Learning Solution Similarity in Preference-based CBR

Amira Abdel-Aziz[1], Marc Strickert[1], and Eyke Hüllermeier[2]

[1] Department of Mathematics and Computer Science
University of Marburg, Germany
{amira,strickert}@mathematik.uni-marburg.de
[2] Department of Computer Science
University of Paderborn, Germany
eyke@upb.de

Abstract. This paper is a continuation of our recent work on preference-based CBR, or Pref-CBR for short. The latter is conceived as a case-based reasoning methodology in which problem solving experience is represented in the form of contextualized preferences, namely preferences for candidate solutions in the context of a target problem to be solved. In our Pref-CBR framework, case-based problem solving is formalized as a preference-guided search process in the space of candidate solutions, which is equipped with a similarity (or, equivalently, a distance) measure. Since the efficacy of Pref-CBR is influenced by the adequacy of this measure, we propose a learning method for adapting solution similarity on the basis of experience gathered by the CBR system in the course of time. More specifically, our method makes use of an underlying probabilistic model and realizes adaptation as Bayesian inference. The effectiveness of this method is illustrated in a case study that deals with the case-based recommendation of red wines.

Keywords: Preferences, distance, similarity learning, probabilistic modeling, Bayesian inference.

1 Introduction

Building on recent research on preference handling in artificial intelligence, we recently started to develop a coherent and generic methodological framework for case-based reasoning (CBR) on the basis of formal concepts and methods for knowledge representation and reasoning with *preferences*, referred to as Pref-CBR [6, 8, 5]. A preference-based approach to CBR appears to be appealing for several reasons, notably because case-based experiences naturally lend themselves to representations in terms of preferences over candidate solutions. Moreover, the flexibility and expressiveness of a preference-based formalism well accommodate the uncertain and approximate nature of case-based problem solving.

Like many other CBR approaches, Pref-CBR proceeds from a formal framework consisting of a problem space $\mathbb{X}$ and a solution space $\mathbb{Y}$. Yet, somewhat less common, it assumes a similarity (or distance) measure to be defined not

only on $\mathbb{X}$ but also on $\mathbb{Y}$. Moreover, it assumes a strong connection between the notions of *preference* and *similarity*. More specifically, for each problem $\boldsymbol{x} \in \mathbb{X}$, it assumes the existence of a theoretically *ideal* solution[3] $\boldsymbol{y}^* \in \mathbb{Y}$, and the less another solution $\boldsymbol{y}$ differs from $\boldsymbol{y}^*$ in the sense of a distance measure Δ_Y, the more this solution is preferred.

As a consequence, the performance and effectiveness of Pref-CBR is strongly influenced by the distance measure Δ_Y: The better this measure captures the true differences between solutions, the more effective Pref-CBR will be. In this paper, we therefore extend our framework through the integration of a *distance learning* module. Thus, the idea is to make use of the experience collected in a problem solving episode, not only to extend the case base through memorization of preferences, but also to adapt the distance measure Δ_Y.

The rest of the paper is organized as follows. By way of background, and to assure a certain level of self-containedness, we recall the essentials of Pref-CBR in the next section. Our approach to learning solution similarity in Pref-CBR is then presented in Section 3 and illustrated by means of two simulation studies in Section 4. The paper ends with some concluding remarks and an outlook on future work in Section 5.

2 Preference-based CBR

Preference-based CBR replaces experiences of the form "solution $\boldsymbol{y}$ (optimally) solves problem $\boldsymbol{x}$", as commonly used in CBR, by weaker information of the form "$\boldsymbol{y}$ is better (more preferred) than $\boldsymbol{z}$ as a solution for $\boldsymbol{x}$", that is, by a preference between two solutions *contextualized* by a problem $\boldsymbol{x}$. More specifically, the basic "chunk of information" we consider is symbolized in the form $\boldsymbol{y} \succeq_{\boldsymbol{x}} \boldsymbol{z}$ and suggests that, for the problem $\boldsymbol{x}$, the solution $\boldsymbol{y}$ is supposedly at least as good as $\boldsymbol{z}$.

As argued in [5], this type of knowledge representation overcomes several problems of more common approaches to CBR. In particular, the representation of experience is less demanding: As soon as two candidate solutions $\boldsymbol{y}$ and $\boldsymbol{z}$ have been tried as solutions for a problem $\boldsymbol{x}$, these two alternatives can be compared and, correspondingly, a strict preference in favor of one of them (or an indifference) can be expressed. To this end, it is neither required that one of these solutions is optimal, nor that their suitability is quantified in terms of a numerical utility.

2.1 A Formal Framework

In the following, we assume the problem space $\mathbb{X}$ to be equipped with a similarity measure $S_X : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$ or, equivalently, with a (reciprocal) distance measure $\Delta_X : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_+$. Thus, for any pair of problems $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X}$, their similarity is

[3] As will be explained in more detail in Section 2, this solution might be fictitious and perhaps impossible to be materialized.

denoted by $S_X(\boldsymbol{x}, \boldsymbol{x}')$ and their distance by $\Delta_X(\boldsymbol{x}, \boldsymbol{x}')$. Likewise, we assume the solution space $\mathbb{Y}$ to be equipped with a similarity measure S_Y or, equivalently, with a (reciprocal) distance measure Δ_Y. In general, $\Delta_Y(\boldsymbol{y}, \boldsymbol{y}')$ can be thought of as a kind of adaptation cost, i.e., the (minimum) cost that needs to be invested to transform the solution $\boldsymbol{y}$ into $\boldsymbol{y}'$. As will become clear later on, our framework suggests a natural connection between distance and similarity, which involves a parameter $\beta \geq 0$ and is of the following form:

$$S_Y(\boldsymbol{y}, \boldsymbol{y}') = \exp\big(-\beta \cdot \Delta_Y(\boldsymbol{y}, \boldsymbol{y}')\big) \in (0, 1] \tag{1}$$

