Outcome Prediction of DOTA2 Based on Naive Bayes Classifier

Kaixiang Wang
School of Computer Science
Communication University of China
Beijing, China
wangkaxiang1996@163.com

Wenqian Shang
School of Computer Science
Communication University of China
Beijing, China
shangwenqian@163.com

***Abstract*—Although DOTA2 is a popular game around the world, no clear algorithm or software are designed to forecast the winning probability by analyzing the lineups. However, the author finds that Naive Bayes classifier, one of the most common classification algorithm, can analyze the lineups and predict the outcome according to the lineups and gives an improved Naive Bayes classifier. Using the DOTA2 data set published in the UCI Machine Learning Repository, we test Naive Bayes classifier's prediction of respective winning probability of both sides in the game. The results show that Naive Bayes classifier is a practical tool to analyze the lineups and predict the outcome based on players' choices.**

Keywords—Naive Bayes; Classification Algorithm; DOTA2

I. INTRODUCTION

Defense of the Ancients 2 (DOTA2) is an independent multiplayer online game, which developes from a map in the WarCraft3. DOTA2 swept the world with more than 50 million players. There have been over 50 million players of DOTA2, and hundreds of clubs and players participate in its global professional leagues every year, whereas no one has predicted the outcome of a game by analyzing the lineups in electronic sports field by now.

There are two teams, known as Radiant and Dire, that respectively consist of five players in each game. Players control one of 113 heroes (by December, 2016), each of which can be chosen only once. According to the Principle of Permutation and Combination, there are

$$C_{113}^{5}\times C_{108}^{5}\times\frac{1}{2}\approx 4.69\times 10^{17} \quad (1)$$

different lineups in the game. Different lineups have different features, as each hero has its own strengths and weaknesses. Thus, not only professional players but also new players need scientific analysis of lineups eagerly.

Bayes classifier, which is based on Bayes theorem, has a solid mathematical foundation and stable efficiency of classification. Since the algorithm was invented over 250 years ago, it has been in an unparalleled position in both mathematics and machine learning fields. Comparing Naive Bayes classifier with other machine learning algorithms including decision tree and neural network algorithms, Michie (1994) et al found that Naive Bayes classifier is competitive in learning algorithms in some specific cases. Bayes classification algorithm has two categories: one is Bayes network classifier, which is still developing, when considered the possible links between the various attributes; the other is Naive Bayes classifier, which assumes that the attributes are mutually independent.

In DOTA2, heroes are independent of one another, so there is no dependency and relevance among heroes.What's more, the game will not end in a draw, but absolutely decides victory and defeat. Therefore, we choose Naive Bayes classifier in our experiment. To examine the feasibility of it, we use the DOTA2 data set published in the UCI Machine Learning Repository to test Naive Bayes classifier's prediction of winning probability.

II. BAYES CLASSIFIER AND IMPROVEMENT

A. Bayes Classifier

A naive Bayes Classifier(nBC) is a probabilistic classifier which applies the Bayes' theorem with a strong (naive) assumption: it is assumed that the features describing the objects to be classified are statistically independent each other [8].

Suppose $x(c, a_1, a_2, \dots, a_n)$ is an item to be classified, which includes n feature variables $a_1, a_2, \dots, a_n$ .Bayes Classsifer is based on Bayes theorem, so we know:

$$P(C|a_1, a_2, \dots, a_n) = \frac{P(a_1,a_2,\dots,a_n|C)}{P(a_1,a_2,\dots,a_n)}. \quad (2)$$

Naive Bayes classifier is more advanced under the condition that all attributes are mutually independent. Equation (2) can be simplified to

$$P(C|a_1a_2\dots a_n) = \prod_{i=1}^{n} P(a_i|C). \quad (3)$$

Compared with other classifers, Naive Bayes classifier has the following features:

- When the attributes are mutually independent, this classification is accurate.
- Not only a small amount of parameters need to be estimated in Naive Bayes Classification model, but it is less sensitive to missing data than Bayes network classifier, so the algorithm of Naive Bayes Classification is relatively simple.

978-1-5090-5507-4/17/$31.00 ©2017 IEEE
ICIS 2017, May 24-26, 2017, Wuhan, China

- Naive Bayes algorithm is based on the conditional independent assumption, and the effect of the attribute value on a given class is independent of other attribute values.

B. Improvement

If the data item to be classified by Naive Bayes classifier satisfies the following conditions:

- The data item has many attributes, but the number of attributes the number of the attributes that determine the classification are significantly less than the number of attributes in total.
- Attributes are mutually independent.
- The attribute takes only two values or similar.

