Indexing by Metric Adaptation and Representation Upgrade in an Emotion-based Agent Model*

Rodrigo Ventura and Carlos Pinto-Ferreira
Institute for Systems and Robotics
Instituto Superior Técnico, TULisbon
Av. Rovisco Pais, 1
1049-001 Lisbon, PORTUGAL
{yoda,cpf}@isr.ist.utl.pt

Abstract

Following recent neurophysiological research, one important role of emotions consists in providing a mechanism for adequate and efficient response to relevant stimuli. In this paper we propose a methodology for implementing such a mechanism, based on a previously presented emotion-based agent model. This model is founded on a double knowledge representation paradigm: a stimulus reaching the agent is processed under two different and simultaneous perspectives — a simple (termed perceptual) and a complex (termed cognitive) — from which two differing representation schemata are derived. This paper addresses a twofold strategy for the construction of a perceptual representation. The first one consists in adapting a perceptual metric, with the goal of approximating it to the cognitive metric. The second one has the goal of upgrading the perceptual representation with additional components (e.g., features). Techniques borrowed from nonmetric Multidimensional Scaling are used to approach these goals.

1 Introduction

The research presented here is a contribution for the development of an emotion-based autonomous agent model, originally proposed in [3]. Briefly, the agent model is based on a double-representation of stimuli: a simple representation termed *perceptual image*[1], designed for fast processing and immediate response to urgent situations, and a complex representation termed *cognitive image,* thus slow to process. These two representations are extracted and processed, simultaneously, by the two levels of the architecture: the perceptual and the cognitive levels. The parallelism of the processing is essential, so that quick response to urgent situations is not compromised by the slow processing of the cognitive level. These two representations are then stored in memory, together with a third representation, termed *desirability vector.* This vector characterizes stimuli according to a set of dimensions relevant to the agent, such as dangerous/safe, interesting, demanding urgent action, threatening, and so on. The memory is thus formed by associations between perceptual and cognitive images, marked by the corresponding desirability vectors. Once the agent faces a new situation, it matches the incoming stimulus with the agent memory, thus retrieving the associated images and the marking.

This model was inspired by the somatic marker hypothesis of António Damásio [2]. According to this hypothesis, the brain is able to associate in memory cognitive mental imagery with representations of incurred bodily changes, and later, to enact these bodily representations after the recollection of that mental imagery [2].

The agent model hypothesizes that the representations matching mechanism proceeds according to two steps. In the first step, a perceptual image is obtained from the stimulus and matched against the perceptual images in memory. For the ones yielding a closer match, the agent, in the second step, matches the cognitive image extracted from the stimulus with those indexed by the closest perceptual images. This mechanism is termed *indexing.* Considering that the cognitive matching mechanism is an operation more complex than the perceptual one, this mechanism allows for a narrowing of the candidate cognitive images, thus providing an efficient algorithm to find cognitive matches.

This indexing mechanism was previously formulated and theoretically analyzed, under the assumption that the matching of the cognitive and perceptual images are performed in metric spaces [4]. Given a stimulus $s \in \mathcal{S}$,

*This work was partially supported by FCT (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds.

[1] The term *image* is here utilized in a broad sense (a percept).

the agent extracts two kinds of representations: a perceptual image $i_p \in \mathcal{I}_p$, and a cognitive one $i_c \in \mathcal{I}_c$. The sets $\mathcal{I}_p$ and $\mathcal{I}_c$ contain all possible perceptual and cognitive images. Each one of them is equipped with a metric function, denoted by $d_p : \mathcal{I}_p \times \mathcal{I}_p \rightarrow \mathbb{R}_0^+$ and $d_c : \mathcal{I}_c \times \mathcal{I}_c \rightarrow \mathbb{R}_0^+$ respectively, mapping pairs of images to distances, interpreted as degrees of mismatch. The memory is assumed to be formed by pairs[2] of cognitive and perceptual images $\langle i_c^k, i_p^k \rangle$ $(k = 1, \ldots)$. The goal of the indexing mechanism is then to find the memory pair which cognitive image minimizes its distance to the one extracted from the stimulus, employing the perceptual representation to do so in an efficient manner.

