

FIPS : a Financial Planification System by Case Based Reasoning

From: AAAI Technical Report WS-99-10. Compilation copyright © 1999, AAAI (www.aaai.org). All rights reserved.

Esma Aïmeur and Kamel Boudina

University of Montreal
Departement d'Informatique et de Recherche Operationnelle
C.P. 6128, succursale Centre-ville
Montreal, Canada H3C 3J7

{aimeur,boudina}@iro.umontreal.ca
Tel: (514) 343-6794
Fax: (514) 343-5834

Abstract. Financial analysis is based on complex concepts and rules; its goal is to propose problem-adapted solutions. The evaluation of a particular financial situation has to consider human factors like savers risk tolerance and consumers behavior. It also has to consider political factors like interest rates variations and currency policy. The financial planner has to analyze the client's financial situation to elaborate a financial portfolio adapted to his or her needs. On the grounds of the nature and the diversity of the parameters describing a client's financial profile, we need tools that will memorize and reuse this information in different situations. In order to provide training and evaluation tools for financial analysts, we propose a system called FIPS (Financial Planification System) using Case-Based Reasoning. In FIPS, case-based reasoning is used in the case retrieval process, and also in a reflexive way during the adaptation stage. FIPS proposes to the learner the client's data like financial goals, acceptable risk, income, etc., and expects a balanced financial portfolio suggested by the student. It uses old cases, already treated and memorized, to propose an adapted solution that is compared to the learner's solution. The learner's evaluation is based on the distance between the solution provided by the system and the solution suggested by the learner.

Keywords: Self-training, Evaluation, Case based reasoning system, Reflexive approach, Financial situation.

Introduction

There are different approaches in the training domain. In this paper we are interested in the Case Based Reasoning (CBR) [6] approach in the training and the evaluation of learners. CBR systems attempt to adopt a pragmatic approach, based on the experience elaborated on the solved problems, exactly like a human expert develops experience and becomes subtler in his reasoning.

In the financial analysis domain, the nature of the problems to solve is not adapted to simple application of a set of general rules. In fact, the client's financial situations are very different and specific for each one. So the blind application of rules to treat those situations is not an

appropriate approach. The financial expert has to build a case study for each new client and try to adapt the financial portfolio in order to answer to the client's needs. Because of the differences between the cases, we adopt a CBR approach to develop a software tool called FIPS for the training and the evaluation of financial experts.

The next section presents some tutorial systems that use the CBR approach. After that, we describe our prototype implementation. We also explain the structures defined for the case memory organization, the similarity measure, the indexation and the adaptation. In the third section, we expose our approach for the training and the evaluation process. The last section presents the conclusions of this work, the future perspectives and the possible enhancements of our system.

Case-based tutoring systems

There are several CBR systems that provide training in different knowledge domains [8]. The following case-based tutoring systems are used to help training and evaluating learners [9]. The system DECIDER [4] helps students understand or resolve a pedagogical problem by selecting and presenting appropriate cases from a database that respond to the student's goal. The system HYPO [1] is a case-based tutoring system for law students. The system is used to generate fresh cases for analysis in response to a particular issue of interest as identified by a tutor. We also have other examples of tutorial systems, like GuSS [5] that provide a training of complex social tasks like how to sell products or services. In the next section, we present our prototype structure and the approach adopted to implement it.

Prototype realization

The goal of this work is to develop a system, that proposes to a learner the information describing a client's profile, and to evaluate if the financial portfolio suggested by the student is adapted to that client. The system uses a case

base with an efficient classification of its data. As it adapts past solutions, it builds an adaptation case base. The system's architecture is presented in [2].

Representation

The FIPS system is built on five major modules: indexation, retrieval, adaptation, evaluation and input interface. The different modules are described in the following sections. In the current section, we present the case and solution structures. We also describe, the functional structure of the FIPS system and the memory organization used to represent the case base.

Case structure. In the client's description, we have different information such as annual income, financial goals, fortune, etc. We also have a significant client characteristic, which is his capacity to manage the risk of his investment. After the evaluation of that client's capacity, the financial expert tries to affect a numerical value between 1 and 10 for his risk tolerance.

The case structure and its representation in the system are shown in Figure 1.

