Short-Term Profiling for a Case-Based Reasoning Recommendation System

Esma Aimeur and Mathieu Vézeau

Computer Science Department - University of Montreal
C.P. 6128, Succ. Centre-ville
Montreal, Quebec Canada H3C 3J7

aimeur@iro.umontreal.ca
Phone: 1-514-343-6794
Fax: 1-514-343-5834

Abstract. In this paper, we aim to address a frequent shortcoming of electronic commerce: the lack of customer service. We present an approach to product recommendation using a modified cycle for case-based reasoning in which a new refinement step is introduced. We then use this cycle combined with a heuristic we devised to create a short-term profile of the client. This profile is not stored or reused after the transaction, reducing maintenance. In fact, it allows the client and the system to find an appropriate product to satisfy the client on the basis of available products in a potentially efficient way.

1 Introduction

Electronic commerce on the Internet is steadily gaining importance and promises to revolutionize the way we exchange products and services. However, many problems remain to be solved before we can fully exploit the potential benefits of this new paradigm. One of those problems is the lack of customer service in electronic commerce applications. For now, sales support offered by enterprises to their Internet customers is generally poor, if existent at all. Of course, most sites give their customers the ability to query the available products, by way of catalogs, textual search engines or database interfaces. These tools, however, require the user to expend much effort and can be totally inadequate if the number of products is substantial, if the products are alike or if the consumer does not know the domain very well.

One solution is to use products recommendation systems. Those systems are able to suggest products to clients according to their preferences or specific requirements. This way, those applications contribute to increase customer satisfaction, therefore increasing sales and improving the reputation of enterprises using these systems. One of the promising technologies for the conception of recommendation systems is case-based reasoning (CBR) [1]. The goal of this sub-domain of artificial intelligence is to

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 23-30, 2000.

conceive knowledge-based systems which, to solve new problems, reuse and adapt solutions to prior similar problems.

In this paper, we propose a heuristic to construct a temporary user profile in CBR-based recommendation systems. This approach is able to fulfill one of the most common deficiencies of these systems, that is to say that they react the same way with all users without regard to their respective preferences.

We first explain the proposed heuristic and then give an example for an application of this heuristic in electronic commerce. Finally, we discuss pros and cons of this method and propose future directions.

2 CBR and Short-Term Profiling

In the context of electronic commerce, the CBR cycle can be interpreted as an iterative search process in the multidimensional space of the products. The initial request is used to find a first solution in the set of possibilities, and we expect the user to iterate progressively toward the ideal solution by formulating successive critiques on the different characteristics of the proposed products. This progression is made possible by adding to the cycle a request refinement step during which the system accounts for the critique by automatically modifying the user request and starting a new search. This interpretation implies that unlike traditional CBR systems, where the cycle is generally executed only once by session (search, adapt, evaluate, retain), recommendation systems for electronic commerce, for their part, are used on a basis of many iterations by session (search, adapt, evaluate, retain, adapt, etc.). By session, we mean a phase where the client interacts with the system to find a particular item. A session may contain an arbitrary number of interactions but must end when the client finds a satisfactory product.

Our approach is based on the following assumption: since the client interacts many times with the system in the same session, this situation is propitious to discovering "short-term" preferences of this client. Here we do not speak of sought product characteristics, those characteristics being already clearly specified in the request. We are instead hinting at the relative significance the user attaches to the various parameters describing the sought product. A client looking for a travel package, for instance, could attach more importance to the price than to the destination itself. In that case, the system should respect the user preferences and rely more on the price in its search. This kind of preference is, in our opinion, temporary, hence the qualifier "short-term". Indeed, we believe the importance given to concepts in a product search is more associable with temporary interests and to circumstantial causes than with long-term interests. In the preceding example, it is possible that the user wished to travel at this time of the year but the destination was not important in his eyes. Maybe the user had a limited budget at that time, and though he had a destination in mind, he was willing, if needed, to sacrifice that choice for a price within his budget. That kind of motivation oftentimes depends on the moment and is not necessarily representative of the personality of the user. According to our approach, the user profile is thus valid only for a single session. This contrasts with traditional approaches to user modeling,

where profiles mostly represent long-term interests and preferences, and where those profiles are created and maintained on the basis of many sessions with the system.