The research presented here concerns the following problem: how to construct a perceptual representation (and metric) with the goal of optimizing the indexing efficiency. In other words, the ideal perceptual representation and metric are the ones that yield small perceptual distances iff the corresponding cognitive distances are also small. To do so, two strategies are explored. One corresponds to adapting a perceptual metric, via a set of parameters, such that cognitive proximity implies perceptual nearness:

$$d_c(i_c^1, i_c^2) < d_c(i_c^1, i_c^3) \Rightarrow d_p(i_p^1, i_p^2) < d_p(i_p^1, i_p^3) \qquad (1)$$

for all image pairs $\langle i_c^k, i_p^k \rangle$ $(k = 1, 2, 3)$ obtainable from stimuli, in a given environment. The second strategy addresses the improvement of the perceptual representation, in the following sense. Assuming that the perceptual representation is a vector of features extracted from stimuli, when these features are not sufficiently representative to satisfy (1), the goal is to upgrade the perceptual representation with new, more representative, features. Both of these strategies are approached here using Multidimensional Scaling (MDS) techniques [1].

2 Methodology

A good perceptual representation is one which satisfies the implication (1) for all image pairs encountered by the agent. Note that this goal is similar to the MDS one, once one considers the cognitive distances to be the *dissimilarities,* and the perceptual ones, to be the *distances* among objects, in the MDS terminology [1]. However, there are differences. In the case of the MDS, the metric is given while the object coordinates are sought. In the case of the indexing, the object coordinates (perceptual images) are given, while the (perceptual) metric is subject to adaptation.

We propose to perform a gradient descent, within the framework of the nonmetric MDS, w.r.t. a parameterization of the perceptual metric, instead of w.r.t. the point coordinates. Thus, the perceptual metric is assumed to depend on a vector of parameters. For instance, these parameters can assign a degree of relevance to each feature of the perceptual representation. Regarding the construction of additional perceptual features, we propose to append each perceptual image with a pre-specified amount of additional components. These components represent the values that the new features ought to take, for each one of the perceptual images in the training set. Their values are randomly initialized, and subject to gradient descent as in the nonmetric MDS. Concerning the obtainment of those added components for new stimuli, the idea we advance is to utilize the obtained values to construct a regression model. That regression model can then be used to obtain the new features values for new stimuli.

Let a perceptual image i_p^r, consisting of the concatenation of q numerical features $x_{r1}, \ldots, x_{rq}$, extracted from a given stimulus, with p additional components $y_{r1}, \ldots, y_{rp}$, be denoted by the vector

$$i_p^r = (x_{r1}, \ldots, x_{rq}, y_{r1}, \ldots, y_{rp})^T \qquad (2)$$

These additional components correspond to the values that the new features ought to take for that particular perceptual image. The perceptual metric employed here is parameterized by q coefficients $\theta_1, \ldots, \theta_q$, taking the form

$$d_{rs} = \sqrt{\sum_{i=1}^{q} \theta_i^2 (x_{ri} - x_{si})^2 + \sum_{i=1}^{p} (y_{ri} - y_{si})^2} \qquad (3)$$

This parameterization corresponds to assigning a weight (relevance) to each perceptual feature, before calculating the Euclidean metric. Moreover, when the algorithm assigns a zero weight to a feature, that feature can be deleted from the perceptual representation, since it is irrelevant (w.r.t. the cognitive matching). The additional components are not weighted since it would just add redundant degrees of freedom.

The cost function employed here is the sum of the MDS stress, with a regularization term penalizing the absolute values of the metric parameters. The MDS stress S assesses the fit of the the perceptual distances $\{d_{rs}\}$ to the ones resulting from the isotonic regression, here denoted as $\{\hat{d}_{rs}\}$.

$$S = \sqrt{\frac{S^*}{T^*}} \qquad S^* = \sum_{r,s} \left(d_{rs} - \hat{d}_{rs} \right)^2 \qquad T^* = \sum_{r,s} d_{rs}^2 \qquad (4)$$

The distances $\{\hat{d}_{rs}\}$ satisfy, by construction, two conditions: (a) it preserves the cognitive distances ordering

[2] The association with the desirability vector is not considered here, since it is not an active player in the current formulation of the indexing mechanism.

$(d_c(i_c^r, i_c^s) < d_c(i_c^r, i_c^t) \Rightarrow \hat{d}_{rs} < \hat{d}_{rt})$, and (b) it minimizes S^* above, restricted to (a). The cost function is then the sum of the MDS stress with the regularization term.

$$J = S + \xi \sum_{i=1}^{q} |\theta_i| \tag{5}$$

This latter term, weighted by ξ, is included in the cost function for two reasons. First, if the stress is invariant to a perceptual component, the stress gradient w.r.t. the corresponding weight would be zero, and therefore the initial parameter value would stay at the same value during the descent. The second reason is due to the quadratic contribution of the parameters to the stress: in order to prevent a slow asymptotic convergence to zero (and therefore never reaching zero exactly), the gradient of their absolute values forces them to approach zero faster[3]. In sum, this term contributes to reduce the number of non-zero parameters θ_i, and therefore to permit the deletion of the components with zero weights from the perceptual representation.

