Inferring Case-Based Reasoners' Knowledge to Enhance Interactivity

Pierre-Alexandre Murena[1,2](✉) and Marie Al-Ghossein[1,3]

[1] Helsinki Institute for Information Technology HIIT, Helsinki, Finland
[2] Department of Computer Science, Aalto University, Espoo, Finland
pierre-alexandre.murena@aalto.fi
[3] Department of Computer Science, University of Helsinki, Helsinki, Finland
marie.al-ghossein@helsinki.fi

Abstract. When interacting with a human user, an artificial intelligence needs to have a clear model of the human's behaviour to make the correct decisions, be it recommending items, helping the user in a task or teaching a language. In this paper, we explore the feasibility of modelling the human as a case-based reasoning agent through the question of how to infer the state of a CBR agent from interaction data. We identify the main parameters to be inferred, and propose a Bayesian belief update as a possible way to infer both the parameters of the agent and the content of their case base. We illustrate our ideas with the simple application of an agent learning grammar rules throughout a sequence of observations.

Keywords: User modelling · Machine learning for CBR · Bayesian Inference for CBR

1 Introduction

Many applications strongly rely on the interactivity between a human user and an Artificial Intelligence (AI). In such applications, a human agent performs actions to complete a specific task in cooperation with an AI agent which guides them along the way, either by providing advice, corrections or by intervening directly in the environment [5]. Intelligent Tutoring Systems (ITS) [1] are an example of such applications, where an AI proposes specific learning materials to help a human learner acquire a specific concept.

Despite their differences, all these applications share an important feature: since they involve the collaboration between two agents, the human user and the AI, they require both agents to have a good understanding of their collaborator [4,18]. From the perspective of the AI, this is done in practice by providing the AI agent with a model of the human user. In the case of ITS, such a model could describe what the learner knows [6,19] or how they acquire knowledge [16]. Alternatively, in model-based recommender systems, a user profile is used to represent their tastes and preferences, based on which items will be recommended by the AI agent.

© Springer Nature Switzerland AG 2021
A. A. Sánchez-Ruiz and M. W. Floyd (Eds.): ICCBR 2021, LNAI 12877, pp. 171–185, 2021.
https://doi.org/10.1007/978-3-030-86957-1_12

Case-Based Reasoning (CBR) has been involved in a long tradition of contributions to the field of interactive systems, some of which made in the domain of education [7], by suggesting how an AI could optimally interact with a user. This paper takes a rather different position: We propose a novel interactive framework that models the human user as a CBR agent, having thus principles from CBR dictate how the user acquires and reuses knowledge from previous observations. Using such a user model enables taking into account different effects that go along with the learning experience, such as the memorization, forgetting, and adaptation of previous observations.

Modeling the user as a CBR agent raises various technical challenges, including the question of how to infer the characteristics of the user from their behavior, in particular when these characteristics are not stationary and evolve throughout the interaction. When the case base of the user is known, it does not seem challenging to infer the other characteristics, such as the similarity metric used for retrieval or the parameters of the adaptation [17]. The main difficulty arises when the content of the case base is unknown to the AI agent. In this paper, we propose to alleviate this uncertainty using Bayesian belief update for a joint inference of the content of the case base and of the CBR characteristics. Although this methodology shows good performances, we also discuss that it would be illusory to expect a full inference, since some CBR configurations cannot be distinguished only based on their outcomes.

The remainder of this paper is organized as follows. In Sect. 2, we introduce a formalization of the interactive process using Partially Observable Markov Decision Processes (POMDP). This formalization introduces the user model as a latent variable and highlights the need for an AI agent to infer the parameters of this model. Section 3 discusses how a CBR agent can be used to model the human user. We then identify the parameters of this CBR agent that need to be inferred during the interaction. The inference itself is described in Sect. 4: after presenting the general principle, we develop a simplified case where the CBR is assumed to be deterministic and we discuss the algorithmic implementation of this procedure. These principles are then applied to specific applications, the results of which are presented in Sect. 5. We conclude the paper with a discussion on the perspectives offered by the presented techniques.

2 Problem Statement: Interaction with a CBR Agent

Let $\mathcal{P}$ be a problem space and $\mathcal{S}$ be a solution space. We call a *case* a tuple $(x, y) \in \mathcal{P} \times \mathcal{S}$. The problem of a CBR agent is to infer a plausible solution $y^{tgt} \in \mathcal{S}$ to a problem $x^{tgt} \in \mathcal{P}$.

We consider an agent, the *user*, taking some decisions based on the observation of a sequence of cases. Given a case base CB, the sequential process can be described as follows: the user observes a problem $x_t \in \mathcal{P}$ and takes a decision ω_t in reaction to this problem. We note that this decision ω_t is not necessarily equal to the estimated solution $\hat{y}_t$ to the problem x_t, but is related to it. Then, the user may eventually observe the true solution y_t.