We can ignore the data that does not affect the classification so as to reduce the complexity of computations. Assuming V is a value of a dependent attribute, equation 3 can be rewritten as follow:

$$P(C|x) = P(C)\times\prod_{i=1}^{n} P(a_i = V\,|C). \tag{4}$$

III. Experiment And Analysis

A. Preprocess Data

The training set has 92,649 data in total, each of which has 117 attributes, shown as table I. The first attribute represents the result of a game with the value of 1 or -1. 1 means victory for the team Radiant and -1 represents victory for the team Dire. The second attribute to the fourth are the game ID information, having no impact on the result. Attribute 5 to the 117 represent the selection of heroes with the value of 1, 0 and -1. If the hero is selected by Radiant, the value will be 1, and -1 for team Dire. In addition, 0 means the hero has not been selected by any team.In this way, we retain game result attribute and hero selection attributes after processing through $\mathrm{DataFrame}(\mathrm{GameResult}, a_1, a_2, \ldots, a_{113})$.

TABLE I. Data Item

No.	*ID1*	*ID2*	*ID3*	*H1*	*H2*	*H3*	*…*	*H112*	*H113*
1	223	8	2	1	0	-1	…	0	-1

Figure 1 illustrates the frequencies of heroes in the training set, and figure 2 shows the teams' percentage of wins.

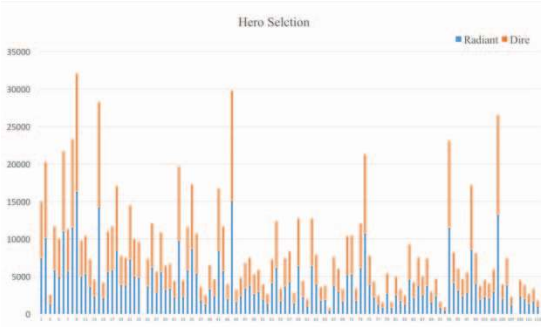

Fig. 1. Hero Selection in Training Set

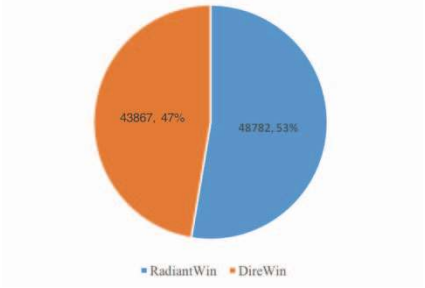

Fig. 2. Total Number of Wins for Each Team

B. Analyze Goal

To get forecast result, we calculate

$$P(\mathrm{Result}|a_1a_2\ldots a_{113}) = \frac{P(a_1a_2\ldots a_{113}|\mathrm{Result})}{P(a_1a_2\ldots a_{113})}. \tag{5}$$

As mentioned before, it can be simplified to

$$P(\mathrm{Result}|a_1a_2\ldots a_{113}) = \prod_{i=1}^{113} P(a_i|\mathrm{Result}). \tag{6}$$

C. Calculate Prior Probability

If the selection of heroes is a_i in a piece of data to be classified, its prior probability will be

$$P(a_i|\mathrm{Radiant}) = \frac{\mathrm{Count}(a_i)}{\mathrm{Count}(\mathrm{RadiantWin})}, \tag{7}$$

$$P(a_i|\mathrm{Dire}) = \frac{\mathrm{Count}(a_i)}{\mathrm{Count}(\mathrm{DireWin})}. \tag{8}$$

D. Calculate Posterior Probability

Assuming the result is x, we calculate probability of x is 1 or -1:

Winning probability for Radiant,

$$P(\mathrm{RadiantWin}|x) = P(\mathrm{RadiantWin})\times\prod_{i=1}^{113} P(a_i|\mathrm{RadiantWin}), \tag{9}$$

Winning probability for Dire,

$$P(\mathrm{DireWin}|x) = P(\mathrm{DireWin})\times\prod_{i=1}^{113} P(a_i|\mathrm{DireWin}). \tag{10}$$

E. Calculate Relative Winning Probability

Relative winning probability for Radiant:

$$P(\mathrm{Radiant}) = \frac{P(\mathrm{Radiant}|x)}{P(\mathrm{Radiant}|x)+P(\mathrm{Dire}|x)}, \tag{11}$$

Relative winning probability for Dire:

$$P(\mathrm{Dire}) = \frac{P(\mathrm{Dire}|x)}{P(\mathrm{Radiant}|x)+P(\mathrm{Dire}|x)}. \tag{12}$$

The Python code in Figure 3 calculates prior probabilities, posterior probabilities and relative winning probabilities.

```python
#Piror Probability
def Prior(data,result):
    count = 0
    for item in data:
        #first column is result
        if item[ResultColumn] == result:
            count += 1
    return count/len(data)

#Posterior Probability
def Posterior(data,ai,i,result):
    count1 = 0
    count2 = 0
    for item in data:
        if item[ResultColumn] == result:
            count1 += 1
            if item[i] == ai:
                count2 += 1
    return count2/count1


def CalculateProbability(dataTrain,dataTest):
    count = 0
    for item in dataTest:

        PRadiant = Prior(dataTrain,1)
        PDire = Prior(dataTrain,-1)

        for i in range(2,len(item)):
              PRadiant *= Posterior(dataTrain,item[i],i,1)
              PDire *= Posterior(dataTrain,item[i],i,-1)

        Radiant = PRadiant/(PRadiant+PDire)
        Dire = PDire/(PRadiant+PDire)
```

