BETTING STRATEGY FOR THE CARD GAME TICHU

ERIC YANG
Stanford Physics department
Draft version December 14, 2018

ABSTRACT

The goal of this project is to come up with the optimal betting strategy for the game *Tichu*. *Tichu* is a card game that includes elements of *Bridge*, *Poker*, and *Big Two*. An essential element of the game is to correctly predict when you can go out first and make bets (call Tichu/Grand Tichu bets) to score points. In this project, we analyze the results from 15000 matches to come up with models that predicts whether the Tichu/Grand Tichu bets will succeed by looking at the starting hand. With the trained models, we then run simulations to compute the expected score of calling Grand Tichu vs. calling Tichu bets. This allow us to come up with the optimal strategy of when to call Tichu and when to call Grand Tichu. To obtain a strategy that is applicable by a human player, we reduce the input features to some basic features easily calculated by a human while retaining most of the prediction power. This is done by taking the features with the largest coefficients in the Logistic Regression model. The final result is that it is advantageous to call Grand Tichu bets when $N_{Ace} + 3 \times N_{dragon} + 3 \times N_{phoenix} + 3 \times N_{bombs} \geq 2$ in the first 8 cards hand, and it is on average better to call Tichu bets when $2 \times N_{Ace} - 2 \times N_{dog} + 6 \times N_{dragon} + 6 \times N_{phoenix} + 5 \times N_{bomb} + N_{straight} - N_{singleton\ small\ cards} \geq 7$. Where N_* is the number of each pattern or card present in the hand. The code for this project can be found on https://github.com/hungiyang/Project_Tichu.git.

1. INTRODUCTION

Tichu is a card game that includes elements of *Bridge*, *Poker*, and *Big Two*. The general goal of the game is to be the first to get rid of all the cards in ones hand. Two teams of two players play against each other to accumulate points, and the first team to reach 1000 points wins the match. At each round, a total of 56 cards are being dealt (each players getting 14 cards). When the first 8 cards for each player are dealt, players can look at the 8 cards and choose whether or not to call Grand Tichu, which is making a bet of 200 points that they will go out first. After all 14 cards per player are dealt, there is another option of calling Tichu, which is making a bet of 100 points that he or she will go out first (tic 2018). There are additional points that can come from the cards such as 5, 10, K, dragon, phoenix, but the majority of the points come from the Tichu/Grand Tichu bets. Therefore, we will ignore points from cards and just focus on points from Tichu/Grand Tichu bets in this project.

We are interested in predicting the game outcome based on the initial cards being dealt. In particular, we want to know the probability of making the Tichu/Grand Tichu bets based on the information of the initial cards of individual players. With a trained model to predict the probability of making the bets, we can run simulations to determine the best strategy of when to call Grand Tichu and Tichu bets.

2. DATASET AND FEATURES

The owner of the online tichu server https://onlinetichu.com is kind enough to provide the gameplay data recorded from their server. We have access to 14765 matches and 84127 rounds (A match is the whole process of playing up to 500 or 1000 points, and a round is each time cards are being dealt). There are 20484 Grand Tichus and 38057 Tichus bets called in this data set.

2.1. *Grand Tichu success rate prediction*

The first supervised learning problem that we want to solve is to predict the Grand Tichu success probability given the 8 cards hand. The input feature is the 8 cards hand in some kind of representation, and the output of the model is the prediction of whether Grand Tichu is made (being the first player to go out).

2.2. *Tichu success rate prediction*

The second supervised learning problem is to predict the Tichu success probability given the full 14 cards hand. In the actual gameplay, there are game mechanics such as card swapping and the option to call Tichu anytime before playing the first card. In our analysis, we ignore these complications and just take the 14 cards after swapping as the input. The input feature is some kind of representation of the 14 card hand, and the output is the probability of making Tichu (being the first player to go out).

FIG. 1.— An example of a 14 card hand. We want to predict the success rate of making Tichu with an input hand like this.

3. METHODS AND DATA PROCESSING

3.1. *Algorithms*

We applied classification algorithms including Naive Bayes, Logistic Regression, Random Forest, and Boosting.

Naive Bayes assumes that the features (hand cards or features) are conditional independent given outcome (making Tichu or not).

$$P(x_1, x_2, ...x_n|y) = \prod_{i=1}^{n} P(x_i|y) \tag{1}$$

where x_i can be the number of some card or pattern, and y is the outcome of making Tichu or not.

Logistic Regression uses a Sigmoid function to determine the probability of each outcome given a weighted sum of all the features. The weight θ is fitted to get the largest likelihood.

$$P(y=1|x) = h_\theta(x)$$
$$P(y=0|x) = 1 - h_\theta(x)$$
$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

Random Forests is an algorithm that combines multiple decision trees to get an estimator. For each decision tree, different bootstrap samples are used, and at each decision node, different subsets of features are used. These procedure tends to reduce the chances of overfitting.

