A Bayesian Ensemble Regression Framework on the Angry Birds Game

Nikolaos Tziortziotis, Georgios Papagiannis and **Konstantinos Blekas** [1]

Abstract. An ensemble inference mechanism is proposed on the Angry Birds domain. It is based on an efficient tree structure for encoding and representing game screenshots, where it exploits its enhanced modeling capability. This has the advantage to establish an informative feature space and modify the task of game playing to a regression analysis problem. To this direction, we assume that each type of object material and bird pair has its own Bayesian linear regression model. In this way, a multi-model regression framework is designed that simultaneously calculates the conditional expectations of several objects and makes a target decision through an ensemble of regression models. The learning procedure is performed according to an online estimation strategy for the model parameters. We provide comparative experimental results on several game levels that empirically illustrate the efficiency of the proposed methodology.

1 INTRODUCTION

Angry birds was first launched five years ago by Rovio(TM), and since then it has become one of the most popular games nowadays. The objective is to get rid of the pigs, which are usually protected in structures made of different kinds of building materials, by killing them. This is achieved by taking control of a limited number of various birds' types, which the player launches to the targets (e.g. building blocks or pigs) via a slingshot. It must be noted that different types of birds are available with some of them being more effective against particular materials, while some other have special features as will be discussed later. The received return at each level is calculated according to the number of pigs killed, the number of the unused birds as well as to the destruction on the structure that achieved. Roughly speaking, the fewer birds are used as well as the more damage to the structures achieved, the higher the received return.

Due to its nature (e.g. large state and action spaces, continuous tap timing, various objects' properties, noisy object detection, inaccurate physical models), Angry Birds constitute a really challenging task. During the last two years, a number of works have been proposed which are focused on the development of AI agents with playing capabilities similar to those exhibited by human players. The Angry birds competitions[2] poses several challenges for building various AI approaches. A basic game platform [5] is provided by the organisers, that makes use of the Chrome version of the Angry Birds and incorporates a number of components such as, computer vision, trajectory planning, game playing interface which can be freely used for the agent construction.

Two different machine learning techniques, the Weight Majority algorithm and the Naive Bayesian Network, have been applied in [8] for selecting the most appropriate shot at each time step. However, the depicted feature space is extremely large since it incorporates a large amount of information about the scene of the game. In addition, it requires a preprocessing step over the input data in order to separate them among positive (shots in winning games) and negative (shots in losing games) examples. In [4, 6] a qualitative spatial representation and reasoning framework has been introduced that is capable of extracting decision rules according to structural properties. Finally, a model based approach has been presented in [9] which tries to learn the environmental model. Then, a number of trajectories are tested in the approximated model by performing a maximum impact selection mechanism.

In this work, we propose a Bayesian ensemble regression framework for designing an intelligent agent for the Angry Bird domain. The main advantages of our approach lies on two aspects:

- Firstly, a novel tree structure is proposed for mapping scenes of game levels, where the nodes represent different material of solid objects. This state representation is informative as incorporates all the necessary knowledge about game snapshots, and simultaneously abstract so as to reduce the computational cost and accelerate the learning procedure. This tree representation allows the construction of an efficient and powerful feature space that van be used next for the prediction.
- Secondly, an ensemble learning approach [7] is designed where every possible pair of 'object material' - 'bird type' has its own Bayesian linear regression model for calculating the expected reward. An ensemble integration framework based on the UCB algorithm [1] is employed using the predictions to obtain the final ensemble prediction. Then, an online estimation procedure is performed in order to adjust the regression model parameters. Finally, an appropriate Gaussian kernel space has been constructed by using a clustering procedure to a randomly selected data collection.

The remainder of paper is organised as follows. The general framework of our methodology is described in Section 2. In particular, the proposed tree structure which is the main building block in our approach, together with the ensemble mechanism of linear regressors are presented. Furthermore, some issues are discussed about the *feasibility* property of tree nodes, as well as about the *tap timing* procedure. To assess the performance of the proposed methodology we present in Section 3 numerical experiments on the 'Poached Eggs' game set and give some initial comparative results with the *naive* agent provided by the organisers. Finally, in Section 4 we provide conclusions and suggestions for future research.

