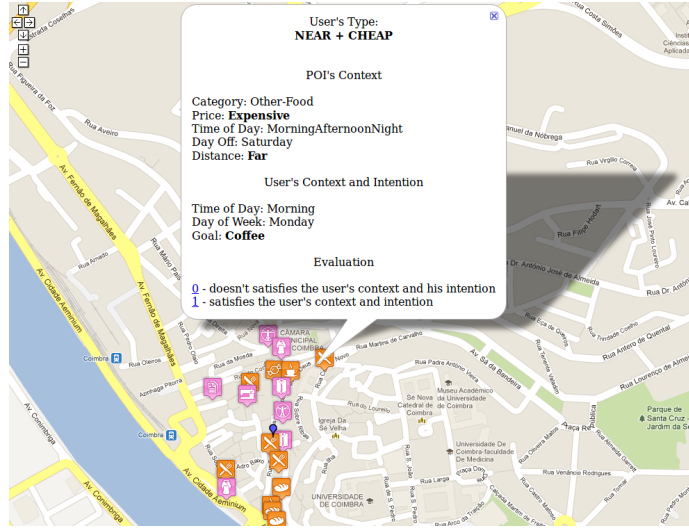


Fig. 3. Manual evaluation example.

Clicking in each POI's icon, the judges could see an information window describing the context of each situation, together with the user's intention. With this information, the judges could perform their evaluation, clicking in the option they prefer. The POIs' names were omitted to avoid that the judges' personal opinion influenced the evaluation. It was important to do this to prevent discrepancy between the judges preferences and the user's profile that they were evaluating. Each human judge was asked to assign one of the following values to each POI, according to the current user's context and POI's context:

- 0 - if the POI does not satisfy the user's context and his intention;
- 1 - if the POI satisfies the user's context and his intention.

The EA among the judges (for the 15 runs) in the G_1 (group one) resulted in 99.4%, 100% for the G_2 , and finally 99.4% for the G_3 . Despite of the small set of judges, the EA among them, means they have very similar opinions, for all the stereotypes evaluated, validating the data that will be used in the following sections.

4.2 Preliminary Results

To make the evaluation test, it was chosen the 10 times tenfold cross-validation [16]. This test was performed over the instances used in the training with the recommendations given by the algorithms for the 3 types of users, for the 15 runs. Table 1 presents the percentage of correctly and incorrectly classified instances, and also the statistics for the classifiers used (BayesNet, J48 pruned, J48 unpruned and Naïve Bayes, presented as **BN**, **J48_p**, **J48_u** and **NB**, respectively).

Table 2 shows in detail the accuracy of the classifiers for the user stereotype u_3 . For each prediction class (**CI**), it is presented the percentage of true positive

Table 1. Classifiers' statistics for the three user stereotypes.

		u_1				u_2				u_3			
		BN	J48 _p	J48 _u	NB	BN	J48 _p	J48 _u	NB	BN	J48 _p	J48 _u	NB
Correctly classified instances		99.14	98.57	100	99.43	97.43	100	100	97.71	99.71	99.43	99.43	99.71
Kappa statistic		0.98	0.97	1	0.99	0.95	1	1	0.95	0.99	0.99	0.99	0.99
Mean error	absolute	0.05	0.02	0	0.05	0.05	0	0	0.06	0.03	0.006	0.006	0.03
Root mean squared error		0.10	0.11	0	0.11	0.14	0	0	0.15	0.08	0.06	0.06	0.08
Relative error	absolute	9.25	3.37	0	10.93	9.45	0	0	11.09	5.92	1.36	1.36	6.90
Root squared error	relative	21.18	23.05	0	22.60	28.43	0	0	30.53	15.98	12.60	12.60	16.73
Total number of instances		350				350				350			

(**TP**), false positive (**FP**), precision (**P**), recall (**R**), F_1 score (**F₁**) and ROC Area (**ROC A.**). The results shows high accuracy for the classes \emptyset and **1**, this is due to the fact that the number of instances in the training dataset was balanced.

Table 2. Cross-validation's statistics for the user stereotype u_3 .

