Case-based recommender system inspired by social insects

Fabiana Lorenzi[1 2], Daniela Scherer dos Santos[2] and Ana L. C. Bazzan[2]

[1]Universidade Luterana do Brasil
Av. Farroupilha, 80001
92420-280 Canoas, RS

[2]Instituto de Informática
UFRGS
Caixa Postal 15064
91.501-970 Porto Alegre, RS, Brasil

lorenzi@ulbra.tche.br,{dsradtke,bazzan}@inf.ufrgs.br

Abstract. *Case-based recommender systems can learn about user preferences over time and automatically suggest products that fit these preferences. In this paper we present such a system, called CASIS. In CASIS we combined case-based reasoning approach with a metaphor from colonies of social insects, namely the honey bee dance. In nature, this metaphor is used to indicate the best nectar source among honey bees. Similarly we use it to retrieve the most similar case to the user's query. Our results show that this combination is effective when used in the retrieval step of the recommendation cycle: the most similar case is found by the "bees".*

1. Introduction

Recommender systems are being used in e-commerce web sites to help the customers selecting products more suitable to their needs. The growth of Internet and eCommerce has brought the need for such a new technology [Schafer et al. 2001], and a number of research projects have focused on these systems [Resnick et al. 1994], applying different approaches. One of these approaches, called knowledge-based recommender system, allows the system to learn about user preferences over time and automatically suggest products that fit the learned user model.

This paper focuses on case-based recommender systems which is classified as a knowledge-based approach. This is realized in the CASIS system, a combination of two approaches: case-based reasoning and swarm intelligence through the honey bees dance metaphor. The former allows to retrieve solution cases and use them to solve new problems. The latter approach is inspired by social insects: honey bees dance to indicate the best nectar source. This combination would still be classified as a case-based recommender system according to the unifying framework presented in [Lorenzi and Ricci 2004]. The motivation of combining these approaches is to improve the recommendation to the user.

The next section provides a brief introduction to case-based recommender systems and to swarm intelligence based approaches. Section 3. describes the proposed approach and section 4. shows the experiment results. Finally section 5. shows the conclusions and the future work.

2. Related Work

2.1. Case-Based Recommender Systems

Case-Based Reasoning (CBR) is a problem solving methodology that deals with a new problem by first retrieving a past, already solved similar case, and then reusing that case for solving the new problem [Aamodt and Plaza 1994]. In a CBR recommender system (CBR-RS) a set of suggested products is retrieved from the case base by searching for cases similar to a case described by the user [Burke 2000].

CBR can support the recommendation process in a number of ways. In the simplest approach, the CBR retrieval is called taking as input a partial case defined by a set of user preferences (attribute-value pairs), and a set of products matching these preferences are returned to the user. In this process we can identify some basic tasks such as the input (where the user provides his/her requirements), the products retrieval (where the system searches for products according to user requirements), and the output, where some recommendation is given to the user.

In the simplest application of CBR to recommendation, the user is supposed to look for some product to purchase. He/she inputs some requirements about the product and the system searches the case base for similar products (by means of a similarity metric) that match the user requirements. A set of cases is retrieved from the case base and these cases can be recommender to the user. If the user is not satisfied with the recommendation he/she can modify the requirements, i.e. build another query, and a new cycle of the recommendation process is started.

The case retrieval is typically the main step of the CBR cycle and the majority of CBR recommender systems can be described as sophisticated retrieval engines. For example, in the Order-Based Retrieval [Bridge and Ferguson 2002] the system uses special operators to retrieve a lattice of cases, or in the Compromise-Driven Retrieval [McSherry 2003] the system retrieves similar cases from the case base but also groups the cases, putting together those offering to the user the same compromise, and presents to the user just a representative case for each group.

2.2. Swarm intelligence

The use of the social insect metaphor to solve computer problems such as combinatorial optimization, communications networks or robotics is increasing [Bonabeau et al. 1999]. Social insects living in colonies, e.g. ants, termites, bees, and wasps [Wilson 2000] distinguish themselves by their organization skill without any centralized control [Camazine et al. 2003, Gordon 1996]. Organization emerges from interactions among individuals, between the individuals and the environment, and from behaviors of the individuals themselves [Bonabeau et al. 1999].

Several approaches exist; the complete list is outside the scope of this paper. However, the following works are related to our approach: [Finbarr and Bridge 2004, Solnon 2002]. The former has implemented and compared six Ants Colony Optimization (ACO) algorithms [Dorigo et al. 1999] to solve constraint satisfaction problems.