[1] Department of Computer Science & Engineering, University of Ioannina, Greece, email: {ntziorzi,gpapagia,kblekas}@cs.uoi.gr

[2] https://aibirds.org/

2 PROPOSED STRATEGY

Our work is based on the project Angry Bird Game Playing software (version 1.31). The proposed methodology is focused on establishing an efficient state space representation, so as to incorporate all the useful information of objects from Angry Birds levels as recognized by the game vision system. In addition, a decision making mechanism has been designed using an Bayesian ensemble regression framework in order to discover the optimum policy and obtain the final ensemble prediction.

Figure 1 illustrates briefly the proposed approach. A step-by-step description is the following:

1. Construct the **tree structure** of the game scene and evaluate each node.
2. Examine the **feasibility of nodes** in terms of their ability to be reached and become possible targets.
3. Calculate the expected reward of each feasible node (target) according to a **Bayesian ensemble regression** scheme, which takes into account the type of object material, as well as the bird. The optimum target is then selected.
4. Perform shooting according to a **tap timing** procedure.
5. Adjust the model parameters of the selected regressor using an online **learning procedure**.

Next, we give a detailed description of the main building blocks of our methodology.

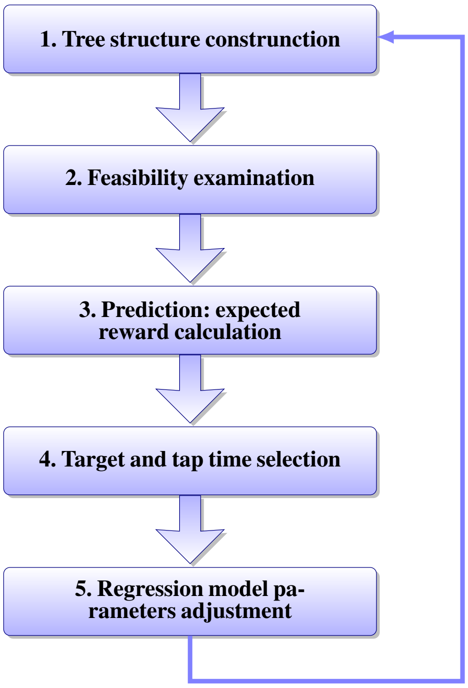

1. Tree structure construnction
2. Feasibility examination
3. Prediction: expected reward calculation
4. Target and tap time selection
5. Regression model parameters adjustment

Figure 1. Flow diagram of the proposed method

2.1 An advanced tree-structure for the Angry Birds scene representation

The input in our scheme is the game scene consists of a list of (dynamic or static) objects together with some measurements of them, as taken by the Angry Bird vision system. We have considered seven (7) types of materials for objects presented in the game:

- Ice/Glass (I)
- Wood (W)
- Stone (S)
- Rolling Stone (RS)
- Rolling Wood (RW)
- Pig (P)
- TNT (T)

Our state space representation follows a *tree-like structure* of the game scene using spatial abstractions and topological informations. In particular, we construct a tree where each node represents a union of adjacent objects of the same material. This is done in an hierarchical fashion (*bottom-up*). The root node is considered as a virtual node that communicates with *orphans* nodes, i.e. nodes which do not have any other object above, see for example nodes: s_{11}, s_{15}, s_{91} in Fig. 2.

Then, we evaluate each node (s) of the tree using three quantities:

- $x_1(s)$: **Personal weight** calculated as the product of the area $Area(s)$ of the object with a coefficient c_s which is related to the type of the objects, i.e. $x_1(s) = Area(s) \times c_s$. All types of object have the same value for this coefficient, $c_s = 1$, except for the types of Pig (P) and TNT (T) which have a larger value of $c_s = 10$.
- $x_2(s)$: **Parents cumulative weight** calculated by the sum of personal weights of the node's parents, $\mathcal{P}(s)$, in the tree, i.e. $x_2(s) = \sum_{s' \in \mathcal{P}(s)} x_1(s')$.
- $x_3(s)$: **Distance** (in pixels) to the nearest pig, normalized to $[0, 1]$. This is made dividing the original distance by 100, where we assumed that 100 pixels is the maximum distance in the scene among objects and pigs.

The above strategy introduces an appropriate and powerful feature space for all the possible targets. An example of this mechanism is presented in Fig. 2 where illustrates the produced tree structure for the scene of the first level of the game's episode. In addition, Table 1 gives the features of the constructed tree nodes.