In preference-based CBR, problems $\boldsymbol{x} \in \mathbb{X}$ are not associated with single solutions but rather with preferences over solutions, that is, with elements from a class of preference structures $\mathfrak{P}(\mathbb{Y})$ over the solution space $\mathbb{Y}$. Here, we make the assumption that $\mathfrak{P}(\mathbb{Y})$ is given by the class of all weak order relations $\succeq$ on $\mathbb{Y}$, and we denote the relation associated with a problem $\boldsymbol{x}$ by $\succeq_{\boldsymbol{x}}$. More precisely, we assume that $\succeq_{\boldsymbol{x}}$ has a specific form, which is defined by an *ideal* solution $\boldsymbol{y}^* = \boldsymbol{y}^*(\boldsymbol{x}) \in \mathbb{Y}$ and the distance measure Δ_Y: the closer a solution $\boldsymbol{y}$ to $\boldsymbol{y}^*$, the more it is preferred. Thus, $\boldsymbol{y} \succeq_{\boldsymbol{x}} \boldsymbol{z}$ iff $\Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*) \leq \Delta_Y(\boldsymbol{z}, \boldsymbol{y}^*)$, for which we shall also use the notation $[\boldsymbol{y} \succeq \boldsymbol{z} \,|\, \boldsymbol{y}^*]$. In conjunction with the regularity assumption that is commonly made in CBR, namely that similar problems tend to have similar (ideal) solutions, this property legitimates a preference-based version of this assumption: *Similar problems are likely to induce similar preferences over solutions.*

We like to mention that the solution $\boldsymbol{y}^*$ could be purely fictitious and does not necessarily exist in practice. For example, anticipating one of our case studies in Section 4 as an illustration, a solution could be a red wine characterized by a certain number of chemical properties. One could then imagine that, for a specific person, there is an ideal (most preferred) wine in a space of all theoretically conceivable (i.e., chemically feasible) red wines, even if this wine is not a practically possible solution, either because it is not produced or not offered by the wine seller—finding another wine that closely resembles it is then the best one can do. More formally, this means that we assume a set of theoretically conceivable solutions $\mathbb{Y}^*$ that contains the space of practically reachable solutions $\mathbb{Y}$ as a subset.

2.2 Case-based Inference

The key idea of preference-based CBR is to exploit experience in the form of previously observed preferences, deemed relevant for the problem at hand, in order to support the current problem solving episode; like in standard CBR, the *relevance* of a preference will typically be decided on the basis of problem similarity, i.e., those preferences will be deemed relevant that pertain to similar problems. An important question that needs to be answered in this connection is the following: Given a set of observed preferences on solutions, considered representative for a problem $\boldsymbol{x}_0$, what is the underlying preference structure $\succeq_{\boldsymbol{x}_0}$ or, equivalently, what is most likely the ideal solution $\boldsymbol{y}^*$ for $\boldsymbol{x}_0$?

We approach this problem from a statistical perspective, considering the true preference model $\succeq_{\boldsymbol{x}_0} \in \mathfrak{P}(\mathbb{Y})$ associated with the query $\boldsymbol{x}_0$ as a random variable with distribution $\mathbf{P}(\cdot \,|\, \boldsymbol{x}_0)$, where $\mathbf{P}(\cdot \,|\, \boldsymbol{x}_0)$ is a distribution $\mathbf{P}_\theta(\cdot)$ parametrized by $\theta = \theta(\boldsymbol{x}_0) \in \Theta$. The problem is then to estimate this distribution or, equivalently, the parameter θ on the basis of the information available. This information consists of a set

$$\mathcal{D} = \left\{ \boldsymbol{y}^{(i)} \succ \boldsymbol{z}^{(i)} \right\}_{i=1}^{N} \tag{2}$$

of preferences of the form $\boldsymbol{y} \succ \boldsymbol{z}$ between solutions.

The basic assumption underlying nearest neighbor estimation is that the conditional probability distribution of the output given the input is (approximately) locally constant, that is, $\mathbf{P}(\cdot \,|\, \boldsymbol{x}_0) \approx \mathbf{P}(\cdot \,|\, \boldsymbol{x})$ for $\boldsymbol{x}$ close to $\boldsymbol{x}_0$. Thus, if the preferences (2) are coming from problems $\boldsymbol{x}$ similar to $\boldsymbol{x}_0$ (namely from the nearest neighbors of $\boldsymbol{x}_0$ in the case base), then this assumption justifies considering $\mathcal{D}$ as a representative sample of $\mathbf{P}_\theta(\cdot)$ and, hence, estimating θ via maximum likelihood (ML) inference by

$$\theta^{ML} \;=\; \operatorname*{argmax}_{\theta \in \Theta} \mathbf{P}_\theta(\mathcal{D}) \;\; . \tag{3}$$

An important prerequisite for putting this approach into practice is a suitable data generating process, i.e., a process generating preferences in a stochastic way.

Our data generating process is based on the idea of a discrete choice model as used in choice and decision theory [11]. Recall that the (absolute) preference for a solution $\boldsymbol{y} \in \mathbb{Y}$ supposedly depends on its distance $\Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*) \geq 0$ to an ideal solution $\boldsymbol{y}^*$, where $\Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*)$ can be seen as a "degree of suboptimality" of $\boldsymbol{y}$. As explained in [10], more specific assumptions on an underlying (latent) utility function on solutions justify the following model of discrete choice, which can be seen as a generalization of the well-known Bradley-Terry model:

$$\mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z}) = \mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^*) \;=\; \frac{S_Y(\boldsymbol{y}, \boldsymbol{y}^*)}{S_Y(\boldsymbol{y}, \boldsymbol{y}^*) + S_Y(\boldsymbol{z}, \boldsymbol{y}^*)} \;\; , \tag{4}$$