Fig. 1. Case Description

Description of the solution components. The system proposes different financial portfolios like solutions for the problems to solve. The combination of the different assets has to take into account the client's goals and the current economical situation. Before the introduction of the rules that will serve to adapt a solution for a particular client, we will introduce some notions linked to the financial analysis. A financial portfolio is composed of three great categories; shares, fixed yield values (bonds, debentures) and specie or quasi-specie (treasury bills) [3]. For each category, we'll see the related factors:

Shares: To evaluate the future share values, we use four great indicator categories. The fundamental indicators which are represented by the companies profits, the technical indicators which are related to the curve of the stock indicators (DowJones, TSE300...), the economical

indicators which are the GNP, retail business, unemployment rate, etc.

Fixed yield values: The fixed yield values are divided in three major categories: the long, short and medium term values. To evaluate the future productivity of the fixed yield values, we must analyze the future tendency of the interest rates.

Specie and quasi-specie: The specie productivity is evaluated depending on the anticipated interest rates at the time of the fixed yield values analysis.

Indexation in FIPS

The case base is represented by a tree. The tree leaves are the pages making up the case base. Each node points to another node or directly to a page. A virtual address assigns a page number to each node. The system evaluates the virtual address for each new case and the hash-code function returns its page number. In the FIPS system, each field describing a case has a weight, which represents its importance. The consequence of the case virtual addressing is the creation of a hierarchy of indexes. The index with the highest level corresponds to the field with the highest weight.

Case retrieval

In the FIPS system, each field describing a case presents a neighborhood expressed in percentage. In other words, for an attribute value v_i , all the values v_j with a distance from v_i lower than a certain level δ_i are in the neighborhood of v_i . The work of Wess and Ritcher [11] and the work of Wess and Globig [10] inspire our approach. The distance between the values v_i and v_j is the absolute value of the arithmetic difference. If the values of the different attributes describing two cases are in the same neighborhood then the cases are members of the same class.

However, it is possible to have two cases with just a subset of their attributes in the same neighborhood. In this situation, we have to take into account the attribute weights. It follows that the distance between a case c and another one q is calculated like shown below:

$$\text{dis}(c,q) = \sum_{a=1,n} w_a * \text{dis}_a(q_a, c_a)$$

w_a : Weight of the attribute a . dis_a : Local distance for the attribute a .

Retrieval case algorithm. The retrieval case algorithm is described as follows:

Read the case q and choose from each field (attribute) a number of bits proportional to the attribute's weight in order to obtain a virtual address
 Determine in the tree, the node N which contains similar cases to the case q
 Read the physical page number p stored in the node N
 Load p from the disk

```

Read the first case c in the page p
Initialize retrieved_case to c and max_similarity to 0
Do while Not (end of page )
Compute the similarity degree between c and q from the
distance between c and q :
  Similarity(c,q) = 1/dis(c,q)
  If similarity(c,q) > max_similarity
  Then max_similarity = similarity(c,q)
  Retrieved_case = c
  Endif
EndDo
Read the solution S corresponding to the case
retrieved_case
Adapt the solution S. (see adaptation algorithm )

```

Case adaptation

In this section, we present the rule and case based adaptation process. The originality in the approach of the system FIPS is the use of case-based reasoning in a reflexive way. In fact, the CBR approach is used to retrieve similar cases and the old adapted solutions. In other words, the system keeps a trace of the solution transformations in order to reuse them [7].

Adaptation case base. The case base in the system FIPS is a set of pairs $\langle C_i, S_i \rangle$, C_i are the cases and S_i are the associated solutions. To treat a new case, the system will find in the base, the most similar case C_i and will adapt the solution S_i using the rules presented in 3.4.3. The transformation of the solution S_i will give us the solution S_i' which is the solution proposed by the system for the new case. Finally, the system will store the pair $\langle S_i, S_i' \rangle$ in the adaptation case base.

Indexation and distance of the adaptation cases. In the adaptation base, the pairs $\langle S_i, S_i' \rangle$, are indexed on the different fields of the solution S_i . There is a field for each type of financial value. However, there is an additional field corresponding to the distance between the case C_i and the new case to solve. Like in the case base, the fields of the elements in the adaptation base have different weighing and we also have a hierarchy of indexes. The same approach is adopted for the field's neighborhood in the adaptation base. Like in the case base, the distance between two cases in the adaptation base is the absolute value of the arithmetical difference.

