AI-PROJECT FINAL REPORT

Introduction to New Eleusis:

New Eleusis is a card game where there are 4-6 players involved. The game is theoretically based on logical induction, which player try to work out a strategy for defining a move as legal or illegal for a given play. This game is conceptualization of simulation of scientific research. The game in general involves a dealer who thinks of one particular rule, which govern the player's move for a given play. Other players try to develop their hypothesis by playing each with using a particular strategy and then try to come up a rule which would directly or indirectly match the rule of the dealer.

Introduction to Concepts Used in developing New Eleusis Player:

Following is the list of concepts that have been used in developing the New Player:

1. Logical Induction
2. Decision Trees

A. Logical Induction:

Logical Induction is a technique that extends the deductive logic to which has been most widely studied by the philosophers and logicians in recent years. This Inductive technique is based on the probability function, which measure the degree of evidence supporting a hypothesis.. This concept is developed on the backdrop of baye's theorem which represents the implication of the hypothesis for a given evidence claims, and influences the degree to which a given hypotheses are supported by those evidence claims[3] .

Induction technique which has been used in developing New Eleusis player is based on Induction by Model fitting. The model here denotes a syntactic or functional skeleton which is fleshed out by the induction algorithms to form a rule. The model that have been used in developing Eleusis and New Eleusis is primarily based on below two model are decomposition and Periodic.[4].

a. Decomposition Model

Decomposition model specifies the rule takes the form of an exclusive disjunction of if-then rules. The condition parts of the rules are referred to previous and previous two cards that are to be predicted. The action part of this model describes the if-then rules that describe correct play of the player.

b. Periodic Model:

A rule of this model describes the layout as periodic function. The layout is split into phases according to the length of the proposed period. Each phase described in this model has a conjunctive description.

c. **Disjunctive Normal Form(DNF):**

The disjunctive normal form technique has been one major element in defining Aq algorithm. This technique has been used to in Aq algorithm to guide it towards symmetric disjunctive terms acting as a heuristic.

B. Decision Tree

A decision Tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences including chance event outcomes resource costs, and utility[6]. The concept of decision trees helps in identifying a strategy which would most likely reach to a goal state. This technique is a descriptive means for calculating conditional probabilities. The decision tree is lineared into decision rules, where the outcome of the leaf node and condition along the path form a conjunction in the if the clause. Thus in general the rule are created is following form:

If condition1 and condition2 and condition 3 then outcome*.[6]*

ID3 also known as Iterative Dichotomiser 3 is an algorithm to generate possibly efficient decision tree. This algorithm is invented by Ross Quinlan[1] which is basically used for hypothesis testing using inductive methods. This algorithm is based on top down approach of greedy search for a given training data table. We define values from attributes of an unknown object to determine appropriate classification according to decision tree rules. We can obtain generalization from unobserved instances based on features are correlated with target labels. When we have sufficient amount of training dataset, this method is prefered for devising the model. The resulting decision tree provides a representation of the concept that appeals to human because it renders the classification process self-evident.It is appealing to humans that self evident base process of classification is significantly effective.[7]

The Instance is represented as attribute-value pairs and which is further quantified according to value and represented as a node in decision tree. It is possible that training data may be erroneous, this issue can be overcome by pruning of the path within the tree. Following are the setups that are covered with ID3 procedure[6].

1. Calculate the entropy of every attribute using the data set S
2. Split the set S into subsets using the attribute for which the resulting entropy (after splitting) is minimum (or, equivalently, information gain is maximum)
3. Make a decision tree node containing that attribute
4. Recurse on subsets using remaining attributes.

The entropy of the data is computed by taking summation over product of probability and log of inverse product of probability of every instance in attribute.

$$H(p) = -p\log(p)$$

Further, we will split the data based on entropy, we will select attribute with smallest entropy. Higher the entropy value higher the chances of getting classifier better. Alternately, we can also compute information gain than that of entropy for attribute selection in ID3. The information gain is compute by taking the difference between entropy of before and after splitting with respect to attribute X. Thus, it represents how much a certain attribute contributes towards classification or reduces the uncertainty before and after the split.

$$IG(X,Y) = H(p)=H(X)-p\log(p)$$

The largest information gain obtained for the attribute is used to split the set S over iteration.