In the context of Intelligent Tutoring Systems, a teacher can aim to teach the human learner a grammar rule by showing a sequence of examples. At each step, the teacher suggests a problem (x_t) and the user suggests a corresponding solution ($\hat{y}_t$). In this example, the teacher observes directly the estimated solution ($\omega_t = y_t$). However, other more sophisticated applications require a strict distinction between ω_t and y_t. For instance, in a medical context [9], a case can be given by the medical observations of a patient (x_t) and the medical diagnosis (y_t). However, based on their diagnosis, a physician will take a decision related to the suitable prescription (ω_t). In an interactive system with an AI assistant, only the prescription would be visible to the AI which should then be able to infer the reasoning process of the physician, for their diagnosis and prescription.

A strong hypothesis made by our work is that the mapping $x_t \mapsto \hat{y}_t$ is computed by the agent based on CBR. This hypothesis will be exploited further in Sect. 4 when estimating the parameters of the decision-making.

Whereas the introduced framework focused only on one agent, the user, its more general setting includes additionally the AI agent that may be responsible for selecting the problems. One way to formalize the decision-making of such an artificial agent in interaction with the human user is offered by the Partially Observable Markov Decision Processes (POMDPs), which have been used in various interaction applications such as teaching [12], dialogues [20] or human-robot interaction [2]. A POMDP is defined as a tuple (S, A, R, T, Ω, O), where S is the set of possible states, A the set of actions (in our context, the cases to present), R a reward function (describing what the AI aims to achieve), $T : S \times A \times S \rightarrow [0, 1]$ the state-transition ($T(s_1, a, s_2)$ measures the probability of transition from state s_1 to state s_2 by playing action a), Ω the set of observations and $O : S \times A \times \Omega \rightarrow [0, 1]$ the observation probability ($O(s, a, \omega)$ measures the probability of observing ω when action a is played in state s).

In our context, the state s corresponds to the description of the parameters of the user, which affect their own decision-making. With this POMDP formalization, the user's decision-making is described by the observation probability function O, which assesses the probability of the user in a state s to take decision ω_t based on the problem x_t selected by the AI.

In the following, we denote by $s^{(t)}$ the user state at time t. This description is given as a vector containing all the parameters necessary for a representation of the user. For a given i, we note $s_i^{(t)}$ to refer to the i-th component of the vector, and $s_{-i}^{(t)}$ to designate the vector of all components s_j for $j \neq i$.

An important challenge when solving POMDPs is that the parameters of the user cannot be directly observed by the AI, and some may evolve during the interactive process (e.g. the content of the case base). These changes in the state are described by the transition probability T. To alleviate this issue, it is important for the system to be able to infer the value of the states in an online manner, while keeping track of the uncertainty. The remainder of this paper will propose a description of how to define the relevant states when the user bases their decision on CBR, and how the parameters can be inferred in practice.

3 Modeling the User as a Case-Based Reasoner

To guarantee an optimal interaction with the user, it is necessary that the definition of the state yields a *forward* model of the user: Given $s^{(t)}$, it must be possible to simulate the user future behavior. In this paper, we assume that the user makes decisions following a CBR. This requires in particular the description of how the user memorizes and reuses previous cases to solve new problems.

Table 1. Summary of parameters to infer by the teacher

Knowledge container	Parameters to infer
Case base	Content of the case base Parameters of case retention Parameters of the forgetting model
Domain knowledge	Background knowledge of domain constraints
Similarity knowledge	Similarity measure Parameters of the similarity measure
Adaptation container	Algorithm used for adaptation Parameters of the algorithm Rules (for a rule-based algorithm)

Using the definition of the knowledge containers for CBR [14], we propose to split the user's model into four components:

(1) *The case base*, denoted by CB^U is the collection of memorized cases. It is updated upon time by adding or removing elements from the collection. During an interaction, it is important to consider how new cases are added to the case base, but also how cases are removed from the case base. In particular, when considering human users, removing a case can be motivated by a conscious desire to update the case base, but also by unconscious phenomena such as forgetting [11]. The AI agent must be able to have an estimation of the content of the case base in order to choose the most appropriate actions. This requires in practice to infer the parameters of the memorization and forgetting phenomenon.

(2) *The domain knowledge* provides a set of rules dictated by the domain and which constrain the search for a solution to the given problems. These rules can be understood as the background knowledge that the user may or may not have. When the teacher is able to identify potential domain knowledge, it needs to infer whether the user does have it. In case the user does not, the AI can adapt its actions to make such rules understandable, or, in practical applications, the AI may provide explicit explanations [3].

(3) *The similarity knowledge* describes the factors used to assess the similarity between cases, and is used in particular when retrieving cases from CB^U to

solve a new problem x_t. For this knowledge container, the inference focuses mostly on the similarity measure used by the learner. In case a finite number of similarity measures can be used, the inference consists in finding out which one is actually preferred by the learner. In more advanced cases, the inference can also focus on parameters of the similarity measure such as the weight coefficients [13].

