Feature Analysis for Modeling Game Content Quality

Noor Shaker, Georgios N. Yannakakis, *Member, IEEE*, and Julian Togelius, *Member, IEEE*

***Abstract*— One promising avenue towards increasing player entertainment for individual game players is to tailor player experience in real-time via automatic game content generation. Modeling the relationship between game content and player preferences or affective states is an important step towards this type of game personalization. In this paper we analyse the relationship between level design parameters of platform games and player experience. We introduce a method to extract the most useful information about game content from short game sessions by investigating the size of game session that yields the highest accuracy in predicting players' preferences, and by defining the smallest game session size for which the model can still predict reported emotion with acceptable accuracy. Neuroevolutionary preference learning is used to approximate the function from game content to reported emotional preferences. The experiments are based on a modified version of the classic Super Mario Bros game. We investigate two types of features extracted from game levels; statistical level design parameters and extracted frequent sequences of level elements. Results indicate that decreasing the size of the feature window lowers prediction accuracy, and that the models built on selected features derived from the whole set of extracted features (combining the two types of features) outperforms other models constructed on partial information about game content.**

I. Introduction

In order to maximize the entertainment value of a game, we need accurate, reliable and computationally efficient models of what makes a game, or some aspect of a game, fun. (The same argument can be made for other affective properties than fun, or for e.g. pedagogical or instructional properties.) Many theories exist regarding why we play games and what makes computer games fun [1], [2], [3]. However, these theories are mostly qualitative and tend to apply to games in general rather than to specific aspects of games. This means we still have to make several auxiliary assumptions if we want to develop algorithms that design or adapt games automatically.

Until recently, optimization of game aspects based on empirically derived models has been focused on the impact of non player character (NPC) behavior [4] and the adjustment of NPC behavioral parameters for maximizing satisfaction in games [5]. The focus of most research has been on dynamic game balancing which aims to prevent players feeling frustrated because the game is too hard or becoming bored because the game is too easy.

A parallel research direction that has received increased attention recently is the automatic generation of game content. Procedural Content Generation (PCG) has been used to generate game content via algorithmic means with or without human designer interference. The classic example of the early use of PCG is the early eighties' game *Rogue*, a dungeon-crawling game in which levels are randomly generated every time a new game starts. However, only recently have approaches from artificial and computational intelligence begun to be explored in the context of creating central game elements such as levels and maps. A recent overview of some commonly used techniques can be found in [6], [7]. PCG can be used offline to generate complex content such as environments; making the game development process more efficient, and online, allowing the generation of endless variations of the game, making the game infinitely replayable and opening the possibility of generating player-adapted content [8]. The literature on personalized and player-adaptive PCG is so far scarce, as it is a new research direction [7]. A few attempts can be found on incorporating players' emotions into the game in a closed-loop manner where player's emotion is actively manipulated to ensure engagement [9]. Existing work [10], [11] demonstrates the power of using affective player models to generate in-game situations of high interest and satisfaction for the players. The reader may refer to [7] for a taxonomy and survey on experience-driven PCG (EDPCG).

A closely related, and partly overlapping, research direction has emerged recently focusing on adapting game content using computational models of player emotion built from the interaction between the player and the game [8], [12]. The very first step towards designing a player experience-centered adaptive game is to detect the player's emotional state and model its relation to game content.

We consider analysing the relationship between the player's emotional state and game content to be of utmost importance for making automatic content generation techniques more usable and for building better approaches for game adaptation. The focus and main contribution of this paper is the analysis of the interplay between game content and players reported preferences in platform games.

The approach proposed extends and draws upon earlier work on modeling player experience in *Super Mario Bros* [8], [13]. We extend this work through (1) expanding the space of level design parameters by investigating six controllable features of level design; (2) designing the experiment with a smaller game window size and collecting players' preference data for more variants of the game; (3) constructing the computational model of player experience based on a new, significantly larger data set of 600 human players, using the same methodology for modeling player experience as in [8], [13] but focusing on modeling the unknown function between players' preferences of experience and game content; (4) investigating sequence representations for level design features

NS, GNY and JT are with the Center for Computer Games Research at the IT University of Copenhagen (nosh@itu.dk, yannakakis@itu.dk, juto@itu.dk).

