Personal Tour: a multi-agent recommender system of travel packages

Fabiana Lorenzi
Invenio Software Inteligente &
Universidade Luterana do Brasil, Brazil

Stanley Loh
Invenio Software Inteligente &
Universidade Luterana do Brasil, Brazil and
Universidade Católica de Pelotas, Brazil

Mara Abel
Universidade Federal do Rio Grande do Sul, Brazil

ABSTRACT
This chapter describes the Personal Tour: a multi-agent recommender system designed to help users to find best travel packages according to their preferences. Personal Tour is based on the collaboration of multiple agents exchanging information stored in their local knowledge bases. Based on the paradigm of the Distributed Artificial Intelligence, a user recommendation request is divided into partial recommendations handled by different agents, each one maintaining incomplete information that may be useful to compose a recommendation.

Keywords: tourism, multi-agent recommender systems, recommender systems

INTRODUCTION

Users of e-Commerce Web sites still face the information overload problem. Recommender Systems (RSs) have been proposed and developed to cope with this problem (Resnick et al., 1994), and more in general to support the information selection and decision making processes on e-Commerce Web sites. These systems are being applied in different domains (Gunawardana & Meek, 2009).

Multi-agent recommender systems are being used for managing information from different sources in domains where the knowledge is distributed. Agents are able to retrieve, filter and use information that may be relevant to recommendation decision process.

This paper presents Personal Tour, a multi-agent recommender system that helps users to find travel packages according to their preferences. Agents can exploit knowledge about previous recommendations in order to determine solutions that suit the wishes and needs of a customer. They are able to aggregate information and match the recommendations with the information that the user is looking for.

Personal Tour has features that help to generate better recommendations, such as:

- Agents are able to perform parts of the recommendation (for example, locomotion ways) in order to cooperate for presenting the final recommendation to the user (the whole travel package);
- Agents are able to exchange information with other agents from the community when necessary;
- Agents become experts in specific part of the recommendation;
- Agents have specific knowledge (for example, alternative flights from different companies) and they are able to search for information needed for the recommendation in their own knowledge bases.

This paper is organized as follow: section 2 presents the related work in multi-agent recommender systems, section 3 presents the Personal Tour and its components, section 4 shows some experiments done in order to validate the system and finally, the last section presents some conclusions and future work.

2. RELATED WORK

Multi-agent models have been applied to retrieve, filter and use information relevant to the requested recommendations. MAPWEB (Camacho et al., 2006), for example, is a multi-agent approach that plans travels according to the preferences of the user. It has 4 different agents: UserAgent, that is responsible for the communication between the user and the system; the PlannerAgent, that is responsible for planning the travel; the Webbot, that is responsible for searching information in Internet; and the CoachAgent that acts like a coach for the group of agents, controlling them and assigning tasks to them. The agents are able to store the generated plans as cases and use these cases to build new plans.

The disadvantage of MAPWEB is the fact that the CoachAgent controls and manages the tasks and it controls the possible communications and indicates who must help who. Moreover, there is no process to validate the knowledge of the agents. Agents may work with outdated information during the planning process generating bad recommendations. In Personal Tour there is no central knowledge or task manager, so the division of tasks is made by common agree among the agents in the system and according to the specialty of each agent.

SmartClient (Torrens et al., 2002) is another multi-agent system applied in the tourism domain that helps the user to plan a flight route by representing the space of solutions (recommendations) as constraint satisfaction problems. Users define the departure city, the cities they want to visit and the travel dates. With these preferences in hand, the system builds a constraint satisfaction network able to exploit the possible routes. This approach has a couple of disadvantages. First, it collects route information from the server only once to avoid several costly accesses. That limits the search space because and it is not possible for the user to modify the preferences and to refine the query. Secondly, the domain variables involved in the constraint network are fixed by the system and it is not possible to explore solutions that are not originally in the search space. In Personal Tour users may criticize the recommendation received, refining the query through the modification of the informed preferences.

MAPWEB and SmartClient differ from Personal Tour in three aspects. First, in Personal Tour, an agent is fully autonomous and it can decide which task it will perform according to the specific knowledge it has. Second, agents are able to become experts in some part of the recommendation through their confidence degrees. Third, agents are able to assume information during the recommendation process in cases of missing information necessary to generate a recommendation.