Adaptation rules in the system FIPS. To elaborate a financial portfolio, we have to take into account two major aspects. The first one is the financial situation and the financial goals for the client, and the second one is the overall economic climate. The adaptation of the retrieved solution will be done in a first step using the rules related to the overall economic climate and in a second step, using the rules related to the distance between the retrieved case and the current one.

Example of economic climate rules

R1: If Increasing interest rates **then** reduce the term of the fixed yield values **endif**

Example of rules related to the client financial situation

R2: If the client goals are the safety and the income **then** reduce the percentage of shares and increase the percentage of fixed yield values **endif**

The rules presented before are used by the system FIPS for the first adaptations. The results of those adaptations are stored in the adaptation base in order to be reused.

Adaptation Algorithm. The adaptation algorithm is the following:

```

Read the case newC and retrieve the nearest case Ck
Read the solution Sk related to the case Ck and search
for an adaptation rule in the rule set RS
If there is a rule corresponding to the difference
between newC and Ck
Then Apply rule to Sk and obtain Sk'
Insert the pair (Sk , Sk ') in the adaptation base (AB)
and return the adapted solution Sk '
Else Retrieve the solution Sk' which is the closest to
the solution Sk in the AB
Take and return the solution Sk '' from the pair (Sk' ,
Sk '')
Endif

```

Training algorithm. The goal of our approach is to develop the learner's capabilities of memorization and solution adaptation. The training algorithm is composed of several steps:

```

(a) Propose a case q from the case base to the learner
(b) Retrieve in the case base the solution S' associated
to the case q
(c) Read the solution S suggested by the learner
(d) Evaluate the distance dis between S and S'
(e) Evaluate and display the similarity degree sim from
the distance dis. (sim = 1/ (dis+1) )
if sim < Acceptable_Rate then Goto (a) Endif

```

Financial portfolio		
	percentage	amount
specie (15%-20%)		
government issues	4	
companies issue	11	
Total	15	
fixed yield values 40%-60%		
short term	13	
medium term	17	
long term	21	
total	51	
shares 20%-40%		
safe values	9	
future values	8	
risk capital	6	
speculative values	34	
total	100	

Fig. 2. Example of solution suggested by a learner

Evaluation algorithm. After several tests are successfully passed, the system proposes an opposite approach when it provides a solution and expects from the learner a profile description. If the last test is successful then the learner evaluation is positive and the training process was well assimilated by the learner.

```

(1) Do while #Step < StepNumber
    Training case (see the training algorithm)
EndDo
(2) Propose a solution S to the learner
(3) Read the case q suggested by the learner
(4) Retrieve in the case base the case q'
(5) Evaluate the distance dis between q and q'
(6) Evaluate the similarity degree (from the distance dis. (sim = 1 / (dis+1))
if sim < Acceptable_Rate then Goto (a) Endif

```

Example

In this section, we see a training and an evaluating example to illustrate what we presented previously. Let C be a case described by the attributes:

$C = \{ \text{Age, civil situation, Salary, risk tolerance, financial goals} \}$

Let S be a solution:

$S = \{ \text{Specie, short term, medium term, long term, safe values, future values, risk capital, speculative values} \}$

Let $Case1$ be a case to propose to a learner (**step(a)** in the training algorithm):

$Case1 = \{ 38, \text{single}, 120000, 60\%, \text{safety} \}$

Let $S1$ be the solution retrieved by the system for the case $Case1$: (**step(b)**)

$S1 = \{ 14\%, 11\%, 19\%, 21\%, 8\%, 9\%, 9\%, 9\% \}$

Let $S2$ be the solution suggested by the learner : (**step(c)**)
 $S2 = \{ 12\%, 13\%, 17\%, 23\%, 7\%, 8\%, 11\%, 9\% \}$

The distance between $S1$ and $S2$ will be evaluated by the system : (**step(d)**)

$$dis(S1, S2) = \sum_{a=1,8} w_a * disa(S1a, S2a) = 12.5$$

$$(2+2+2+2+1+1+2+0)=150$$

The evaluation of the student's solution is the value of the similarity degree between $S1$ and $S2$:

$$sim = 1 / (1 + dis(S1, S2)) = 1 / (1 + 150) = 66\% \text{ (step(e))}.$$

The system proposes n different cases (this number depends on the learner's profile) to the learner and evaluates his solutions as shown for the solution $S2$. If the value of the learner's solutions is greater than an acceptable rate (for example 65%), the system skips to the learner's evaluation level (**step(2)** in the evaluation algorithm), and proposes a solution (for example $S1$) and checks if the case proposed by the learner (**step(3)**) is similar to the case $Case1$ (**steps(4,5,6)**). Depending on the answer provided by the learner, the system evaluates the success of the training process (**step(7)**).

