Evolving Random Forest for Preference Learning

Abou-Zleikha, Mohamed; Shaker, Noor

Published in:
Applications of Evolutionary Computation

DOI (link to publication from Publisher):
10.1007/978-3-319-16549-3_26

Publication date:
2015

Document Version
Accepted author manuscript, peer reviewed version

Evolving Random Forest for Preference Learning

Mohamed Abou-Zleikha and Noor Shaker

Audio Analysis Lab, AD:MT, Aalborg University, Denmark
IT University of Copenhagen, Copenhagen, Denmark.
moa@create.aau.dk, nosh@itu.dk

Abstract. This paper introduces a novel approach for pairwise preference learning through a combination of an evolutionary method and random forest. Grammatical evolution is used to describe the structure of the trees in the Random Forest (RF) and to handle the process of evolution. Evolved random forests are evaluated based on their efficiency in predicting reported preferences. The combination of these two efficient methods for evolution and modelling yields a powerful technique for learning pairwise preferences. To test the proposed methodology and compare it to other methods in the literature, a dataset of 1560 sessions with detail information about user behaviour and their self-reported preferences while interacting with a game is used for training and evaluation. The method demonstrates ability to construct accurate models of user experience from preferences, behavioural and context data. The results obtained for predicting pairwise self-reports of users for the three emotional states *engagement*, *frustration* and *challenge* show very promising results that are comparable and in some cases superior to those obtained from state-of-the-art methods.

1 Introduction

Despite the recent advancement in methodologies for capturing users' affects, self-reporting is still widely used due to its reliability, directness and being less intrusive than other objective-based methods for collecting affects. Pairwise preferences is a popular method to collect self-reports [6, 12]. Constructing user models from pairwise preferences is not an easy task due to the scarcity of machine learning and dataminig techniques that can be directly applied to preference data. Most of the methods employed deal with this problem as learning a global ranking function [8]. This, however, usually leads to less accurate models than those trained to learn pairwise preferences directly. This is mainly because of the nature of the problem which limits the possibility of accurately transforming preference data into rankings due to the relational structures of the data according to which ranking between subjects may not be commensurate (i.e. the existence of pairwise preference data points of the form: instance A is preferred over B, instance B is preferred over C and C is preferred over A).

Up until now, Neuroevolutionary Preference Learning (NPL) [24] has shown the best results in learning preference data for cognitive modelling. In a comparison study, this method gave more accurate results than those obtained by other methods [24]. This method has therefore been extensively used in several studies where preference data is available [24, 13, 21, 20]. These studies demonstrated that reported preferences can be

predicted with high accuracies from information about the interaction between the user and the system as input and their reported affects as output using NPL.

In this paper, we propose a novel evolutionary-based method for learning pairwise preferences. Few other machine learning methods have been tested for this purpose, the results obtained by NPL, however, are mostly significantly better. The method proposed in this paper achieves higher results than those obtained by NLP in similar experimental setting. Furthermore, most of the proposed methods suffer from a limited expressive power in terms of the difficulty in interpreting the models and analysing the input-output relationship, a drawback that could be overcome using small RFs. Finally, constructing models of pairwise preferences through evolving random forest is an interesting problem that has not yet been explored yet.

The method we propose in this paper is novel (1) in terms of the technique used; we are not aware of any previous attempts for evolving random forest to learn a specific problem, and (2) in utilising the proposed approach for learning pairwise preferences. For comparison purposes, we are using the same dataset collected from players playing Super Mario Bros (SMB) that has already been used in several studies for modelling player experience from preference data using the NPL framework [20].

2 Preference Learning

Preference learning has received increasing attention in the machine learning literature in recent years [8]. The ranking problem has been categorised into three main types, namely label ranking, instance ranking and object ranking [8]. We focus on object ranking in this paper. Within object ranking, the goal is to learn a ranking function $f(.)$ that produces a ranking of a given subset of objects given their pairwise preferences. More formally, given a set of instances Z and a finite set of pairwise preferences $x_i \succ x_j; (x_i, x_j) \in Z$ x Z, find a ranking function $f(.)$ that returns the ranking of this set Z. Here, $x_i \succ x_j$ means that the instance x_i is preferred to x_j.

