Building predictors for the game mia

Anonymous ACL submission

Abstract

This term paper deals with the topic of predicting different artificial intelligence approaches based on the game mia. It starts with an short introduction to artificial intelligence. Then, the description of the game mia on which the project is based on. Afterwards, the strategies of the different implemented artificial intelligences (AI) are explained. It follows an explanation of our approach predicting the behavior of the different AIs. Finally, the results of the experiments are discussed.

1 Introduction

Artificial intelligence has become an increasingly important field in computer science and other areas such as automotive (self-driving cars) or security (face-detection). In computer games artificial intelligence reaches new stages of success (Google bot AlphaGo for the game go). In this term paper we now want to answer the question if it is possible to predict different artificial intelligence approaches based on the game mia. The main achievement would be the accuracy of correct predictions if a player (AI) will to lie or tells the correct value. If the actual value is lesser or greater seems not as important as prediction liars properly.

2 The Game

Mia is a simple dice game that is played with two dices and a flat bottomed container (or a dice cup). At the beginning each player has a certain amount of lives (e.g. five). The first player rolls the dices but keeps their values hidden from the other players. He then can decide if he wants to tell the truth to the next player and announce a value that was was actually rolled. Alternatively he can lie and announce a greater or lesser value than the rolled one. But each player has to announce a greater value than the previous player. The next player (who still has not seen the actual values) can now believe the passer, call the passer a liar and look on the dice or pass the dice to the next player (still without looking) announcing a higher value. A player looses a life if he called the previous one a liar and looked on the values to find out that they are what the previous player has announced or even higher. Otherwise the previous player looses a life. The higher value of the roll is multiplied by then and then added to the other die (a 4 and a 2 is 42). The **scoring** is from highest to lowest: 21 (Mia), 11, 22, 33, 44, 55, 66, 65, 64, 63, 62, 61, 54, 53, 52, 51, 43, 42, 41, 32, 31. If a player announces mia the next player either believes him, give up (without looking at the dices) and looses one life. Or he may look at the dice. If it was actually mia then he looses two lifes if it was not, the previous player looses a life. (For further information see (mia, 2016).)

3 Implementation of the Game

4 Setup and Strategies

To measure the performance of the predictor different artificial intelligence approaches were used. For data acquisition we used a homogeneous set of players in our game implementation and let it generate game data. For each turn we reported:

- player number
- previous player number
- if the turn is the first one in the round (new game)
- new announced value
- actual rolled value

- previous announced value

- does the player look onto the dices or not

Since the actual rolled value is mostly hidden from the players. We decided to learn the predictor the players strategies only based on: new game, last announced value and new announced value (this seems most realistic). We then create liar labels for the datasets due to comparison of the new announced value and the actual rolled value for training purposes. Each dataset is split up to training data and test data. So we train the predictor for the different approaches by using cross validation (determination of parameter C) on the training data and measure the performance (generalization) on the test data. The test set is 30 percent of the data set, the others are training data. In the end we will compare the results of the implemented AIs detecting lies (take the *look at* label and the predictions of our predictor).

We examined three different types of artificial intelligence approaches. First the statistical approach where the AI acts very straight forward. Then an AI with a certain degree of randomness in its call and look behavior. Last we implemented a learning AI with different calling behaviors to examine if the predictor is able to learn the strategy of a learning player. Last we wanted to know how many samples it takes until there is convergence of accuracy in the predictor.

4.1 Statistic Approach

4.2 Approach with certain degree of randomness (Primitive Random)

This AI will call with a certain probability a value that is greater than the actual rolled value if the value is lower than the one of the previous player. Also the AI will look on the dices with a probability that arises with decreasing beat probability of a announced value (a 31 will not be looked at with probability near 100 percent, 21 is looked at with 100 percent).

4.3 A SVM learning approach

4.3.1 SVM variant 1

4.3.2 SVM variant 2

4.3.3 SVM variant 3

4.3.4 SVM variant 4

5 The Predictor

To predict the different strategical approaches we used a support vector machine (SVM) predictor with radial basis functions. A linear predictor would not have been sufficient due to the complexity of data, but with the SVM-predictor adequate results could be expected. By using SVM we are interested in separating two (or more) classes by a separating hyperplane with maximal margin. The margin is defined with respect to the training points as the minimal distance between the hyperplane and a training point.(See (von Luxburg, 2016) and (Schoellkopf and Smola, 2002, 187–227).)

6 Discussion of Results

An overview of the maintained results can be seen in table 1. In all but one case the learned predictor was able to predict the lier labels better than the AIs. Moreover we can conclude that the predictor was able to learn the strategy of the different approaches quite well.

strategy	**sdetect**	**C**	**pdetect**
statistic	**0.473**	**0.0001**	**0.863**
primitive	**0.585**	**0.5**	**0.849**
SVM1	**0.964**	**2.0**	**0.971**
SVM2	**0.810**	**0.6**	**0.797**
SVM3	**0.844**	**3.0**	**0.971**
SVM4	**0.642**	**2.0**	**0.989**

Table 1: Obtained results in the experiments. Strategy, strategies liar detection (sdetect), parameter C for predictor, predictors liar detection (pdetect) on test data.