where $S_Y(\boldsymbol{y}, \boldsymbol{y}^*)$ is defined as

$$S_Y(\boldsymbol{y}, \boldsymbol{y}^*) \;=\; \exp\big(-\beta \cdot \Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*) \big)$$

according to (1). This similarity can be seen as the degree to which $\boldsymbol{y}$ resembles the ideal solution $\boldsymbol{y}^*$; likewise, $S_Y(\boldsymbol{z}, \boldsymbol{y}^*)$ is the degree to which $\boldsymbol{z}$ is close to ideal. Thus, the probability of observing the (revealed) preference $\boldsymbol{y} \succ \boldsymbol{z}$ depends on the degree of optimality of $\boldsymbol{y}$ and $\boldsymbol{z}$, namely their respective closeness to the ideal solution: The less optimal $\boldsymbol{z}$ in comparison to $\boldsymbol{y}$, the larger the probability to observe $\boldsymbol{y} \succ \boldsymbol{z}$; if $\Delta_Y(\boldsymbol{z}, \boldsymbol{y}^*) = \Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*)$, then $\mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z}) = 1/2$.

The coefficient β can be seen as a measure of precision of the preference feedback. For large β, $\mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z})$ converges to 0 if $\Delta_Y(\boldsymbol{z}, \boldsymbol{y}^*) < \Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*)$ and to 1 if $\Delta_Y(\boldsymbol{z}, \boldsymbol{y}^*) > \Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*)$; this corresponds to a deterministic (error-free) information source. The other extreme case, namely $\beta = 0$, models a completely unreliable source reporting preferences at random.

The probabilistic model outlined above is specified by two parameters: the ideal solution $\boldsymbol{y}^*$ and the (true) precision parameter $\beta^* \in \mathbb{R}_+$. Depending on the context in which these parameters are sought, the ideal solution might be unrestricted (i.e., any element of $\mathbb{Y}$ is an eligible candidate), or it might be restricted to a certain subset $\mathbb{Y}_0 \subseteq \mathbb{Y}$ of candidates.

As mentioned before, the parameter vector $\theta^* = (\boldsymbol{y}^*, \beta^*) \in \mathbb{Y}_0 \times \mathbb{R}^*$ can be estimated from a given set (2) of observed preferences using the maximum likelihood principle. Assuming independence of the preferences, the likelihood of $\theta = (\boldsymbol{y}, \beta)$ is given by

$$\ell(\theta) = \prod_{i=1}^{N} \mathbf{P}\left(\boldsymbol{y}^{(i)} \succ \boldsymbol{z}^{(i)} \,|\, \theta\right) \tag{5}$$

The ML estimation $\theta_{ML} = (\boldsymbol{y}^{ML}, \beta^{ML})$ of θ^* is given by the maximizer of (5):

$$\theta_{ML} = \left(\boldsymbol{y}^{ML}, \beta^{ML}\right) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathbb{Y}_0,\, \beta \in \mathbb{R}_+} \ell(\boldsymbol{y}, \beta) \tag{6}$$

The problem of finding this estimation in an efficient way is discussed in [10].

2.3 CBR as Preference-guided Search

Case-based inference as outlined above realizes a "one-shot prediction" of a promising solution for a query problem, given preferences in the context of similar problems encountered in the past. In a case-based problem solving process, this prediction may thus serve as an initial solution, which is then adapted step by step. An adaptation process of that kind can be formalized as a search process, namely a traversal of a suitable space of candidate solutions that is guided by preference information collected in previous problem solving episodes.

To this end, we assume the solution space $\mathbb{Y}$ to be equipped with a topology that is defined through a *neighborhood structure*: For each $\boldsymbol{y} \in \mathbb{Y}$, we denote by $\mathcal{N}(\boldsymbol{y}) \subseteq \mathbb{Y}$ the neighborhood of this candidate solution. The neighborhood is thought of as those solutions that can be produced through a single modification of $\boldsymbol{y}$, i.e., by applying one of the available adaptation operators to $\boldsymbol{y}$.

Our case base **CB** stores problems $\boldsymbol{x}_i$ together with a set of preferences $\mathcal{P}(\boldsymbol{x}_i)$ that have been observed for these problems. Thus, each $\mathcal{P}(\boldsymbol{x}_i)$ is a set of preferences of the form $\boldsymbol{y} \succ_{\boldsymbol{x}_i} \boldsymbol{z}$, which are collected while searching for a good solution to $\boldsymbol{x}_i$.

We conceive preference-based CBR as an iterative process in which problems are solved one by one; our current implementation of this process is described in pseudo-code in Algorithm 1. In each problem solving episode, a good solution for a new query problem is sought, and new experiences in the form of preferences are collected. In what follows, we give a high-level description of a single problem solving episode:

(i) Given a new query problem $\boldsymbol{x}_0$, the K nearest neighbors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ of this problem (i.e., those with smallest distance in the sense of Δ_X) are retrieved from the case base **CB**, together with their preference information $\mathcal{P}(\boldsymbol{x}_1), \ldots, \mathcal{P}(\boldsymbol{x}_K)$.

(ii) This information is collected in a single set of preferences $\mathcal{P}$, which is considered representative for the problem $\boldsymbol{x}_0$ and used to guide the search process.
(iii) The search for a solution starts with an initial candidate $\boldsymbol{y}^\bullet \in \mathbb{Y}$, namely the "one-shot prediction" (6) based on $\mathcal{P}$, and iterates L times. Restricting the number of iterations by an upper bound L reflects our assumption that an evaluation of a candidate solution is costly.
(iv) In each iteration, a new candidate $\boldsymbol{y}^{query}$ is determined, again based on (6), and given as a query to an underlying oracle. The oracle is a (possibly imperfect) information source that compares $\boldsymbol{y}^{query}$ with the current best solution $\boldsymbol{y}^\bullet$. The preference reported by the oracle is memorized by adding it to the preference set $\mathcal{P}_0 = \mathcal{P}(\boldsymbol{x}_0)$ associated with $\boldsymbol{x}_0$, as well as to the set $\mathcal{P}$ of preferences used for guiding the search process. Moreover, the better solution is retained as the current best candidate.
(v) When the search stops, the current best solution $\boldsymbol{y}^\bullet$ is returned, and the case $(\boldsymbol{x}_0, \mathcal{P}_0)$ is added to the case base.