This section describes swarm intelligence and specifically self-organization among honey bees. Among these, the colony selects the best nectar source available through simple behavioral rules. The process of dispatching bees into the surrounding countryside to gather the colony's food is called foraging. Bees travel up to 10km from the hive to collect nectar. They return with nectar and information about the nectar source [Camazine et al. 2003].

The bee has three behavior options in the foraging process [Camazine et al. 2003]:

1. to share the nectar source information by dancing, a behavior in which a bee communicates to other bees the direction, distance, and desirability of the food source, trying to recruit new bees to that food source.
2. to continue foraging without recruiting other bees.
3. to abandon the food source and go to the area inside the hive called the dance floor to observe dancing bees and select its next food source.

In [Seeley et al. 1991] the authors presented an experiment showing the recruitment of nectar forager bees to feeders containing sugar solutions. Two food sources are presented to the colony at 8:00 a.m. at the same distance from the hive: source A is characterized by a sugar concentration of 1.00 mol and source B by a concentration of 2.5 mol. At the noon, the sources are exchanged: source A is now characterized by a sugar concentration 2.5 mol and source B by a concentration of 0.75 mol. In both situations the better source is more visited. This experiment showed that the foraging decision of each bee is based on very limited information of its own particular visited source. This simple behavior allows the colony to select the best quality source of nectar.

Based on these observations, colonies select the best quality source through the rate of dancing and abandonment based upon nectar source quality. Camazine and Sneyd [Camazine and Sneyd 1991] developed a mathematical model which demonstrates how the properties of the system emerge automatically from the dynamic interactions among the constituent components.

The recruitment of other bees to forage a good nectar source is considered a positive feedback in the honey bees organization [Tarasewich and McMullen 2002]. The positive feedback is also a step of the case-based recommender system's cycle and its importance is well recognized in many researches (see [McGinty and Smyth 2002] for more details).

In the next section we show how the metaphor from honey bees behavior can be applied to the development of case-based recommender systems.

3. The CASIS system

3.1. The mathematical model

The CASIS system is based on a mathematical model described by Camazine and Sneyd [Camazine and Sneyd 1991] which demonstrates how the properties of the system emerge automatically from the dynamic interactions among the bees. To validate this mathematical model, the authors considered two nectar sources (A and B). Figure 1 shows the flow diagram with the scenerio. It is divided in seven compartments where the bees can follow one of the behaviors:

- H_A and H_B: unload nectar from food source A and B, respectively;
- D_A and D_B: dance for food source A and B, respectively;
- A and B: forage at food source A and B, respectively;
- F: follow a dancer (after having watched the dancer).

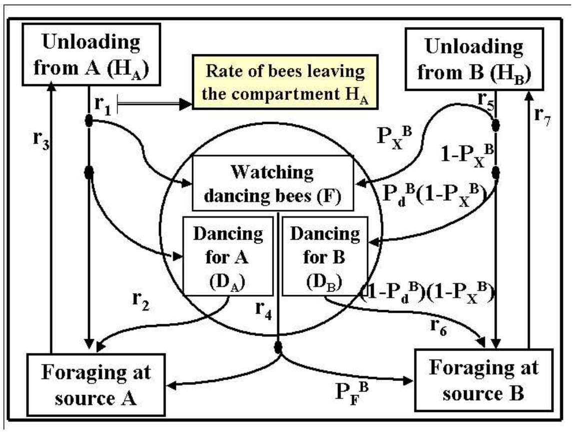

Figure 1. Camazine and Sneyd's mathematical model [Camazine and Sneyd 1991]

Two factors affect the proportion of the total forager force in each of the seven compartments: the rate at which a bee moves from one compartment to another r_{1-7}, and a probability that a bee takes one or other fork at five branch points. These forks are presented in figure 1 as black points. We give the formulas only for source B, for the sake of example.

After the bee unloaded the nectar in the hive, it finds the first decision fork: abandon the source B (P_X^B) or continue in the source B ($1 - P_X^B$). If it chooses to continue in the source, then it can decide also if it recruits new bees for this source ($P_d^B(1 - P_X^B)$) or not. If the bee has decided to abandon the source it can watch other dancers and decide which one it will follow for nectar source (P_F^B). This probability is estimated by:

$B(P_F^B) = \frac{D_B d_B}{D_B d_B + D_A d_A}$.

where:

- d_A and d_B are the proportions of time that the foragers actually dance; it must be said that only a portion of a bee's time in the dance area is actually spent on dancing, and that the dancing time is proportional to the quality of the nectar source (thus the best the source, the higher the chance that other bees will observe the dance and go to that source);
- D_A and D_B are the number of bees in the source A and B respectively.