(4) *The adaptation container* encompasses information that is used for adapting the solution of a retrieved case $(x, y) \in CB^U$ to a new problem x_t. In a rule-based system, the adaptation container contains the rules used to perform the adaptation. The inference of the adaptation container requires then to infer which of these rules are used by the user. In a more general case, the rules are replaced by general parameters and/or algorithms for the adaptation.

A summary of the user model parameters to infer is provided in Table 1. In the context of this paper, we will ignore the parameters related to case retention and case forgetting and the inference techniques proposed in the next section cannot apply directly to them. The inference of these parameters will have to be studied in future works.

4 Inference of the CBR Parameters

4.1 General Principle

A common way to deal with the fact that the states in POMDPs are unobserved, is to evaluate the states using a Bayesian belief update. It can be shown that, in this case, the POMDP is equivalent to a *belief-MDP*. The idea is to estimate the parameters in two steps. First, we estimate the posterior of the state $s^{(t-1)}$ after interaction $t - 1$, using the information obtained at time t:

$$p(s^{(t-1)}|x_t, \omega_t) \propto p(\omega_t|x_t, s^{(t-1)})p(s^{(t-1)}) \tag{1}$$

where $p(s^{(t-1)})$ is the prior over the state. The value taken as a prior for the next interaction is obtained by applying the transition function T of the POMDP:

$$p(s^{(t)}) = \mathbb{E}_{p(s^{(t-1)}|x_t,\omega_t)}\left[T(s^{(t-1)}, (x_t, y_t), s^{(t)})\right] \tag{2}$$

Although this formulation is the soundest, it is difficult to use in practice when inferring the parameters of a CBR agent, because of the very large dimension of the state space, which must contain all possible case bases. It is applicable though when the case-base of the learner is known.

As a solution, we propose in the following to use marginal distributions over each parameter independently instead of the full joint distribution. This simplification, which is used for computational reasons, yields a loss in terms of the richness of potential correlations between parameters of the model. However, depending on the problem, it is possible to consider some groups of variables together to keep track of some correlations.

When considering the marginals, we consider the update of each of the components s_i of the state s. The belief update is then given by:

$$p(s_i^{(t-1)}|x_t, \omega_t) \propto \mathbb{E}_{s_{-i}^{(t-1)}} \left[p(\omega_t|x_t, s_i^{(t-1)}, s_{-i}^{(t-1)})\right] p(s_i^{(t-1)}) \tag{3}$$

where the expected value over the components $s_{-i}^{(t-1)}$ is computed based on the probabilities $p(s_{-i}^{(t-1)})$. For simplicity purposes, we will write the terms $s_i^{(t-1)}, s_{-i}^{(t-1)}$ simply as $s^{(t-1)}$, which is technically correct but loses the intuition that the term $s_i^{(t-1)}$ corresponds to the quantity being updated and $s_{-i}^{(t-1)}$ to the variables of the expected value.

Since the observation is not directly produced by the CBR agent, we decompose the likelihood $p(\omega_t|x_t, s^{(t-1)})$ into two terms, accounting for (i) the result $\hat{y}_t$ of the CBR and (ii) how this result is used to yield observation ω_t:

$$p\left(\omega_t|x_t, s^{(t-1)}\right) = \sum_{\hat{y}} \underbrace{p\left(\omega_t|x_t, s^{(t-1)}, CBR(x_t, s^{(t-1)}) = \hat{y}\right)}_{\text{choice of the response given the result of the CBR}} \underbrace{p\left(CBR(x_t, s^{(t-1)}) = \hat{y}\right)}_{\text{result of the CBR}} \tag{4}$$

where the notation $CBR(x_t, s^{(t-1)})$ designates the result of the CBR for problem x_t with $s^{(t-1)}$ as parameters (including the content of the case base).

4.2 Inference of the Parameters for a Deterministic CBR

We consider as an illustration the specific case where the learner is a deterministic CBR, i.e. that the retrieval and adaptation are both deterministic functions. In addition, we assume that the learner's output ω_t is the result of the adaptation, which implies that:

$$p\left(\omega_t|x_t, s^{(t-1)}, CBR(x_t, s^{(t-1)}) = \hat{y}\right) = \mathbb{I}(\omega_t = \hat{y}) \tag{5}$$

where $\mathbb{I}(x) = 1$ if x is true, and $\mathbb{I}(x) = 0$ otherwise. In this context, it can be shown that $p\left(\omega_t|x_t, s^{(t-1)}\right) = \mathbb{I}\left(\omega_t = CBR(x_t, s^{(t-1)})\right)$, and eventually:

$$p(s_i^{(t-1)}|x_t, \omega_t) \propto p_{s_{-i}^{(t-1)}}\left(\omega_t = CBR(x_t, (s_i^{(t-1)}, s_{-i}^{(t-1)})\right) p\left(s_i^{(t-1)}\right) \tag{6}$$