Phase -I player Design:

Phase - I consisted of designing a player which would formulate a 'Rule' that describes whether a card is legal to play. For inducting such a rule, the player would work on forming multiple hypothesis, using the hypothesis the player would determine its next move or play. As player-I has an infinite hand of cards, dealer plays the initial card by formulating a rule as called as "god rule". There after the dealer governs or supervises the play of the New Eleusis player and classify them as legal and illegal. As the boundary condition where relaxed in phase-I the game involved announcing the rule after finishing a minimum of 20 card play by the player. The upper constraint on the player in phase -I was to return the best hypothesis before 200 card plays.

The player in phase-I was designed based on supervised learning approach. The decision trees was the predictive model that was used in defining and analysing the hypothesis of any given play for the New Eleusis player in phase -I. As the game is designed in such a way that at a given time only three cards are taken into consideration for defining any hypothesis by the New Eleusis player. The decision tree formed in phase -I was based on ID3 algorithm which would provide the best possible hypothesis that would be suitable for given card play based on a the current board state.

The player created phase one was accurately able to guess the "god's Rule", within a stipulated number of moves.

Phase -II player.

In this phase we will allow other players which we call as adversaries, to compete with themselves and our player. There are some specification associated with the game such as that every player will hold 14 cards, every player will come up with a belief for the hypothesis at the end of the game and a player calls for the prediction, final winner will be decided on basis of score function which will be computed bases of the belief they had in the round. Motive of each player is to compute the prediction and prune the hypothesis for the efficient solution faster and the one who is confident enough will call for the testing.

In continuation with phase-I to enhance the capabilities of our player which would involve predicting the hypothesis by considering adversarial player's moves, thus we can term as a learning agent. Here, we have three modules devised for out player for prediction. The first module would contain simple rules which the player would use in initial game moves. In the second module the hypotheses are devised to form more advanced form of complex rule sets which the player can use once the games advances into further stages. Third module is created with complex hypotheses set which is combination of first and second module hypotheses. We can not consider the hypothesis as fully bounded, as there could be multiple rules which could be satisfied for the rule made by dealer. Also, a rule can be induced by a single hypothesis. The confidence of the learning agent can be cross-validated based on a "bluff' move where the agent would make an incorrect move in order to strengthen the current hypothesis, thus which would help the player to predict better results.

Basic Difference made in designing Phase-I and Phase -II players.

As phase -I was single player, cards in hand of the player was not taken into consideration, however in phase-II since it was a multiplayer game and also the cards present in hand of a given player were 14 at a time, we had to keep to record of the cards in hand during each move during the game in phase -II. Further, the changes were made in playing strategy of the New Eleusis player for phase -II to involve the three adversarial players in the game. The phase -II player also devised a slightly new score function as compared to player designed in phase -I, as we took into consideration score of the other player as well. Lastly, we also took into consideration the board state of our own New Eleusis player as well as the board state of the other adversarial players.

C4.5 is an extension of ID3 algorithm. Decision trees created using C4.5 algorithm can be used for classification. It creates a decision tree based on training data in the same way as ID3 algorithm. It uses concept of entropy and information gain while building decision tree. In general algorithm for building a decision tree consists of calculating information gain for every attributes. For our new eleusis player attributes are considered as hypothesis. Here hypothesis can be fully formed rule or a part of a rule. Basic single argument rules can be easily represented as a hypothesis , but complicated rules are created with combination of two or more hypothesis. While building decision tree, best attribute is selected based on the information gain calculated. Hypothesis with maximum information gain is selected for splitting the decision tree[2].
There are certain improvements in C4.5 compared to ID3. C4.5 can handle continues valued attributes. For continuous valued attributes C4.5 splits the attributes using a threshold values. It has very efficient pruning techniques, it goes through the tree after creation and replaces the branches which are not useful.