We make the hypothesis that the various critiques formulated by the client in his search process for the "ideal product" are a good source of information to construct a temporary profile automatically. In fact, in the kind of applications we consider, we expect the client to make heavier use critiques and automatic adjustments of requests as a way to search instead of the explicit formulation of requests. Critiques therefore are an important source of interactions between the individual and the system. We can suppose, for instance, that a client who frequently criticizes a particular aspect such as the price attaches greater importance to this concept and wishes the system considers it for the rest of the session. We can also suppose that the order in which critiques are made is also an indicator of the client's short term requirements. Instead, we have analyzed the usage of a subtler mechanism that could appear in a more transparent way to the user. This heuristic first supposes that the global similarity of a case with the request is expressed as a combination of local similarities between attributes as well as with weights indicating the relative importance of those attributes. This is the case in many CBR systems, including those using the classical *nearest neighbors* method. It is also the case with the "Case Retrieval Nets" technique [3], popular in electronic commerce CBR applications.

Giving this informal definition is useful at this point: the *deficiency* of an attribute measures to what degree the system considers this attribute to be susceptible to a user critique. The proposed heuristic can then be expressed as such: *if the client does not criticize the most deficient of the considered cases, then the importance of the most deficient aspects must decrease and the importance of the criticized aspect must increase*. The intuitive justification behind this heuristic is that clients have a tendency to first criticize the aspects to which they attach more importance. Therefore, we suppose the user always criticize the most important aspect for him that has not yet been optimized. Hence, if the user does not criticize the aspect the system expects to (i.e., the most deficient), then the relative importance of the concepts as maintained by the system is incorrect and must be updated. In fact, if the relative importance of the attributes more deficient than the one being criticized had been lower than the importance of that one, then the proposed result would have been likelier to satisfy the client request.

We now give a short mathematical formalization of the proposed heuristic, in the context of the nearest neighbors method. According to this method, the *global similarity* between a case c and the current request q is expressed by:

$$S = \sum_{i=1}^{K} \omega_i S_i(q_i, c_i) \quad \textit{where} \quad \omega_i \geq 0 \forall i \quad \textit{and} \quad \sum_{i=1}^{K} \omega_i = 1$$

where S is the global similarity, K is the number of parameters of each case, ω_i is the weight given to the parameter i, and S_i is the measure of the global similarity between the parameter i of the request and the one of the considered case.

Local similarity measures largely depend on the application domain, but they all have the same use: to return an estimate between 0 and 1 indicating the similarity between a particular attribute of a case and its equivalent in the request. Knowing the nature of used local similarity measures is not useful for this analysis, because the

approach is general and does not depend nor influences these measures. These measures are considered to be "scientific" references that indicate the absolute similarity between two attributes. We shall see some examples of local similarity measures in the next section.

Here, the ω_i weights represent the relative importance of the concepts. These are the quantities that we want to modify when the user does not criticize the most deficient attribute. Following a critique, the first task of the system consists of computing the deficiency of each attribute. We introduce a mathematical definition of the *deficiency D*:

$$D_i = \omega_i \left(1 - S_i\left(q_i, c_i\right)\right)$$

The reader will notice that the bigger the weight and the weaker the local similarity, the higher the deficiency of the attribute. Once the deficiency of each parameter *i* is computed, we cover one by one each parameter whose deficiency is higher than the criticized parameter, which we call the *critical deficiency* D_*. For all those parameters, we reduce their weight by a value equal to their deficiency:

$$\omega_i \leftarrow \omega_i - (D_i - D_*) \qquad \forall i \qquad D_i > D_*$$

Finally, there had been at least one parameter whose deficiency is higher than the critical deficiency. The weight of the criticized parameter finds itself increased by an amount equivalent to the sum of all reductions subjected by the weights of the parameters that lost importance, such as the sum of all weights stays equal to the unit:

$$\omega_* \leftarrow \omega_* + \sum_i (D_i - D_*) \qquad \forall i \qquad D_i > D_*$$