The preference-based guidance of the search process is realized in (iii) and (iv). Here, our case-based inference method is used to find the most promising candidate among the neighborhood of the current solution $\boldsymbol{y}^\bullet$, based on the preferences collected in the problem solving episode so far. By providing information about which of these candidates will most likely constitute a good solution for $\boldsymbol{x}_0$, it (hopefully) points the search into the most promising direction.

3 Distance Learning in Pref-CBR

This section is devoted to the main extension of our Pref-CBR framework, namely the distance adaptation component in line 24 of Algorithm 1. In our framework, we assume preference information to be produced according to the probabilistic model (4). Therefore, it is natural to approach the distance learning problem from a probabilistic point of view. Correspondingly, we shall propose a Bayesian method to tackle this problem.

3.1 A Local-Global Representation of Distance

We begin with a simplifying assumption on the structure of the distance measure Δ_Y, namely that it adheres to the *local-global principle* [2] and takes the form

$$\Delta_Y(\boldsymbol{y}, \boldsymbol{y}^*) = \sum_{i=1}^{k} \alpha_i \cdot \Delta_i(\boldsymbol{y}, \boldsymbol{y}^*) \ , \tag{7}$$

where $\Delta_1, \ldots, \Delta_k$ are local distances pertaining to different properties of solutions, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ is a partition of unity (i.e., the coefficients α_i are non-negative and sum up to 1). We assume the Δ_i to be known, whereas the α_i, which are modeling the importance of the local distances, are supposed to be unknown. Learning the distance measure (7) is thus equivalent to learning these parameters.

Algorithm 1 Pref-CBR Search(K, L, J)

Require: K = number of nearest neighbors collected in the case base
L = number of queries to the oracle
J = number of preferences used to guide the search process

$\mathbb{X}_0 \leftarrow$ list of problems to be solved ▷ a subset of $\boldsymbol{X}$
$Q \leftarrow [\cdot]$ ▷ empty list of performance degrees
CB $\leftarrow \emptyset$ ▷ initialize empty case base
while $\mathbb{X}_0$ not empty **do**
  $\boldsymbol{x}_0 \leftarrow$ pop first element from $\mathbb{X}_0$ ▷ new problem to be solved
  $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\} \leftarrow$ nearest neighbors of $\boldsymbol{x}_0$ in **CB** (according to Δ_X)
  $\{\mathcal{P}(\boldsymbol{x}_1), \ldots, \mathcal{P}(\boldsymbol{x}_K)\} \leftarrow$ preferences associated with nearest neighbors
  $\mathcal{P} \leftarrow \mathcal{P}(\boldsymbol{x}_1) \cup \mathcal{P}(\boldsymbol{x}_2) \cup \ldots \cup \mathcal{P}(\boldsymbol{x}_k)$ ▷ combine neighbor preferences
  $\boldsymbol{y}^\bullet \leftarrow \mathrm{CBI}(\mathcal{P}, \mathbb{Y})$ ▷ select an initial candidate solution
  $\mathbb{Y}^{vis} \leftarrow \{\boldsymbol{y}^\bullet\}$ ▷ candidates already visited
  $\mathcal{P}_0 \leftarrow \emptyset$ ▷ initialize new preferences
  for $i = 1$ to L **do**
    $\mathcal{P}^{nn} = \{\boldsymbol{y}^{(j)} \succ \boldsymbol{z}^{(j)}\}_{j=1}^{J} \leftarrow J$ preferences in $\mathcal{P} \cup \mathcal{P}_0$ closest to $\boldsymbol{y}^\bullet$
    $\mathbb{Y}^{nn} \leftarrow$ neighborhood $\mathcal{N}(\boldsymbol{y}^\bullet)$ of $\boldsymbol{y}^\bullet$ in $\mathbb{Y} \setminus \mathbb{Y}^{vis}$
    $\boldsymbol{y}^{query} \leftarrow \mathrm{CBI}(\mathcal{P}^{nn}, \mathbb{Y}^{nn})$ ▷ find next candidate
    $[\boldsymbol{y} \succ \boldsymbol{z}] \leftarrow \mathrm{Oracle}(\boldsymbol{x}_0, \boldsymbol{y}^{query}, \boldsymbol{y}^\bullet)$ ▷ check if new candidate is better
    $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \{\boldsymbol{y} \succ \boldsymbol{z}\}$ ▷ memorize preference
    $\boldsymbol{y}^\bullet \leftarrow \boldsymbol{y}$ ▷ adopt the current best solution
    $\mathbb{Y}^{vis} \leftarrow \mathbb{Y}^{vis} \cup \{\boldsymbol{y}^{query}\}$
  end for
  $q \leftarrow$ performance of solution $\boldsymbol{y}^\bullet$ for problem $\boldsymbol{x}_0$
  $Q \leftarrow [Q, q]$ ▷ store the performance
  CB $\leftarrow$ **CB** $\cup \{(\boldsymbol{x}_0, \mathcal{P}_0)\}$ ▷ memorize new experience
  Adapt distance measure Δ_Y
end while
return list Q of performance degrees