AQ algorithm is also a machine learning methodology which we wanted to develop but ended up using decision trees for simplicity. AQ algorithm uses set of hypothesis given in its prolog database and induces a rule. AQ requires a model which needs the type of hypothesis in its input/database and relation between them.

It is based on an algorithm for determining quasi-optimal solutions i.e. optimal or suboptimal solutions to general covering problems of high complexity. AQ is a form of supervised learning. Therefore input data for AQ contains labeled data which is then generalized into sets of examples with respect to counterexamples. AQ is specifically helpful in finding rules for multivariate events. The algorithm learns from positive and negative patterns of attribute values.

Such patterns can be generalizations of the individual events of being totally complete and consistent to accepting a trade off of coverage to gain simplicity. This difference is based upon AQ's varying mode of operation.

AQ is a machine learning classifier, its learning process can proceed in these two modes -

1. Theory Formation Mode - In this mode the algorithm learns rules that covers all positives and does not cover any of the negatives i.e. the rules are complete and consistent with regards to the data. To be able to use this mode for the algorithm, the training data should be error free and consistent.
2. Pattern Discovery Mode - This mode was introduced after the original AQ methodology was implemented. Pattern Discovery mode is used to mine large and noisy datasets. The patterns learned could be inconsistent or incomplete with respect to the training data. However, this mode is useful for determining strong patterns in the data.

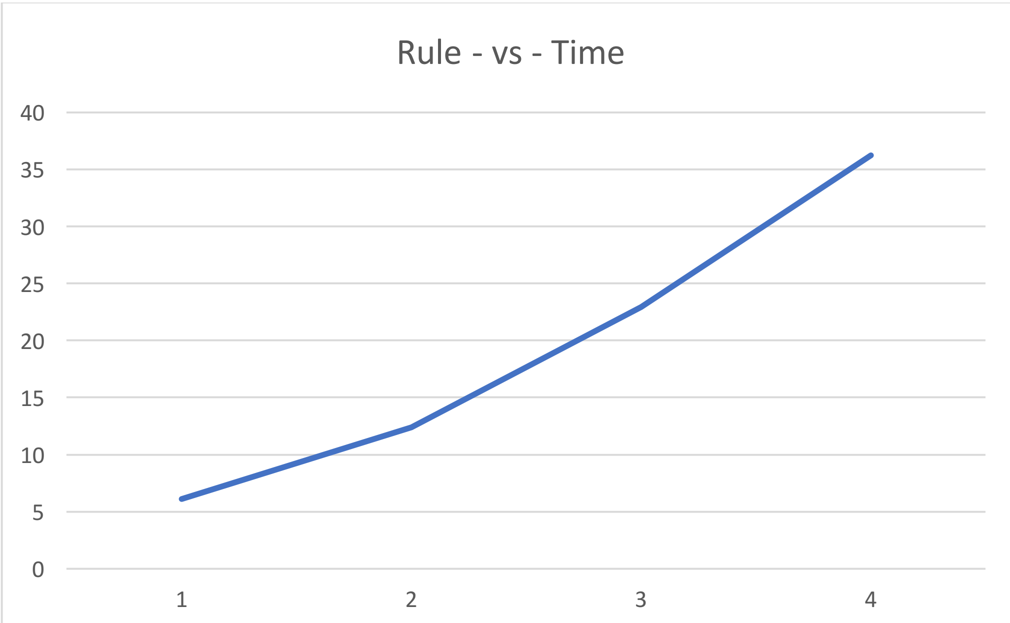
AQ implementation cannot handle continuous attributes.

For the New Eleusis game, given set of legal and illegal cards played, AQ classifies rules that covers all legal cards and do not cover any illegal cards for the current game state. Since the data is fully visible and correct, the mode to use would be Theory Formation mode.