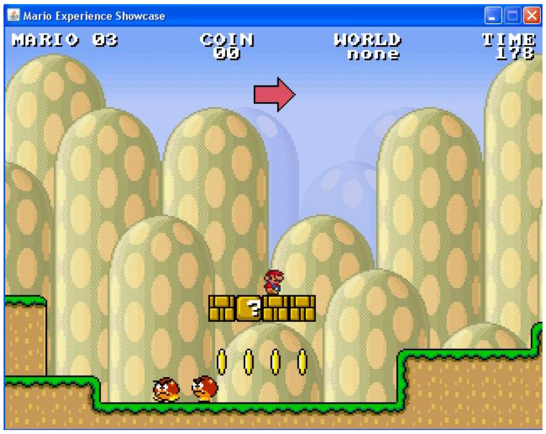


Fig. 1. Snapshot from Super Mario Bros game.

and (5) investigating the impact of the size of game session on the accuracy of predicting players' emotional state.

It should be noted that in this paper we are *not* concerned with the impact of playing style (as measured by player metrics) on entertainment value. Rather, we are investigating novel methods of finding out as much as possible about player preferences *from the game content only*, fully aware that this will lead to lower accuracy than would have been possible if player metrics were included as model inputs. In a future study, based on the same study dataset and building on the methodological findings reported in this paper, we will include this information.

The ultimate aim of the project which this study is part of, is to tailor player experience in real-time via automatic game content generation based on computational models of in-game player experience.

II. Testbed Platform Game

The testbed platform game used for our study is a modified version of Markus Persson's *Infinite Mario Bros* which is a public domain clone of Nintendo's classic platform game *Super Mario Bros*. The gameplay in Super Mario Bros consists of moving the player-controlled character, Mario, through two-dimensional levels. Mario can walk and run, duck, jump, and shoot fireballs. The main goal of each level is to get to the end of the level. Auxiliary goals include collecting as many coins as possible, and clearing the level as fast as possible. For more details about the game and our modifications the reader may refer to [14].

III. Data Collection

Before any modeling can take place, we need to collect data from players which will be used to train the model. For this purpose data from hundreds of players has been collected over the Internet. The following sections describe the types of data that has been used for the work done in this paper.

1) Controllable features of the game: These are used to generate the levels. These were varied to make sure several variants of the game are played and compared.
2) The player's reported experience of playing the game: The player experience is measured through a 4-alternative forced choice questionnaire presented to the player after playing a pair of games with different controllable features, asking the player to report the preferred game for three affective states; engagement, challenge and frustration.

Below we give a detailed description of the features collected.

A. Controllable Features

The level generator of the game has been modified to create levels according to the following six controllable features:

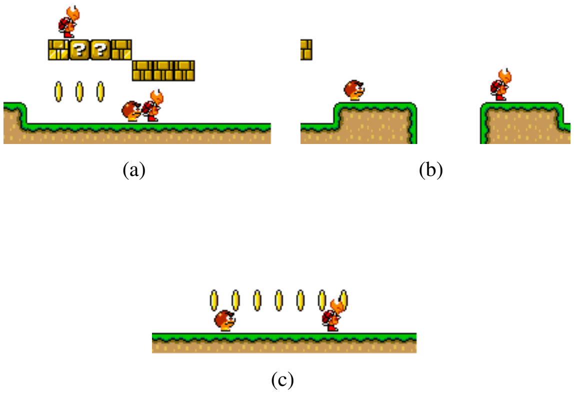
- The number of gaps in the level, G.
- The average width of gaps, $\bar{G}_w$.
- The number of enemies, E. This parameter controls the number of goompas and turtles scattered around the level, changing the level difficulty.
- Enemies placement. The way enemies is placed around the level determined by three probabilities which sum to one.
  - Around horizontal boxes, P_b: Enemies are placed on or under a set of horizontal blocks (a number of blocks placed horizontally without connection to the ground).
  - Around gaps, P_g: Enemies are placed within a close distance to the edge of a gap.
  - Random placement, P_r: Enemies are placed on a flat space on the ground.

  Fig. 2 illustrates positioned enemies by giving different values for P_b, P_g and P_r. Fig. 2.(a) shows enemies placed by setting P_b to 80%. Fig. 2.(b) illustrates the result of setting P_g to 80%, and Fig. 2.(c) is the result of $P_r = 80\%$.
- The number of powerups, R. Mario can collect powerup elements hidden in boxes to upgrade his state from little to big or from big to fire.
- The number of boxes, B. We define one variable to specify the number of the two different types of boxes that exist in Super Mario. We call these two types *blocks* and *rocks*. Blocks contain hidden elements such as coins or powerups. Rocks may hide a coin, a powerup or they can be empty. Mario can smash rocks only when he is in big mode.

The selection of these particular controllable features was done after consulting game design experts, and with the intent to cover the features that have the most impact on the investigated affective states. Please note that the two first features appeared in our previous studies [13], [8].

Two states (low and high) are set for each of the controllable parameters above except for enemies placement which has been assigned three different states allowing more control over the difficulty and diversity of the generated levels.