Table 1. The feature vectors along with the feasible and type labels for the 16 tree nodes of Fig. 2.

Nodes	Features					
	Level	Type	Feasible	Personal Weight (x_1)	Above Weight (x_2)	Distance (x_3)
s_{11}	1	Wood	**True**	65	0	0.818
s_{12}	1	Wood	**True**	312	3557	0.501
s_{13}	1	Wood	False	156	7656	0.660
s_{14}	1	Wood	False	312	3557	0.501
s_{15}	1	Wood	False	65	0	0.818
s_{21}	2	Ice	False	162	3682	0.504
s_{22}	2	Ice	False	130	3682	0.504
s_{31}	3	Wood	False	125	3557	0.341
s_{41}	4	Wood	False	318	3239	0.151
s_{51}	5	Wood	**True**	318	377	0.164
s_{52}	5	Wood	False	72	1777	0.082
s_{53}	5	Wood	False	318	377	0.198
s_{61}	6	Pig	**True**	1400	377	0.170
s_{71}	7	Wood	**True**	156	221	0.431
s_{81}	8	Stone	**True**	156	65	0.521
s_{91}	9	Wood	**True**	65	0	0.651

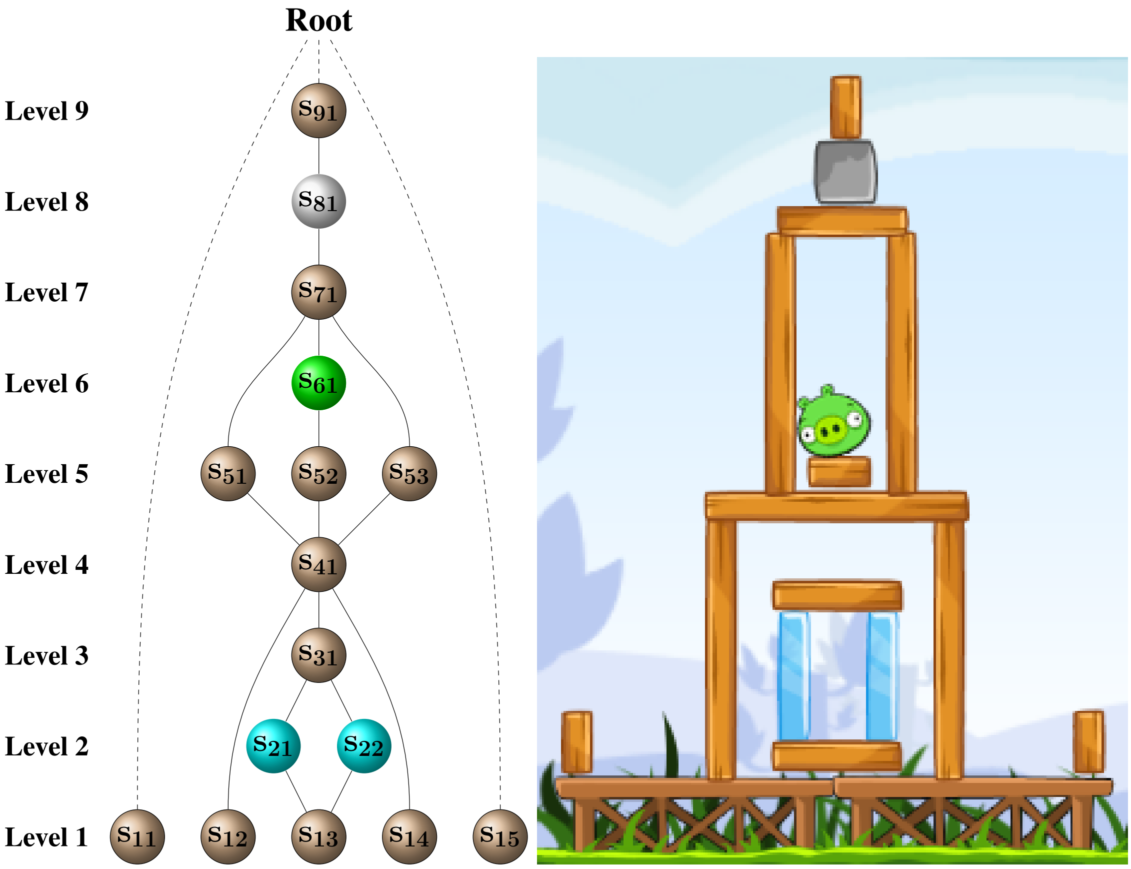


Figure 2. The proposed tree structure consisting of 16 nodes at the first game level.

