Prediction of Sports Performance based on Genetic Algorithm and Artificial Neural Network

Biao Xu

Physical education department, Suzhou University of Science and Technology, Suzhou, 215011

Email: biaoxu_suzhou@163.com

Abstract

In order to predict sports performance more accurately, we investigated the recent algorithms, and proposed a hybrid prediction system based on genetic algorithm and artificial neural network (GANN). This is the first paper using the GANN to predict sport performance. We created a database of 1000 samples by questionnaire and physical test. The dataset was divided into two categories of training set (30%) and test set (70%). The experiments demonstrate the MSE of our method achieves as small as 26.637 on the training set and 32.334 on the test set. Moreover, the training time only costs 1.563s.

Keywords: Sports performance, Genetic algorithm, Prediction, Artificial Neural Network

1. Introduction

A prediction or forecast is a statement about the way things will happen in the future [1]. Sports performance prediction is determined by many factors, such as the morale of a player and a team, skill and the training strategy and so on [2]. The performance prediction is related to many people, including the coach, fans and the athletes. It also raises interesting to the researchers. We explore the use of advanced non-linear modeling techniques such as neural networks [3] to provide an analysis in such decision situations.

Stekler et al. [4] found that a large quantity of effort was spent on predicting the results of sporting events, but researchers focused only on the characteristics of sports forecasts. On the contrary, many researchers paid attention to the efficiency of sports betting markets. As it turned out, it was possible to derive a considerable amount of information about the forecasts and the forecasting process from studies that had tested the markets for economic efficiency. Moreover, the huge number of observations provided by betting markets made it possible to obtain robust tests of various forecasting hypotheses. Their paper was concerned with a number of predicting topics in horse racing and several team sports. The first topic involved the type of forecast that is made: picking a winner or forecasting whether a particular team would beat the point spread. Different evaluation procedures would be tested and alternative predicting methods (models, experts, and the market) compared. Iyer et al. [5] supposed that the selection for international sports competitions required forecasting performance of individual athletes. They employed the neural networks to rate players and select specific players for a competition. Cricket as an example, they employed neural networks to predict each cricketer’s performance in the future based upon their past performance. They classified cricketers into three categories which are performer, moderate and failure. They collected data on cumulative player performance from 1985 onwards until the 2006-2007 season. The neural network models were
progressively trained and tested using four sets of data. The trained neural network models were then applied to generate a forecast of the cricketer’s near term performance. Min et al. [6] proposed a framework for sports prediction based on Bayesian inference and rule-based reasoning, together with an in-game time-series approach. The framework consisted of two major components: a rule-based reasoner and a Bayesian network component. The two different approaches cooperated in forecasting the results of sports matches. It was motivated by the observation that sports matches are highly stochastic, but meanwhile, the strategies of a team could be approximated by crisp logic rules. Plessner et al [7] presented a social-cognitive overview of empirical work on judging sport performance. It followed the basic steps of social information processing such as perception, encoding/categorization, memory processes, and information integration. The application of a social cognition approach provided insights into the processes that underlie biases in judgments of sport performance and, thus, some hints on how to prevent them. In addition, they proposed possible future applications of social cognition concepts in sports judgment research.

In this paper, we proposed a method based on genetic algorithm and BP network model to forecasting the sports performance. The paper is organized as following: section 2 will describe the basic back-propagation Neural Network, Section 3 explains the algorithm employed in this paper, section 4 will demonstrate the experiment, Section 5 is the conclusion and our future work.

2. The Artificial Neural Network

The BP is a type of supervised learning neural network. A general model of the BP has a structure as shown in Figure 1.

![Figure 1. The architecture of an Artificial Neural Network](image)

In Figure 1 we find that there are three layers exist in BP: input layer, hidden layer, and output layer. Two nodes of each adjacent layer are directly connected called as a link [8]. Each link has a weighted
value as the relation degree between two nodes [9]. Assume that there are n input neurons, m hidden neurons, and one output neuron [10], we can infer a training process described by the following equations to update these weighted values, which can be divided into two steps [11]:

1) Hidden layer stage: The outputs of all neurons in the hidden layer are calculated by following steps:

\[
net_j = \sum_{i=0}^{n} v_{ji}x_i \quad j = 1, 2, \cdots, m \tag{1}
\]

\[
y_j = f_H(\text{net}_j) \quad j = 1, 2, \cdots, m \tag{2}
\]

Here \( \text{net}_j \) is the activation value of the \( j \)th node, \( y_j \) is the output of the hidden layer, and \( f_H \) is called the activation function of a node, usually a sigmoid function as follow:

\[
f_H(x) = \frac{1}{1+\exp(-x)} \tag{3}
\]

2) Output Stage: The outputs of all neurons in the output layer are given as follows:

\[
O = f_O(\sum_{j=0}^{m} \omega_{oj}y_j) \tag{4}
\]

Here \( f_O \) is the activation function, usually a line function [12]. All weights are assigned with random values initially, and are modified by the delta rule according to the learning samples traditionally.