The CBR process can be divided here into two main steps: the retrieval, denoted by $Ret(x_t)$, which outputs the closest case(s) to x_t, and the adaptation, denoted by $Ad(x_t, \mathcal{R})$, which consists in adapting the retrieved cases $\mathcal{R}$ to solve problem x_t. It can then be observed that:

$$\begin{aligned} & p_{s_{-i}^{(t-1)}}\left(\omega_t = CBR(x_t, (s_i^{(t-1)}, s_{-i}^{(t-1)})\right) \\ & \quad = \sum_{\mathcal{R} \subset CB^U} p_{s_{-i}^{(t-1)}}\left(\omega_t = Ad(x_t, \mathcal{R}) \middle| s_i^{(t-1)}\right) p_{s_{-i}^{(t-1)}}\left(Ret(x_t) = \mathcal{R} \middle| s_i^{(t-1)}\right) \end{aligned} \tag{7}$$

We note that, in Eq. 7, the sum over all possible results of the retrieval will be in practice reduced to those yielding a correct result during the adaptation (otherwise, the probability that $\omega_t = Ad(x_t, \mathcal{R})$ is 0).

4.3 Probability of Retrieval for kNN

In the case where the retrieval is operated by a k Nearest Neighbor algorithm, the probability $p_{s_{-i}^{(t-1)}}(Ret(x_t) = \mathcal{R})$ can be evaluated as follows.

We first consider that the similarity metric and its parameters are fully known. For simplicity and without loss of generality, we assume in the following that the cases are ordered by decreasing similarity to x (in particular, (x_1, y_1) is the most similar to x). In practice, this can be obtained using a permutation σ reordering the cases. Then the probability that $kNN(x_t)$ outputs $(i_1, \ldots, i_k)$, where $i_1 < \ldots < i_k$, is given by:

$$p\big(kNN(x_t) = (i_1, \ldots, i_k)\big) = \prod_{j \in (i_1, \ldots, i_k)} p(\lambda_j^{(t)} = 1) \prod_{\substack{j=1 \\ j \notin (i_1, \ldots, i_k)}}^{i_k} p(\lambda_j^{(t)} = 0) \qquad (8)$$

where $\lambda_i^{(t)} \in \mathbb{B}$ indicate whether case (x_i, y_i) belongs to the user's case base. Note however that the $\lambda_i^{(t)}$ are components of the vector $s^{(t)}$.

When there is uncertainty over the similarity metric and/or its parameters, we can obtain the probability of retrieval by using the law of total probability over these values. Note that the probability in Eq. 7 is computed over the variables $s_{-i}^{(t-1)}$ only, variable $s_i^{(t-1)}$ being fixed and corresponding to the variable being updated. The computational complexity of computing this probability under uncertainty depends on the number of similarity measures and parameters to consider. In particular, this operation requires additional attention in continuous parameter spaces.

4.4 Discussion on the Inference Process

The Bayesian inference described in this section is very general and applicable to any situation where the behaviour of the CBR system can be modelled. In particular, we showed how Eq. 4 can be used to assess situations where the exact output of the reasoning is not observed. In terms of implementation however, it is noticeable that the presented techniques can quickly become computationally very expensive, as soon as the number of parameters of the models increases. The variable separation suggested in Eq. 3 goes into the direction of lowering the dimension of the state space, but this dimension is obviously not the only cause of complexity. For instance, Eq. 7 requires to sum over all possible retrieval results, the number of which grows exponentially with the number of cases to retrieve. In the experimental section, we will consider the simple case of 1 neighbor only, in order to keep reasonable space exploration. For future works however, more advanced inference and approximation techniques will be needed, in particular Monte-Carlo techniques or likelihood-free inference.

5 Application: Teaching Word Inflection

5.1 Presentation of the Application

As an illustrative example, we consider the application of teaching a grammar rule to a learner. In order to teach a new grammatical concept in a foreign language, a commonly used method is to present some examples to the learner, as well as exercises during which the learner aims to solve a series of problems. The purpose of the application considered here is to mimic this teaching procedure.

We focus on the simple case of word inflection, which is the transformation of a word, called *stem*, into an alternative form, called *inflection*. Such transformations are typical of conjugation or declension. In the simplest case, a single rule applies to all stems, but there exist multiple word classes with their specific transformations, and the learner must be able to memorize all these transformations and to know when to apply them. As a typical example, the Institute for the Languages of Finland identifies 51 different declension groups in the Finnish language, which differ mostly in a change of radical. For instance, although the genitive case is obtained by suffixing a -n to the radical, the formation of the radical from the stem varies from one group to the other. We cite here a few examples following the schema (Stem, Radical, Genitive case): ("kissa", "kissa-", "kissan"), ("korpi", "korve-", "korven"), ("rakkaus", "rakkaude-", "rakkauden"), ("Sibelius", "Sibeliukse-", "Sibeliuksen").

When considering this scenario, a case (x, y) is given by the stem (for instance "kissa") and the corresponding genitive form (here "kissan"). The goal of the teacher, as described in Sect. 2, is then to propose an optimal sequence of cases to the learner (which can be seen as exercises).