2.2 Feasibility examination

The next step to our approach is to examine each node in terms of its possibility to be reached. Infeasible situations could be happened as the bounding boxes of objects in the scene may not be able to perfectly fit these structures and they often have irregular non-convex shapes. In addition, it is possible some obstacles and stable structures such as mountains, to be inserted between the slingshot and the target. Therefore, an examination step is required at each node so as to ensure that the corresponding trajectories can reach the target.

It must be noted that two different trajectories are calculated, a direct shot (angle $<= 45°$) and a high arching shot (angle $> 45°$). Both of them are examined in order to estimate the tree's nodes feasibility, see Fig. 3. If there is at least one shot that could reach that node (target) directly, we label it as *feasible* (Fig. 3(a)), otherwise the tree's node is labeled as *infeasible* (Fig. 3(b)). In the case where both trajectories are accepted, priority is given on the direct shot due to its effectiveness. Finally, in the case of the white bird a node is considered as feasible if it can be reached by bird's *egg* (Fig. 4), as opposed to the other types of birds.

2.3 Ensemble of linear regression models

In our approach we convert the problem of selecting an object for shooting into an ensemble regression framework. We consider the reward values as the real target values t_n of samples (feature vectors) $\boldsymbol{x}_n$ which are observed sequentially. They correspond to noisy measurements of the output of an M-order linear regression model together with an additive noise term ϵ_n:

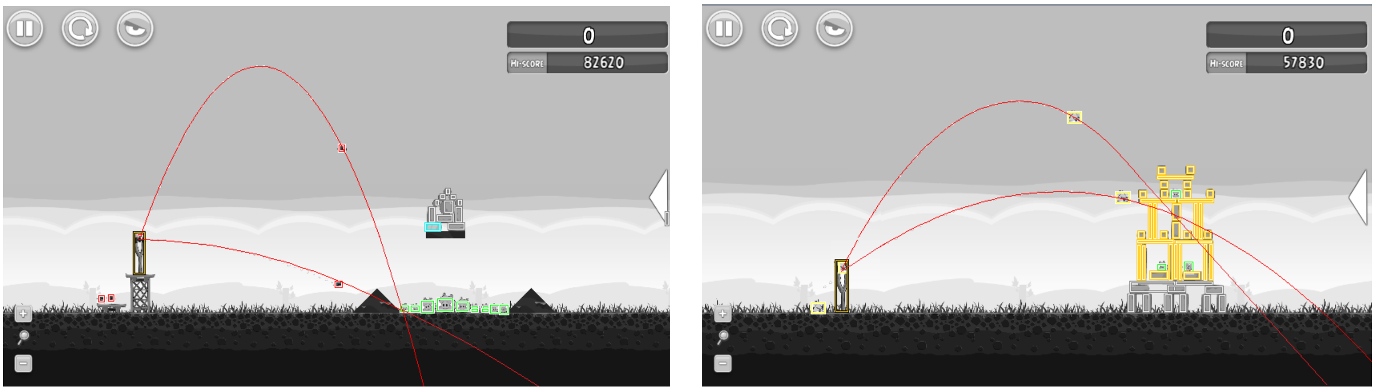
$$t_n = \sum_{i=1}^{M} w_i \phi_i(x_n) + \epsilon_n = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_n) + \epsilon_n ,$$

where $\boldsymbol{w} = (w_1, \ldots, w_M)^\top$ is the vector with the M unknown regression parameters. The above equation represents the reward as a linearly weighted sum of M fixed basis functions denoted by $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x}))^\top$. The error term ϵ is assumed to be zero mean Gaussian with variance $1/\beta$, i.e. $\epsilon \sim \mathcal{N}(0, \beta^{-1})$.