The total number of pairwise combinations of these states is 96. This number can be reduced to 40 by analysing the



(a)

(b)

(c)

Fig. 2. Enemies placement using different probabilities: high probability is given to placement around horizontal boxes, P_b (a), around gaps, P_g (b), and to random placement, P_r (c).

dependencies between these features and eliminating the combinations that contain independent variables.

Other features of the levels have been given fixed values such that the number of cannon and flower tubes = 1, the type of background = over ground, the number of coins = 7, the number of coins hidden in boxes = half the total number of boxes and the number of stairs around the gaps = half the number of gaps.

B. *Reported Player Experience*

We designed a game survey study to collect subjective affective reports expressed as pairwise preferences of subjects playing different variants (levels) of the test-bed game by following the experimental protocol proposed in [10]. According to the protocol, each subject plays a predefined set of two games. The games played differ in the levels of one or more of the six controllable features presented previously. After completing a session of two games, players are asked to report their emotional preferences for three emotional states; engagement, challenge and frustration, using a 4-alternative forced choice (4-AFC) protocol [14]. Note that the affective modeling procedure followed in this paper focuses only on reported engagement.

IV. Experimental Protocol

Data from Super Mario Bros players is collected over the Internet. A Java applet has been created and placed on a web page[1], which has been advertised over social networks, mailing lists and blogs. The applet is connected to an online SQL database that is used to collect data about game content, player's behaviour and reported experience.

The database initially contains all possible pairs marked as "unplayed". Whenever a game session starts, the software connects to the database and asks for an unplayed pair to load. Once two levels' are chosen from the database, they are loaded and the player is ready to play. When a session of two games is completed, the players are asked to report their preferences. The gameplay statistics and preferences are then stored to the database and the pair is marked as "played". The list of played pairs is reset if there are no more pairs available in the database to play (all pairs were marked as "played"),

The game sessions presented to players have been constructed using a level width of 100 Super Mario Bros units (blocks), about one-third of the size usually employed when generating levels for Super Mario Bros game in our previous experiments [13], [8]. The selection of this length was due to a compromise between a window size that is big enough to allow sufficient interaction between the player and the game to trigger the examined affective states and a window which is small enough to set an acceptable frequency of an adaptation mechanism applied in real-time aiming at closing the affective loop of the game [15].

As mentioned earlier, the combinations of the different states of the controllable features result in 40 different levels, The minimum number of experiment participants required so that each possible configuration is played at least once is determined by $C_2^{40} = 780$, this being the number of all combinations of 2 out of 40 levels. The analysis presented in this paper is based on the 600 game pairs that have been collected so far. The collected data has been preprocessed to remove the pairs with unclear preferences (those pairs where both games are equally preferred or not preferred for engagement) yielding 485 pairs with clear preferences for reported engagement.

The process of collecting data is still in progress, and once a substantial enough number of players has participated in the experiments we plan to go through the analysis of the effect of game content and playing characteristics on reported preferences based on the whole dataset.

V. Level Segmentation

The purpose of segmenting the level is to identify the size of the level segment that generates the best prediction accuracy of engagement and to determine the smallest possible segment for which the model can still predict reported engagement with acceptable accuracy. That segment size can then potentially be used to set the frequency of a real-time adaptation mechanism for the purpose of maximising the engagement value of the game (as in [5], [8]).

We start the process by calculating the models' performance over the entire game session. The level is then divided into two equal segments and the values for all controllable features for these two segments are recalculated. We then retrain the models presenting the two segments' controllable features as inputs (2 * 6 features). Each level is then further divided into three equal segments and the models are evaluated on individual segment and on combination of segments assuming that the expressed whole-game engagement preferences remain constant across those segments. No performance improvement has been obtained by further division of the level, and the focus of the remaining of this paper is on levels divided for up to three segments.

For the remaining of this paper we will use the term *window* to refer to the whole game session (a level with

[1] http://noorshaker.com/participate_in_experiments.htm

a width of 100), and the term *segment* to refer to parts of a window.

VI. Content-Driven Preference Learning

Based on the data collected in the process described above, we try to approximate the function from the controllable game level features (e.g. number of gaps) to reported emotional preferences using neuroevolutionary preference learning. We proceed in a bottom-up fashion, starting with a simple nonlinear models, then trying more complex models.

Learning is achieved through preference learning using artificial evolution of neural networks (neuroevolution) [16]. In one of the authors' previous work on preference learning algorithms, neuro-evolution has been found to be more effective than a number of other approaches including large margin classifiers and bayesian learning [10]. Multilayer perceptrons (MLPs) are utilized for learning the relation between the controllable features (ANN inputs) and the value of the engagement preference (ANN output) of a game. Since there are no prescribed target outputs a genetic algorithm (GA) was used to train the MLP using a fitness function that measures the difference between the players' reported emotional preferences and the relative magnitude of the corresponding model (ANN) output. More details of the method used can be found in [10].