The learner model we propose is a CBR framework based on the notion of Kolmogorov complexity [8], inspired by the work of Murena et al. (2020) [10] on morphological analogies. Kolmogorov complexity [8] is a theoretical tool measuring how complex the generation of a string is. Intuitively, the character string "0000000000" is less complex than "0110111010" because it can be generated by a simple program. More formally, the complexity of a binary string $x \in \mathbb{B}^*$, denoted $K_M(x)$, is defined as the length of the shortest program, on a reference Turing machine M, that outputs x.

This definition relies on the choice of a reference Turing machine M; theoretical results show that this is not a real issue because of invariance properties, and most applications, including the one of interest here, fix a simple machine to make $K(.)$ computable. For the case of analogies on words, Murena et al. (2020) [10] introduce a simple description language based mostly on the concatenation of character strings. The programs allow the definition of functions with variables, which can be used for instance to assess repetitions of patterns. Although the choice of this language is a parameter *in se*, we consider it as fixed and optimal. Inferring the optimal description language could be an interesting and challenging future direction.

5.2 Implementation of a Case-Based Reasoning Learner

We now propose a full description of the CBR learner we use in the context of this application.

Case Base. The case base CB is a set of tuples (x, y), where x is the stem and y the inflected form. We consider only finite case bases. We consider a probabilistic retention, where a case is retained with a given probability (see Sect. 5.3, Experiment 3). As stated before, we do not consider the inference of the parameters of the case retention.

Domain Knowledge. The domain knowledge is given as a set of rules which determine the validity of a solution and/or affect the adaptation. We consider here the understanding $h \in \mathbb{B}$ of the *vowel harmony* rule in Finnish, which states that the groups of vowels a/o/u and ä/ö/y cannot coexist in a word; according to it, the solution of the analogical equation "maa:maalla::pää:x" will be corrected from "pääll$\underline{\text{a}}$" into "pääll$\underline{\text{ä}}$".

Similarity Knowledge. The retrieval is highly dependent on a distance function between existing cases and a new problem: $d : \mathcal{X} \times \mathcal{Y} \times X$. We identify three main candidate functions, all based on complexity. The first candidate exploits the idea that adaptation knowledge can play a role in the retrieval phase [15]:

$$d_0(a : b, c) = \min_d K(a : b{::}c : d) - K(a : b) \tag{9}$$

where $K(a : b)$ removes the impact of the complexity of the source case. Distance $d_1(a : b, c)$ is similar, but has $K(a)$ as a regularizer. The third considered distance measures how close the structures of a and c are: $d_1(a : b, c) = K(a{::}c) - K(a)$

The retrieval phase is then implemented as a k-nearest neighbors procedure, where the neighbors are defined according to the chosen distance function. The domain knowledge is then given by two parameters: $d \in \{d_0, d_1, d_2\}$, the chosen distance function, and k, the number of neighbors, chosen to be equal to 1 in this paper. The adoption of higher values of k will be explored in future work.

Adaptation Knowledge. The retrieved case $\{(a, b)\}$ is reused for solving the new problem c by solving the analogical equations $a : b{::}c : x$, using the algorithm proposed in [10], which states that the solution x of the analogical equation minimizes the complexity $K(a : b{::}c : x)$. This algorithm is non-parametric.

Discussion. Altogether, these four knowledge containers fully define the learner's CBR model. We notice that the only free parameters considered in this application are the understanding of vowel harmony ($h \in \mathbb{B}$) and the distance function used for the retrieval ($d \in \{d_1, d_2\}$). Other parameters (for instance k the number of neighbors) could be considered in more sophisticated models. In addition to the inference of these parameters, the teacher must also infer the content of the case base.

5.3 Empirical Evaluation

This section presents different experiments for evaluating the process of inference of the CBR parameters, proposed in Sect. 4, in the context of the application of teaching word inflections. We carried out three sets of experiments focusing on different aspects of the inference.

In all of these experiments, the specific task considered is the one of teaching to derive the inessive case of a Finnish word given its nominative case. The list of Finnish words considered is extracted from the one provided by the Institute for Languages in Finland (Kotus)[1]. This list of words also includes characteristics of each word, in particular its group that dictates in part how the radical is formed based on the nominative case. The inessive case was automatically scraped from the Wiktionary dictionary[2]. In the experiments, we considered only words belonging to the 48th type, which contains a large diversity of stem-to-radical transformations.

The main idea of the experiments is to simulate the interaction between a learner, modeled as a CBR agent with fixed parameters s_{true}, and an AI agent trying to infer these parameters, over a number of steps. The true CBR model is used to simulate the user's answers and the evaluation of the parameter inference is based on how close the estimated parameters are from their true values. In addition, we also evaluate the ability of the estimated CBR model to reproduce the true behavior of the user. To measure this ability, we introduce a score metric that is measured at each step t and defined as follows:

$$score^{(t)} = \mathbb{E}_{s^{(t)}} \left[\frac{1}{|CB_{test}|} \sum_{(x,y) \in CB_{test}} \mathbb{I} \left(CBR(x, s^{(t)}) = CBR(x, s_{true}) \right) \right] \tag{10}$$

where CB_{test} is a test case base that is introduced for the sole purpose of evaluating the capacity of reproducing the user behavior on a new set of problems.