Specifically, we have considered Gaussian kernels as basis functions following the next procedure: At first we have gathered a number of data (feature vectors) from different scenes of the game. Then, we performed an agglomerative hierarchical clustering procedure to them, where we have applied the standardized Euclidean distance for the merging procedure. Finally, we have selected a number M of clusters, where we calculated their mean m_{ik} and variance s_{ik}^2 for any feature ($k = \{1, 2, 3\}$). Therefore, kernel functions have the following form:

$$\phi_i(\boldsymbol{x}) = \exp\left(- \sum_{k=1}^{3} \frac{(x_k - m_{ik})^2}{2 s_{ik}^2} \right) .$$

It must be noted that the number of clusters was not so crucial for the performance of the method. During our experimental study we have found that a number of $M = 150$ clusters was adequate.



(a)

(b)

Figure 3. Tree's node feasibility examination. (a) Represents a feasible node (pig) as it is reachable by at least one trajectory. The direct shot is infeasible due to the fact that a mountain is interposed between the slingshot and the target. (b) An infeasible node (wood) is represented as it is not directly reachable due to the tree structure.

Consider a sequence of observations (input vectors) $\{\boldsymbol{x}_k\}_{k=1}^{n}$ along with the corresponding targets $t_{1:n} = \{t_k\}_{k=1}^{n}$. Therefore, given the set of regression parameters $\boldsymbol{w}, \beta$ we can model the conditional probability density of the targets $t_{1:n}$ with the normal distribution, i.e.

$$p(t_{1:n}|\boldsymbol{w},\beta) = \mathcal{N}(t_{1:n}|\Phi_n \boldsymbol{w}, \beta^{-1} I_n)\ ,$$

where matrix $\Phi_n = [\boldsymbol{\phi}(\boldsymbol{x}_1), \boldsymbol{\phi}(\boldsymbol{x}_2), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n)]^\top$ is called the design matrix of size $n \times n$ and I_n is the identity matrix of order n.

An important issue, when using a regression model is how to define its order M, since models of small order may lead to underfitting, while large values of M may lead to overfitting. One approach to tackle this problem is through the Bayesian regularization method that has been successfully employed at [11, 2]. According to this scheme, a zero-mean (spherical) Gaussian prior distribution over weights w is considered:

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, a^{-1}I),$$

where the hyperparameter α is the common inverse variance of all weights and I is the identity matrix. In this direction we can obtain the posterior distribution over the weights $\boldsymbol{w}$, which is also Gaussian, as:

$$p(\boldsymbol{w}|t_{1:n},\alpha,\beta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_n, \Sigma_n)\ ,$$

where its mean and covariance are given by

$$\boldsymbol{\mu}_n = \beta \Sigma_n \Phi_n^\top t_{1:n}\ \ ,\ \ \Sigma_n = (\beta \Phi_n^\top \Phi_n + aI)^{-1}.$$

Then, when examining a test point (node) $\boldsymbol{x}_*$ we can calculate the prediction and obtain its corresponding target t_* according to the predictive distribution. In the Bayesian framework, this is based on the posterior distribution over the weights,

$$p(t_*|t_{1:n},\alpha,\beta) = \mathcal{N}(t_*|\boldsymbol{\mu}_n^\top \boldsymbol{\phi}(\boldsymbol{x}_*), \beta_*)\ ,$$

where

$$\frac{1}{\beta_*} = \frac{1}{\beta} + \boldsymbol{\phi}(\boldsymbol{x}_*)^\top \Sigma_n \boldsymbol{\phi}(\boldsymbol{x}_*).$$

Our framework follows an ensemble approach in the sense that we have a separated regression model for each pair of material object and bird type. Totally, there are $7 \times 5 = 35$ different parametric linear regression models, each one has its own set of regression parameters $\theta_j = \{\boldsymbol{w}_j, \beta_j\}$. Thus, every time we select a regressor for estimating the expected reward per each possible target (node).

In our approach, we have translated the selection mechanism into a multi-armed bandit problem which offers a trade-off between exploration and exploitation during learning. In particular, we have applied the Upper Confidence Bound (UCB) algorithm [1] for choosing the next arm (bird-material type regressor) to play. The selection mechanism is restricted only to the feasible nodes of the current tree. According to the UCB, each arm maintains the number of times (frequency) that has been played, denoted by $n_{f(q)}$, where $f(q)$ corresponds to the type of the regression model for the specific node q and the bird type used. The algorithm greedily picks the *arm* $f(j^*)$ as follows:

$$j^* = \arg\max_{q} \left\{ \left(\boldsymbol{\mu}_{n_{f(q)}}^{f(q)}\right)^\top \boldsymbol{\phi}(\boldsymbol{x}_q) + C\sqrt{\frac{2\ln N}{n_{f(q)}}} \right\}\ ,$$

where N is the total number of plays so far, $\boldsymbol{x}_q$ is the feature vector of a node and $\boldsymbol{\mu}_{n_{f(q)}}^{f(q)}$ is the current estimation of the regression coefficients that corresponds to the ensemble of the specific bird-material type pair. Finally, C is a constant of the UCB decision making process (during our experiments we have used $C = 3000$).