In order to express the gradient of the stress, one can consider a parameter vector Λ containing all variables subject to the gradient descent.

$$\Lambda = [\lambda_1 \cdots \lambda_{q+np}]^T = [\theta_1 \cdots \theta_q | y_{11} \cdots y_{np}]^T \tag{6}$$

The gradient is then obtained from the partial derivative of the cost w.r.t. each parameter λ_k

$$\frac{\partial J}{\partial \lambda_k} = \frac{\partial S}{\partial \lambda_k} + \xi \,\mathrm{sgn}(\lambda_k) \tag{7}$$

$$\frac{\partial S}{\partial \lambda_i} = S \sum_{r,s} \left(\frac{d_{rs} - \hat{d}_{rs}}{S^*} - \frac{d_{rs}}{T^*} \right) \frac{\partial d_{rs}}{\partial \lambda_i} \tag{8}$$

As before, the summation above is performed for $r = 1, \ldots, (n-1)$ and $s = (r+1), \ldots, n$.

If λ_k corresponds to a metric parameter θ_l, then

$$\frac{\partial d_{rs}}{\partial \theta_k} = \frac{(x_{rk} - x_{sk})^2}{d_{rs}} \, \theta_k \tag{9}$$

otherwise, if it corresponds to a component y_{ui}, then

$$\frac{\partial d_{rs}}{\partial y_{ui}} = \frac{y_{ri} - y_{si}}{d_{rs}} \, (\delta^{ru} - \delta^{su}) \tag{10}$$

where δ^{ij} is the usual Kronecker function (1 iff $i = j$, 0 otherwise).

Taking into account these considerations, and based on the standard nonmetric MDS algorithm [1], we propose the following one:

1. Start with an initial variables vector Λ. For instance, the metric parameters θ_k can be initialized to all ones, and the additional components $\{y_{ri}\}$ randomly distributed with a uniform distribution;
2. Normalize the metric parameter vector $\Theta = (\theta_1, \ldots, \theta_q)^T$ to unit norm, since the stress is invariant to scaling of this vector. The additional components $\{y_{ri}\}$ are, however, not normalized[4];
3. Compute the distances set $\{d_{rs}\}$ using the parameterized perceptual metric (3);
4. Perform the isotonic regression to obtain the set of distances $\{\hat{d}_{rs}\}$;
5. Compute the cost; if its value is below a threshold ϵ, stop the algorithm (stopping criterion);
6. Find the gradient of the cost function (5) w.r.t. the variables vector Λ;
7. Perform a step of the gradient descent method;
8. Go to step 2.

3 Results

To validate the proposed methodology, a simple test-bed was devised. Random points $\mathbf{x} \in \mathbb{R}^c$ (simulating stimuli) were uniformly drawn from an hypercube of unit side length. The cognitive images $i_c \in \mathbb{R}^c$ were set to the components of $\mathbf{x}$ multiplied by fixed coefficients $[w_1, \ldots, w_c]$, randomly chosen between 0 and 2 prior to each run

$$i_c = \mathrm{diag}(w_1, \ldots, w_c)\, \mathbf{x} = \mathbf{W}\mathbf{x} \tag{11}$$

These coefficients introduce different degrees of relevance to the components of i_c. The perceptual images were obtained by concatenating two vectors: the p first components of $\mathbf{x}$ multiplied by a second set of fixed coefficients $[v_1, \ldots, v_c]$ (for $p \leq c$), also randomly chosen between 0 and 2); and n random numbers (noise) between 0 and 1. Thus, the perceptual images have $p + n$ components.

$$i_p = \left(\frac{[\mathrm{diag}(v_1, \ldots, v_p) | \mathbf{0}]\; \mathbf{x}}{\mathbf{u}} \right) = \left(\frac{\mathbf{V}\mathbf{x}}{\mathbf{u}} \right) \tag{12}$$

where $\mathbf{0}$ stands for a matrix of zeros of appropriate dimension, and $\mathbf{u}$ for the noise vector. The random weights in $\mathbf{W}$ and $\mathbf{V}$, randomly drawn from the $[0; 2]$ interval, together with the numbers c, p, and n, define a *world,* represented

[3] Numerically this makes parameters close to zero to oscillate around zero, so, they are set to zero once they become negative. The implementation further forces them to stay at zero thereafter.

[4] Otherwise it would constrain *a priori* the relative weights of the additional components w.r.t. the original features in (3). Normalizing the parameters vector prevents its norm from growing or shrinking because of numerical errors. Moreover, because of (3), the additional components do not grow/shrink arbitrarily.

by a tuple $\langle c, p, n, \mathbf{W}, \mathbf{V} \rangle$. The cognitive distances are calculated using the Euclidean distance, while the perceptual ones employ the metric (3).