The proposed heuristic automatically modifies the weights as the user criticizes the products, so as to obtain an increasingly faithful representation of his short-term profile. This, of course, supposes that at each time the client always criticizes the attribute with which he is the least satisfied. In that case, the method offers a way to accelerate the convergence toward a recommendation that can fulfill the needs of the client. Moreover, the adjustment of weights can be useful when the successive critiques of the client always lead him in the same dead-end, that is in an iteration where the system cannot find any product matching the critiques formulated to date. In that case, the client must explicitly reinitiate a request, but he still profits from the adjusted weights according to his requirements, so that we can expect him to find faster than the first time a subset of interesting products.

In the next section, we will instead see an example of an application of the method explained here.

3 Application Example

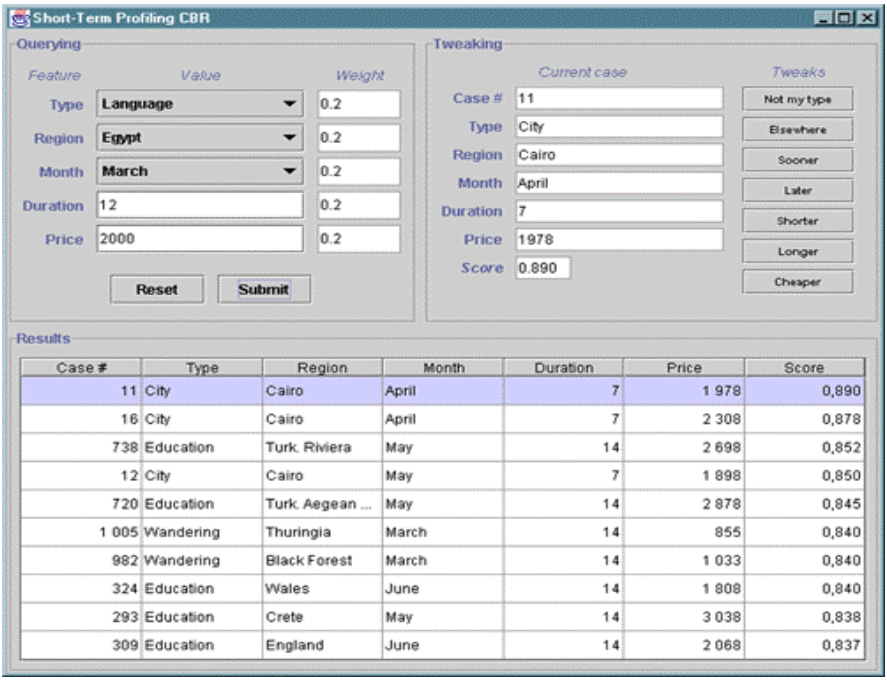
In order to illustrate the method proposed with a concrete example, we have implemented a prototype system for the recommendation of travel packages. For this,

we will use the "travel agency" case base, freely available from the AI-CBR site [5]. The cases contained in that base come from a real application: the "Virtual Travel Agency". The case base consists of 1470 predefined travel packages (i.e., non-configurable), each of them described by about ten attributes such as the type (bathing, skiing, etc.), the price, the duration, the destination and others. For our example, we chose the following five attributes: the type, the region, the month, the duration and the price. For each attribute, we have defined a simple measure of local similarity that produces a value between 0 and 1. The measures for nominal attributes (type, region, month) are tables indicating similarities between all possible combinations. An example can be found in Figure 1.