2.4 Tap Timing

After selecting the target among the feasible nodes of tree, the tap timing procedure is then executed. Using the trajectory planner component of the game playing framework the corresponding tap time is calculated and a tap is performed right before the estimated collision point. In our approach the tap time strategy depends on the type of birds used:

- For the red and black birds (*Bomb* birds are the most powerful among the birds) no tapping is performed.
- Blue birds (*the Blues*) split into a set of three similar birds when the player taps the screen. The agent performs a tap in an interval between the 65% and 80% of the trajectory from the slingshot to the first collision object.
- Yellow birds (*Chuck*) accelerate upon tapping which performed between 90% and 95% of the trajectory in the case of high-arching shots (angle $> 45°$). In the case of direct shots (angle $<= 45°$),

tap time has been selected randomly between 85% and 90% of the trajectory.

- White birds (*Matilda*) drop *eggs* in the target below them. In this case tapping is executed when the bird lies above the target (see, Fig. 4). As experiments have shown, this strategy is very efficient for handling this specific type of birds.

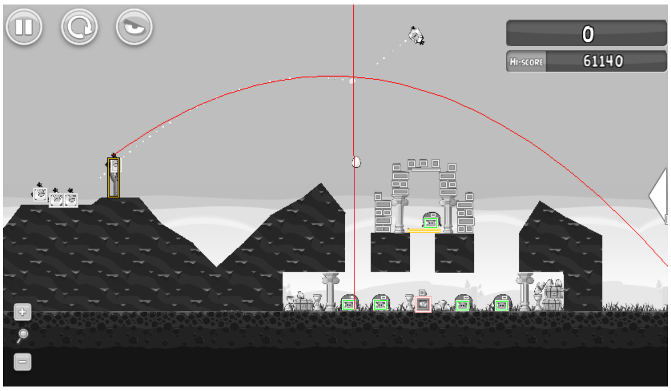


Figure 4. Tap timing procedure for the white bird.

2.5 Online learning of model parameters

The final step of the proposed scheme is the learning procedure. Due to the sequential nature of data, we have followed a recursive estimation framework for updating the regression model parameters [2]. This can be considered as an online learning solution to the Bayesian learning problem, where the information on the parameters is updated in an online manner using new pieces of information (rewards) as they arrive. The underlying idea is that at each measurement we treat the posterior distribution of previous time step as the prior for the current time step.

Suppose that we have selected a regressor, $k \triangleq f(j^*)$, for making the prediction upon an object that has a feature vector $\boldsymbol{x}_{n_k+1}$. After the tapping procedure we receive a reward t_{n_k+1}. The recursive estimated solution is obtained by using the posterior distribution conditioned on the previous n_k measurements $t_{1:n_k}$:

$$p(\boldsymbol{w}_k|t_{1:n_k}) = \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{\mu}^k_{n_k}, \Sigma^k_{n_k}).$$

The new received observation (reward) t_{n_k+1} follows the distribution $p(t_{n_k+1}|\boldsymbol{w}_k) = \mathcal{N}(t_{n_k+1}|\boldsymbol{w}_k^T\boldsymbol{\phi}(\boldsymbol{x}_{n_k+1}), \beta_k)$. Thus, we can obtain the posterior distribution of weights as:

$$\begin{aligned} p(\boldsymbol{w}_k|t_{1:n_k+1}) &= p(t_{n_k+1}|\boldsymbol{w}_k)p(\boldsymbol{w}_k|t_{1:n_k}) \\ &= \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{\mu}^k_{n_k+1}, \Sigma^k_{n_k+1}) \,, \end{aligned}$$

where the Gaussian parameters can be written in a recursive fashion as:

$$\begin{aligned} \Sigma^k_{n_k+1} &= \left[(\Sigma^k_{n_k})^{-1} + \beta_k\boldsymbol{\phi}(\boldsymbol{x}_{n_k+1})^T\boldsymbol{\phi}(\boldsymbol{x}_{n_k+1})\right]^{-1}, \\ \boldsymbol{\mu}^k_{n_k+1} &= \Sigma^k_{n_k+1}\left[\beta_k\boldsymbol{\phi}^T(\boldsymbol{x}_{n_k+1})t_{n_k+1} + (\Sigma^k_{n_k})^{-1}\boldsymbol{\mu}^k_{n_k}\right]. \end{aligned}$$