	TP	FP	P	R	F_1	ROC A.	CI
BN	99.50	0.49	100	99.50	99.75	99.85	\emptyset
	100	0.00	99.33	100	99.67	99.85	1
J48_p	98.61	0.13	99.07	98.61	98.84	99.29	\emptyset
	98.50	0.14	99.78	98.51	98.14	99.29	1
J48_u	98.61	0.14	99.07	98.61	98.84	99.29	\emptyset
	98.51	0.15	97.78	98.51	98.14	99.29	1
NB	99.50	0.49	100	99.50	99.75	99.85	\emptyset
	100	0.00	99.33	100	99.67	99.85	1

4.3 Performance Evaluation

In order to observe the relation between the manual evaluation and the output values given by the classifiers, the correlation coefficients between them were computed using the Spearman's coefficient, where $\rho : -100 \leq \rho \leq 100$ (see equation 2).

$$\rho(m_i, x_i) = \frac{\sum_i (m_i - \bar{m})(x_i - \bar{x})}{\sqrt{\sum_i (m_i - \bar{m})(x_i - \bar{x})}} \quad (2)$$

Table 3. Correlation coefficients’ results for the three user stereotypes.

	BN	J48_p	J48_u	NB	B
u_1 H ₁	48.50	49.00	49.00	48.50	100
H ₂	48.50	49.00	49.00	48.50	100
H ₃	49.25	49.72	49.33	49.25	98.78
EA _{G₁}	48.50	49.00	49.00	48.50	100
B	48.50	49.00	49.00	48.50	–
H ₄	49.01	48.56	48.56	47.34	100
H ₅	48.67	48.21	48.21	46.99	99.39
u_2 H ₆	49.01	48.56	48.56	47.34	100
EA _{G₂}	49.01	48.56	48.56	47.34	100
B	49.01	48.56	48.56	47.34	–
H ₇	49.43	56.50	56.50	49.43	100
H ₈	49.43	56.50	56.50	49.43	100
u_3 H ₉	48.24	55.27	55.27	48.24	97.27
EA _{G₃}	49.43	56.50	56.50	49.43	100
B	49.43	56.50	56.50	49.43	–

To evaluate the predictions’ results, it was created a baseline (**B**) that classifies each instance based on a set of rules (the same rules used to create the training datasets, see equation 1). This baseline acts like a trusted base of comparison with the ML algorithms predictions. Table 3 shows the correlation outputs for the three user stereotypes, as well as the exact agreement for each group of judges (**EA_{G_n}**, where n corresponds to the user stereotype).

The high correlation (≈ 100) between the judges (**H_n**) and the baseline (**B**) resulted in a perfect monotone increasing relationship, what can be seen as an evidence of the trustiness of their feedback. Although the ML algorithms have lower correlation compared with the baseline, as expected, all of them have positive correlation, revealing in specific situations correlation scores higher than 56% (which can be seen as a good hint to support this approach).

To test the algorithms’ accuracy, F_1 scores were calculated for the 3 types of users (u_1 , u_2 , u_3). Table 4 presents the results with the mean (\bar{x}) and the standard deviation (σ) for the 15 runs. In order to avoid some of the ambiguity that could arise when considering only the feedback given by one judge, F_1 was calculated by using the **EA** of each group (i.e., the **EA_{G_n}**, where n corresponds to user stereotype).

Again, it was used a baseline (**B**, see equation 1) to compare the accuracy between the algorithms. For the sake of clarity, the runs represent different user’s intentions, more specifically, runs: $\{r_1, r_2, r_6, r_7, r_{11}, r_{12}\}$ = goal **coffee**; $\{r_3, r_8, r_{13}\}$ = goal **lunch**; $\{r_4, r_9, r_{14}\}$ = goal **dinner**; and $\{r_5, r_{10}, r_{15}\}$ = goal **party**. As we can see, higher values are obtained for the goals **lunch**, **dinner** and **coffee**

Table 4. F_1 results (%) for the three user stereotypes.