Summarizing, the probabilities are:

- P_X^A and P_X^B: probabilities of abandoning A and B respectively, per foraging trip;
- P_d^A and P_d^B: probabilities of dancing for A and B respectively;
- P_F^A and P_F^B: probabilities of following a dancer for A and B respectively.

3.2. The approach

In our approach, Camazine and Sneyd's mathematical model was adapted and combined with case-base reasoning to a case based swarming intelligence recommender system (called CASIS).

In the simulation, each bee has the following features:

- probability of abandoning the nectar source;
- probability of recruiting more bees by dancing;
- probability of keep following the source.

Each nectar source is considered a case and the set of sources is the case base. Given a new user query, the metaphor calls for honey bees leaving the nest looking for nectar sources. As shown in Camazine and Sneyd's model, each bee's choice is calculated by using probabilities. The most important is the abandoning's probability where the bee checks the quality of the nectar source and decides whether or not to continue in this source. In our system, the original model was modified and the bee can forage for several different sources (not only two as in the original model).

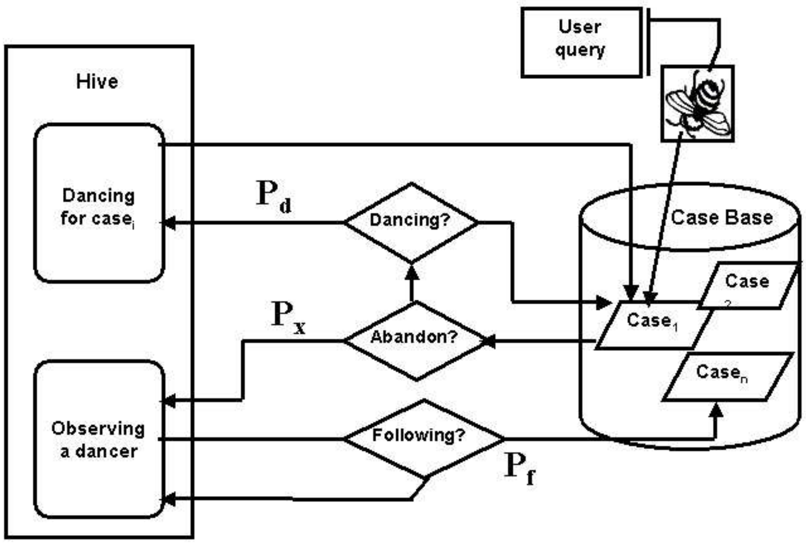

Figure 2. CASIS's approach – adapted from Camazine and Sneyd

Figure 2 shows the adapted model with the hive having two compartments (dancing or observing), the case base (representing the available sources), and the diamonds representing the bee's decisions possibilities. To decide what to do, some probabilities are calculated:

- P_x^i: probability of abandoning case i, per foraging trip;
- P_d^i: probability of dancing for case i;
- P_F^i: probability of following a dancer for case i.

The probability of abandoning a case was adapted in our model. In a traditional case-based recommender system, the new query is compared with all cases in the case base and the most similar is shown to the user. But in CASIS, the similarity is used to help the bee to make its decision to continue in the nectar source or not. P_x^i is the distance between the new query and the cases. The cases with the smallest distance to the new query are the most similar, i.e. they represent the best nectar sources.

The probability of following a dancer was also adapted because we are not taking into account the time the bee spend dancing. Thus, $P_F^i = \frac{D_i}{\sum_{j=1}^{n} D_j}$ where D_i is the number of dancers bees to the source *i* and *n* is the total number of bees.

3.3. Validation

One scenario from tourism was used to validate the approach. The recommender cycle starts with a simple query where the user specifies some preferences. The next step is the retrieval. The bees start to look for the best cases and after a number of iterations the best one is recommended to the user.

The review step was implemented in a different way here. Normally, a recommender system allows the user to refine the query and to start a new recommendation cycle. However, in this experiment we use the honey bees feedback to search cases to recommend, meaning that one recommendation cycle is composed by many iterations. It is important to notice that the probabilities regarding each bee are recalculated in each iteration, using the similarity between the case and the query.