Various methods have been presented in the literature for the task of object ranking. Methods based on large-margin classifiers [7, 10, 1], Gaussian processes [9, 3], and neuroevolution [24] have been investigated to learn the ranking function. Neuroevolutionary preference leaning proved to have a powerful approximation capability and to build efficient models of player experience in similar setups to the one at hand [18, 17, 13, 24]. Other supervised learning methods such as standard backpropagation [23], rank support vector machine [11], Cohens method [4], linear preference learning [19] and pairwise preference leaning [8] have also been employed to learn pairwise preferences with various success. There exists a number of other attempts where the problem of pairwise preference learning is converted into learning a global classifier and therefore standard ranking method can not be applied [8]. This paper introduces a new paradigm for learning pairwise preferences and presents a test case where the suggested method demonstrated efficient learning capabilities.

3 Random Forest

A random forest is a tree-based non-parametric classification and regression approach [2]. The principle is to grow a number of trees on a random selection of samples in a training set. Each tree is a non-pruned decision tree. During the construction of each tree and when adding a new node, a randomly selected feature subset is chosen from the set of input features. The features in this subset are then investigated and the one with the best splitting results is chosen.

When the random forest is used for a classification task, the trees are treated individually and each of them is processed to predict the target class. The final classification result is then calculated as the majority vote of the predictions obtained by the individual trees. For a regression task, each tree predicts one target value and the forest prediction output is then calculated as the average of the predicted values of all its trees.

In this work, and since we are learning pairwise preferences, the random forest model is used as a regression model.

Compared to a single decision tree, the random forest assembles several trees that are trained on randomly selected subsets of the data. Because of the use of this method for training, random forest usually demonstrates better generalisation capabilities and higher degree of stability as well as achieving better performance than other classification and regression methods [5].

Formally, a random forest, RF, is a set of decision trees

$$RF = \{t_1, t_2, ..., t_T\} \tag{1}$$

where t_i is the i^{th} individual tree and T is the total number of trees in a forest.

Each tree, t_i in the forest is trained independently given a bagged version of the training data. Given a set of input features $F = f_1...f_n$, the j^{th} node is split using the feature that maximises the information gain:

$$f_j = \arg\max_{f \in F_j} I(X_j, f) \tag{2}$$

where F_j is a randomly selected feature subset of F at node j, X_j is the data at node j and $I(X_j, f)$ is the information gain function.

One of the implementation used for the information gain is a function that aims at decreasing the impurity of the split data. For a regression task, impurity is defined as the mean squared prediction error between the predicted and the actual value.

Unlike a single decision tree, no pruning is applied, and as a result each tree grows until either a maximum depth is reached or the information gain becomes smaller than a predefined threshold.

4 Grammatical Evolution

Grammatical Evolution (GE) is an evolutionary algorithm based on Grammatical Programming (GP) [16]. The main difference between GE and GP is the genome representation; while a tree-based structure is used in GP, GE relies on a linear genome representation. Similar to general Genetic Algorithms (GAs), GE applies fitness calculations for every individual and it applies genetic operators to produce the next generation.

The population of the evolutionary algorithm is initialised randomly consisting of variable-length integer vectors; the syntax of possible solution is specified through a context-free grammar. GE uses the grammar to guide the construction of the phenotype output. The context-free grammar employed by GE is usually written in Backus Naur Form (BNF). Because of the use of a grammar, GE is capable of generating anything that can be described as a set of rules. GE is used previously to evolve single decision trees to solve a classification problem [14]. In this paper, we focus on the problem of preference learning and we evolve a random forest to learn this function.

Each chromosome in GE is a vector of codons. Each codon is an integer number used to select a production rule from the BNF grammar in the genotype-to-phenotype mapping. A complete program is generated by selecting production rules from the grammar until all non-terminal rules are mapped. The resultant string is evaluated according to a fitness function to give a score to the genome.

In this paper, a design grammar is defined to specify the structure of possible solutions to our problem (trees in our case). The grammar is then employed by GE to evolve a random forest according to a predefined fitness function. GE is employed because it combines the advantages of an evolutionary algorithm and due to the simple nature of the design grammar that allows and easy way of defining, interpreting and manipulating the structure of the solutions.