Relying upon earlier successful parameter tuning experiments [10], [12], a population of 1000 individuals was used, and evolution run for 100 generations. A probabilistic rank-based selection scheme was used, with higher ranked individuals having higher probability of being chosen as parents. Finally, reproduction was performed by uniform crossover, followed by Gaussian mutation of 5% probability.

A. Optimizing Neural Networks Topologies

The experiment designed to optimize the topology of MLP affective models is as follows. We trained MLPs containing a maximum of two hidden layers. We start with a simple MLP topology of one hidden layer of two neurons, we then increase the number of neurons up to eight by adding two neurons at each step. Further, we investigate MLPs with two hidden layers, with up to ten and eight neurons in the first and second layer, respectively; again, the number of hidden neurons starts at two and increases by adding two neurons at each step; this sums to 25 different MLPs topologies which are tested for each input vector.

B. Neural Networks Input Representation

The ANN networks have been trained to predict players' preferences from game content. In the following sections we describe the two types of ANN input vectors that have been used to represent the content of the levels.

1) Controllable Features Statistics: For each segment the statistical values for the controllable features presented in section III-A have been calculated. All feature values are uniformly normalized to the range [0,1] using the standard max-min normalization.

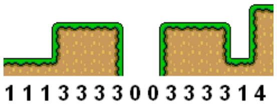


Fig. 3. Snapshot from a level and the corresponding platform structure sequence representation.

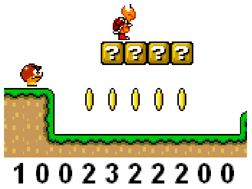


Fig. 4. Snapshot from a level and the corresponding enemies and decorations sequence representation.

All normalized values are included into the input of multilayer perceptron models of emotional preferences and MLPs' topologies are optimized for maximum prediction accuracy. The performance of MLPs is measured through the average classification accuracy of the model in three independent runs using 3-fold cross validation.

2) Sequences as Features: A modified version of the SPADE algorithm [17] has been implemented to extract frequent subsequences of different game content from all levels.

The content of the levels has been converted into numbers representing different types of game content. Three different representations of game content have been investigated.

- Platform structure, S: A sequence of integer numbers that represents the height of the ground along the level. Fig. 3 presents part of a level and the corresponding platform structure sequence representation.
- Enemies placement, E_p: A bit-string sequence that represents the initial placement of enemies along the level has been generated for each level. A boolean variable is used to represent the existence (1) or non-existence (2) of enemies.
- Enemies and items placement, D: The term items refers to the coins and the different types of boxes scattered around the level. The existence and non-existence states for enemies and items have been combined together resulting in four different possible values 0, 1, 2 and 3 corresponding, respectively, to non-existence of either enemies or items, the existence of an enemy, the existence of an item, and existence of an enemy and an item. Fig. 4 illustrates an example level segment where the above-mentioned four states are presented.

Different subsequence lengths and minimum support thresholds (i.e. the minimum number of times the subsequence must occur in the data to be counted as frequent) values have been explored. All subsequences used in the experiments of this paper are of length 3 and have been extracted using a minimum support threshold of 20, meaning

TABLE I

THE NUMBER OF FREQUENT SUBSEQUENCES OF LENGTH THREE EXTRACTED FROM THE LEVELS USING A MINIMUM SUPPORT OF 20

Sequence	# frequent subsequences
S	35
E_p	7
D	12

TABLE II

A SUBSET OF THE FREQUENT SUBSEQUENCES OF LENGTH THREE OF D AND THE CORRESPONDING OCCURRENCES OF EACH OF THEM IN ONE EXAMPLE LEVEL

Frequent subsequences	#of occurrence
000,020,022,100,200,220,222,232,322	80,0,1,2,2,2,3,0,0

that each subsequence should occur at least in half of the levels to be considered frequent. Table I presents the number of frequent subsequences of length 3 that have been found in the 40 levels for the three types of sequences.

The number of occurrences of each of the subsequences of level 3 is calculated for each level. These values are then presented as inputs to the ANNs after uniformly normalizing them to the range [0,1]. Table II presents a subset of the frequent subsequences of length three and the number of occurrences of each of them for one example level.

VII. EXPERIMENTS

The rest of this paper describes a number of experiments that have been carried out to: 1) identify the features that convey the most useful information about game content; 2) investigate the size of a game session that yields the best performance in predicting players' reported preferences and 3) define the smallest game session size for which the model can still predict reported engagement with acceptable accuracy. As the data used to construct the model is based on pairwise preference, the baseline (majority vote) predictor accuracy is in all cases 50%.

