Supervised Learning Methods in Video Game Dataset

Sichao Liu
6328594244
sichaoli@usc.edu

1. Problem definition

In this project, I studied the dataset of a video game called Starcraft2. Data samples are gathered through the UCI[1] data repository. My job is to impose classification method on this dataset and find the most affecting attributes in each league (game level). To start with, I will use Decision Tree and Bayes Classification to train the data. Then using cross-validation to compare these two approaches and conclude the key attributes. Using the modern classification methods, providing a reasonable explanation for the difference between each league is the final goal of this project.

2. Background

StarCraft2 is a famous RTS[2] (real-time strategy) game that attracted millions of players all over the world by Blizzard Entertainment. The participants maneuver units and position their structures under their control to secure the map or destroy their opponents' assets. Additionally, they will use mouse to check the units and hotkeys to accelerate their performance. All these principles are limited under a requirement for expend accumulated resources, which is the same for both participants at the start of the game. It is widely acknowledged that players will improve a lot after a great amount of practice. A players' total performance can be summarized as their control in the game, or their age, hours spent on the game per week outside the game.

Luckily, the game itself includes a league system that evaluates the players'

performance level. Players' leagues are divided into seven categories, ranging from Bronze, Silver, Gold, Platinum, Diamond, Master, and GrandMaster. Players always want to see their main difference between their current league and their ideal league. Is the action per minute players can make matters? Or players are just influenced by their total amount of hours spend on this game? Both these questions have greatly raised my interest to study further on this topic. The process will be a valuable attempt for future prediction about unknown data and also provide a meaningful guide for players who want to improve their gaming level quickly.

3. Dataset

This dataset is mainly about the players' performance and their playing habits in each game. The records were collected in StarCraft2 on September 2013. It has 3395 instances in total. Each instances consists of 20 attributes. Some are as follows:

GameID/LeagueIndex/Age/HoursPerWeek/TotalHours/APM/UniqueHotkeys /Total

MapExplored/WorkersMade

APM: Action Per Minute (continuous)
UniqueHotkeys: Number of unique hotkeys used per timestamp (continuous)
TotalMapExplored: The number of 24x24 game coordinate grids viewed by the player per timestamp (continuous)

The dataset is in .csv format. The data category consists of integer and real number.

Below is a quick look at these values in Excel format in figure 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	GameID	LeagueIndex	Age	HoursPerWeek	TotalHours	APM	SelectByHotkey	AssignToHo	UniqueHotk	MinimapAtt	MinimapRig	NumberOfPA	GapBetweenI	ActionLaten	ActionsInPa	TotalMapEx	WorkersMad	UniqueUnit	ComplexUni	ComplexAbi	litiesUsed
2	52	5	27	10	3000	143.718	0.003515189	0.0002197	7	0.00010985	0.00039232	0.00484904	32.6677	40.8673	4.7508	28	0.0013966	6	0	0	
3	55	5	23	10	5000	129.2322	0.003303812	0.0002946	4	0.0029406	0.00043244	0.00430706	32.9194	42.3454	4.8434	22	0.0011935	5	0	0.00020757	
4	56	4	30	10	200	69.9612	0.001101091	0.0003357	4	0.00029362	0.00046141	0.00292576	44.6475	75.3548	4.043	22	0.00074455	6	0	0.00018876	
5	57	3	19	20	400	107.6016	0.001033542	0.0002131	1	5.33E-05	0.00054341	0.00378255	29.2203	53.7362	4.9155	19	0.0004262	7	0	0.00038368	
6	58	3	32	10	500	122.8908	0.001136014	0.0003273	2	0	0.00132856	0.0023683	22.6885	62.0813	9.374	16	0.0011745	4	0	1.93E-05	
7	60	2	27	6	70	44.457	0.00097839	0.0002562	2	0	0	0.00242471	76.4405	98.7719	3.0965	16	0.00037221	6	0	0	
8	61	1	21	8	240	46.9962	0.000820114	0.0001685	6	0	4.49E-05	0.0019886	94.0227	90.8311	4.1017	16	0.00067296	5	0	0	

Figure 1 Original data format

4. Method involved

4.1 Decision Tree

Decision tree is a decision support tool used for classification and regression in

supervised learning. The goal is to create a model (tree) that predicts the value of a target by simple decision rules from the data sample. Data are classified/sorted according to specific feature values, which become increasingly specific. From the training dataset, one can calculate the entropy of the occurrence of each attribute, then find the best attributes to split the data. Also, since the trees can be visualized, this is a good feature for me to interpret.