5 Evolving Random Forest

Standard implementations of RF are not applicable to solve the problem of preference learning since for any two given instances A and B in a pair, there is no specific target output to calculate the information gain at each node and to perform data splitting. This form of constrained-classification problems can only be solved using preference learning methods [8]. In contrast of the incremental approach that is usually used for building the trees, a holistic approach is proposed. In this approach, we build the whole trees randomly, then a set of modifications is performed on the forest trees targeting to improve the performance of the generated model. In our framework, we utilise grammatical evolution to evolve a forest so that for each pair in the pairwise dataset, the prediction value of a preferred sample is higher than that of the non-preferred one, i.e. $f(x_i) > f(x_j)$ if $x_i > x_j$ and $f(x_i) < f(x_j)$ if $x_i < x_j$. A Design Grammar (DG) is defined to describe the structure of possible solution (a tree) in the evolution process. The system is trained and tested using a dataset consisting of a set of pairs, each of which is associated with a preference value assigned by at least one user. Each of the evolved trees is evaluated using a fitness function that measures the number of correctly classified pairs, i.e. the number of pairs in which the output produced by the method matches the reported preference in the dataset.

5.1 Design Grammar

The DG is defined in a way that allows the construction of binary trees where each node is selected from one of the input features available. A simplified version of the DG is presented in Fig. 1. According to the grammar, the tree starts with an *internal*

node which has a condition and two child nodes. Each of the child node can be either a leaf or another internal node. The condition in an internal node is a splitting condition based on one of the input features and a threshold. The leaves return a regression value between 0 and 100.

```
<Tree> := <Internal_Node>
<Internal_Node> :=  <Condition> <Child_Node>
                    <Child_Node>
<Child_Node> := <Leaf>  | <Internal_Node>
<Condition>:= <Feature> <Threshold>
<Feature> := feature1 |  feature2 | ...
<Leaf> := <Regression_value>
<Threshold> := [0,1]
<Regression_value> := [0,100]
```

Fig. 1. A simplified version of the grammar that specifies the structure of the trees in a forest.

5.2 The Evolution Process

GE is used to evolve RFs given the design grammar that defines the structure of the trees. The evolution process implemented can be described as follows: an population of N tree is initialised randomly according to the grammar. During the evolution process and at each generation, the following process is repeated:

Step 1 Each member (a tree), t, of the population gets a bagged version of the dataset for evaluating its prediction accuracy.
The input data is organised in d-tuples (where d is the number of input features). The data is then presented to each tree in the population in pairs: A and B for the preferred and non-preferred instances, respectively. Each tree outputs a real regression value for each instance presented, namely $y_{j.A}$ and $y_{j.B}$ for the pair j. If the output of the tree for the pair matches that in the dataset, i.e. there is a consistency between the sign of $y_{j.A} - y_{j.B}$ and the actual reported pairwise preference for the pair j, then we state that there is an 'agreement' between the output and the reported preferences. In the opposite case, we state that there is a 'disagreement'.

Step 2 Each tree t in the population is evaluated via a fitness function f_t that calculates the number of instances with an 'agreement' between the output and the reported preferences.

Step 3 The population is ranked and the best n trees are chosen.

Step 4 A roulette-wheel selection scheme is used as the selection method.

Step 5 The Montana and Davis crossover is applied to selected parents for generating two offspring. Gaussian mutation occurs in each gene of each offspring's genome.

Step 6 The performance of the best n trees is checked on a validation dataset. These trees replace the old forest if the performance obtained is better.

The algorithm terminates when a predefined total number of generations is reached. The best n trees obtained in the final population are then evaluated on a testing set and the performance obtained is reported as the modelling performance.

6 Case Study: Player Data in Super Mario Bros

The dataset used for our experiments consists of rich information about game content, player behaviour, and self-reports of hundreds of players playing a modified version of an open source clone of the popular game *Super Mario Bros*.

We implemented a content generator for creating variations of levels. We further designed a post-experience game survey to collect subjective affective reports expressed as pairwise preferences of subjects playing two levels of the game following the 4-alternative forced choice experimental protocol proposed in [24]. Data from gameplay and questionnaires was collected from players over the Internet via a crowd-sourcing experiment. Complete games were logged enabling complete replays. The following three types of data were extracted from raw logs and replays: content, gameplay and annotated (self-reported) player experience of the three emotional states: *engagement*, *frustration* and *challenge*. For a complete detailed list of the different features extracted the reader may refer to [20].