For each experiment, different MLP topologies have been investigated as discussed in section VI-A. The analysis presented in the following sections is based on the best networks (in terms of performance, size and standard deviation over five runs). The statistical analysis presented is based on 20 runs for each of the best network chosen. Significant effect is determined by $p < 0.05$.

A. MLPs Performance on Full Information about Game Content

Statistical features from the whole game session have been extracted and included as inputs to MLP models. The models have been evaluated on features from the windows and features from segments to which the windows have been divided. Fig. 5 illustrates the performance of the ANN models with respect to the number of the segments used. Since we are dividing the window for up to three segments, the number of inputs for the MLPs is 6, $6 * 2 = 12$ or $6 * 3 = 18$ when evaluated on 1, 2 and 3 segments, respectively.

TABLE III

THE TOPOLOGY AND PERFORMANCE OF THE BEST MLP MODELS EVALUATED ON FULL AND PARTIAL INFORMATION ABOUT GAME CONTENT. THE MLP PERFORMANCE PRESENTED IS THE AVERAGE PERFORMANCE OVER 20 RUNS.

Training and evaluation data	MLP topology	MLP performance
Full window	6-2-2-1	63.16%
Two segments	12-2-1	61.43%
Three segments	18-4-4-1	59.97%
1st segment out of 2	6-2-2-1	59.07%
2nd segment out of 2	6-10-2-1	59.13%
1st segment out of 3	6-2-8-1	60.04%
2nd segment out of 3	6-4-6-1	58.45%
3rd segment out of 3	6-10-6-1	57.41%
1st and 2nd segments out of 3	12-8-1	60.49%
1st and 3rd segments out of 3	12-10-6-1	60.80%
2nd and 3rd segments out of 3	12-10-8-1	58.90%

The best networks found vary in size and topology. Results presented in Table III show that the performance is degraded by segmenting the data. The accuracy for the MLP evaluated on the whole game window is 63.1%. When two segments are used for evaluating, the performance decreases to 61.4%. Further dividing the windows into three segments resulted in a further decrease in the MLP performance to 59.8%.

To check weather partitioning the level causes a significant decrease in networks performance, we check for a statistically significant effect ($p < 0.05$) between the performance of the networks. Results show that evaluating the networks on full information about the level calculated from different number of segments to which the level has been divided yields significant decrease in the models' accuracies in predicting players' reported engagement. These results suggest that information is lost due to partitioning the window, and this loss causes a decrease in the performance.

B. MLPs Performance on Partial Information about Game Content

To define the smallest game session size for which the model can still predict reported emotion with acceptable accuracy, we evaluate MLPs on features extracted from different segments' size. We start by dividing the windows into two segments and train the MLP models on each segment at a time (6 features as inputs). Comparing the results obtained with the result of the model evaluated on both segments ($6 * 2 = 12$ features as inputs) we found that using features extracted from both segments for evaluating yields better prediction accuracy than when evaluating on features extracted from one segment at a time. Table III presents the topologies and the prediction accuracies for models evaluated on different number of segments. The statistical analysis shows that this performance decrease is significant.

To further investigate the effect of the size and choice of the segment that gives the most useful information, we partition the windows into three segments and evaluate the