Boosting takes a collection of weak classifier and weight their results in some way to create a stronger classifier.

Naive Bayes and Logistic Regression are our baseline model in which no hyper-parameter tuning is required after the features are specified.

3.2. *Data Processing and Feature Selection*

For the Tichu classification problem, our samples are biased strongly towards strong hands because in actual games players only call Tichu when they are confident in going out first. To make the model prediction better at evaluating weaker hands, I add in the following cases also as training data for the Tichu classification.

1. When a B team player calls Tichu and no one on A team calls Tichu. If the player on A team goes out first, consider him as a y=1 Tichu sample.
2. When a B team player calls Tichu and no one on A team calls Tichu. If both players on A did not go out first, consider both players' hand on A team to be y=0 Tichu samples.

The reasoning for this is that when an opponent calls Tichu and no one on your team did, the number one priority is to stop them from making Tichu. Therefore, all players on your team is on contending mode to go out first. If you or your teammate succeed, your hand can be considered a sample that can make Tichu (y=1 label). if both of you failed, they both hands on your team can be considered as hands that can't win Tichu (y=0 labels). Adding case 2 above in particular give us sample coverage of weaker hands so that we can be more confident about our prediction of weak hands.

For Grand Tichu classification, since there is the card passing part that would be significantly affected by the whether Grand Tichu is called, we only use the hand where Grand Tichu is actually called as training data.

3.3. *Detect Pattern as features*

In theory if the model is sophisticated enough and with enough data, the hand input itself contains all the information. However, since we only have few tens of thousands of samples, it is not quite sufficient to learn these connection on its own. Therefore, we tried to search for playable patterns such as 4 of a kind, royal flush, straights, three of a kind, pairs, etc and add them to the input vector.

Below we list the patterns that we search for either Grand Tichu or Tichu hand.

1. number of ace (0-4)
2. number of phoenix (0-1)
3. number of dragon (0-1)
4. number of dog (0-1)
5. number of bombs
6. number of three of a kind
7. largest three of a kind
8. number of pairs
9. largest pair
10. longest straight length
11. largest straight
12. number of tractors
13. number of single small cards

where tractors are consecutive pairs such as 3344 or 334455, bombs are four of a kind or straight flush, and single small cards are cards that are smaller than Q and does not belong to a straight, a pair, or a three of a kind.

Besides finding the model that gives the most accurate prediction, we also want to have a model that is easy enough for human players to calculate during actual gameplay. Therefore, we also looked at the importance of these features in our models and trained a model that only kept the minimal features that are easy to compute as a human without loosing too much prediction accuracy.

4. SUPERVISED LEARNING RESULTS

4.1. *Grand Tichu prediction*

For Grand Tichu, the noise is large and adding pattern detection essentially does not help the prediction accuracy. Below we list the results of combination of different subset of features vs. different models. The metric we choose to evaluate our model is the AUC, which is the area under the ROC (receiver operating characteristic curve). The ROC curve plots the true positive percentage vs. the false positive percentage. It shows how the model performs under different decision threshold. If the model is randomly guessing, the AUC will be 0.5, whereas if the model can classify all the samples correctly, the AUC will be 1.0.

The basic way to represent the hand (raw hand) is a length 56 vector of 0 and 1 indicating whether the hand contains that card or not. The compressed hand counts the number of cards of each value. This way we encode the symmetry information of suits and can potentially help the models. The Full Pattern are the 13 features listed in sec 3.3.

In Tab.1, the 4 features selected in minimal Pattern takes the 4 features that has the largest coefficients in the Logistic Regression results from the full 13 pattern fit. The four most important features are number of Aces N_A, dragon N_{dragon}, phoenix $N_{phoenix}$, and bombs N_{bomb}. Dragon and Phoenix are the individually strongest cards followed by aces, and bombs are the most powerful pattern in the game. Therefore, it is not surprising that they have the largest coefficients in the Logistic Regression fit. The coefficients for the four features for the minimal pattern fit are listed in Tab.2 when the coefficient for N_A is normalized to one.

	LR	NB	AdaBoost	RF
Raw hand	0.619	0.605	0.591	0.612
Compressed hand	0.624	0.619	0.609	0.627
Minimal Pattern (4 features)	0.626	0.552	0.625	0.627
Full Pattern (13 features)	0.628	0.582	0.622	0.620
Full Pattern + Compressed hand	0.624	0.595	0.618	0.622

TABLE 1
AUC FOR DIFFERENT MODEL AND FEATURES COMBINATION. LR: LOGISTIC REGRESSION. NB: NAIVE BAYES. RF: RANDOM FOREST. IT SEEMS THAT THE MINIMAL FEATURE HAS CONTAINS MOST OF THE PREDICTIVE POWER, AND ALL MODELS HAVE SIMILAR PERFORMANCE.