One of the primary reasons for choosing the Mario dataset is because it offers rich information in terms of the features collected (30 different gameplay and content features). The other reason is that it contains information about players' pairwise self reports of three different emotional states permitting a thorough analysis of the capabilities of the modelling approach.

Evolutionary algorithms are well known for their efficiency in handling large input spaces and decision trees are popular as powerful classifiers and therefore evolving them for the task at hand promises models of high accuracies. However, due to the complex nature of self-reporting in general (being noisy because of their subjectivity) and the difficulties in accurately modelling pairwise preferences [24], the problem still introduces interesting challenges and constitutes an interesting direction for research.

7 Experimental Setup

A dataset of 1560 gameplay sessions (780 pairs) is used for training, validation and testing. The dataset was preprocessed to remove the instances where players show no clear preferences (the answers to the questionnaires were either that both games were equally liked or disliked). The sizes of the resultant datasets are 597, 531 and 629 pairs, for engagement, frustration and challenge, respectively. These datasets were split into 70% for training and 15% for validation and testing. Significance analysis is performed using t-test.

The existing GEVA software [15] was used as a core to implement the needed functionalities. The experimental parameters used are the following: 50 runs each ran for 100 generations with a population size of 3000 individuals, the ramped half-and-half

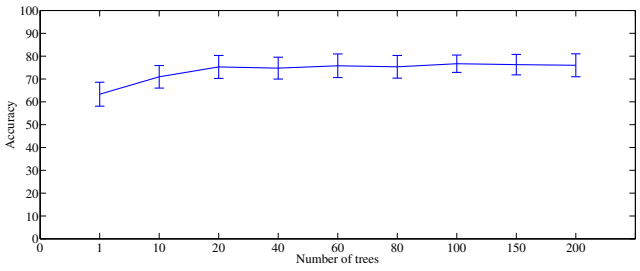

Fig. 2. The averages and standard deviations of the accuracies obtained from 50 forests of different number of trees for predicting frustration.

initialisation method. Tournament selection of size 30, int-flip mutation with probability 0.3, one-point crossover with probability 0.5, and 0 maximum wraps were allowed. All parameters have been experimentally tuned.

8 Analysis

In the following sections we describe a number of analysis conducted to test the effect of different configurations on the modelling performance. We investigate factors such as the depth and the number of trees and we provide a preliminary analysis of the expressive power of the models.

8.1 Number of Trees

We ran several experiments to investigate the best number of trees to form a forest. Fig. 2 presents the average accuracies obtained for predicting reported frustration from 50 runs of the algorithms with different number of trees ranging between 1 and 200. The results show that the accuracy increases as we add more trees to the forest up to the point where the forest consist of 100 trees after which a slight drop in the performance is observed. The significance test showed statistical better performance for the models with number of trees > 60 compared to those with a smaller number of trees (with no significant difference between the models' performance in each group). Given these results, the rest of this paper focuses only on analysis of models consisting of 100 trees as those yield the highest accuracies and provide a reasonable trade-off between the size and the performance of the models. Note that we accept with generalisation that this setup will also give good results for the other affective states while we acknowledge that it may not be the best configurations.

8.2 Depth of Trees

In order to investigate the effect of the depth of the evolved trees in a forest on the modelling performance, we run two experiments with forests of 100 trees where the

maximum depth of trees is 10 or 20. The experiment is repeated for 50 times for each of the three emotional states and the average accuracies obtained are presented in Table 1. The statistical analysis showed no significant difference between the accuracies obtained by the two configurations for the three emotional states ($p - value > 0.05$). Therefore the analysis for the rest of the paper will focus on evolving models with tree depth equals to 10 as those are easier to analyse, faster to evolve, less complex and yield comparable accuracies to those of more sophisticated models.

Table 1. The averages and standard deviations of the accuracies obtained from 50 forests of different tree depths for predicting engagement, frustration and challenge.