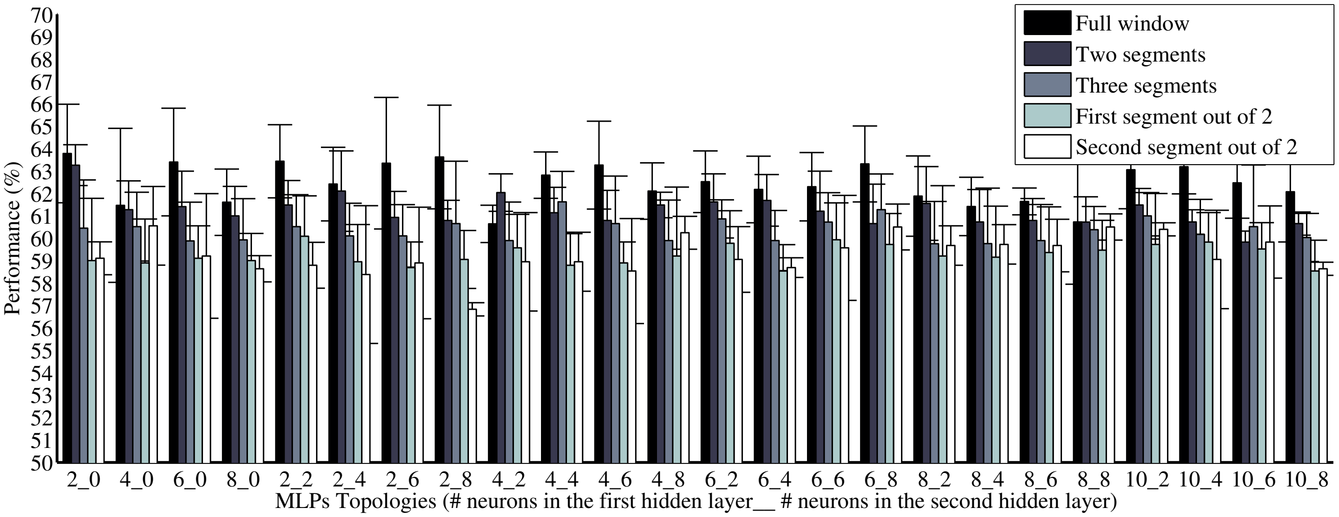


Fig. 5. The performance and topologies of MLP models evaluated on full and partial information of game content using statistics from the game window and from two and three segments to which the window has been divided. The performance presented is the average over five runs

networks using one segment out of three at a time. The models' topologies and accuracies are presented in Table III. As can be seen from the results, the MLP model evaluated on features from the whole session (the three segments together) and the model evaluated on features from the first segment only have a slightly better performance than the one obtained when evaluating on the second segment which is, in turn, slightly better than the performance of the model evaluated on the third segment only.

The statistical analysis shows no significant decrease in the performance between the model evaluated on features extracted from the 3 segments together and the performance of the model evaluated on features from the first segment. However, a significant degradation in the performance was obtained between the model evaluated on full data from the level and the two models evaluated on features from the second or third segments. This suggests that the information contained in the first segment helps more in building better predictors of players' reported preferences than the information contained in the second and third segment.

To further investigate this results, we evaluate MLP models on all possible combinations of these three segments. The results, depicted in Table III show that the models evaluated on features extracted from the first and second segments and from the first and third segments performs better than the model evaluated on the second and third segments.

By statistically analysing these results we found a significant decrease in the models' performance when evaluating on the first and third segments and when evaluating on the second and third segment, while no significant effect was found when evaluating on the first and second segments. This indicates that the information contained in the third segment is less useful to predict players' reported preferences that the information contained in the first and second segments. For further analysis, we investigate for significant effect between the model evaluated on the whole level and the model evaluated on the first and second segments out of the three segments together, the result shows a significant performance decrease between these two models.

In general, the statistical analysis of the models' evaluated on full or partial information from a different number of segments suggest that partitioning the level causes a significant decrease in the accuracy of predicting player's reported engagement. This suggests that there might be information loss because of decomposing the data and that this loss causes a performance decrease.

C. Sequences as Input Features for MLPs

We investigate another form of content representation which we use as input to the MLP models. Sequences capturing different information about content have been extracted following the method described in section VI-B.2. MLP models of different number of inputs and different topologies (see section VI-A) have been evaluated on the three different types of sequences. Since the number of frequent subsequences varies between S, E_p and D as can be seen from Table I, and since we are evaluating the MLP models on the number of occurrences of each of these frequent subsequences in each level; the number of inputs to the MLP models varies between 35, 7 and 12 for S, E_p and D, respectively. The performance and topologies of the best-performing ANNs are presented in Table IV.

The results shown in Table IV indicate that the model evaluated on frequent subsequences of information about enemies and items, D, outperform the other models evaluated on platform structure, S, and on enemies placement alone, E_p.

By statistically analysing the results, we obtained a significant performance decrease between the model evaluated on D and the models evaluated on S and E_p.

TABLE IV

THE BEST-PERFORMING MLP MODELS EVALUATED ON OCCURRENCES OF FREQUENT SUBSEQUENCES OF LENGTH THREE EXTRACTED FROM THE 40 LEVELS

Training and evaluating data	MLP topology	MLP performance
Platform structure	12-10-6-1	62.00%
Enemies placement	7-8-4-1	54.03%
Enemies and items placement	35-8-8-1	59.54%

It's worth noting that the results obtained from the network evaluated on D have a slightly lower performance than the best ones obtained using statistical features for training and evaluating the networks.

The result suggests that better model could be built by combining statistical and sequential forms of content representation.

D. Statistics and Sequences as Input Features for MLPs

In order to build a better model of players' preferences and squeeze the most useful information about game content we combine the best two models obtained from evaluating on statistics and frequent subsequences of game content.

More specifically, we construct new models based on controllable features statistics extracted from the whole game session, and occurrences of frequent subsequences of D. We proceed in constructing the topologies of the MLP models following the methodology presented in section VI-A, but since our input feature space is rather big, Sequential Forward Selection (SFS) is utilized to find the features subset that yields the best performance and save computational effort.