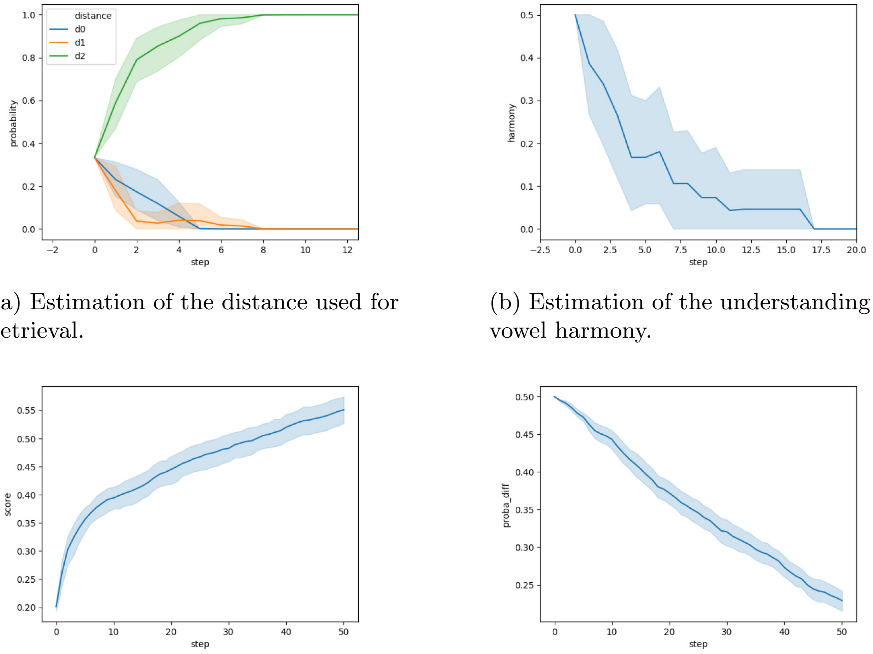
Experiment 1: Parameter Evaluation with a Fixed Case Base. The first set of experiments focuses on the special case of parameter inference when the case base of the user is fixed and does not evolve throughout the interaction. This setting would remove any potential impact of the dynamic character of parameters on the inference process as described by Eq. 2, which would itself be the subject of Experiment 3.

Under this condition, we denote by CB^U the fixed case base of the user that is itself a subset of a larger (also fixed) case base, denoted by CB and containing all possible cases that the user may have observed or learned. We set the size of CB^U to 30 and that of CB to 100. We consider an interaction session of 50 steps, during which the user does not retain any observation but only provides answers to the problems based on its content and parameters. The experiment

[1] www.kotus.fi. The link to the list of Finnish words: kaino.kotus.fi/sanat/nykysuomi.

[2] www.wiktionary.org.

is run 20 times and we sample at each run a different CB^U and CB from the complete list of words described above. The parameters of the true user CBR model are set as follows: $h = 0$ and $d = d_2$. The different priors are taken as uniform distributions over the set of related parameters.



(a) Estimation of the distance used for retrieval.

(b) Estimation of the understanding of vowel harmony.

(c) Evolution of the predicted score compared to the real user.

(d) Average error in the prediction of the word probabilities.

Fig. 1. Results of Experiment 1, considering the case where the user case base is fixed and measuring the quality of parameter estimation.

As an indication in terms of computational time, running such an experiment (including the 20 runs) takes up to one hour on a machine with one processor Intel Core i5 2.3 GHz and 8G of RAM. Such an experiment also includes measuring the score at each step, which is a costly operation, and does not involve advanced optimization or parallelization of execution.

Results. Figure 1 shows the results obtained for this experiment. Figure 1a displays the evolution of the probability of each of the potential distance measures over the number of steps. The estimation of the distance reaches the true value after a few number of steps and the figure only shows the first few steps of the interactive process, after which the values relatively stabilize. Following a similar idea, Fig. 1b shows the estimation of the h parameter related to the understanding of the vowel harmony concept. Its value drops over the number of steps until reaching zero, also showing that it is able to reach the desired value after observing the user solutions. This shows that, even in the absence of a complete certainty over the case base of the user, it remains possible to estimate the parameters of the CBR model.

Figure 1c presents the score metric (Eq. 10) evaluated at each step of the experiment on a fixed CB_{test} of size 100, sampled at the beginning of the experiment from the complete list of words. The score increases over the number of steps, starting from a score of 0 as no proper estimation of the user model has been done at $t = 0$ and the AI agent cannot reproduce the user behavior. The increase of the score metric throughout the experiment shows that the behavior of the estimated user model gets closer to the one of the true user model, which suggests that the estimation quality improves. This idea can also be derived from Fig. 1d where the curve plotting the average difference between the estimated probability of a word from the case base and its true probability, decreases over time. However, and even after a large number of steps, this error does not reach 0: it can be observed that some cases in CB are given a probability of about 0.5. This phenomenon can be explained by an impossibility to discriminate between different words, which are seen by the inference as having a completely similar role, and therefore as completely indiscernible. We discuss further this question of indiscernibility in the next set of experiments.