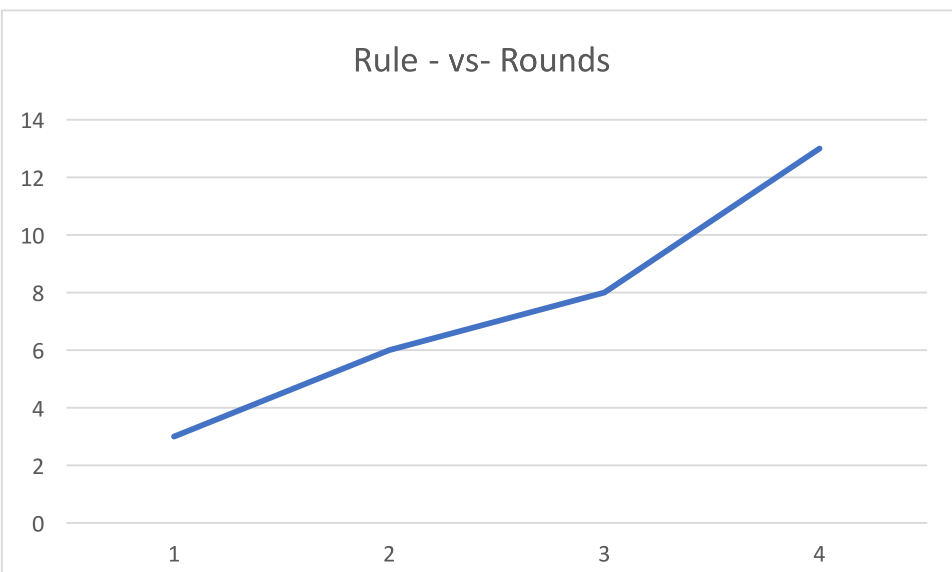
The evaluation New Eleusis player was evaluated by combining all function possible from our set of hypothesis. We use these functions for testing the accuracy of the player in predicting the correct rule. The other effective strategies we implemented was by playing our agent against other learning agent developed by our classmates. This strategy helped us in measuring the performance and comparing the accuracy of the prediction of the "God Rule". We also used the scores evaluated from the testing phase from to measure the performance of our player which helped us in identifying the rules which our player was not able to predict and also, improving our coding strategy for predicting better and faster rules.

Further, the player was tested against complex rule set to determine the time it takes to predict the correct and multiple run were performed to improve the predictive results by the player. The decision tree approach that was implemented was slightly tweaked from the general/ textbook approach in order to make our player to perform better and faster. The decision table formed is initially filled with the basic rule while the player plays then the rule are formed in the combination of 'or', ''if' and 'and', such that these rules are formed and then appended to the decision table. Thus, after every moved a particular set of rules are formed and are appended to the decision table and accuracy for each of the rule is calculated. The rule are then sorted in terms of the accuracy in descending order. The hypothesis that formulates to maximum accuracy is chosen by the New Eleusis player for predicting it next moves and also predicting the final "God's Rule".

For the experimental purpose, we used checked for the runtime of our program versus the length of a particular rule to see the performance graph of the player.



This graph denotes the time taken by our rule to predict the rule according to its difficulty.
As the rule becomes complicated the time taken by our player increases.



This graph shows number of rounds taken by our player to predict the rules. We deduce that as difficulty of our rule increases number of rounds taken by our player to predict is increased.

REFERENCES:

[1] Quinlan, J. Ross. "Induction of decision trees." *Machine learning* 1.1 (1986): 81-106
[2] https://en.wikipedia.org/wiki/C4.5_algorithm
[3] https://plato.stanford.edu/entries/logic-inductive/#6
[4] http://web.engr.oregonstate.edu/~tgd/publications/aaai80-eleusis.pdf
[5] http://onlinelibrary.wiley.com/doi/10.1002/wics.78/full
[6] http://users.ics.forth.gr/~potamias/hy577/rules.pdf
[7] https://en.wikipedia.org/wiki/Decision_tree
[8] Utgoff, Paul E., and Carla E. Brodley. "An incremental method for finding multivariate splits for decision trees." Proceedings of the Seventh International Conference on Machine Learning. 1990.
[10] https://en.wikipedia.org/wiki/Decision_table