3. THE PERSONAL TOUR RECOMMENDER SYSTEM

Personal Tour is a multi-agent recommender system designed to help users to find best travel packages according to their preferences. It is based on the collaboration of multiple agents exchanging information stored in their local knowledge bases, following the Distributed Artificial Intelligence paradigm.

This multi-agent recommender system was projected to be used in travel agencies where travel agents have to deal with specific customers' needs and broad knowledge about tourism options, have to exchange information among them and many times they have to suppose information about customers in order to generate recommendation of a travel package. Due the multi-agent feature, the system may be used for different travel agents at same time, solving several recommendation requests in an asynchronous way.

A user recommendation request is divided into partial recommendations (different travel services) handled by different agents, each one maintaining incomplete information that may be useful to compose a recommendation and accomplishing part of the whole task.

3.1 Cycle of Recommendation

A cycle of recommendation in Personal Tour, as shown in Figure 1, starts collecting the preferences from the user through the main interface (step 1) for each travel service (flight, hotel and attractions). These preferences are used to create the tasks of the recommendation that will be performed by the agents of the system (step 2). In step 3, agents choose tasks to solve (according to their confidence in each travel service) and from this moment they start to perform the tasks searching for the information necessary to compose the recommendation. Agents have two ways of searching information: **local**, when agents search in their knowledge base (step 4a); or **in the community**, when agents cooperate, communicating to each other to get the information (step 4b).

The local knowledge base is composed by historical data from other customers' travels, storing each recommendation presented to the customer (information about flights, hotels and attractions generated by the system to each request of the user). This information is structured as cases following the Case-based Reasoning approach, where the description of the problem is the user's request and the description of the solution is the recommendation presented by the agents.

After the task is solved, agents change the status of the performed task (step 5) and the results of all tasks are returned to the interface (step 6) that is responsible for presenting the whole recommendation to the user (step 7). To complete the cycle of recommendation, the user evaluates the recommendation received and this evaluation is stored in the agents' knowledge base (step 8).

The combination of agents with the decomposition of the customer request in small parts (tasks) is a feature that allows Personal Tour to present a complete recommendation to customers and this is an important feature to a recommender system applied in the tourism domain.

After receiving the customers preferences, Personal Tour creates the list *L* of tasks and the available agents are able to choose tasks to perform. An interesting feature of Personal Tour is the fact that the available tasks are interdependent but agents may perform them asynchronously. For example, tasks of hotel and attractions are dependent of the flight information and they should be performed after receiving the information provided by flight task. However, agents have an assumption component that helps to deal with this issue (this will be further explained in section 3.5).

Figure 1: A recommendation cycle in Personal Tour

3.2 Customer Profile

The customer model in Personal Tour is composed by the preferences of the customer for each travel service informed in the main interface and it is represented by $U_t = \{i_1, ..., i_m\}$ where each i_m represents a preference.

Every time the customer asks for a new recommendation, the current preferences are stored in a new XML file. This file is named with the customer name and the current date. The set of all XML files of the customers compose the customer profile and may be used to infer information about the customer when necessary. Figure 2 shows an example of the XML file with a request of the customer.


```xml
<object.Profile>
  <user>
    <name>John</name>
    <password>XXXXX</password>
  </user>

  <travel>
    <destination>Paris</destination>
    <home>Rio de Janeiro</home>
    <dateTravel>2010-10-22</dateTravel>
    <dateReturn>2010-10-29</dateReturn>
    <passangers>2</passangers>
  </travel>

  <flight>
    <flightClass>Economic</flightClass>
    <type>night</type>
    <stops>false</stops>
    <connections>false</connections>
  </flight>

  <hotel>
    <category>tourist</category>
    <pool>indoor</pool>
    <pet>false</pet>
    <type>twin</type>
    <wifi>true</wifi>
  </hotel>

  <attraction>
    <type>
      <string>Monument</string>
    </type>
```

</attraction>
</object.Profile>

Figure 2: Example of the XML file with a customer request.

3.3 Agents

Personal Tour consists of a set of agents defined as $C = \{a_1, a_2, ...,a_n\}$, where we assume that agents work in a cooperative way and they know each other.