In order to evaluate the results, a measure of performance called *eval-order* was introduced, aiming at assessing how well the indexing mechanism would behave, for a particular perceptual metric. This assessment is performed using a test set disjoint from the training set employed in the gradient descent (cross-validation). Inspired by the *N-best* indexing algorithm described in [4], the eval-order is defined in the following way: given a cognitive and perceptual images pair $\langle i_c, i_p \rangle$, determine all perceptual distances from it to images in the perceptual memory (*i.e.,* the training set); then, after sorting all these images w.r.t. the perceptual distances, determine which n-th image pair $\langle i_c^k, i_p^k \rangle$ on the resulting ordered list has the minimum cognitive distance to $\langle i_c, i_p \rangle$. In the ideal case, it corresponds to the first one, and thus an eval-order of 1 (one). Higher values correspond to worse performance.

The features in the perceptual images were all (training and test sets) normalized to zero mean and unit variance, prior to any experiment. Unless otherwise stated, the parameterization Θ of the perceptual metric was initialized to all ones. The additional components, when used, were initialized with a uniformly distributed random configuration, as in the nonmetric MDS algorithm.

In the first phase of experimentation, no additional perceptual components were considered, and the cognitive and perceptual dimensions were made equal ($c = p$). The algorithm was run for 100 generated training sets with the same world parameters, each one containing 100 training patterns (and thus 4950 dissimilarities among them). The world dimensions were $c = p = 10$ and $n = 3$. For each training set, a test set containing 100 patterns was also generated, for posterior eval-order assessment. Figure 1 shows the results: for each component index, the first bar represents the respective weight from matrix $\mathbf{W}$ (labeled W), while the second one corresponds to the mean value of the corresponding metric parameter along all runs, with standard deviation error bars (labeled result). Both vectors are normalized to unit norm, in order to be comparable. Note that the resulting weights faithfully represent the relative importance of the $\mathbf{x}$ coordinates in the cognitive metric. The observed extinguishing of the third weight is due to the combined effect of its diminished importance (*i.e.,* low value in $\mathbf{W}$), and the penalization of non-zero weights in (5). Moreover, the last three components (noise) were all zero, thus showing a successful capability of identifying irrelevant features.

Concerning the eval-order assessment, the results are shown in table 1. These are consolidated values, obtained in the following way: for each run, a training set and a test set were randomly generated, as above-mentioned; then, the weights obtained in each run were tested against the test

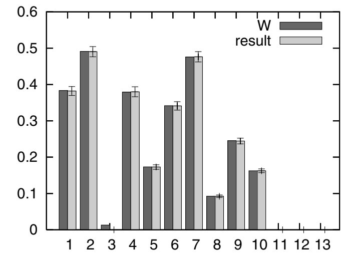


Figure 1. Weights obtained by the algorithm.

set (cross-validation), calculating the mean, minimum, and maximum values of the obtained eval-orders for all images in the test set. The results shown here correspond to the mean of these means[5] (central tendency), the minimum of all minima, and the maximum of all maxima (worst case of eval-order). These results show a significant improvement of the eval-order performance after using the metric weights found by the algorithm. Namely, the worst case (maximal eval-order) went down from 92 to just 2. Note that the test set has 100 image pairs, therefore, the worst possible eval-order value is 100.

metric	mean	min	max
unweighted	11.4	1	92
weighted	1.00	1	2

Table 1. Obtained eval-order performance values.

The algorithm was run from several initial conditions, in order to determine the sensitivity of the solution w.r.t. local minima. Apart from 11% of outlier runs, the metric weights converged to the correct values. These outlier runs were found to be caused by weights initialized close to zero.

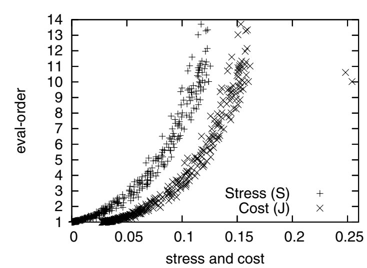
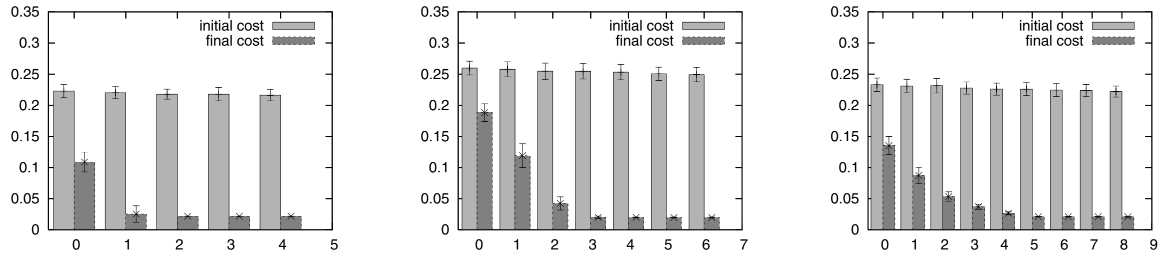


Figure 2. Sampling of the cost (and stress) values w.r.t. the eval-order.