Fig. 3. Python Code 1

F. Improve Classifier

As we mentioned in part II, ignoring unselected heroes enables us to reduce the quantities of prior probabilities to be calculated from 113 to 10, which greatly reduce computational complexity.Figure 4 shows the optimized code.

```python
#Optimizing
def CalculateProbability(dataTrain,dataTest):
    count = 0
    for item in dataTest:

        PRadiant = Prior(dataTrain,1)
        PDire = Prior(dataTrain,-1)

        for i in range(1,len(item)):
            if (item[i]!=0)
                PRadiant *= Posterior(dataTrain,item[i],i,1)
                PDire *= Posterior(dataTrain,item[i],i,-1)

        Radiant = PRadiant/(PRadiant+PDire)
        Dire = PDire/(PRadiant+PDire)
```

Fig. 4. Python Code 2

G. Analyze Result

After testing 92649 pieces of data in the training set, we find that the accuracy of training set on training set is 85.33%.

Test set has 10294 pieces of data. The accuracy of test set on training set is lower than that of traing set on traing set. However, with the quantity of training set increasing, the correct rate tends to be stable at 58.99%.

TABLE II. ACCURACY OF 2 DATASET

	Training Set on Training Set	***Test Set on Training set***
Accuracy	85.33%	58.99%

It can be verifed that the difference of the accuracy between original classifier and optimized classifier is at most 1%, which means the optimized classifier is feasible. This is shown in figure 3.

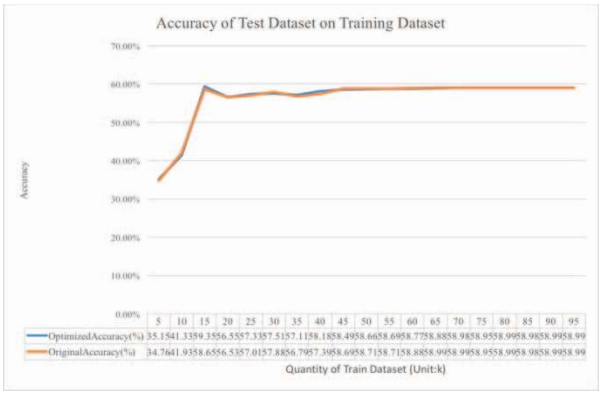

Fig. 5. Accuracy of Test Daset on Training Dataset

IV. CONCLUSION

Although the independent assumptions are often inaccurate in fact, some of the properties of the Naive Bayes classifier make it surprisingly effective in practice. The author provides a way to analyze lineups and froecast winning probability in Dota2 with Naive Bayes classifier, introduces the basic idea of how to analyze the game the Naive Bayes classifier and verifies the feasibility of analyze the game with quantitative data in Naive Bayes classifier model.

ACKNOWLEDGMENT

This paper is supported by National Training Programs of Innovation and Entrepreneurship for Undergraduates " Mobile Service System of Communication University of China for Students".

I would also like to express my sincere gratitude to Jiaqi Sun, my friend in Sun Yat-sen University, for revising the English version of this paper.

REFERENCES

[1] J.M. Li, L.H. Sun, Q.R. Zhang and C.S. Zhang, Application of naive Bayes classifier to text Classification, Journal of Harbin Engineering University, Vol. 24 No.1, Feb. 2003.

[2] Nino Prasetyo Hamal Pratama, Eko Mulyanto Yuniarno and Supeno Mardi Susiki Nugroho, Fuzzy Controller based AI for Dynamic Difficulty Adjustment for Defense of the Ancient 2 (DotA2), 2016 International Seminar on Intelligent Technology and Its Application (ISITIA), pp. 95-100.

[3] Mahit Mertiya and Ashima Singh, Naive Bayes and Adjective Analysis for Sentiment Detection on Twitter, 978-1-5090-1286-2, ISBN 2016.

[4] Z. Fu, Q. Chen, H. Zhang and S. Liu, Application of Naive Bayes Classifier in Stampede Risk Early-warning of Large-scale Activities, 2016 International Conference on Industrail Information - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), pp.174-180.

[5] Peter Harrington, Machine Learning in Action, Posts & Telecom Press.

[6] Tom M. Mitchell, Machine Learning, China Machine Press.

[7] Zhu Jun and Hu Wenbo, Recent Advances in Bayesian Machine Learning, Jounal of Computer Research and Development, 2015, pp.16-26.

[8] C. De Stefano, F. Fontanella, A. Scotto di Freca, A Novel Naive Bayes Voting Strategy for Combining Classifiers, 2012 International Conference on Frontieres in Handwriting Recognition, pp.467-472.