Tree depth	Engagement	Frustration	Challenge
10	$63.01\% \pm 3.96$	$77.68\% \pm 4.70$	$72.66\% \pm 3.79$
20	$64.83\% \pm 3.79$	$76.66\% \pm 5.02$	$74.41\% \pm 4.48$

8.3 Expressive Power

One of the advantages of evolving decision trees is that they demonstrate powerful classification ability while preserving the expressive power being easy to interpret. In order to investigate the expressivity of the proposed approach, to emphasise the need for RF and for comparison purposes, we will discuss models of only one tree and compare them to more complex models of 100 trees. One-tree models are chosen since they are the easiest to interpret.

The results obtained from the comparison for predicting frustration can be seen in Fig. 2. The performance obtained for the best evolved tree, although relatively high (the average performance obtained is 63.34%), is significantly lower than the one achieved by100-tree forests ($p - value = 2.88 * 10^{-26}$). The results suggest that there is a trade-off between the models' expressivity and performance. In the next section, we present a methodology implemented to facilitate analysis of expressivity of forests of more than one tree thus preserving the performance while permitting high-level understanding.

9 Feature Importance

In order to understand the constructed models, we performed a preliminary analysis that helps shedding some light on the features important for predicting each emotional state. This is done by calculating the number of occurrences of each of the 30 input features in the forests constructed. The experiment is conducted with forests of 100 trees of depth 10 and repeated 50 times. Fig. 3 presents the average number of times each feature is presented in each forest constructed. As can be seen, some features appear more often than others, and those are considered more important for predicting a particular affective state. It is interesting to note that the importance of features differs along the emotional state. For example, some features, such as 23 and 30 are more important for predicting frustration than engagement.

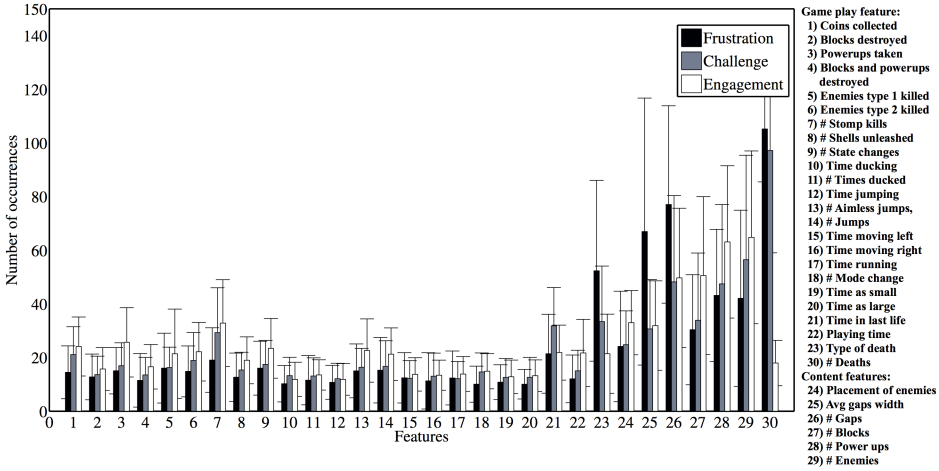

Fig. 3. The averages and standard deviations of the number of occurrences of the gameplay and content features in 50 forests evolved for the three emotional states.

In general, it appears that the last eight features are the most important for predicting frustration. Most of these features (24-29) are related to content aspects while feature 30 stores the number of times the player died in the game and feature 23 stores the reason of the death (killed by an enemy or fell in a gap).

The same set of features is also important for predicting reported preferences of challenge along with the two features 7 and 21 which stand for the number of enemies the player killed and the amount of time spent during the last trail (note that the player is given three lives to play), respectively.

The subset of features for predicting engagement consists of features with less number of occurrences than those of frustration and challenge. The set includes the feature 7 and features 24-29.

It is worth noting that most content features (24-29) are important for predicting the three states and that the number of player's death in the game is of significant importance for predicting frustration and challenge. This indicates that elicited affective states can be predicted up to a good degree from information about the design of the game and that there are key gameplay features that are of significant prediction power.

10 Comparison and Analysis

According to the results of our previous analysis, models with high accuracies can be evolved for predicting the three cognitive states using the proposed methodology. The averages and best accuracies obtained from 50 independent runs to evolve 50 forests of 100 trees of depth 10 for the three emotional states are presented in Table 2.