Figure 3: Architecture of the agent

An agent is able to perform a part of the recommendation in response to a user query. Figure 3 shows the agent architecture with its main components that are:

- **Knowledge**: an agent has its knowledge base where it stores all recommendations performed. The knowledge is stored in the knowledge base as cases and each customer request generates 3 cases (a case of flight, a case of hotel and a case of attraction);

- **Assumptions**: an agent is capable of assuming some information in order to perform a task. These assumptions are related to the data that is not available as the time the agent needs it during the recommendation process; for example, agent1 is performing a task of hotel and it needs information about the flight (that is being performed by agent3). If it does not receive the information from agent3, if may assume this information in order to solve the task;

- **Specialization**: agents become experts in some type of task and this specialization helps agents to decide which task they should perform during the recommendation process.

When an agent does not have information necessary to complete the chosen task, we say that agents cooperate. In Personal Tour, this cooperation is considered a capability of the agent and it is given by the communication among agents that exchange information in order to complete its part of the recommendation.

3.4 Local Knowledge/Case Base

Each agent has a case-base where all the tasks performed are stored as cases. Each case is composed by the description of the problem (that is represented by the set of preferences informed by the user) and the description of the solution (represented by the recommendation generated by the agent and the evaluation given by the user to this recommendation).

Each case is composed by:

- **The request of the user**: the query is the set of needs and preferences chosen by the user.

- **The recommendation**: the recommendation generated by the agent to the travel service.

3.5 Assumptions

As tasks are interdependent, some of them should be performed before others. However, as the agents work asynchronously, sometimes the order of tasks is not respected by the agents and it may cause lack of information during the recommendation process. Assumptions allow agents to reason with incomplete information (by making guesses), i.e., they represent knowledge that the agent supposes to believe.

When an agent is performing a task that depends on the result from another one, it may assume information rather than waiting for the information from other agent. For example, if $agent_1$ is performing hotel task, it needs information about the flight in order to know which hotel is better to recommend but the flight task is being performed by $agent_2$. It may happen that $agent_2$ take too long to solve its task or it may become unavailable during the recommendation process. These problems can generate a large, even more than expected, delay in the answering time which is not good for the system.

In order to use assumptions during the recommendation process it is necessary to formulate the set of assumptions that agents may access. We propose two different methods to generate the set of assumptions during the recommendation process:

- **The most popular option in the community of users**: if the customer has no profile in the system yet, then the agent sums the number of occurrence of each option for each attribute in the past travels of all customers stored and the most popular option will be used by the agent as assumption;

- **Similar cases**: the agent searches for the most similar customer in its case base and uses the options to the new customer. In order to find the most similar customer, the agent uses the similarity measure, comparing attributes of the new customer with all customers in its knowledge base. Here, it is necessary to define a threshold that represents how much the case is similar to the new customer.

In Personal Tour the hotel and attraction tasks depend on the information from the flight task. Regarding the hotel task, for instance, the agent needs information about the arrival time in order to accomplish its recommendation.

The use of assumptions in the recommendation process guarantees that agents are able to complete their tasks even when there is lack of information during the recommendation process and in an acceptable time for the user, without jeopardizing the performance of the system.

3.6 Specialization

In Personal Tour, an agent may become expert in a specific travel service during the recommendation cycles. The agent has one confidence degree for each travel service (flight, hotel and attractions). These confidence degrees represent how much the agent is expert in each travel service and agents use them to choose the tasks to perform.

The confidence degrees of each agent are updated according to the new evaluations received from the customer. The new evaluation may increase (if the customer liked the recommendation presented by the agent) or decrease (if the customer did not like it) the agent confidence degree for the travel service.

If the user's evaluation was positive, then the confidence degree will be increased which means that the agent will become more expert in the travel service. On the other hand, if the user's evaluation was negative, the confidence degree will be decreased.

3.6.1 Customers Evaluations

The customer has an important role in the recommendation process evaluating the recommendations presented by the system. After receiving the final recommendation, the customer evaluates each item of each travel service with a rate: “I like it” (represented by 1) or “I did not like it” (represented by 0). This evaluation is used to update the confidence degrees of the agents. The confidence degrees are increased or decreased according to the received evaluations.