The bees' behavior always allow them to recommend something to the user thus avoiding empty recommendations. This is due to the fact that the system always recommend the best nectar source, i.e. the case which has the highest number of bees in the end of the iterations. The revise and retain steps were not implemented in this experiment.

4. Results

The implementation was done using the SeSAm tool[1] for agent-based simulation.

In our model the parameters of the simulation are the number of cases, the number of bees, and the number of iterations. Although we have ran the system with different sets of parameters, the case base was the same in each experiment. It has 100 cases and each case represents a tourism package from Brazilian tourism operators. Attributes of the cases (packages) are: destination, length, and price.

In the first experiment, we used 2500 bees (which is considered a high number) and the system was allowed to run until the best case emerges. Figure 3 shows the results of this first experiment. We plot only the most visited cases during the simulation in order to avoid a dense graphic. Among these, only three cases have high similarity with the user query. The figure shows that, in the beginning, bees consider various cases because they have little information about them (as it is the case with real nectar sources which are visited randomly). By time t=20 the best case emerges as *case 04* and this is in fact the best package the user can have (in the case base, *case 04* is the one with the highest similarity to the user's query).

[1] available for download at http://www.simsesam.de

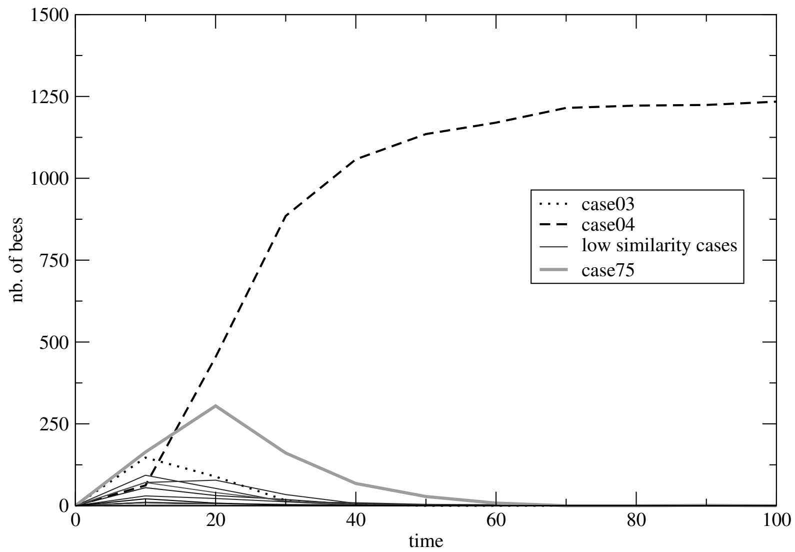

Figure 3. Number of bees visiting each case along time: experiment with 2500 bees

In the next experiment, shown in figure 4, we used the same user query but we reduced the number of bees to 500, allowing the simulation to run up to 500 iterations. Here, an interesting situation arises: bees started to visit *case 69* because it had the highest similarity *in the beginning*. However, as soon as *case 04* was founded as a better case (around t=150), bees start abandoning visiting *case 69* and prefer to visit *case 04*. This shows that the approach is effective.

Moreover, the approach is also robust to dynamic changes in the queries: during the iterations, it is possible to notice that the bees' behavior is automatically clustered around cases according to the user's preferences. For example, in the tourism scenario, users' preferences usually change with seasons of the year: during summer beaches are favorite spots, whereas in winter people normally look for skying or tropical resorts. Once the user's query changes, so does the clustering of bees.

5. Conclusions and Future Work

This paper presented the CASIS system, a case-based recommender systems that uses swarm intelligence in the retrieval step. This use is justified by the success social insect have in dynamic changing environments.

Normally only the pheromone metaphor is used, which is suited for optimization problems. The motivation of our work was to explore a different and more suitable metaphor for CBR: the honey bees dance. The positive feedback of the social insects can be considered as a common point between the bee's behavior and the case-based recommender systems.