For the statistical approach convergence takes place around 4000 samples if we allow one percent of deviation in both directions (see fig. 7). In fig. 6 the convergence can be seen more easily. It is about 4000 samples. Afterwards there is only little variation of the accuracy. For the SVM approaches can be seen that variant 1 and variant 3 converge to the same value of accuracy (0.97). Convergence for both variants is about 2000 samples. Variant 2 of the SVM approach has a lower lever of accuracy meaning that this one is harder to predict correctly than the others (under the given data). The convergence of variant 2 is at around 3000 samples. Variant 4 has the highest rate of accuracy with 98 percent and converges about 2500 samples (compare fig. 5).

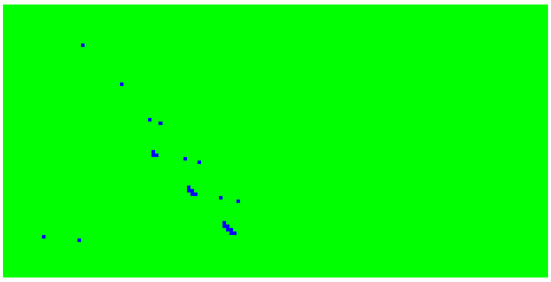

Figure 1: caption svm1

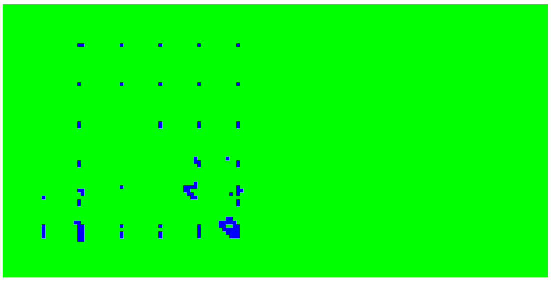

Figure 2: caption svm2

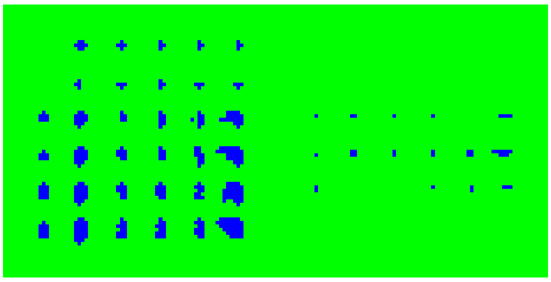

Figure 3: caption svm3

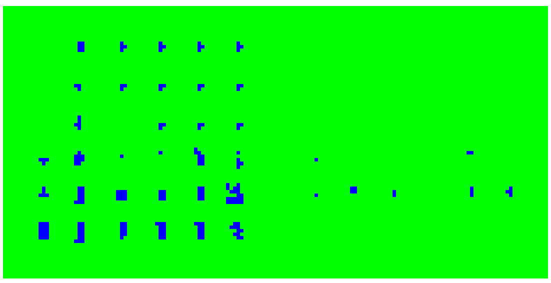

Figure 4: caption svm4

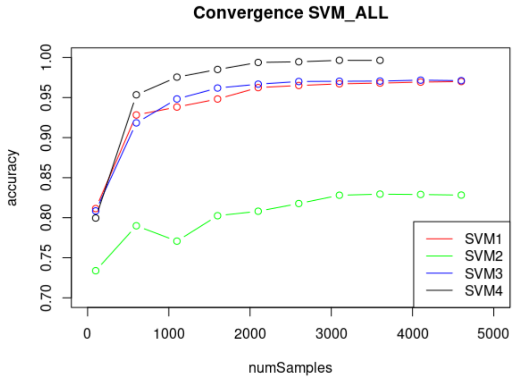

Figure 5: caption convergence all

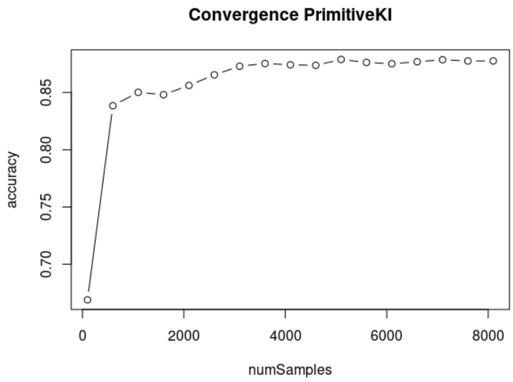

Figure 6: caption convergence all

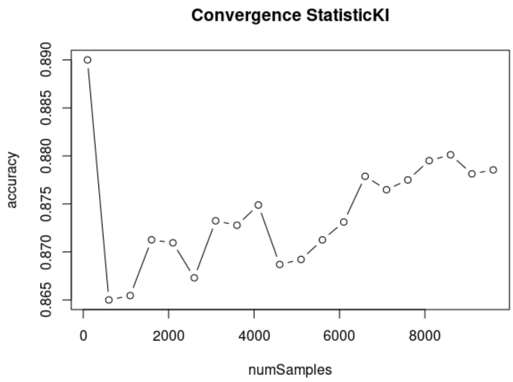

Figure 7: caption convergence all

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

References

[mia2016] 2016. Mia (game). https://en.wikipedia.org/wiki/Mia_(game).

[Schoellkopf and Smola2002] Bernhard Schoellkopf and Alexander J. Smola. 2002. *Learning with Kernels*. Massachusetts Institute of Technology, Cambridge, Massachusetts, London, England.

[von Luxburg2016] Ulrike von Luxburg. 2016. Machine learning: Algorithms and theory. Scriptum for the lecture.

A Supplemental Material