Thus, the evaluation model is represented by a vector $E = \{e(i_1, r_1), ..., e(i_m, r_m)\}$, that describes the evaluation of the user about the recommendation item received (r_m), considering the preference informed by the user in the query (i_m). The final evaluation rate to each travel service is a value between the range [0, 1].

Equation 1 shows how the final evaluation rate ($v(t_n)$) is calculated to each travel service, where *m* is the number of attributes for the travel service and $e(i_j,r_j)$ is the evaluation rate for the attribute i_j.

<equation 1>

(1)

We consider good recommendation when $v_{tn} = 0.5$, i.e., when user has rated as “I liked it” at least half of the attributes recommended.

3.6.2 Updating the Confidence Degrees

The task evaluation is then used in the agent confidence degree computation so that the agent increases the confidence regarding a travel service when it solves the task in a better way.

As the confidence degree represents how much an agent is becoming an expert in a specific travel service, it is used by the agent to choose the task it will perform. Thus, agents consider their confidence degrees when they choose the next task to perform. When *L* is available, the agent will choose one task that belongs to the travel service with the greater confidence degree. For example, if its confidence degree is 0.4 in flight, 0.6 in hotel and 0.2 in attraction, the agent will choose a hotel task to perform because it has more expertise in hotel tasks than others.

We use equation 2 to update confidence degrees where:

- z is the number of evaluations received for the travel service t_n performed by a_i;
- Θ is the number of days elapsed since the evaluation;
- $\mathcal{T}$ is a constant that defines the weight of the evaluation $v(t_n)$ in the update of the confidence degrees of the agents.

<equation 2>

(2)

$T^{tn}_{ai,\ ai}$ returns a value between 0 and 1, where 0 represents the minimum confidence degree of a_i in the travel service $_t$ and 1 represents the maximum confidence degree of a_i in the travel service $_t$. $\mathcal{T}$ has to be set according to the number of days (Θ) the evaluation lasts.

An important feature of the specialization applied in Personal Tour is that when updating the confidence degrees in each agent it considers that most recent evaluations have more influence than old evaluations.

3.7 Cooperation

In Personal Tour, agents are able to cooperate with each other, exchanging information during the recommendation process. When an agent does not find in its knowledge base the information necessary to generate the recommendation requested in the task, it may communicate with other agents, asking for the information. For example, if a_3 is solving an attraction task and it does not have information about attractions in Lisbon City, it may ask to other agents in the community.

In this cooperation process, the agent sends a message to all available agents in the community asking for the information. Then, it chooses the first agent that answered and stores the received information in its knowledge base.

However, it might happen that nobody in the community has the information and nobody will answer. In this case, the agent will starts its assumption component in order to generate the recommendation. We defined a waiting time of 30s, before uses assumptions.

4. EXPERIMENTS

Experiments were done to validate the Personal Tour. The first step was the knowledge acquisition where 300 cases were obtained from real customers of a travel agency, to create the case-bases of the agents. Each case represents one travel and is composed with flight, hotel and attractions information that corresponds to 900 performed tasks.

As different users may use the system at same time, more than 3 agents are necessary in the system to have a good performance. In our experiments, 10 agents were created in the community. The cases obtained from the travel agency were randomly stored in the knowledge bases agents. The agents are not seen by the user and they run background in the system.

Figure 4 shows the main interface where the user interacts with Personal Tour. In order to generate a recommendation, agents need to know the needs of the user, such as, destination, departure date and number of passengers. In this first version, only one destination may be selected. This issue is being worked for the next version.

As mentioned previously, a travel package must be composed by three travel services: flight, hotel and attractions (but the customer may choose which travel services he wants in the travel package). An important feature of the Personal Tour is that these preferences are dynamic and the user may insert a new feature in the request moment.

Figure 4: Personal Tour main interface

The Personal Tour was run in a travel agency for 2 months (June and July/2010) in order to validate the recommendations generated by the agents. During these experimental months, the recommendations requested by the customers were generated in two ways in parallel: by a human travel agent (the expert of the travel agency) and by the Personal Tour (by the agents of the system). The human travel expert is a travel agent that works in a travel agency for more than 15 years and is expert in travel packages recommendations.