Conclusion

A rule-based system is powerless in front of any non-planned problem. The set of rules guiding an expert system are fixed and offer no evolution whatsoever. In other words, this type of system has no capacity to go beyond its predefined rules and enhance its knowledge of the application domain in which it operates. Contrary to a rule based system, a case based system is capable of learning by rendering the solution available for use in any future problem, thus adding a learning mechanism to the process of problem solving. This technique offers a certain advantage over the rule-based approach.

The degree of local similarity between two values of an attribute figuring in two cases reflects the local distance separating these two values. The similarity between the two cases is translated into a composed (rather than a single) similarity, which takes into account the local attribute's similarity as well as the weight associated to them.

The originality of this work is the recursive use of the CBR approach. In fact, the CBR mechanism is used to retrieve similar cases and the old adapted solutions; words, for each solution adaptation, the system keeps in memory a trace of that operation in order to reuse it in the future. Therefore the adaptation has a mixed approach with the use of a Case-Based Reasoning and a rule based process.

The approach of FIPS for the learner training is based on the weights of the attributes in the description of the solutions. Therefore, we have a good memory organization with a hierarchical structure and a mechanism built on attributes with a strong power of discrimination. The case memory is dynamically modified while the system evolves. The case based adaptation process gives the system the

possibility to learn more on how to adapt its solutions after each new adaptation.

On the other hand, an interesting enhancement for FIPS could be the addition of a base of student's profiles. For a new learner, the system can use a CBR approach to retrieve an old similar student profile in order to know what kind of examples can be used to provide a training adapted to the new student.

FIPS has been tested successfully with a base of one hundred cases and gives good results in financial analysis. We conclude by highlighting the fact that all the modules in FIPS are easily reusable in other application domains, like medical diagnostic and mechanical-failure detection.

References

- Aleven, V., Ashley, K. D., "Automated Generation of examples for a tutorial in Case-Based Argumentation", *Proceedings of The Second International Conference on Intelligent Tutoring Systems (ITS 92)*, 575-584. Edit. By C. Frasson; G. Gauthier; G.L. McCallan. Berlin: Springer-Verlag. Montréal, 1992.
- Boudina, K. "Le raisonnement à base de cas dans la planification financière", Master Thesis in Computer Science, the University of Montreal, 1998
- CCVM "Cours sur le Commerce des Valeurs Mobilières au Canada", *Institut Canadien des Valeurs Mobilières*, 1995.
- Farrel, R., "Intelligent Case Selection and Presentation", *Proceeding of the Tenth International Joint Conference on Artificial Intelligence. IJCAI-87*,1,74-76. Milan, Italy, 1987.
- Kass, A., Burke, R. D., Blevis, E., Williamson, M., "The GuSS project: Integrating instruction and practice through guided social simulation", Northwestern University, Institute for the Learning Sciences Technical Report no.34, 1992.
- Kolodner, J. L., "Case-Based Reasoning". Morgan Kaufmann Publishers, 1993.
- Leake, D. B., Kinley, A., Wilson, D., "Learning to Improve Case Adaptation by Introspective Reasoning and CBR". *Case-Based Reasoning. Experiences, Lessons and Future Directions*. 185-197. Ed. Leake, D. B., AAAI Press/ MIT Press, 1996.
- Shiri, M., Aïmeur, E., Frasson, C. "Student Modelling by Case Based Reasoning" *ITS-98, Fourth International Conference on Intelligent Tutoring Systems*, Lecture Notes in Computer Sciences, no. 1452, B.P. Goettl, H.M. Halff, C.L. Redfield, V.J. Shute Editors, Springer Verlag, San Antonio, Texas, pp. 394-403, August 1998.
- Watson, I., "Applying Case-Based Reasoning: Techniques for Enterprise Systems", Morgan Kaufmann Publishers, San Francisco, California. University of Salford. United kingdom, 1997.
- Wess, S., Globig, C., "Case-Based and Symbolic Classification Algorithms", University of Kaiserslautern, Germany., 1994
- Wess, S., Ritcher, M., "Similarity, Uncertainty and Case-Based Reasoning in PADTEX". University of Kaiserslautern, 1993.