	BN	J48 _p	J48 _u	NB	B	BN	J48 _p	J48 _u	NB	B	BN	J48 _p	J48 _u	NB	B
r_1	26.67	37.50	37.50	26.67	100	31.58	30.00	30.00	31.58	100	26.67	28.57	28.57	26.67	100
r_2	37.50	47.06	47.06	37.50	100	40.00	38.10	38.10	40.00	100	26.67	28.57	28.57	26.67	100
r_3	<u>76.19</u>	<u>76.19</u>	<u>76.19</u>	<u>76.19</u>	100	<u>78.57</u>	<u>78.57</u>	<u>78.57</u>	<u>78.57</u>	100	81.82	85.71	85.71	81.82	100
r_4	<u>76.19</u>	<u>72.73</u>	<u>72.73</u>	<u>76.19</u>	100	<u>78.57</u>	<u>78.57</u>	<u>78.57</u>	<u>78.57</u>	100	81.82	81.82	81.82	81.82	100
r_5	15.38	13.33	13.33	15.38	100	21.05	21.05	21.05	11.11	100	26.67	26.67	26.67	26.67	100
r_6	40.00	40.00	40.00	40.00	100	40.00	33.33	33.33	40.00	100	40.00	57.14	57.14	40.00	100
r_7	40.00	40.00	40.00	40.00	100	33.33	33.33	33.33	33.33	100	40.00	57.14	57.14	40.00	100
r_8	66.67	66.67	66.67	66.67	100	75.00	75.00	75.00	75.00	100	54.55	75.00	75.00	54.55	100
r_9	66.67	57.14	57.14	66.67	100	75.00	75.00	75.00	75.00	100	54.55	54.55	54.55	54.55	100
r_{10}	66.67	57.14	57.14	66.67	100	57.14	57.14	57.14	57.14	100	66.67	66.67	66.67	66.67	100
r_{11}	72.73	69.57	69.57	72.73	100	75.00	75.00	75.00	75.00	100	<u>87.50</u>	<u>87.50</u>	<u>87.50</u>	<u>87.50</u>	100
r_{12}	53.57	60.00	60.00	55.56	100	51.72	63.64	63.64	55.56	100	42.86	72.73	72.73	50.00	100
r_{13}	60.00	60.00	60.00	60.00	100	57.14	57.14	57.14	57.14	100	36.36	44.44	44.44	36.36	100
r_{14}	60.00	57.14	57.14	60.00	100	60.00	57.14	57.14	60.00	100	36.36	36.36	36.36	36.36	100
r_{15}	60.00	57.14	57.14	60.00	100	63.16	57.14	57.14	63.16	100	71.43	71.43	71.43	71.43	100
	\mathbf{u}_1					\mathbf{u}_2					\mathbf{u}_3				
\bar{x}	54.55	54.11	54.11	54.68	100	55.82	55.34	55.34	55.41	100	51.59	58.29	58.29	52.07	100
σ	18.56	16.31	16.31	18.56	0.00	18.96	19.65	19.65	20.36	0.00	21.43	21.29	21.29	21.30	0.00

(see the underline values in the r_3 , r_4 and r_{11} , respectively), and lower values are obtained for the goal **party** (see for instance r_{15}). This happens because the goal **party** is only valid at night and a lower number of POIs suits that goal, which lead the classifiers to perform worse in these situations. Contrarily, the goal **coffee** is valid in all times of day, resulting in a lot more instances and, consequently, the classifiers' improvement is faster. For example, for the r_{11} , \mathbf{u}_3 , all the algorithms have 87.50%.

In general, the algorithms performed similarly, however some of them had a higher mean, for a specific user stereotype. For instance, **NB** had better F_1 scores for \mathbf{u}_1 , for $\mathbf{u}_2 \rightarrow \mathbf{BN}$, and for $\mathbf{u}_3 \rightarrow \mathbf{J48}_p$ and $\mathbf{J48}_u$.

To sum-up, ML algorithms can be a powerful technique, in location-based services, to predict which content will be interesting for a determined user. Nevertheless, with more data and more usage the recommendations' accuracy could be improved.

5 Conclusions

In this paper, we discussed the combination of context and intention-awareness with RS, applied in a location-based application. We pointed out what advantages are earned in using, besides the context, the user's intentions, and how to integrate both into a location-based RS. We also presented our system's

architecture and described its advantages. ML techniques were used to train the classifiers, more precisely Naïve Bayes, BayesNet and J48 (pruned and unpruned).

Additionally, we created an experimental set-up to evaluate the algorithms' performance, for three types of users. Firstly, a 10 times tenfold cross-validation test was made. Secondly, in order to observe the relation between the manual evaluation and the output values given by the PAAs, the correlation coefficients between them were computed. Finally, we performed an information retrieval task consisting on the identification of correct recommendations, given by the ML algorithms. All of them had high accuracy, revealing in specific situations F_1 scores higher than 87%.