Both recommendations were showed to the customers that evaluated them. As shown in section 3.6.1, the user evaluates all the attributes of the recommendation with a rate 0 or 1 (negative/positive), in each travel service (flight, hotel and attractions). The evaluation by attribute will be used in future improvements of Personal Tour where agents will generate a new recommendation according to the negative rates of the user and update the user profile. Next, we asked the customers to evaluate the recommendations received (the travel package) and to indicate which one they prefer to buy (with the rate “I would like to buy it” or “I would not like to buy it”).

Thus, the number of recommended travel packages that the customers would like to buy was considered as a metric. From the 73 travel packages recommended by the Personal Tour, 66 travel packages were rated with “I would like to buy it”, i.e., and 61 travel packages recommended by the expert were indicated as “I would like to buy it” which represents 83.56%.

Table 1 shows the percentage of purchasable travel packages during the experimental months of Personal Tour. We can see that agents had performance superior to the expert in the generation of the recommendations.

Table 1: Percentage of purchasable travel packages in the experimental months

Method of recommendation	Percentage of purchasable travel packages
Personal Tour	90.41
Human Expert	83.56

We claim that the results of the purchasable travel packages using the system was better due the fact that the specific knowledge is distributed over agents that are capable of cooperate and assume information during the recommendation process. In real travel agency, the recommendation is performed by only one human travel agent, in a centralized way. Distributing the knowledge in different agents and letting them become experts in travel services improve the quality of the recommendations.

ACKNOWLEDGEMENTS

This work was partially supported by the project PERSONAL TOUR: an intelligent software for personalized tourism packages, under support from CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil, grant n. 561408/2010-6) and from FINEP (Agência Financiadora de Estudos e Projetos - Brasil, program PRIME).

CONCLUSIONS AND FUTURE WORK

This paper presented a multi-agent recommender system, called Personal Tour, applied to the tourism domain in order to recommend travel packages. A recommendation is divided in tasks and each agent is responsible to perform some tasks. Agents work in a cooperative way to recommend travel packages to the user.

Agents become experts in a specific travel service over time. We can say that agents become travel agents, where each one has specific knowledge and the cooperation among them results in good recommendations. This feature helps the system to mimic what happens in a travel agency where each travel agent has specific knowledge about a travel service and the cooperation among all them generates the final recommendation to the customer.

Personal Tour has an assumption component that enables agents to assume information during the recommendation process when there is lack of information to generate a recommendation.

The system was tested in a real travel agency and the results obtained after two months of use showed that Personal Tour is capable of improving the recommendations presented to customers, due the fact that agents have their knowledge bases and they cooperate during the recommendation process.

Decomposing the problem and distribute it to several different agents that become more and more specialized can yield good recommendations, even when applied to tourism that is a complex domain that needs specific knowledge distributed over different sources.

Personal Tour is applied in the tourism domain but we believe that the approach may help customers in other applications that deal with dynamic and distributed knowledge to generate recommendations.

As future work we want to develop a mechanism to validate the assumptions used by agents in the recommendation process. In this version of Personal Tour, agents generate and manipulate assumptions but we do not consider what happen when an assumption received a negative evaluation and this is important to improve the process.

REFERENCES

Camacho, D.,Molina, J. M., Borrajo, D., Aler, R. (2002). Multi-agent plan based information gathering. Applied Intelligence, 25(1), 59–71.Hingham, MA.

Gunawardana, A. & Meek, C.(2009). *A unified approach to building hybrid recommender systems.* In Proceedings of the third ACM conference on Recommender systems - RecSys’09, (pp. 117–124) New York:ACM.

Macho, S., Torrens, M., Faltings, B. (2000) *A multi-agent recommender system for planning meetings.* In WORKSHOP ON AGENT-BASED RECOMMENDER SYSTEMS (WARS’2000).

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. (1994). *GroupLens: an open architecture for collaborative filtering of netnews.* In ACM CONFERENCE ON COMPUTER-SUPPORTED COOPERATIVE WORK, (pp.175–186).

Torrens, M., Faltings, B., Pu, P. (2002) SmartClient: constraint satisfaction as a paradigm for scaleable intelligent information systems. *CONSTRAINTS: an International Journal*, *7*(1), p.49–69.