3.2 Bayesian Distance Learning

Adopting the above representation of the distance measure Δ_Y, our choice model (4) is now given by

$$\mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z}) \;=\; \frac{S_Y(\boldsymbol{y}, \boldsymbol{y}^*)}{S_Y(\boldsymbol{y}, \boldsymbol{y}^*) + S_Y(\boldsymbol{z}, \boldsymbol{y}^*)} \tag{8}$$

with

$$S_Y(\boldsymbol{y}, \boldsymbol{y}^*) = \exp\left(-\sum_{i=1}^{k} \gamma_i \cdot \Delta_i(\boldsymbol{y}, \boldsymbol{y}^*)\right) \tag{9}$$

and $\gamma_i = \beta \cdot \alpha_i \geq 0$. Thus, learning $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)$ means learning β and $\boldsymbol{\alpha}$ simultaneously. In fact, these parameters can be recovered from $\boldsymbol{\gamma}$ as follows:

$$\begin{aligned} \beta &= \gamma_1 + \gamma_2 + \ldots + \gamma_k \\ \alpha_i &= \gamma_i / \beta \end{aligned}$$

For simplicity, suppose that $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)$ only assumes values in a finite (or at least countable) set $\Gamma \subset \mathbb{R}_+^k$; this allows us to work with probability distributions instead of density functions. For example, Γ could be a suitable discretization of a continuous domain, such as a grid on a hypercube.

Since the true $\boldsymbol{\gamma}$ (used by the oracle) is assumed to be unknown, we model our belief about the parameters γ_i in (9) in the form of a probability distribution

$$\mathbf{P} : \Gamma \to [0, 1] \;\;,$$

i.e., for each vector $\boldsymbol{\gamma} \in \Gamma$, $\mathbf{P}(\boldsymbol{\gamma})$ denotes the prior probability of that vector. Unless specific (prior) knowledge is available, this probability can be initialized by the uniform distribution over Γ.

Now, suppose a preference $p = [\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^*]$ to be revealed by the oracle. Since the oracle is supposed to generate preferences according to (8), this observation provides a hint at the true value of $\boldsymbol{\gamma}$. More specifically, it can be used for performing a Bayesian inference step to update our belief about $\boldsymbol{\gamma}$:

$$\mathbf{P}(\boldsymbol{\gamma} \,|\, p) = \frac{\mathbf{P}(p \,|\, \boldsymbol{\gamma})\mathbf{P}(\boldsymbol{\gamma})}{\mathbf{P}(p)} \;\;, \tag{10}$$

where $\mathbf{P}(p \,|\, \boldsymbol{\gamma})$ is given by (8). Concretely, this means realizing the following update for each $\boldsymbol{\gamma} \in \Gamma$:

$$\mathbf{P}(\boldsymbol{\gamma}) \;\leftarrow\; \frac{1}{C} \cdot \mathbf{P}(\boldsymbol{\gamma}) \cdot \mathbf{P}(p \,|\, \boldsymbol{\gamma}) \;\;, \tag{11}$$

where $\mathbf{P}(p \,|\, \boldsymbol{\gamma}) = \mathbf{P}(\boldsymbol{y} \succ \boldsymbol{z})$ is given by (8) and C is a normalizing constant assuring that the (posterior) probability degrees sum up to 1. To the best of our knowledge, there is no parameterized family of distributions that is conjugate with (8), so that the posterior (10) needs to be computed numerically.

3.3 Integration with Pref-CBR Search

As mentioned before, the adaptation of Δ_Y as outlined above is integrated in our Pref-CBR search procedure in line 24 of Algorithm 1. Thus, the idea is to update the belief about $\boldsymbol{\gamma}$ (and hence about Δ_Y, which is uniquely determined by this parameter) after each problem solving episode, making use of the newly observed preferences. Here is a summary of the main steps:

- Suppose our current belief about $\boldsymbol{\gamma}$ to be specified in the form of a probability $\mathbf{P}$ on Γ; in the beginning, this could be the uniform distribution, for example.
- In a single problem solving episode (lines 5–23 of Algorithm 1), Pref-CBR Search is used to solve a new problem $\boldsymbol{x}_0$. This requires a concrete distance Δ_Y, and therefore a concrete parameter vector $\boldsymbol{\gamma}$, which is used to "mimic" the (ground-truth) similarity measure of the oracle. To this end, we can reasonably choose the expectation according to our current distribution, which is considered as our current "best guess" of the true vector:[4]

$$\widehat{\boldsymbol{\gamma}} = \frac{1}{|\Gamma|} \sum_{\boldsymbol{\gamma} \in \Gamma} \boldsymbol{\gamma} \cdot \mathbf{P}(\boldsymbol{\gamma}) \tag{12}$$

  Using the distance measure (7) and choice model (8) parameterized by $\widehat{\boldsymbol{\gamma}}$ or, more specifically, the induced parameters

$$\widehat{\beta} = \widehat{\gamma}_1 + \ldots + \widehat{\gamma}_k \ , \tag{13}$$

$$\widehat{\alpha}_i = \widehat{\gamma}_i / \widehat{\beta} \ , \tag{14}$$

  the Pref-CBR search procedure is performed as usual.
- Upon termination of a problem solving episode (line 20 of Algorithm 1), Pref-CBR Search yields a solution $\boldsymbol{y}^\bullet$, which is not necessarily the truly ideal solution $\boldsymbol{y}^*$ but at least an approximation thereof. Moreover, Pref-CBR Search returns a set of preferences $\mathcal{P}_0$ that have been collected during the search process. These preferences can now be used for updating our belief about $\boldsymbol{\gamma}$.[5] To this end, the adaptation (10) is carried out for each of the preferences in $\mathcal{P}_0$. More specifically, for each preference $\boldsymbol{y} \succ \boldsymbol{z}$ observed in the last episode, a learning step is carried out with $[\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^\bullet]$.

It is important to note that contextualizing an observed preference $\boldsymbol{y} \succ \boldsymbol{z}$ by $\boldsymbol{y}^\bullet$ instead of $\boldsymbol{y}^*$ makes our distance learning method *approximate* and may possibly affect its efficacy. In fact, since preferences of the form $\widehat{p} = [\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^\bullet]$ can be seen as "noisy" versions of the true preferences $p = [\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^*]$, our method is actually learning from noisy data. We shall return to this issue in the experimental section further below.