The above equations constitute a recursive estimation procedure for the regression model parameters. In the beginning of the estimation (i.e. step 0) all the information we have about the model parameters $\boldsymbol{w}_k$, is the prior distribution $p(\boldsymbol{w}_k)$ which is assumed to be zero mean Gaussian ($\boldsymbol{\mu}^k_0 = \mathbf{0}$) with spherical covariance matrix ($\Sigma^k_0 = a^{-1}I$). A last note is that, the sequential nature of estimation allows us to monitor the effect of learning progress to parameters.

3 EXPERIMENTAL RESULTS

A series of experiments has been conducted in an attempt to analyze the performance of the proposed agent (**AngyBER**) in the Angry birds domain. Due to the low complexity of the general framework of our agent, the experiments took place in a conventional PC[3].

Our analysis was concentrated mainly on the first 21 levels of the freely available 'Poached Eggs' episode of Angry Birds. During the learning phase of the AngryBER agent, a complete pass of the previously mentioned episode was executed more than once (in our study we have passed the episode 10 times). For comparison purposes, we have used the default *naive* agent, as well as the published results of the participant teams of the last IJCAI 2013 Angry Birds competition, since they are provided by the the organizers of the competition[4]. During testing, we have tried to follow the instructions mentioned in the competition rules, by setting a time limit of 3 minutes per level on average, that is, a total time of 63 minutes for the 21 levels. It must be noticed that our agent requires approximately forty (40) minutes for a successfully episode completion.

The depicted results are presented in Table 2 that gives statistics about the performance of the AngryBER agent, i.e. mean values and stds of the score reached per game level. Note that (after learning) we have made 10 independent runs of the episode. More specifically, mean and standard deviation of the score received per level, averaged over 10 runs. Furthermore, the maximum and minimum received score per level is also given.

The first remark that stems from our empirical evaluation is that our AngryBER agent achieves to pass every level with success at each run. Apart from a small fraction, AngryBER achieves to gain quite large scores in the majority of levels. That is interesting to be noted is the fact that our agent obtains the highest score in seven (7) levels as highlighted in Table 2, comparing with the results of all other agents of the last year's competition. At the same time, the mean accumulative score received per episode is approximately equal to the highest total score achieved among all the other agents.

Another impressive characteristic of the proposed scheme is its ability to speed-up learning process and to discover near optimal policies quite fast. We believe that this is happened due to the tree structure representation in combination with the ensemble strategy. This allows AngyBER agent to be specialized at each possible pair material-bird type, recognizing the special bird's behavior on specific materials. Last but not least, it must be noted that we have conducted a number preliminary experiments on Levels 22-42, where the results were similar making the generalization ability of our approach more evident.

4 CONCLUSIONS AND FUTURE WORK

In this work, we presented an advanced intelligent agent for playing the Angry Birds game based on an ensemble of regression models. The key aspect of the proposed method lies on the efficient representation of state space as a tree structure and the exploitation of its superior modeling capabilities to establish a rich feature space. An ensemble scheme of Bayesian regression models is then presented, where

[3] Intel Core 2 Quad (2.66GHz) CPU with 4GiB RAM

[4] https://aibirds.org/benchmarks.html

Table 2. Performance statistics of the proposed agent in the first 21 levels of the 'Poached Eggs' episode

