Decision Support in AmI Sport Environments

J. Vales-Alonso, P. López-Matencio, Juan J. Alcaraz and J. García-Haro
Universidad Politécnica de Cartagena
Campus Muralla del Mar, Spain
Email: {javier.vales,pablo.lopez,juan.alcaraz,joang.haro}@upct.es

Abstract—In this paper, we comparatively analyze a set of decision making methodologies that have been successfully applied to improve athletes’ training in real-time. The decision-making processes are based on the environmental data as well as on the athletes’ biometrics in a scenario with runners in cross country circuits. Data acquisition is performed by means of a WSN which measures temperature and circuits slopes, whereas heart rate is monitored for each athlete. In all cases, the goal was to select the best tracks in the circuit to control the heart rate of the athlete within a given interval. Two broad types of decision-making procedures have been studied: (i) optimal classifiers, and (ii) dynamic programming. Results show a notable performance increase of these methods over heuristics, as well as the importance of environmental sensor data.

I. INTRODUCTION

Wireless Sensor Networks (WSN) processing capabilities are growing rapidly, and they are getting increasingly embedded and seamlessly integrated in the physical world. WSN-assisted environments will be sensitive and responsive to the presence of people, allowing the development of Ambient Intelligence (AmI) services. Among the activities where these technologies can be applied, sports may be one of the most benefited. For instance, in outdoors sports such as cross country running and jogging, practitioners commonly use wearable computing devices, which can provide useful telemetry about runner biometrics and practice-related events [1]. Some of the potentially useful data that can be gathered by the system are the hearth rate (HR), track routes, speeds and distances, and so forth. Often, the complexity of the data collected requires subsequent analysis by a human coach, sometimes with the aid of specific software.

In common training systems, performance is only evaluated at the end of the training session, and sensed data are usually incomplete because generally only human biometrics are analyzed. In consequence, it is not possible to make real-time decisions during the training session, (e.g. to change the running pace or the track route) allowing the athlete to better accomplish their training objectives under variable biometrical. Moreover, nowadays training systems are limited because they do not take into account the direct influence on athletes performance of the environmental conditions (e.g. temperature or humidity). For simplicity we center our attention on the influence of ambient temperature [2] and track hardness [3] in the training decisions (i.e. HR control).

Fig. 1 depict our AmI system approach. Athletes train in a field with track alternatives. Each track has a different hardness degree and the weather conditions may vary (temperature). The goal is to select a suitable track for the runner at each path junction, in order to fulfill an overall training program. In our system we aim at controlling the HR of the user. Different training HR ranges are possible (e.g. aerobic, weight control, etc.). Before training starts, athletes or coaches select the desired HR profiles during exercise, e.g. perform tracks in fat-burn HR range. We denote this kind of training as homogeneous, as it involves a single HR ranges.

As training conditions can rapidly vary, the system would be expected to make real-time decisions to meet the user needs. Therefore, constant user and environmental monitoring
This paper presents a comparative resume of several decision-making processes for outdoor sports. Three classification techniques are comparatively investigated:

- k-Nearest-Neighbors (k-NN) [5]
- Splines of $(m, s)$ degree [4]
- Dynamic programming (DP)

Whereas the two first classification techniques make decisions for a single-step horizon, *i.e.* they look for the best track decision to fulfill only the next step in the training program, the DP methodology allows to select a series of tracks which maximizes the correct distribution of the HR intensity levels during the whole training period. This is a multi-step decision making process, since it is assumed that decision at step $n$ will affect HR at steps $n+1$, $n+2$, and so on.

### II. Single-step Decision Engines

The k-NN classifier has been recently utilized to predict a variety of heart diseases [6]. Numerical approximation techniques based on splines are extensively applied in engineering areas such as signal processing [7]. Among the different spline techniques, we selected $(m, s)$-splines [8] for their favorable characteristics to our research: they allow to face multivariable problems, the problem domain is not required to be a mesh grid (that is, data used to compute interpolation can be at any point in space), and the computational load is low.

The goal of the single-step classification engines consist in maintaining the runners’ HR in a given target range. For our experiments, aerobic (cardio-training) was the selected target.

The k-NN method is *memory-based*, because, in contrast to other statistical methods, requires no training, thus, no model to fit. It belongs to the category of Prototype Methods. It operates on the intuitive idea that close objects are more likely to be in the same category. Thus, k-NN predictions are based on a set of prototype examples that are used to predict new (or unseen) data based on the majority vote (for classification tasks) and averaging (for regression) over a set of k-nearest prototypes.

The following input features are considered in our decision engine implementation:

- Next and current track hardinesses (“hard” or “easy”)
- Next and current track environmental temperatures (“high” or “low”)
- Current average and variance of the heart rate

The method operates as follows. Given a data set consisting of $N$ pairs $(p_1, g_1) \ldots (p_N, g_N)$ where $p_i$ is the feature information vector and $g_i$ is the corresponding scalar class label. Each of the $N$ pairs is called *prototype*, or *scenario*, and, every data $p_i$ belongs to the feature space of dimension $P$. In our experiment specification a feature space of dimension 6 consisting of the previous features. Besides, two classes are possible (“In range” or “Out of range”) corresponding to a track where the HR is in the objective range or outside, respectively. Thus, given a query point $p_0$, the classification problem consists of finding the $k$ training points $p(r), r = 1, \ldots, k$ closest in distance to $p_0$, and then classify it, (*i.e.* assign a class label) using majority vote among the $k$ neighbors. If vote results in a tie then it would be resolved at random. In
Finally, let us mention the tests performed by using numerical multi-variable interpolation of the HR as decision engine methodology. In this approach (see [4]), the goal is estimating the future average HR if a track is selected, instead of obtaining a direct classification of that track. Then, if the average HR is within the desired interval, the track is considered as suitable.