[4] An alternative would be to choose the mode of the distribution instead of the mean.

[5] In principle, this set of preferences can be enriched further, assuming that each solution adopted in a later stage of the search process is preferred to each earlier solution.

3.4 Related Work

The learning and adaptation of similarity or distance measures has been studied intensively in the literature, not only in CBR but also in related fields like machine learning. Yet, our approach has a number of properties that distinguish it from most others: similarity is learned in the solution space, not in the problem space; training information is purely qualitative and based on paired comparisons; learning is done within the framework of Bayesian inference, making use of a probabilistic model.

Similarity learning in CBR has almost exclusively focused on learning similarity in the problem space. This is also true for the work of Stahl [12, 16, 13, 15], which nevertheless shares a number of commonalities with our approach. In particular, he also considers the learning of weights in a linear combination of local similarity functions [12, 14], albeit based on different types of training information and using other learning techniques. Our own previous work [3] is related, too, as it learns from qualitative feedback in the form of preferences. Essentially, this is what Stahl in [13] refers to as *relative case utility feedback*.

Appropriate metrics are also essential for the performance of distance-based methods such as nearest neighbor estimation, which are used for classification, regression, and related problems. Metric learning has therefore been studied quite intensively in machine learning and pattern recognition. While Mahalanobis distance metric learning has received specific attention in this regard, more involved problems such as nonlinear metric learning, local metric learning, semi-supervised metric learning, and metric learning for structured data have been tackled more recently. We refer to [1] for a comprehensive and up-to-date survey of the metric learning literature.

A Bayesian approach to distance metric learning has been proposed in [17]. Here, the authors estimates a posterior distribution for the distance metric from labeled pairwise constraints, namely equivalence constraints (pairs of similar objects) and inequivalence constraints (pairs of dissimilar objects). Worth mentioning is also the Bayesian approach to preference elicitation by [9]. Although it is concerned with utility instead of distance learning, the authors proceed from training information in the form of paired comparisons, and assume preferences to be generated by the Bradley-Terry model. Their model is still a bit simpler than our model (4) and permits the derivation of closed-form Bayesian updates (using a suitable family of conjugate priors).

4 Experiments

4.1 Synthetic Data

To illustrate our Bayesian approach to distance learning, independently of its use within the Pref-CBR framework, we conducted some very simple experiments for the case $\mathbb{Y} = [0,1]^2$, $\Delta_1(\boldsymbol{y}, \boldsymbol{y}') = |y_1 - y_1'|$, $\Delta_2(\boldsymbol{y}, \boldsymbol{y}') = |y_2 - y_2'|$, and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$. For simplicity, we also assumed β to be known and only learned $\boldsymbol{\alpha}$.

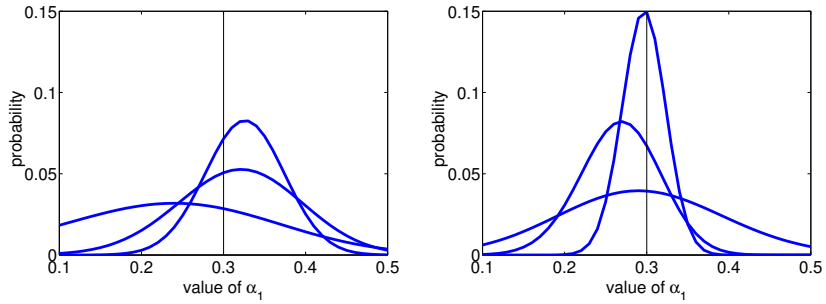

Fig. 1. Probability distributions for the parameter α_1 after $N = 50$, $N = 200$ and $N = 500$ examples, with $\beta = 5$ (left) and $\beta = 10$ (right).

To this end, we generated triplets $(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{y}^*) \subset \mathbb{Y}$ uniformly at random and derived exemplary preferences $[\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^*]$ or $[\boldsymbol{z} \succ \boldsymbol{y} \,|\, \boldsymbol{y}^*]$ according to our probabilistic model (4). Starting with a uniform prior on the simplex $\{(\alpha_1, \alpha_2) \,|\, \alpha_1, \alpha_2 \geq 0,\ \alpha_1 + \alpha_2 = 1\}$, N updates (11) were realized based on N random preferences of that kind.

Figure 1 shows typical examples of the marginal distributions for α_1 after $N = 50$, $N = 200$ and $N = 500$ examples. As expected, the distributions are fluctuating around the true value of α_1 (here taken as 0.3) and become more and more peaked with increasing N. Moreover, comparing the distributions on the left and the right panel, it can be seen that learning becomes easier for larger values of β: for $\beta = 5$, the distributions are less peaked than for $\beta = 10$. Since β reflects the reliability of the oracle, this is again in agreement with our expectation. The same effect can also be observed in Figure 2 (left), which shows the boxplots for the mean value estimator (14); 100 of such estimators were derived from distributions for $N = 100$ and different values of β. Again, the larger β, the more precise the estimate of α_1 (and correspondingly of α_2).

As explained above, our Pref-CBR framework only produces an estimate $\boldsymbol{y}^\bullet$ of the ideal solution $\boldsymbol{y}^*$. Therefore, our distance learning method is based on preferences $[\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^\bullet]$ that can be seen as "noisy" versions of the true preferences $[\boldsymbol{y} \succ \boldsymbol{z} \,|\, \boldsymbol{y}^*]$. To simulate this property, we generated triplets $(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{y}^*) \subset \mathbb{Y}$ as above and set $\boldsymbol{y}^\bullet = \boldsymbol{y}^* + (\epsilon_1, \epsilon_2)^\top$, where ϵ_1 and ϵ_2 are (independent) normally distributed random variables with mean 0 and standard deviation σ. The observed preference (either $\boldsymbol{y} \succ \boldsymbol{z}$ or $\boldsymbol{z} \succ \boldsymbol{y}$) was then generated with our model (4) using the true $\boldsymbol{y}^*$, while distance learning was done using this model with the estimate $\boldsymbol{y}^\bullet$.