In the future, we are planning numerous improvements to this work, such as: the use of new information sources, as well as their aggregation; take into account new attributes (e.g., POI's quality by considering the number of check-ins and the users' reviews); implement and compare other ML algorithms; analyse other users' profiles; and allow the user to change what values fit in each attributes (e.g., what price is considered cheap, as well as allow the user to select his budget). We think that with more data and more usage the recommendations' accuracy could improve. Furthermore, we plan to analyse the system accuracy when applying selective attention metrics, such as surprise [13], in the recommendation outputs. Finally, we intend to make the application available to the community in order to get more feedback and also to test our system in other situations.

Acknowledgments

Work funded by Fundação para a Ciência e Tecnologia — Project PTDC/EIA-EIA/108675/2008, and by FEDER through Programa Operacional Factores de Competitividade do QREN — COMPETE:FCOMP-01-0124-FEDER-010146.

References

1. Adomavicius, G., Mobasher, B., Ricci, F., Tuzhilin, A.: Context-Aware Recommender Systems. *AI Magazine* 32(3), 67–80 (2011)
2. Balabanović, M., Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. *Commun. ACM* 40(3), 66–72 (1997)
3. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context Relevance Assessment and Exploitation in Mobile Recommender Systems. *Personal and Ubiquitous Computing* pp. 1–20 (2011)
4. Biancalana, C., Flamini, A., Gasparetti, F., Micarelli, A., Millevolte, S., Sansonetti, G.: Enhancing Traditional Local Search Recommendations with Context-Awareness. In: *User Modeling, Adaption and Personalization*. pp. 335–340. Springer (2011)
5. Billsus, D., Pazzani, M.J.: Learning Collaborative Information Filters. In: *15th Int. Conf. on Machine Learning*. pp. 46–54. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)

6. Costa, H.: A Multiagent System Approach for Emotion-based Recommender Systems. PhD proposal, University of Coimbra, Coimbra, Portugal (2012)
7. Costa, H., Furtado, B., Pires, D., Macedo, L., Cardoso, A.: Context and Intention-Awareness in POIs Recommender Systems. In: 6th ACM Conf. on Recommender Systems (RecSys'12), 4th Workshop on Context-Aware Recommender Systems (CARS'12). ACM (2012)
8. Costa, H., Macedo, L.: Emotion-Based Recommender System for Overcoming the Problem of Information Overload. In: 11th Int. Conf. on Practical Applications of Agents and Multi-Agent Systems (PAAMS'13), Workshop on User-Centric Technologies and Applications (CONTEXTS'13). Springer (2013)
9. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: 16th Int. Conf. on World Wide Web. pp. 271–280. ACM, NY, USA (2007)
10. Huang, H., Gartner, G.: Using Context-Aware Collaborative Filtering for POI Recommendations in Mobile Guides. In: Advances in Location-Based Services. pp. 131–147. Lecture in Geoinformation and Cartography, Springer, Vienna, Austria (2012)
11. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM* 40(3), 77–87 (1997)
12. Macedo, L.: Selecting Information based on Artificial Forms of Selective Attention. In: 19th European Conference on Artificial Intelligence (ECAI'10). pp. 1053–1054. IOS Press (2010)
13. Macedo, L.: A Surprise-based Selective Attention Agent for Travel Information. In: 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'10), 6th Workshop on Agents in Traffic and Transportation (ATT'10). pp. 111–120 (2010)
14. Melville, P., Mooney, R.J., Nagarajan, R.: Content-Boosted Collaborative Filtering for Improved Recommendations. In: 18th National Conf. on AI. pp. 187–192. AAAI, CA, USA (2002)
15. van Meteren, W., van Someren, M.: Using Content-Based Filtering for Recommendation. In: ECML/MLNET, Workshop on ML and the New Information Age. pp. 47–56. Barcelona, Spain (2000)
16. Refaeilzadeh, P., Tang, L., Liu, H.: Cross-Validation. In: Encyclopedia of Database Systems, pp. 532–538. Springer (2009)
17. Schein, A., Popescul, A., Ungar, L., Pennock, D.: Methods and Metrics for Cold-Start Recommendations. In: 25th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. pp. 253–260. ACM, NY, USA (2002)
18. Schilit, B.N., Theimer, M.M.: Disseminating active map information to mobile hosts. *Network, IEEE* 8(5), 22–32 (1994)
19. van Setten, M., Pokraev, S., Koolwaaij, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS. In: 3rd Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems. pp. 235–244. Springer, Berlin (2004)
20. Woerndl, W., Eigner, R.: Collaborative, Context-Aware Applications for Inter-networked Cars. In: 16th IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. pp. 180–185. IEEE, DC, USA (2007)