	N_A	N_{dragon}	$N_{phoenix}$	N_{bomb}
Coeff	1.0	2.8	2.8	3.4
Approx	1	3	3	3

TABLE 2
COEFFICIENTS FOR THE 4 MOST IMPORTANT FEATURES FOR GRAND TICHU PREDICTION.

It is interesting that Dragon and Phoenix are almost three times as important as Ace for calling Grand Tichu. We did not expect that from our personal gameplay experience.

We round the coefficients to the nearest integer to obtain a Grand Tichu index $I_g = N_A + 3N_{dragon} + 3_{phoenix} + 3N_{bomb}$ that is easy to compute during actual game play. If we train a Logistic Regression using only I_g, the AUC is ≈ 0.62, and $I_g \approx 0.87$ correspond to 50% probability of making Grand Tichu. This means that when your 8 cards hand has one Ace, there is above 50% chance to go out first if you call Grand Tichu.

In the Fig.3 below, we show the training set and test set AUC for the four models using the input feature of 13 identified patterns plus the compressed card hand.

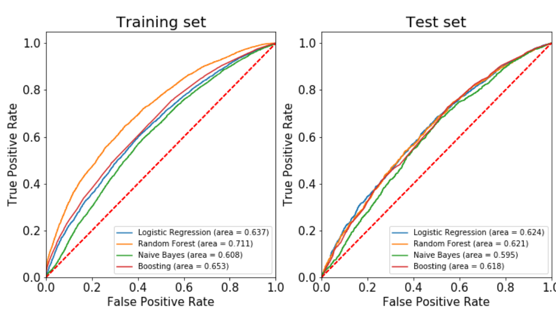

FIG. 2.— With input as 14 identified patterns plus card hand. Naive Bayes performs worse than the other models.

4.2. *Tichu prediction*

For Tichu bet predictions, we use a similar approach as for Grand Tichu. However, since the randomness is smaller in the Tichu problem because the information of all 14 cards are available, identifying the patterns becomes more useful. In the Tichu prediction, the minimal features we decide to keep are number of ace N_A, dog N_{dog}, dragon N_{dragon}, phoenix $N_{phoenix}$, bombs N_{bomb}, staights $N_{straight}$, and individual small cards N_{small}. The coefficients in the Logistic Regression when these features are used are listed in Tab.3 (Coefficients for N_A normalized to 2.0).

The results of training with different model/features combinations are shown in Tab.4 below.

In the Tichu predictions, we see that the accuracy improves when the patterns are added as features. The full pattern alone

	N_A	N_{dog}	N_{dragon}	$N_{phoenix}$	N_{bomb}	$N_{straight}$	N_{small}
Coeff	2.0	-1.9	5.9	5.7	4.7	1.0	-1.1
Approx	2	-2	6	6	5	1	-1

TABLE 3
COEFFICIENTS FOR THE 6 MOST IMPORTANT FEATURES FOR TICHU PREDICTION.

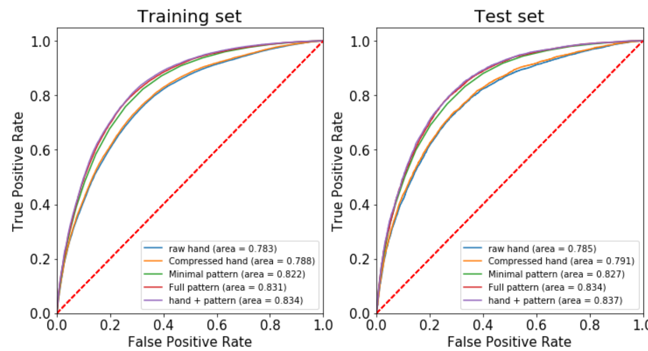

FIG. 3.— With input as 14 identified patterns plus card hand. Naive Bayes performs worse than the other models.

	LR	NB	AdaBoost	RF
Raw hand	0.785	0783	0.786	0.783
Compressed hand	0.791	0.789	0.804	0.802
Minimal Pattern (6 features)	0.827	0.827	0.82	0.827
Full Pattern (14 features)	0.834	0.74	0.832	0.837
Full Pattern + Compressed hand	0.837	0.80	0.833	0.835

TABLE 4
AUC FOR TICHU PREDICTION WITH DIFFERENT MODEL/FEATURE COMBINATIONS.

performs as well as full pattern plus hand as input. The predictive power of just the minimal 6 most important input (N_A, N_{dog}, N_{dragon}, $N_{phoenix}$, N_{bomb}, $N_{straight}$, N_{small}) is slightly worse than using the full pattern, but still significantly better than just using the hand as input. Another interesting observation is that the models achieve better performance using compressed hand as input vector compared to the raw hand. This is likely because the compressed hand encodes the symmetry of the 4 suits, whereas the raw hand represents each card by a binary 0 and 1 and does not differentiate between card value and suit.