The effect of learning from noisy examples can be seen in Figure 2 (right), where we again show boxplots for the mean value estimator (14) based on 100 repetitions of the learning procedure with $N = 100$. As can be seen, the noise level σ does not seem to have a strong influence on the *variance* of the estimation. What is notable, however, is an apparent *bias* of the estimate: The larger σ, the more the estimates of α_1 are moving away from the true value 0.3 toward 0.5.

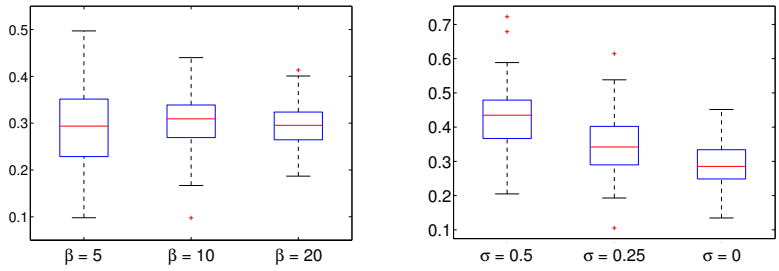

Fig. 2. Boxplots for the mean value estimate of α_1. Left: Different values of the precision parameter β. Right: Different levels of noise in the estimate $\boldsymbol{y}^\bullet$.

Although this result cannot easily be generalized beyond the specific setting of our experiment, a tendency toward uniform weights of the α-coefficients (i.e., $\alpha_1 = \alpha_2 = 0.5$ in our case) is plausible: The more $\boldsymbol{y}^\bullet$ deviates from $\boldsymbol{y}^*$, the more noisy the examples will be for our distance learner—in the limit, they will become purely random, and on average, all local distances Δ_i will seemingly have the same influence then.

4.2 Red Wine Recommendation

In this case study, we applied Pref-CBR to the problem of red wine recommendation. The scenario is as follows: A wine merchant tries to find his best offer for a customer, i.e., that wine in his cellar the customer likes the most. For an average customer, it will be much easier to (qualitatively) compare two wines instead of rating an individual wine—this nicely fits the assumption of our framework. Thus, the merchant can offer different candidate wines to the customer (who plays the role of our oracle), which are always compared to the current favorite. For obvious reason, however, the number of such comparisons needs to be limited.

To simulate this scenario, we made use of the red wine data set from the UCI machine learning repository [7]. This data describes 1599 red wines in terms of different chemical properties; here, we only used three of them, namely sulphates (y_1), pH (y_2), and total sulfur dioxide (y_3), which were found to have the strongest influence on preference [4]. We randomly extracted 500 wines to constitute the wines in the cellar, while the remaining 1500 were used as queries. Thus, a query is a wine that is thought of as the ideal solution of a customer (in this example, problem space and solution space therefore have the same structure).

We defined the distance measures in terms of (7) with the local distances given as $\Delta_i(\boldsymbol{y}, \boldsymbol{y}^*) = |y_i - y_i^*|$, $i = 1, 2, 3$. Moreover, assuming that the different chemical properties have a different influence on taste, we defined the *ground truth* distances $\Delta_X = \Delta_Y$ by setting $\alpha_1 = 0.1$, $\alpha_2 = 0.6$, $\alpha_3 = 0.3$. However, we assume this ground truth measure, which is the target of our similarity learning method, to be unknown. Instead, the measure used in the solution space as

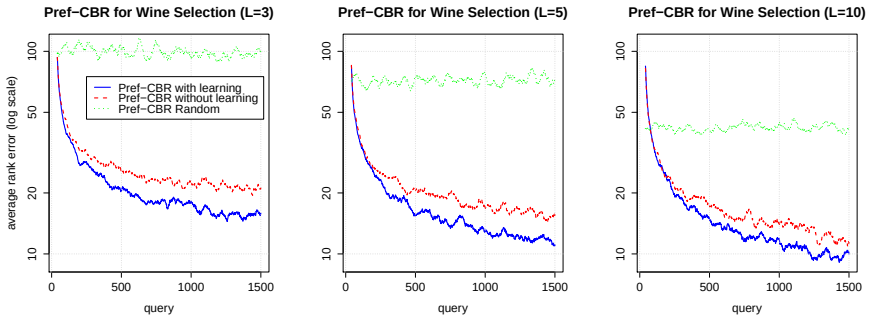

Fig. 3. Evolution of the average rank error in sequential problem solving (each query gives rise to one problem solving episode) for $L = 3, 5$ and 10 queries.

an initial measure subject to adaptation (and in the problem space without adaptation) is the *default measure* with uniform weights $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$.

We used Pref-CBR Search (Algorithm 1) for sequential problem solving, starting with an empty case base. Then, we applied the algorithm to the 1500 query cases one by one and monitored its performance. To measure the quality q of a proposed solution $\boldsymbol{y}^\bullet$ for a problem $\boldsymbol{x}_0$ (line 22), we computed the position of this solution in the complete list of $|\mathbb{Y}| = 500$ wines in the store ranked by (ground truth) similarity to the query $\boldsymbol{x}_0 = \boldsymbol{y}^*$ (i.e., 1 would be the optimal performance). To stabilize the results and make trends more visible, the corresponding sequence of performance degrees produced by a single run of Pref-CBR Search was averaged over 100 such runs.

We compared two versions of Pref-CBR Search, namely with and without (solution) similarity adaptation. Moreover, as a baseline we also used a search strategy in which the preference-guided selection of the next candidate solution in line 15 of Algorithm 1 is replaced by a random selection (i.e., an element from $\mathbb{Y}^{nn}$ is selected uniformly at random). Although this is a very simple strategy, it is suitable to isolate the effect of guiding the search behavior on the basis of preference information.