Query / Case	"bathing"	"city"	"recreation"	"skiing"	...
"bathing"	1.0	0.2	0.5	0.1	...
"city"	0.2	1.0	0.3	0.2	...
"recreation"	0.8	0.3	1.0	0.8	...
"skiing"	0.1	0.2	0.5	1.0	...
...	...	...	...	...	...

Fig. 1. Local similarity measure for the "type" attribute As for numerical quantities such as duration and price, we use standard distance measures (i.e., Euclidean) that we normalize between 0 and 1

Figure 2 illustrates the graphical interface of the application. In the upper-left corner, the user can enter explicit requests used as starting points to the recommendation process. We can specify the desired product characteristics, and we can see at each moment the actual weights. Let us note that in our application, only the automatic weight refinement mechanism can modify those values. In a real application however, it would be interesting to allow the user to override this functionality. The upper-right part of the interface is the location where is displayed the product currently being considered as well as its parameters. It is also from there that the client can formulate its critiques of these parameters, such as "cheaper" for the price, "sooner" or "later" for the month, etc. Finally, the lower part continually displays the ten cases judged the most likely to respond to the client's needs according to the last request (or critique) made. We note that following a request or a critique, the system displays by default as the candidate product the one that obtains the highest score (the highest global similarity). But the client may select and criticize other products in the pool of possibilities if he so desires. That table exists specifically to give the user a broader choice.



Case #	Type	Region	Month	Duration	Price	Score
11	City	Cairo	April	7	1 978	0,890
16	City	Cairo	April	7	2 308	0,878
738	Education	Turk. Riviera	May	14	2 698	0,852
12	City	Cairo	May	7	1 898	0,850
720	Education	Turk. Aegean ...	May	14	2 878	0,845
1 005	Wandering	Thuringia	March	14	855	0,840
982	Wandering	Black Forest	March	14	1 033	0,840
324	Education	Wales	June	14	1 808	0,840
293	Education	Crete	May	14	3 038	0,838
309	Education	England	June	14	2 068	0,837

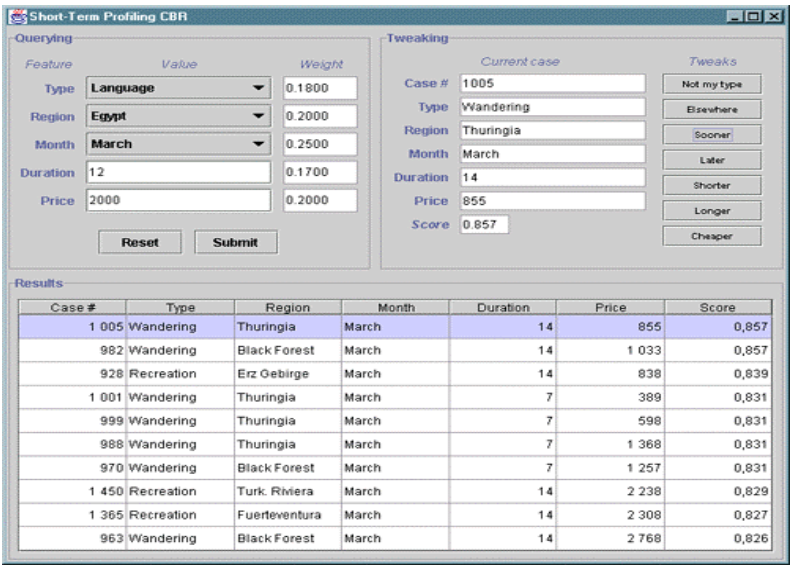
Fig. 2. The application graphical interface

As an example, let us imagine the system is in the state illustrated in Figure 2 and the user wishes to criticize the month: “sooner”. Figure 3. represents the internal state or the system before and after the formulation of this critique.