4.2 Naïve Bayes Classification

Naïve Bayes classifiers are a family of simple probabilistic classifiers based on Bayes' theorem using independence assumptions between the features. The model includes assigning class labels to problem instances, and doing the prediction based on the conditional probabilities. Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. Bayes algorithm can work efficiently fast in supervised learning setting. For most cases, Bayes models uses the method of maximum likelihood to make the prediction.

4.3 Cross-validation

Cross-validation is a model validation technique for assessing the result of statistical analysis to an independent dataset. It is usually used when one wants to estimate how accurately a predictive model will perform in practice. This involves the data to be collected including training dataset, testing dataset, and unknown dataset. The goal of the cross-validation is to define the "test" model in the training process. In order to limit the problem like over-fitting, this approach provides an insight of how we doing the learning job using the above methods. One round of cross-validation involves partitioning a sample of data into complementary subset, performing analysis on one

subset, and validating the analysis on the other subset. Back to my project, this means the data has been partitioned to 500 training data and the other validation data. Using this method will help us compare the accuracy of Decision Tree and Bayes classification.

4.4 Tools and Platform

Project language platform: Python

Pandas[3] is an open source, BSD-licensed library providing high-performance, easy-to-use data structure. Using this library to convert csv dataset to in-memory data structure. A fast and efficient DataFrame object for data manipulation with integrated indexing is what I want. Besides it can quickly transform data to array object, which is the sample input for python scikit-learn.

Scikit-learn [4] is an open source machine-learning library for the Python programming language. It features various classification, regression and clustering algorithms. Since it was designed to interoperate with the Python libraries NumPy and SciPy, I can easily convert data using this package.

5. Experiment analysis and result

5.1 Dataset preprocessing

In this part, the job is to standardize the data to be used for the next step. From the dataset, it contains some missing data, from record 3342 to 3396, some fields of the record is missing, this will not yield a coherent result and will influence the precision of the classification. So I will just leave out these 54 instances.

Each record contains 20 fields in total, some attribute like the Game_ID make sense in the data, but since I am doing the classification work, I care more about the whole

dataset, regardless of a single instance, this field will not be taken into consideration. The challenging part is that some attributes are not completely independent with each other. Take the “MinimapAttacks” and “MinimapRightClicks” attributes for example, when a player notice the attacks on minimap, he will definitely use clicks and hotkeys to see what is happening and react with the best response he can make. This action will also influence his “APM” statistics. Therefore, using all three attributes to classified will influence the outcome because they have much overlapped part. When the records contain related attributes, I will use the standard deviation to measure the distribution of the two and choose the larger one to use for the following classification.

Also, records containing attributes related to PAC[5] (Perception Related Cycled) are left out. This is a derived characteristics about the screen movements that transformed to screen-fixations using dispersion-threshold algorithm[6]. The wellness of using this attribute will largely determined by the algorithm itself. In this project, we focus on the original data from the game. So fields related to PAC will also be emitted.

In all, the remaining fields are LeagueIndex, Age, HoursPerWeek, TotalHours, APM, UniqueHotkeys, and TotalMapExplored.

5.2 Decision Tree prediction

Split the LeagueIndex in another datasheet, Using python to read the data value and transfer them into array object.Luckily, We can use Numpy to convert the DataFrame object to an array object, this is used as the input for the decision tree methods. Then use scikit-learn python module to fit the data. We will also need to specify the classification method in DecisionTree. In this project I am using entropy calculation

to measure the attribute's wellness of splitting. As the tree has been produced, output its figure in pdf format for visualizing.

Note: After testing several times, setting DecisionTree's max depth over 6 or below will result in a decrease in correct prediction rate. Although such change will provide a perfect fit for the training dataset, this will do harm to the validation dataset.