Experiment 2: Impossibility of Differentiating Indiscernible States. The parameter inference takes as evidence the answers given by the user to a problem set. As mentioned above, it seems that some sets of parameters could exhibit the same behavior (same answers) from the user's side. In this set of experiments, we aim to show that two equivalent states cannot be discernible by the inference process.

We consider the two words “kaura” and “käyrä”, having the inessive case as “kaurassa” and “käyrässä” respectively. We focus on the three following states: $s_1 = (\textit{kaura} \in CB^U, \textit{käyrä} \notin CB^U, h = 1)$, $s_2 = (\textit{kaura} \notin CB^U, \textit{käyrä} \in CB^U, h = 1)$, and $s_3 = (\textit{kaura} \in CB^U, \textit{käyrä} \in CB^U, h = 0)$. Since s_1 and s_2 both incorporate vowel harmony, it can be verified that they hold the same information in terms of how to derive the inessive case from the nominative case, and will therefore provide similar answers to problems.

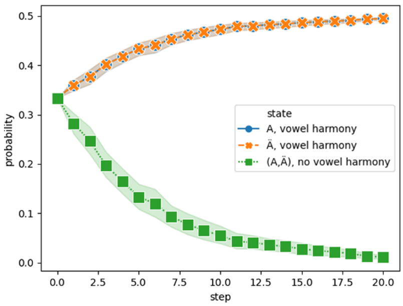
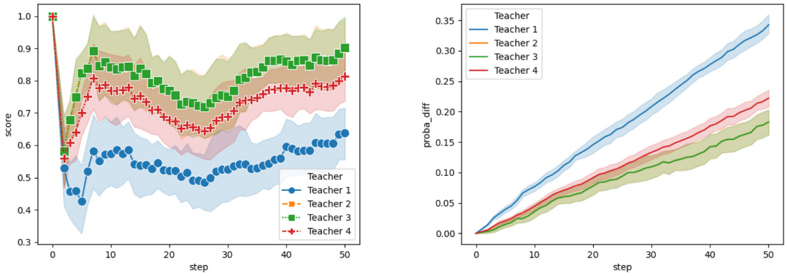


Fig. 2. Results from Experiment 2. The states s_1 and s_2 (blue and orange) yield a similar user's behaviour, and are therefore indiscernible. (Color figure online)

To compare the probability of each of these states given the user answers, we simulate the behavior of a user having a set of parameters equivalent to s_1 on a series of 20 interactions (Note that similar results are obtained with s_2).

Results. Figure 2 shows the evolution of the probability of each state over the number of steps. It can be seen from the plot that the two states s_0 and s_1 have the same probability: It is not possible to differentiate between them or favor one over the other only based on the user answers. The probability of state s_3 decreases over time until reaching 0: if the user were in state s_3, they would retrieve any of the two cases from the case base and would adapt it to form a potentially incorrect answer.



(a) Evolution of the estimated score compared to the real user.

(b) Average error in the prediction of the word probabilities.

Fig. 3. Results of Experiment 3: A learner acquires new data during the interaction with a teacher. The teacher estimates the case base, with the following assumptions on retention: (1) Retain with $p = 0.2$; (2) Retain with $p = 0.6$; (3) Retain with $p = 0.95$; (4) retain with $p = 0.5$ when predicting the correct answer and with $p = 0.8$ otherwise.

Experiment 3: Parameter Inference for a Dynamic Case Base. We complement the results provided in Experiment 1 by showing that an inference is possible even in a context of a sequential interaction. We mimic a teaching interaction between the AI and the user, during which the teacher displays a sequence of problems from a case base CB of size 50. The learner proposes an answer and observes the actual solution. The presented case is then retained with a probability which depends on the learner's answer: $p = 0.8$ when the answer is incorrect, and $p = 0.5$ otherwise. To infer the learner's CBR model, the teacher exploits a fixed transition dynamics. We compare four possible dynamics: three dynamics having a fixed probabilistic retention (with probabilities $p = 0.2$, $p = 0.6$ and $p = 0.95$) and one having the same dynamics as the learner's. The experiments are led in the same conditions as those of Experiment 1 (20 runs, fixed test base of 100 cases for the score).

Results. The experimental results show that the inference of the distance and understanding of vowel harmony is unchanged when using the four transition models. We thus omit to include the corresponding plots. The results presented in Fig. 3 show however that the content of the inferred case base, and consequently the prediction score, are directly affected by the choice of the transition dynamics. In particular, we observe that underestimating the probability of retention (Teacher 1, in blue) causes lower prediction capabilities and more errors on the case base. The teachers with larger probabilities of retention (Teachers 2 and 3, in orange and green) have identical inference of the parameters, which is in particular better than the estimation based on the exact retention model (Teacher 4, in red). These observations highlight the importance of having a good estimation of the transition dynamics. Although this aspect has been ignored in this paper, it is a fundamental and unavoidable future work.