3. Introduction of Genetic Algorithm

In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA) [13], which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

GAs are powerful stochastic search techniques based on the principle of natural evolution [14]. The individuals of GA are encoded in the form of strings [15]. A collection of such string is called a population. Initially, a random population is created, which represents different points in the search space [16]. An objective function is associated with each string that represents the degree of the goodness of the string. Based on the principle of survival of the fittest [17], a few of the string are selected and each is assigned a numbers of copies that go into the mating pool. Crossover and mutation operators are applied on these strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is met [18]. Figure 2 shows the flowchart of GA.
3.1. Initialization

In the beginning, a lot of individual solutions are randomly generated to form an initial population. The population size is decided by the nature of the problem, but usually contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, allowing the entire range of possible solutions.

3.2. Selection

During each successive generation, a proportion of the existing population is selected to generate a new generation. Individuals are selected via a fitness-based process, during which the solutions with high fitness value are more likely to be chosen [19]. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process consumes a lot of time [20].

3.3. Crossover and Mutation

After the selection process, it is time to generate the second generation population of solutions from those selected through genetic operators: crossover and mutation [21].

For each new solution to be produced, a pair of "parent" solutions is chosen for generating from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which shares many of the characteristics of its parents. New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated [22].

These processes finally make the next generation population of chromosomes different from the initial generation. Generally the average fitness will be increased via this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

3.4. Termination

This generational process is repeated until a termination condition has been reached as shown in Figure 2. Usually the conditions are:

- A solution is found that satisfies minimum criteria
  - Fixed number of generations reached;
  - Allocated budget such as the computation time reached;
  - The highest ranking solution's fitness is reached or has reached a plateau such that successive iterations no longer produce better results;
  - Manual inspection;
  - All of the above.

GAs are very different from classical optimization techniques such as gradient-based algorithm in
the following points [23]: 1) GAs make use of the encoding of the parameters not the parameters themselves; 2) GAs work on a population of points not a single one; 3) GAs only use the values of the objective function not their derivatives or other auxiliary knowledge; 4) GAs use probabilistic transition functions and not deterministic ones [24].

3.5. Genetic algorithm Neural Network

In order to employ the genetic algorithm to train the neural network, the fitness function is shown in equation (5)

\[ f(w) = \frac{\sum_{i=1}^{n} (P_i(w) - O_i)^2}{n} \] (5)

In which \( P_i(w) \) and \( O_i \) means the predicted performance and original score of one specific athlete.

4. Experiment

The experiments are carried on the Windows XP operation system with 2GB Hz processor and 1GB memory. We also developed an in-house GUI as friendly interface between human and computers. The GUI can run at any computer with Matlab.

4.1. Database

For the sports performance, the features used for the experiment is shown in Table 1. We select fatigue, weather, experience, training time, weight, height and nutrition facts of the athletes as the
criteria of the prediction [25]. The performance of the athletes are scored within [60,100] [26].

Table 1. Features used for the prediction

<table>
<thead>
<tr>
<th>Feature</th>
<th>Grades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fatigue</td>
<td>Weak (A)</td>
</tr>
<tr>
<td></td>
<td>Normal (B)</td>
</tr>
<tr>
<td></td>
<td>Strong (C)</td>
</tr>
<tr>
<td>weather</td>
<td>Excellent (A)</td>
</tr>
<tr>
<td></td>
<td>Normal (B)</td>
</tr>
<tr>
<td></td>
<td>Bad (C)</td>
</tr>
<tr>
<td>experience</td>
<td>Long (A)</td>
</tr>
<tr>
<td></td>
<td>Normal (B)</td>
</tr>
<tr>
<td></td>
<td>Short (C)</td>
</tr>
<tr>
<td>Training time</td>
<td>Long (A)</td>
</tr>
<tr>
<td></td>
<td>Average (B)</td>
</tr>
<tr>
<td></td>
<td>Short (C)</td>
</tr>
<tr>
<td>weight</td>
<td>Above (A)</td>
</tr>
<tr>
<td></td>
<td>Average (B)</td>
</tr>
<tr>
<td></td>
<td>Less (C)</td>
</tr>
<tr>
<td>Height</td>
<td>Tall (A)</td>
</tr>
<tr>
<td></td>
<td>Average (B)</td>
</tr>
<tr>
<td></td>
<td>Short (C)</td>
</tr>
<tr>
<td>Nutrition facts</td>
<td>Excellent (A)</td>
</tr>
<tr>
<td></td>
<td>Average (B)</td>
</tr>
<tr>
<td></td>
<td>Bad (C)</td>
</tr>
</tbody>
</table>

We collect the data via questionnaire and physical ability tests based on 1000 students randomly selected from Suzhou University of Science and Technology. We choose 30% for training which means that we have 300 training samples and 700 hundred test samples. Figure 3 and Figure 4 shows the performance score for 1000 samples and the histogram of the 1000 samples.

Figure 3. Original Performance Samples in the experiment
4.2. Experiment Result

We randomly select 300 hundred samples from the 1000 original data samples and the remaining 700 data samples will be used for testing. We employ the back-propagation neural network as the training model and genetic algorithm to optimize the weight values [27]. The prediction results compared with the original score are shown in Figure 5. The medium square error (MSE) between prediction score and original score of the training samples is 26.637 and is 32.334 of the test samples value. The training time of 300 data samples is 1.563 seconds.

5. Conclusion and Future work

We employ the BP neural and genetic algorithm for the sports performance prediction. From the experiment result, it demonstrates that we can approximately predict the value. Our future work should collect some real data from different school and different ages. Furthermore, we should optimize the algorithm to make the prediction more accurate. Finally, we should design a user-interface to the wide application.
Figure 5. Prediction result vs Original result

Reference