Below figure 2 is a portion of the decision tree. The whole complete tree will be seen through the attachment. Notice that each record is in an x array corresponding to

[Age, hoursPerWeek, TotalHours, APM, UniqueHotkeys, TotalMapExplored]

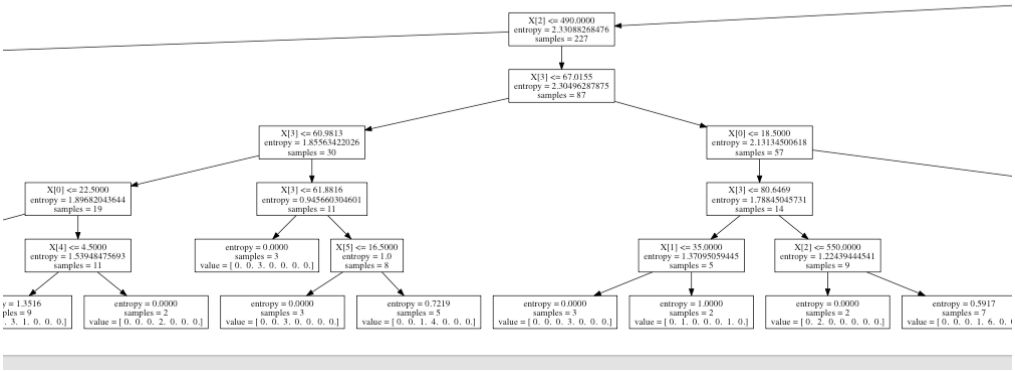

Figure 2 Portion of decision tree

5.3 Naïve Bayes Classification

Similar to the Decision Tree process about preprocessing the data, change the data classification method to Bayes Classsification.

```
'''This is for Bayes Classification'''
from sklearn,naive_bayes import GaussianNB
clf_Bayes = GaussianNB()
clf_Bayes.fit(x,y)
clf_Bayes.predict(z)
```

5.4 Comparison

From the Decision Tree method and Naïve Bayes classification on 2838 instances. The prediction outcome has been showing in the table.

Training Data

	Bronze	Silver	Gold	Platinum	Diamond	Master	GrandMaster	Total
Decision Tree	17	40	114	110	122	94	3	500
Bayes Prediction	10	126	20	164	68	101	11	500
True Value	24	56	87	119	117	90	7	500

Validating Data

	Bronze	Silver	Gold	Platinum	Diamond	Master	Grand Master	Total
DT	55	146	637	633	726	603	38	2838
Correct Predict	26 *47%*	32 *22%*	160 *25%*	217 *34%*	243 *33%*	243 *40%*	3 *11%*	924 *33%*
Bayes	33	627	149	994	385	614	36	2838
Correct Predict	14 *42%*	156 *25%*	31 *21%*	323 *32%*	124 *32%*	285 *46%*	6 *17%*	939 33%
True Value	143	291	466	692	687	531	28	2838

Note: First column of each table means after we applying the classification method, the DT (Decision Tree) and Bayes Classification's prediction occurrence number for each league. "Correct Predict" means if an instance is in league x, then the prediction successfully predict it is in league x. The following percentage means the precision of successful prediction. True Value means the supervised data classified in each league.

6. Conclusion

After training the data, both methods provide similar Correct Prediction rate for the validation dataset. We can see the similar prediction distribution in prediction and the real dataset. From the Decision Tree approach, we can clearly see the x[3], x[2], x[0] attribute is the best splitting attribute in each league, which corresponding to APM, TotalHourSpent and Age. These three attributes have great impact on the player's performance level. TotalMapExplored, HotKeys setting is not playing an important

role in the game as other factors. For players who want to get promotion quickly, considering improve in these aspects will be benifical.

For the prediction part, both methods didn't provide a high Correct Prediction rate. For most cases, methods have difficulty in predicting instances in the boundaries of each league. This can greatly influence the correctness of each input. For Decision Tree, it can be easily created as over-fitting or missing splitting attributes. It can be unstable because small variations in the data might result in a completely different tree being generated. For Bayes classification, since the data is not well distributed and the dataset may not be big enough, especially for players who are in the high level, their occurrence is naturally low comparing to players in other league. Then the frequency-based probability estimate will likely to be inaccurate and the precision and recall will be very low.

7. Reference

[1] Dataset http://archive.ics.uci.edu/ml/datasets/SkillCraft1+Master+Table+Dataset

[2]Bruce Geryk. "A History of Real-Time Strategy Games". GameSpot. Archived from the original on April 27, 2011.

[3] Pandas http://pandas.pydata.org/

[4]Scikit-learn http://en.wikipedia.org/wiki/Scikit-learn

[5]Thompson JJ, Blair MR, Chen L, Henrey AJ (2013) Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning. PLoS ONE 8(9): e75129

[6] Salvucci, D. D., & Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In Proceedings of the Eye Tracking Research and Applications Symposium (pp. 71-78). New York: ACM Press