1) Feature Selection, Sequential Forward Selection: SFS is a bottom-up search procedure where one feature is added at a time to the current feature set. The feature to be added is selected from the subset of the remaining features such that the new feature set generates the maximum value of the performance function over all candidate features for addition.

The set of 18 features (6 controllable features and 12 frequent subsequences of D) is used as input to SFS to extract the minimal features subset that yields the best performance. The performance of each model is measured through the average classification accuracy of a Single Layer Perceptron (SLP) in three independent runs using 3-fold cross validation.

Using SLPs with the subset of selected features as inputs, The model was able to predict players' reported preferences of engagement with 63.2% accuracy.

The selected feature subset consists of seven features (+/- in parenthesis signifies positive or negative correlation): number of poweups (+), enemies placement (+) and the number of the occurrences of the following subsequences; 000 (-), 022 (+), 200 (+), 222 (-) and the subsequence 322 (+).

The analysis of the correlations between the selected features and players' preferences of engagement draws a picture of most players enjoying game that includes many items like free coins, coin blocks, powerups and enemies, but the fact that a positive correlations were found between players' preferences and the two sequences 022 and 200 while players' preferences were negatively correlated with the sequence 222, indicates that players prefer these objects to be distributed rather than allocated close to each other.

2) Selected Features as Inputs to MLPs: Different MLPs topologies have been investigated using the subset of selected features as input. The topology of the best model found consists of two hidden layers with 6 and 8 in first and second hidden layer, respectively. The model is able to predict players' reported emotional preferences with 65.72% accuracy. The statistical analysis shows that this performance is significantly higher than all other models mentioned previously along with the model evaluated on all features without using the feature selection mechanism.

Fig. 6 presents the best-performing ANN using SFS compared to the best-performing models obtained from statistical features from different number of segments and the best models evaluated on occurrences of frequent subsequences.

VIII. DISCUSSION

Using a combination of two types of features of game content, we are able to predict players' reported engagement preferences with acceptable accuracy. (We remind the reader that the baseline accuracy is 50%)

The results show that the ANN engagement preference models built on data derived from the game as a whole gives the best performance over all other models that have been constructed. Thus, the results suggest that the minimum acceptable size of the segment for which the model is able to predict player's reported preferences of engagement with acceptable accuracy is the one that has been chosen in the first place when designing the experiment.

The results indicate that the models performance in general decreases when segmenting the data. A performance decrease was observed when segmenting the window into two and three segments. This suggests that segmenting the data causes information loss and that the loss is minimized when evaluating the models on data from the full window. Another possible explanation for performance degradation when partitioning the windows is that the input feature space expands by segmenting the data (6 input features for the full window, 12 input features for two partitions and 18 input features for three partitions) resulting in a harder problem to learn (the curse of dimensionality).

Note, again, that the generated models are the composite of subjective preferences of several subjects. The models are thus average models, not perfectly adapted for any individual playing the game.

IX. CONCLUSION AND FUTURE WORK

The work reported in this paper presents data-driven computational models that predict players' reported engagement based on level design features. We investigate several types of features and different window lengths in order to investigate