The best performance obtained is from the models for predicting frustration with accuracy up to 86% that is significantly higher ($p - value = 3.60 * 10^{-32}$) than the one obtained from the models for predicting pairwise preferences of engagement, which

Table 2. Averages and best accuracies obtained from 50 runs of the experiments for evolving 50 random forests for the three emotional states.

	Engagement	Frustration	Challenge
$Average$	63.01%	77.68%	72.66%
$Maximum$	73.03%	86.08%	85.11%

Table 3. Averages and best accuracies obtained from 20 runs of the experiments for modelling player experience using neuroevoltionary preference learning for the three emotional states [20].

	Engagement	Frustration	Challenge
$Average$	67.18%	76.50%	74.03%
$Maximum$	73.50%	83.00%	79.10%

have a best accuracy of 73.03%. The frustration models also significantly out perform the models evolved to predict reported challenge preferences ($p-value = 8,05*10^{-6}$). The results indicate that reported pairwise preferences of engagement are the hardest to predict and that the evolved models for predicting reported frustration and challenge are of significant higher performance ($p - value = 4.78 * 10^{-19}$ between the performance of engagement and challenge models).

As discussed previously, one of the main reasons for choosing the Mario dataset is that it has been previously used to construct models of player experience primarily using NPL [20, 22]. In this paper, and to provide a fair comparison between NPL and the proposed approach, we compare the accuracies obtained by our evolved RF models and those obtained by NPL on the same Mario dataset using the same set of input features (a set of 30 statistical features capturing differences in content and players' behaviour as frequencies of events and actions). The averages and the best accuracies obtained in the earlier work by NPL are presented in Table 3 [20].

The comparison of the results obtained shows that reported frustration appears to be the easier to predict by the two approaches while engagement is the hardest. The analysis of the results (Table 2 and 3) shows that our proposed method outperforms NPL for predicting frustration and challenge while comparable results are observed for predicting reported engagement.

The comparison presented confirms the efficiency of the proposed approach and demonstrates its ability to outperform NPL which has so far yield the most accurate results for predicting pairwise preferences in the same and similar problems [20, 13, 24].

11 Conclusions and Future Work

In this paper we present a novel approach for learning pairwise preferences through evolving a random forest. Grammatical evolution is used to evolve trees in a forest where each model is evaluated based on its efficiency when classifying preference data. The method proposed is tested on a big dataset of players' data where each instance of content and player behaviour is annotated with subjects' preferences. We investigated

several setups and analysed the effect of the number of trees and their depth on the modelling accuracy. We further presented a preliminary analysis of the expressive power of our modelling approach. The comparison of the results with those obtained from a state-of-the-art method shows that comparable, and in some cases superior, accuracies can be achieved using the proposed approach.

The analysis of the expressivity of the models presented in this paper constitutes the first step towards more in-depth investigations. For example, in a dataset similar to the one used in this paper, the analysis the relationship between each input feature and reported emotional states is of utmost interest for game analyst and designers. Therefore, future work will include conducting more experiments to improve the readability of the models. One possibility is to use smaller number of trees which can be converted to a set of human-readable rules. Another essential factor when analysing the importance of features that we did not consider in this paper is their depth in the tree. Future direction will include accounting for this factor.

Another important direction, which is the ultimate goal of user experience modelling, is to utilise the constructed models as a ranking function of content given particular user behaviour. The best piece of content can then be used to generate user-adapted experience.

12 Acknowledgement

The research was supported in part by the Danish Research Agency, Ministry of Science, Technology and Innovation; project "PlayGALe" (1337-00172). This work also was supported in part by the Danish Council for Strategic Research of the Danish Agency for Science Technology and Innovation under the CoSound project, case number 11-115328. This publication only reflects the authors views. The authors would like to thank Prof. Georgios Yannakakis and Dr. Héctor P. Martinez for valuable discussions.