6 Conclusion

When interacting with other agents, be it other artificial agents or human users, an AI must be able to understand its teammate to enhance the quality and efficiency of the cooperation. In this paper, we discussed the possibility to use CBR as a paradigm underlying the other agent's behavior. Such a model is particularly interesting when interacting with human users, since it directly incorporates the fact that humans constantly memorize and reuse knowledge from previous experiences. However, it introduces the important challenge of identifying the parameters of such a CBR model based on the observed behavior.

Our first contribution is to clearly identify the dimensions of interest in a CBR model that would need to be inferred (see Table 1). In particular, we discussed that a major but unavoidable challenge is to infer the content of the case base, i.e. what the user knows. This is challenging because of the number of possible configurations for the case base. A second contribution is to demonstrate the feasibility of such an operation: using basic probabilistic tools, we could propose simple algorithms for the inference of the parameters of a CBR agent. For the application of word declension, we succeeded in inferring the parameters used by a CBR user for both the retrieval and the adaptation, when considering fixed and dynamical case bases. However, we also showed that this has limitations: the inference cannot differentiate between different states that exhibit equivalent behaviours, and all the fixed parameters have to be chosen with care. Future research is needed to be able to infer the parameters of case retention, which none of the methods described in our paper can tackle. Furthermore, more advanced techniques will have to be implemented to enable the inference of more complicated models: in particular, *likelihood-free inference* techniques could be valuable tools for approximating more realistic CBR models of human reasoning.

References

1. Anderson, J.R., Boyle, C.F., Reiser, B.J.: Intelligent tutoring systems. Science **228**(4698), 456–462 (1985)
2. Broz, F., Nourbakhsh, I.R., Simmons, R.G.: Planning for human-robot interaction using time-state aggregated POMDPs. In: AAAI, vol. 8, pp. 1339–1344 (2008)

3. Celikok, M.M., Murena, P.A., Kaski, S.: Teaching to learn: sequential teaching of agents with inner states. arXiv preprint arXiv:2009.06227 (2020)
4. Chakraborti, T., Kambhampati, S., Scheutz, M., Zhang, Y.: AI challenges in human-robot cognitive teaming. arXiv abs/1707.04775 (2017)
5. Dafoe, A., Bachrach, Y., Hadfield, G., Horvitz, E., Larson, K., Graepel, T.: Cooperative AI: machines must learn to find common ground. Nature (593) 33–36 (2021)
6. Elsom-Cook, M.: Student modelling in intelligent tutoring systems. Artif. Intell. Rev. **7**(3–4), 227–240 (1993)
7. Kolodner, J.L., Cox, M.T., González-Calero, P.A.: Case-based reasoning-inspired approaches to education. Knowl. Eng. Rev. **20**(3), 299–304 (2005)
8. Li, M., Vitányi, P., et al.: An Introduction to Kolmogorov Complexity and its Applications, vol. 3. Springer, Heidelberg (2008)
9. Marling, C., Whitehouse, P.: Case-based reasoning in the care of Alzheimer's disease patients. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 702–715. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44593-5_50
10. Murena, P.A., Al-Ghossein, M., Dessalles, J.L., Cornuéjols, A.: Solving analogies on words based on minimal complexity transformations. In: International Joint Conference on Artificial Intelligence, IJCAI (2020)
11. Nioche, A., Murena, P.A., de la Torre-Ortiz, C., Oulasvirta, A.: Improving artificial teachers by considering how people learn and forget. arXiv preprint arXiv:2102.04174 (2021)
12. Rafferty, A.N., Brunskill, E., Griffiths, T.L., Shafto, P.: Faster teaching via POMDP planning. Cogn. Sci. **40**(6), 1290–1332 (2016)
13. Richter, M.M.: Similarity. In: Perner, P. (ed.) Case-Based Reasoning on Images and Signals, pp. 25–90. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-73180-1_2
14. Richter, M.M., Michael, M.: Knowledge Containers. Readings in Case-Based Reasoning. Morgan Kaufmann Publishers (2003)
15. Smyth, B., Keane, M.T.: Using adaptation knowledge to retrieve and adapt design cases. Knowl.-Based Syst. **9**(2), 127–135 (1996)
16. Sottilare, R.A., Graesser, A., Hu, X., Holden, H.: Design Recommendations for Intelligent Tutoring Systems: Volume 1-Learner Modeling, vol. 1. US Army Research Laboratory (2013)
17. Stahl, A.: Learning similarity measures: a formal view based on a generalized CBR model. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 507–521. Springer, Heidelberg (2005). https://doi.org/10.1007/11536406_39
18. Tabrez, A., Luebbers, M.B., Hayes, B.: A survey of mental modeling techniques in human-robot teaming. Curr. Robot. Rep. 1–9 (2020)
19. Woolf, B.P.: Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-learning. Morgan Kaufmann (2010)
20. Young, S., Gašić, M., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: a review. Proc. IEEE **101**(5), 1160–1179 (2013)