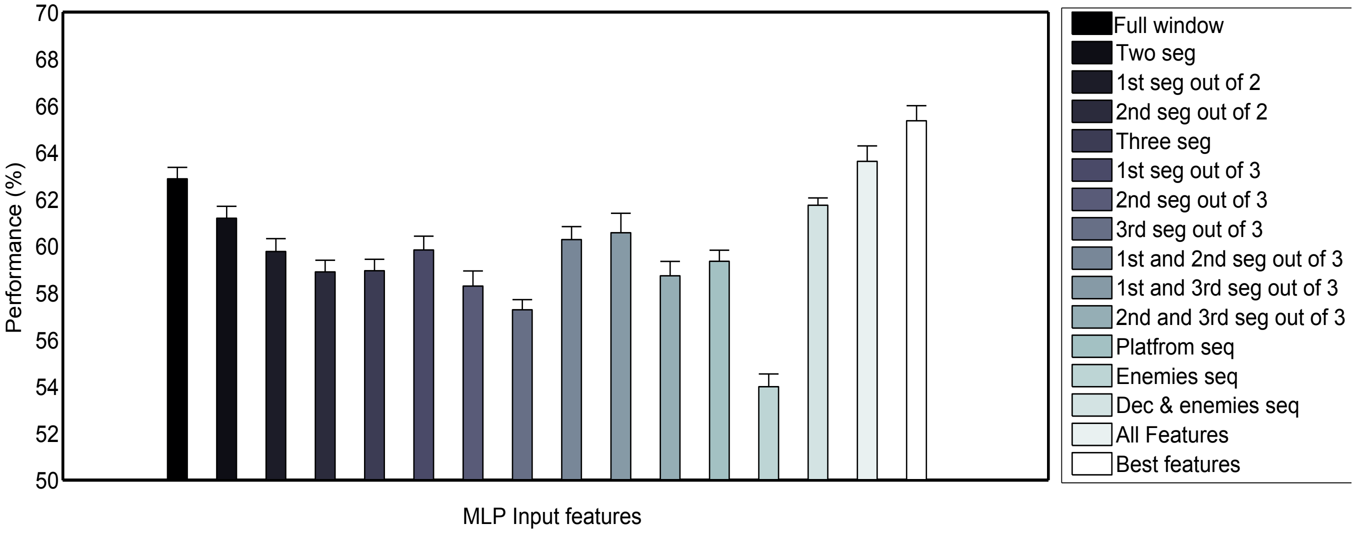


Fig. 6. The performance and topologies of the best MLP models evaluated on a statistical controllable features, number of occurrences of frequent subsequences and a subset of features extracted by SFS.

how to most effectively construct the best player preference predictors from short levels or sections of levels. The best predictors we found were based on selected features based on both directly controllable design parameters and frequent subsequences of level elements.

While we were able to construct predictors with acceptable accuracy, there are several ways to further increase those models' performance. The most obvious improvement is to include players' gameplay characteristics as features when constructing the models (as we have done in our previous work [13] on a smaller dataset with fewer features). The good results obtained by sequence-based features in the current study suggest that features based on frequent sequences of player actions could be effective. Higher accuracy on such predictors will bring us close to our ultimate goal, to be able to modify the levels in real-time adapting the content based on the performance of specific players.

ACKNOWLEDGMENTS

The research was supported in part by the Danish Research Agency, Ministry of Science, Technology and Innovation; project "AGameComIn" (274-09-0083).

REFERENCES

[1] C. Bateman and R. Boon, *21st century game design*. Charles River Media, 2006.

[2] K. Isbister and N. Schaffer, *Game Usability: Advancing the Player Experience*. Morgan Kaufman, 2008.

[3] R. Koster, *A theory of fun for game design*. Paraglyph press, 2004.

[4] G. Andrade, G. Ramalho, H. Santana, and V. Corruble, "Automatic computer game balancing: a reinforcement learning approach," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '05. New York, NY, USA: ACM, 2005, pp. 1111–1112. [Online]. Available: http://doi.acm.org/10.1145/1082473.1082648

[5] G. N. Yannakakis and J. Hallam, "Real-time Game Adaptation for Optimizing Player Satisfaction," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 2, pp. 121–133, June 2009.

[6] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation," in *Proceedings of EvoApplications*, vol. 6024. Springer LNCS, 2010.

[7] G. N. Yannakakis and J. Togelius, "Experience-Driven Procedural Content Generation," *IEEE Transactions on Affective Computing*, 2011.

[8] N. Shaker, G. N. Yannakakis, and J. Togelius, "Towards Automatic Personalized Content Generation for Platform Games," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. AAAI Press, October 2010.

[9] E. Hudlicka, "Affective computing for game design," in *GAMEON-NA'08: Proceedings of the 4th Intl. North American Conference on Intelligent Games and Simulation*, Montreal, Canada, 2008, pp. 5–12.

[10] G. N. Yannakakis, M. Maragoudakis, and J. Hallam, "Preference learning for cognitive modeling: a case study on entertainment preferences," *Trans. Sys. Man Cyber. Part A*, vol. 39, pp. 1165–1175, November 2009. [Online]. Available: http://dx.doi.org/10.1109/TSMCA.2009.2028152

[11] D. Charles and M. Black, "Dynamic player modeling: A framework for player-centered digital games," in *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, pp. 29–35.

[12] G. Yannakakis and J. Hallam, "Real-time Game Adaptation for Optimizing Player Satisfaction," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 2, pp. 121–133, June 2009.

[13] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling player experience for content creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, 2010.

[14] ——, "Modeling player experience in super mario bros," in *CIG'09: Proceedings of the 5th international conference on Computational Intelligence and Games*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 132–139.

[15] K. Höök, "Affective loop experiences - what are they?" in *PERSUASIVE*, ser. Lecture Notes in Computer Science, vol. 5033. Springer, 2008, pp. 1–12.

[16] G. N. Yannakakis and J. Hallam, "Game and player feature selection for entertainment capture," in *CIG'07: Proceedings of the 5th international conference on Computational Intelligence and Games*. Piscataway, NJ, USA: IEEE Press, 2007, pp. 244–251.

[17] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, pp. 31–60, January 2001. [Online]. Available: http://portal.acm.org/citation.cfm?id=599609.599626