Because of the lower dimensionality of the parameters vector, the algorithm still converged to the correct solution using either about 10 training patterns, or about 0.5% of the total number of dissimilarities[6].

[5] This is equivalent to a mean over all image pairs, since all test sets have the same size.

[6] Random sampling from the 4950 dissimilarities originated by the 100



(a) for $c = 6$, $p = 5$, and $n = 3$ (b) for $c = 8$, $p = 5$, and $n = 3$ (c) for $c = 10$, $p = 5$, and $n = 3$

Figure 3. Statistics of initial and final costs (vertical axis) w.r.t. the number of new components, for various world parameters.

The introduction of a strictly monotonic non-linear distortion function f was also tested by setting the cognitive distance to $d_c(i_c^1, i_c^2) = f\left(\|i_c^2 - i_c^1\|\right)$. The results were not altered, as expected, by construction of the nonmetric MDS.

The relationship between the cost values and the eval-order is critical to the success of the approach. The algorithm seeks the reduction of the cost function (5), while the quality of the result is measured by the eval-order performance metric. For this synthetic world, the relationship between the cost and the eval-order during the gradient descent was examined. Figure 2 plots a sampling taken from 25 runs, by sampling randomly 1 out of 5 descent steps. This illustrates how, in this test-bed, smaller cost values lead systematically to better generalization in the test set. This kind of analysis can be useful to assess whether the method is appropriate for a given world, w.r.t. the generalization performance.

The second phase of the experimentation comprised the introduction of new components to the perceptual representation. To do so, the dimension of the cognitive images was made higher than the perceptual one, *i.e.,* $c > p$. Thus, the perceptual metric is performed with less components than the cognitive one. The first impact of this is that, without the introduction of new components, the final cost values were much higher than before, due to lack of fit (previous experiments resulted in final costs between 0.02 and 0.03). Figure 3 shows the obtained initial and final costs, after testing four different generated worlds. The algorithm was run for several amounts of new components for each one of the worlds. The plots display the mean and the standard deviation of the initial and final costs, after 100 runs performed in each world. Error bars denote the standard deviation of the cost values across all runs. The only difference among runs sharing the same world parameters is the initial values for the new dimension coordinates (initialized to random values, as explained above). The training set contained 20 patterns.

These plots corroborate the idea that, once the number of new components reaches $c - p$, the final cost stabilizes on values close to the ones found in previous experiments. This observation suggests a methodology for the estimation of how many new components are required for a given problem of unknown structure: to try successively higher amounts of new components, until the final cost value stabilizes.

Further experimentation showed more interesting results (omitted here due to lack of space). In one of them, a single additional component to the cognitive representation was considered ($c = p + 1$). The algorithm showed the ability to reconstruct, for the perceptual images in the training set, the values of that component. The reconstruction power, measured in term of signal-to-noise ratio (SNR) between the missing component and the recovered dimension yielded values of about 45dB. In another experiment, a linear regression model was employed to extract features for the perceptual representation. The obtained results were also satisfactory.

4 Concluding remarks

Experimentation have shown interesting results, thus illustrating the proposed methodology on a synthetic world. However, due to the lack of exploitable structure of the employed synthetic worlds, whenever the cognitive dimensionality exceeds the perceptual one (including any additional components), the obtained results become significantly degraded because of poor fitting. For instance, noise components get zero weights (because of their irrelevancy), only if a good fit is found.

References

[1] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, UK, 1994.

[2] A. R. Damasio. *Descartes' Error: Emotion, Reason and the Human Brain*. Picador, 1994.

[3] R. Ventura and C. Pinto-Ferreira. Emotion-based agents. In *Proceedings AAAI-98*, page 1204. AAAI, AAAI Press and The MIT Press, 1998.

[4] R. Ventura and C. Pinto-Ferreira. A formal indexing mechanism for an emotion-based agent. In *Proceedings IASTED-2002*, pages 34–40, Malaga, Spain, 2002. ACTA Press.

patterns of the training set.