Level	**AngryBER Agent**			**Naive Agent**	**High scores of IJCAI 2013**
	Mean Scores	**Max Scores**	**Min Scores**		**Angry Birds Competition**
1	28740 ± 165.6	28940	28400	29510	**31210**
2	51370 ± 2875.1	52360	43190	52230	**60400**
3	41917 ± 9.5	41920	41890	40620	**42240**
4	27049 ± 3485.6	29110	20350	20680	**36770**
5	65483 ± 2272.9	**69800**	63350	55160	65850
6	33961 ± 2860.0	35200	26020	16070	**36180**
7	26449 ± 7767.8	45650	20430	21590	**49120**
8	53191 ± 8782.2	57110	28240	25730	**57780**
9	36053 ± 7392.9	**52320**	24410	35490	51480
10	50547 ± 11221.9	65560	37980	32600	**68740**
11	55211 ± 7756.4	**60030**	33490	46760	59070
12	50151 ± 5502.5	54800	36530	54070	**58600**
13	43945 ± 7214.3	**50920**	25200	49470	50360
14	70181 ± 7176.1	**79330**	56620	50590	65640
15	43185 ± 3998.4	51620	38460	46430	**55300**
16	60430 ± 3295.1	63650	53680	55210	**66550**
17	48242 ± 3745.8	52050	39760	48140	**54750**
18	42975 ± 3145.8	48480	40210	49430	**54500**
19	30622 ± 4533.6	**39110**	21130	37920	38460
20	45523 ± 5643.8	54370	38870	36790	**56050**
21	66012 ± 5911.5	**78100**	58760	54240	75870
Total	971237 ±14647	991370	943250	858730	**1134920**

different bird-material type of regressors over the tree are combined and act as ensemble members in a competitive fashion. The best prediction is then selected for the decision making process. Learning in the proposed scheme is achieved in terms of an online estimation framework. Initial experiments on several game levels demonstrated the ability of the proposed methodology to achieve improved performance and robustness compared to other approaches on the Angry Birds domain.

We are planning to study the performance of the proposed methodology to other game levels and test its generalization capabilities more systematically. Since the tree structure is very effective and convenient, another future research direction is to examine the possibility to enrich the feature space with other alternative topological features which can be extracted for the proposed lattice structure. A general drawback in the regression analysis is how to define the proper number of basis functions. Sparse Bayesian regression offers a solution to the model selection problem by introducing sparse priors on the model parameters [11], [10], [3]. During training, the coefficients that are not significant are vanished due to the prior, thus only a few coefficients are retained in the model which are considered significant for the particular training data. This constitutes a possible direction for our future work that may improve further the proposed methodology.

REFERENCES

[1] P. Auer, N. Cesa-Bianchi, and P. Fischer, 'Finite-time analysis of the multiarmed bandit problem', *Machine Learning*, **47**(2-3), 235–256, (2002).

[2] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[3] K. Blekas and A. Likas, 'Sparse regression mixture modeling with the multi-kernel relevance vector machine', *Knowledge and Information Systems (KAIS)*, **39**(2), 241–264, (2014).

[4] L. A. Ferreira, G. A. W. Lopes, and P. E. Santos, 'Combining qualitative spatial reasoning utility function and decision making under uncertainty on the angry birds domain', in *International Joint Conference on Artificial Intelligence*, (2013).

[5] XiaoYu. Ge, S. Gould, J. Renz, S. Abeyasinghe, and P. Zhang J. Keys, A. Wang, 'Angry birds basic game playing software, version 1.31', Technical report, Research School of Computer Science, The Australian National University, (2014).

[6] S. Lin, Q. Zhang, and H. Zhang, 'Object representation in angry birds game', in *International Joint Conference on Artificial Intelligence*, (2013).

[7] J. Mendes-Moreira, C. Soares, A. Jorge, and J. Freire de Sousa, 'Ensemble approaches for regression: A survey', *ACM Computing Surveys*, **45**(1), 1–10, (2012).

[8] A. Narayan-Chen, L. Xu, and J. Shavlik, 'An empirical evaluation of machine learning approaches for angry birds', in *International Joint Conference on Artificial Intelligence*, (2013).

[9] M. Polceanu and C. Buche, 'Towards a theory-of-mind-inspired generic decision-making framework', in *International Joint Conference on Artificial Intelligence*, (2013).

[10] M. Seeger, 'Bayesian Inference and Optimal Design for the Sparse Linear Model', *Journal of Machine Learning Research*, **9**, 759–813, (2008).

[11] M.E. Tipping, 'Sparse bayesian learning and the relevance vector machine', *Journal of Machine Learning Research*, **1**, 211–244, (2001).