Attribute	Weight	Local similarity	Deficiency
Type	0.2	0.8	0.04
Region	0.2	1.0	0.0
Month	0.2	0.9	0.02
Duration	0.2	0.75	0.05
Price	0.2	1.0	0.0

Attribute	Weight	Local similarity	Deficiency
Type	0.18	0.8	0.04
Region	0.2	1.0	0.0
Month	0.25	0.9	0.02
Duration	0.17	0.75	0.05
Price	0.2	1.0	0.0

Fig. 3. System state before and after the “sooner” critique



Case #	Type	Region	Month	Duration	Price	Score
1 005	Wandering	Thuringia	March	14	855	0,857
982	Wandering	Black Forest	March	14	1 033	0,857
928	Recreation	Erz Gebirge	March	14	838	0,839
1 001	Wandering	Thuringia	March	7	389	0,831
999	Wandering	Thuringia	March	7	598	0,831
988	Wandering	Thuringia	March	7	1 368	0,831
970	Wandering	Black Forest	March	7	1 257	0,831
1 450	Recreation	Turk. Riviera	March	14	2 238	0,829
1 365	Recreation	Fuerteventura	March	14	2 308	0,827
963	Wandering	Black Forest	March	14	2 768	0,826

Fig. 4. Proposed results after the "sooner" critique

We note that the criticized attribute, the month, was not the most deficient at the time the critique was formulated. Two other attributes had a higher deficiency: the travel type and the duration. Since the user chose to criticize the month, we deduce this parameter is more important to him and we increase its value at the expense of the type and the duration. In Figure 4, we can see what the interface shows after the critique. The results are more in line with the month constraint, not only because the critique had the effect of filtering all months that did not satisfy "sooner", but also due to the increase of the importance of this concept. The pertinence of the other less important parameters, particularly the type, becomes more arbitrary. Curiously, the duration seem more similar than before yet their importance has diminished. This is only a coincidence due to the fact that travels less similar according to the type happen to be more similar according to the duration.

We have made various trials using this application and we could notice that if, at each moment, we always modified the parameter we considered the most unsatisfactory, then the recommendation system allowed us to find an adequate solution faster with our approach than if the weights were fixed. If we base ourselves on the few thirties of examples we tested, we can roughly estimate a performance gain of 20 to 40 percent on the convergence speed toward an acceptable solution. Of course, that is only an estimate and quantifying the advantages of this heuristic more precisely would require in-depth experiments.

4 Conclusion

In this work, we began by raising the problem of the lack of customer support in electronic commerce applications on the Internet. We then proposed a heuristic

allowing CBR-based recommendation systems to take into account short-term preferences of clients. This heuristic is based on the principle that if a client does not criticize the attribute of a product the system considers the most deficient, then the relative importance of the attributes as represented by the system would benefit from being corrected. Using this method, a CBR system does not react exactly the same way from one session to the other. It can now take into account the fact that according to the person or the situation, the relative importance of the different characteristics may vary. The simplicity, the maintenance-free operation and the automatic nature of the profile creation are three of the principal advantages of the proposed approach. Also, the fact that this heuristic can be combined with long-term modeling methods makes it an ideal candidate for hybridization. However, the method has several limits at this point, which is why a more detailed study would be required before its validity can be definitely ascertained. In particular, better mathematical formalization as well as experimental tests are certainly required.

References

1. Aamodt, A. and Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(1), 39-59.
2. Bartsch-Spörl, B., Lenz, M. and Hübner, A. (1999). Case-Based Reasoning – Survey and Future Directions. In *Proceedings of th 5th Biannual German Conference on Knowledge-Based Systems (XPS-99)*, Springer-Verlag.
3. Lenz, M. and Burkhard, H.D. (1996). Case Retrieval Nets : Basic Ideas and Extensions. In *Proceedings of the 4th German Workshop on Case-Based Reasoning (GWCBR-96)*, Berlin.
4. Wilke, W., Lenz, M. and Wess, S. (1998). Intelligent Sales Support with CBR. In Lenz, M. et al. (editors), *Case-Based Reasoning Technology – From Foundations to Applications*, Springer-Verlag.

URL links

5. **AI-CBR** http://www.ai-cbr.org/theindex.html