The computation of the interpolation function was realized by using splines of order \((m,s)\) [8]. This splines have favorable characteristic to our research: allow to tackle multi-variable problems, the problem domain may be a grid or not, and finally the computational load is reduced. In fact, the interpolant function is a polynomial which can be evaluated by a normal PC in less than 1ms.

The tests were performed using knowledge base similar to the one used in the k-NN approach (that is, the same features were considered). The query points were again retrieved from the knowledge base - removing them for the computation of the interpolated mapping. The conformity of the classification method was measured by means of its degree of discrepancy between the decision engine outcome and the following real HR. That is, if after running in the selected track the average HR was not in the target interval, the classification failed. Otherwise, it was correct. For a knowledge base of 119 records and 11% of query points the ratio of valid decisions were superior to 85%, which also may outperforms in some tests the results obtained with k-NN method. Tests also show that the ratio of success is reduced by interpolating without environmental variables.

### III. Multi-Step Decision Engine

In the previous approaches only the problem of selecting the best decision for one stage is considered, independently of the future evolution. That is, optimization is done for a single-step scenario. We consider now the multi-stage case, where the track is selected in order to fulfill training goals, considering the future evolution of the athletes. This problem can be addressed as a dynamic program. As in the previous section we assume the goal of performing all tracks within a selected HR range \((\bar{HR})\).

For our formulation, let us assume a partition of HR ranges consisting of \(m\)-non-overlapping levels, each one comprising heart rate in the set \(HR^i = [hr_i, hr_{i+1})\) for \(i = 1, \ldots, m\), where \(hr_i\) denotes the lower bound of the heart rate associated to range \(i\). Let us define also the following elements for the DP formulation:

- A training session consists then in a sequence of \(N\) tracks.
- The stage of the system is denoted by \(n = 0, \ldots, N\).
- The state of the system at stage \(n, x_n\), containing all the required information for the CN to take a decision at each stage (e.g. the HR).
- The control applied at each stage, \(u_n\), is the track selected (each one with a different difficulty level). In this case \(u_n = \{\text{hard, easy}\}\).
- The reward obtained at each stage, \(r(x_n, u_n)\), is a function of the state and the control selected. Whenever a
track is traversed in $\hat{HR}$ there must be a positive reward added (+1 in our implementation). Otherwise, there is no reward.

- The value function, $J^n(x_n)$, is the maximum expected reward that can be gathered from stage $n$ to stage $N$, i.e. if the optimal decisions $u_n$ are taken from the current stage to the last one.

Therefore in the general problem, $x_n$ comprises three elements: (1) the athlete’s $HR$, which we denote as $hr_{x_n}$, (2) the selected track and (3) the number of loops remaining of each $HR$ class to accomplish the objective.

The dynamic program can now be formulated. It basically consists computing the value function $J^n(x_n)$ recursively as the sum of the reward expected in state $x_n$ plus the value function for the next stage, $n + 1$.

$$J^n(x_n) = r(x_n) + \max_{u_n} \left\{ E \left\{ J^{n+1} (f_n(x_n, u_n)) \right\} \right\}$$

$\forall n = 0 \ldots N$. To solve this equation, the transition mapping $f(\cdot)$ is modelled as a Markov Decision Process (MDP). In a MDP it is assumed that the probabilities of going from state $i$ to state $j$ when control $u_n$ is applied at stage $n$ are known, yielding to transition probability matrices $P^n(u_n)$. Therefore the recursive equation can be written in the following matricial form:

$$J^n = r + \max_{u_n} \left\{ P^n(u_n) J^{n+1} \right\}$$

In our experiments, we have computed the probability transition matrices by evaluating the relative frequencies of going from each combination of track and $HR$, $(track_i, HR^R)$, to each subsequent combination of track and $HR$, $(track_j, HR^R)$. These matrices have been obtained from the data gathered in the experiments described in the introductory section, to select the optimal policies for the athlete. In addition, this policy is compared to the following sub-optimal ones:

- Easy policy, i.e. selecting always the “easy” track of the circuit.
- Hard policy, i.e. selecting always the “hard” track of the circuit.
- Worst policy, easily computed substituting the $\max$ operator for the $min$ operator in (3).

In these tests, the goal of the the training plan is performing all tracks in the aerobic (cardio-training) regime. Fig. 4 shows the results for $N = 1, \ldots, 10$ using the absolute matrix. As expected, the optimal policy achieves better performance than non-optimal ones. In addition, Fig. 4 depicts a comparison of the optimal policy using either a specific matrix (high and low temperature) or the absolute one. If the specific ones are used, the value function improves significantly, especially in high temperature conditions.

This experiment clearly demonstrates that using environmental information is highly advisable, similarly to the single-step experiments of the previous section.

IV. CONCLUSION

Decision support based on k-NN, and $(m, s)$-spline interpolation classifiers have been tested for single-step decision making. Besides, another set of multi-step techniques have been introduced based on dynamic programming. In this case, the goal was to maximize the performance for all future training steps, rather than considering only a horizon with a single step of decision. Although not perfect, the classification methods achieves a ratio success in the range from 70% (k-NN) to 85% (splines interpolation). DP demonstrates a 70% match in an homogeneous training program. Note that this is similar to the k-NN classification ratio, however both results are not directly comparable since k-NN success ratio includes the out-of-range events, whereas DP do not, and therefore DP results are more impressive. Besides, environmental data improves outcome in all methodologies studied. Results presented in this paper must be considered as an starting point for the development of more sophisticated training methods based on the AmI approach.

REFERENCES