Our experiments' results have shown that using this metaphor the system always

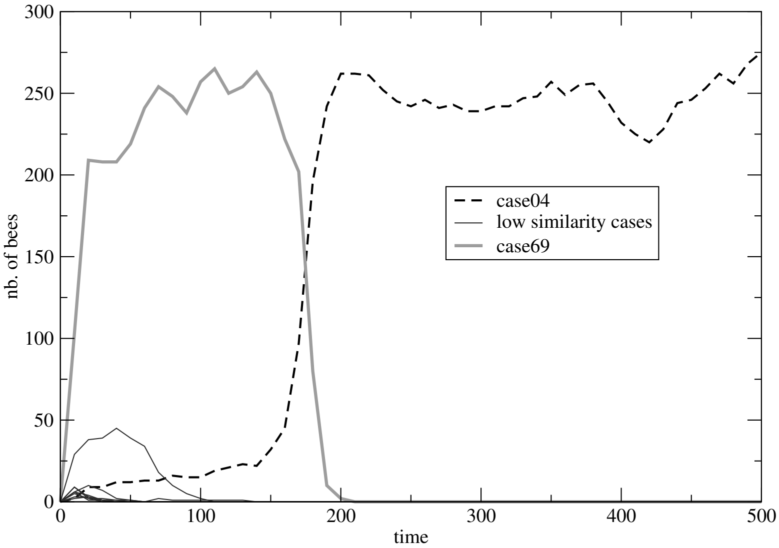

Figure 4. Number of bees visiting each case along time: experiment with 500 bees

return something to recommend to the user, avoiding the user's disappointment with the recommender system.

In the future we intend to explore two main directions. First, we intent to better study the behavior of the bees (e.g. varying their number) in highly dynamical environments, meaning that user's queries change rapidly. We are particularly interested in how robust the system is and how many bees are necessary to guarantee this robustness.

As another future work we want to improve the case representation, saving the previous queries as cases. Depending on the similarity of the new query the bees can start not completely randomly but using information gathered regarding the previous query.

References

Aamodt, A. and Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

Bonabeau, E., Thraulaz, G., and Dorigo, M. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford Univ Press.

Bridge, D. and Ferguson, A. (2002). Diverse product recommendations using an expressive language for case retrieval. In Craw, S. and Preece, A., editors, *Advances in Case-Based Reasoning, Proceedings of the 6th European Conference on Case Based Reasoning, ECCBR 2002*, pages 43–57, Aberdeen, Scotland. Springer Verlag.

Burke, R. (2000). Knowledge-based recommender systems. In Daily, J. E., Kent, A., and Lancour, H., editors, *Encyclopedia of Library and Information Science*, volume 69. Marcel Dekker.

Camazine, S., Deneubourg, J. D., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton.

Camazine, S. and Sneyd, J. (1991). A model of collective nectar source selection by honey bees: Self-organization through simple rules. *Journal of Theoretical Biology*, 149(4):547–571.

Dorigo, M., Di Caro, G., and Gambardella, L. (1999). Ant colony optimization: A new meta-heuristic. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A., editors, *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1470–1477, Mayflower Hotel, Washington D.C., USA. IEEE Press.

Finbarr, T. and Bridge, D. (2004). When ants attack: Comparing ant algorithms for constraint problems. In *Fifteenth Irish Conference on Artificial Intelligence and Cognitive Science*, pages 167–176.

Gordon, D. (1996). The organization of work in social insect colonies. *Nature*, 380:121–124.

Lorenzi, F. and Ricci, F. (2004). Case-based reasoning and recommender systems. Technical report, IRST.

McGinty, L. and Smyth, B. (2002). Comparison-based recommendation. In Craw, S. and Preece, A., editors, *Advances in Case-Based Reasoning, Proceedings of the 6th European Conference on Case Based Reasoning, ECCBR 2002*, pages 575–589, Aberdeen, Scotland. Springer Verlag.

McSherry, D. (2003). Similarity and compromise. In Aamodt, A., Bridge, D., and Ashley, K., editors, *ICCBR 2003, the 5th International Conference on Case-Based Reasoning*, pages 291–305, Trondheim, Norway.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings ACM Conference on Computer-Supported Cooperative Work*, pages 175–186.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153.

Seeley, D., Camazine, S., and Sneyd, J. (1991). Collective decision-making in honey bees: How colonies choose nectar sources. *Behavioral Ecology and Sociobiology*, 28:277–290.

Solnon, C. (2002). Ants can solve constraint satisfaction problems. *IEEE Trans. Evolutionary Computation*, 6(4):347–357.

Tarasewich, P. and McMullen, P. (2002). Swarm intelligence: Power in numbers. *Communications of the ACM*, 45(8):62–67.

Wilson, E. O. (2000). *Sociobiology: The New Synthesis*. Harvard Univ Press.