References

1. Bahamonde, A., Bayón, G., Díez, J., Quevedo, J., Luaces, O., Del Coz, J., Alonso, J., Goyache, F.: Feature subset selection for learning preferences: a case study. In: Proceedings of the twenty-first international conference on Machine learning. p. 7. ACM (2004)
2. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
3. Chu, W., Ghahramani, Z.: Preference learning with gaussian processes. In: Proceedings of the 22nd international conference on Machine learning. pp. 137–144. ACM (2005)
4. Cohen, W.W., Schapire, R.E., Singer, Y., et al.: Learning to order things. J Artif Intell Res 10, 243–270 (1999)
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Microsoft Research Cambridge, Tech. Rep. MSRTR-2011-114 p. 12 (2011)
6. Doyle, J.: Prospects for preferences. Computational Intelligence 20(2), 111–136 (2004)
7. Fiechter, C., Rogers, S.: Learning subjective functions with large margins. In: Stanford University. pp. 287–294. Morgan Kaufmann Publishers (2000)
8. Fürnkranz, J., Hüllermeier, E.: Preference learning. Springer-Verlag New York Inc (2010)

9. Gervasio, M., Moffitt, M., Pollack, M., Taylor, J., Uribe, T.: Active preference learning for personalized calendar scheduling assistance. In: Proceedings of the 10th international conference on Intelligent user interfaces. vol. 5, pp. 90–97. Citeseer (2005)
10. Herbrich, R., Graepel, T., Bollmann-Sdorra, P., Obermayer, K.: Learning preference relations for information retrieval. In: ICML-98 Workshop: text categorization and machine learning. pp. 80–84 (1998)
11. Herbrich, R., Graepel, T., Obermayer, K.: Support vector learning for ordinal regression. In: Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470). vol. 1, pp. 97–102. IET (1999)
12. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 133–142. ACM (2002)
13. Martinez, H., Jhala, A., Yannakakis, G.: Analyzing the impact of camera viewpoint on player psychophysiology. In: International Conference on Affective Computing and Intelligent Interaction and Workshops. pp. 1–6. IEEE (2009)
14. Motsinger-Reif, A.A., Deodhar, S., Winham, S.J., Hardison, N.E.: Grammatical evolution decision trees for detecting gene-gene interactions. BioData mining 3(1), 1–15 (2010)
15. O'Neill, M., Hemberg, E., Gilligan, C., Bartley, E., McDermott, J., Brabazon, A.: Geva: grammatical evolution in java. ACM SIGEVOlution 3(2), 17–22 (2008)
16. O'Neill, M., Ryan, C.: Grammatical evolution. IEEE Transactions on Evolutionary Computation 5(4), 349–358 (2001)
17. Pedersen, C., Togelius, J., Yannakakis, G.N.: Modeling player experience in super mario bros. In: CIG'09: Proceedings of the 5th international conference on Computational Intelligence and Games. pp. 132–139. IEEE Press, Piscataway, NJ, USA (2009)
18. Pedersen, C., Togelius, J., Yannakakis, G.N.: Modeling player experience for content creation. IEEE Transactions on Computational Intelligence and AI in Games 2(1), 54–67 (2010)
19. Runarsson, T., Lucas, S.: Imitating play from game trajectories: Temporal difference learning versus preference learning. In: IEEE Conference on Computational Intelligence and Games (CIG). pp. 79 –82 (2012)
20. Shaker, N., Yannakakis, G.N., Togelius, J.: Crowd-sourcing the aesthetics of platform games. IEEE Transactions on Computational Intelligence and Games, Special Issue on Computational Aesthetics in Games, (to appear) (2013)
21. Shaker, N., Shaker, M., Abuabdallah, I., Zonjy, M., Sarhan, M.H.: A quantitative approach for modeling and personalizing player experience in first-person shooter games
22. Shaker, N., Togelius, J., Yannakakis, G.N.: Towards automatic personalized content generation for platform games. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE). AAAI Press (2010)
23. Tesauro, G.: Connectionist learning of expert preferences by comparison training. In: Advances in neural information processing systems 1. pp. 99–106. Morgan Kaufmann Publishers Inc. (1989)
24. Yannakakis, G.N., Maragoudakis, M., Hallam, J.: Preference learning for cognitive modeling: a case study on entertainment preferences. IEEE Transactions on Systems, Man, and Cybernetics. Part A 39, 1165–1175 (November 2009)