We applied our Algorithm 1 with $K = 5$, $L \in \{3, 5.10\}$, $J = 25$; since the solution space is quite small, we used a global neighborhood structure, i.e., we defined the neighborhood of a solution $\boldsymbol{y}$ as $\mathcal{N}(\boldsymbol{y}) = \mathbb{Y} \setminus \{\boldsymbol{y}\}$. As can be seen from the results in Figure 3, our preference-based CBR approach shows a clear trend toward improvement from episode to episode, as opposed to the random variant of the search algorithm.

More importantly, however, similarity adaptation is clearly beneficial: Making use of the preference information gathered in the first episodes, Pref-CBR Search succeeds in learning the ground truth similarity measure, which in turn leads to better search performance and solution quality. The variant without similarity adaptation finds reasonably good solutions, too, because even the suboptimal (default) measure is guiding the search in a right direction—yet, with

similarity adaptation enabled, the search becomes more effective, and the smaller the number of queries (L), the more pronounced the relative improvement.

5 Summary and Outlook

In this paper, we extended our framework for preference-based CBR by a Bayesian learning method for adapting the similarity (distance) measure in the solution space. As already explained, this measure equips this space with a topology, which in turn influences the search-based problem solving strategy in Pref-CBR. Consequently, a data-driven adaptation of this measure in cases where it does not seem to be optimally predefined can be seen as an important prerequisite for the effectiveness of Pref-CBR. In fact, first experimental studies presented in this paper have shown that an adaptation of similarity can lead to an improved overall performance of the CBR system.

Needless to say, there is still much scope to improve our method. For example, our implementation of Bayesian inference based on a fixed discretization Γ of the parameter space is not very efficient, especially if the vector $\boldsymbol{\gamma}$ to be learned is high-dimensional. Therefore, this approach should be replaced by more sophisticated methods for approximate Bayesian inference.

While the last problem is purely computational, the learning bias that is due to the use of estimated instead of truly ideal solutions is of a more conceptual nature. One way to overcome this problem is to make reasonable statistical assumptions about the relationship between $\boldsymbol{y}^\bullet$ and $\boldsymbol{y}^*$ (like normality of the difference in our simulation study), and to extend our probabilistic model of the data-generating process correspondingly. Realizing Bayesian learning on the basis of this extended model would then explicitly account for the potential sub-optimality of $\boldsymbol{y}^\bullet$.

A nice feature of Bayesian learning is the possibility to incorporate prior knowledge in a simple and elegant way. In our case, this would concern knowledge about the importance of the local distance measures (parameter $\boldsymbol{\alpha}$) and/or knowledge about the reliability of the information source (parameter β). Exploring the usefulness of this option is another point on our agenda.

Finally, instead of only learning the solution similarity S_Y, it would of course make sense to adapt the problem similarity S_X, too. This could be done, for example, on the basis of the preferences collected for different problems $\boldsymbol{x}_i$. In fact, our neighborhood-based approach to case retrieval assumes that similar problems lead to similar preferences on solutions, or, more concretely: if two cases $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar in the sense of S_X, the respective sets of preferences $\mathcal{P}(\boldsymbol{x}_i)$ and $\mathcal{P}(\boldsymbol{x}_j)$ are similar (in a sense to be specified), too. The idea, then, would be to adapt S_X in such a way as to make this assumption as valid as possible.

Acknowledgments

This work has been supported by the German Research Foundation (DFG).

References

1. A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.
2. Hans-Dieter Burkhard and Michael M Richter. On the notion of similarity in case based reasoning and fuzzy theory. In *Soft computing in case based reasoning*, pages 29–45. Springer, 2001.
3. W. Cheng and E. Hüllermeier. Learning similarity functions from qualitative feedback. In *Proceedings ECCBR–2008, 9th European Conference on Case-Based Reasoning*, pages 120–134, Trier, Germany, 2008. Springer-Verlag.
4. Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.
5. C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. *Artificial Intelligence*, 2011.
6. J. Doyle. Prospects for preferences. *Comput. Intell.*, 20(2):111–136, 2004.
7. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
8. J. Goldsmith and U. Junker. Special issue on preference handling for Artificial Intelligence. *Computational Intelligence*, 29(4), 2008.
9. Shengbo Guo and Scott Sanner. Real-time multiattribute Bayesian preference elicitation with pairwise comparison queries, 2010.
10. E. Hüllermeier and P. Schlegel. Preference-based CBR: First steps toward a methodological framework. In *Proc. ICCBR–2011, 19th International Conference on Case-Based Reasoning*, pages 77–91. Springer-Verlag, 2011.
11. M. Peterson. *An Introduction to Decision Theory.* Cambridge Univ. Press, 2009.
12. Armin Stahl. Learning feature weights from case order feedback. In *ICCBR–01*, pages 502–516, Vancouver, Canada, 2001. Springer-Verlag.
13. Armin Stahl. Learning similarity measures: A formal view based on a generalized CBR model. In *ICCBR–05*, pages 507–521, Chicago, USA, 2005. Springer.
14. Armin Stahl and Thomas Gabel. Using evolution programs to learn local similarity measures. In *ICCBR–03*, pages 537–551, Trondheim, Norway, 2003. Springer.
15. Armin Stahl and Thomas Gabel. Optimizing similarity assessment in case-based reasoning. In *Proceedings of the 21th National Conference on Artificial Intelligence, AAAI*, 2006.
16. Armin Stahl and Sascha Schmitt. Optimizing retrieval in CBR by introducing solution similarity. In *Proceedings of the International Conference on Artificial Intelligence, IC-AI*, Las Vegas, USA, 2002.
17. L. Yang, R. Jin, and R. Sukthankar. Bayesian active distance metric learning. In *Proc. UAI, Uncertainty in Artificial Intelligence*, 2007.