For ease of computation during actual games, we round the coefficients of the 6 most important features to the nearest integer to create a Tichu Index defined as $I_t = 2N_A - 2N_{dog} + 6N_{dragon} + 6N_{phoenix} + 5N_{bomb} + N_{straight} - N_{small}$. Training a Logistic Regression model with the Tichu index I_t, we achieved an AUC ≈ 0.83, and an $I_t = 6.4$ correspond to a 50% probability of making Tichu. Therefore, there is a positive expected return to call Tichu when I_g of the hand is 7 or greater.

5. SIMULATION COMPARING GRAND TICHU AND TICHU EXPECTED RETURN

To decide whether to call Tichu, it is sufficient to consider the probability of making the bet. If the probability P_{tichu} is higher than 50%, than it is advantageous to call Tichu. However when deciding to call Grand Tichu, there is the added complication that even when your probability of making Grand Tichu P_{grand} is higher than 50%, the expected score for waiting and see the full 14 cards hand might be higher. For example, at 8 cards, you might think the hand is not bad, but at 14 cards you realize that there is no chance of making

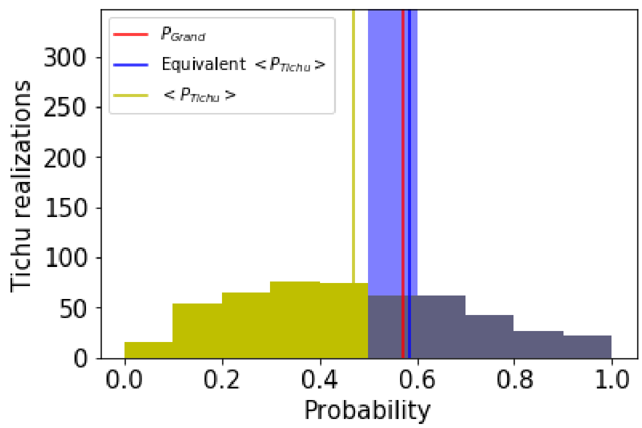

FIG. 4.— An example of the histogram of Tichu success probabilities given a fixed 8 card starting hand. Each count in the yellow histogram is a random realization of the next 6 cards, and the probability is the output of our Tichu Logistic Regressoin model given the 14 card hands. Since when $P_{tichu} < 0.5$, players won't call Tichu, the histogram that is smaller than 0.5 is equivalently at 0.5, since $P_{tichu} = 0.5$ has expected return 0 when you call Tichu. The expected histogram is plotted in blue. The equivalent Tichu probability $\tilde{P}_{tichu}$ is therefore the average of the blue histogram. The red vertical line is the Grand Tichu probability P_{grand}, the yellow line is the mean Tichu probability $\langle P_{tichu}\rangle$, and the blue line is the mean equivalent Tichu probability $\langle \tilde{P}_{tichu}\rangle$

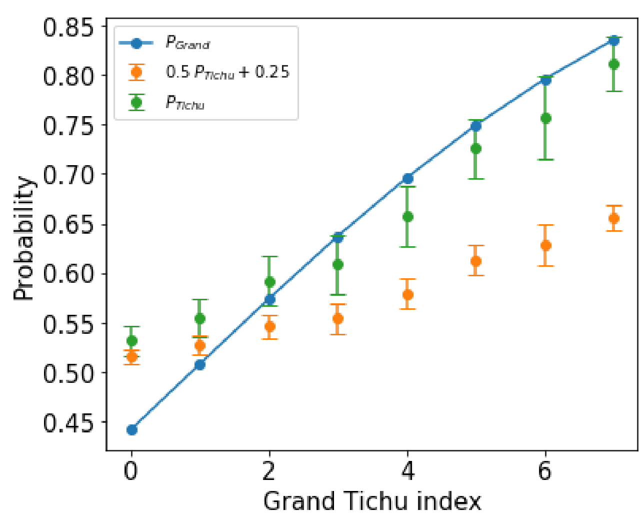

FIG. 5.— P_{grand}, $\langle \tilde{P}_{tichu}\rangle$, and $0.5\langle \tilde{P}_{tichu}\rangle + 0.25$ at different Grand Tichu index I_g

the bet. If you wait until 14 cards, you can eliminate some scenarios where you lose 200 points, however, in scenarios where you actually have a good hand, your return goes down from 200 to 100.

Given an input 8 card hand, to evaluate quantitatively the expected return for calling Grand Tichu vs. waiting for 14 card hand, we run simulations of random realizations of the next 6 cards. Fig.4 is an example of random realizations of Tichu hand success rate. For the hands that have Tichu success probability P_{tichu} less than 50%, we won't be calling Tichu, and the expected return is 0. Therefore it is equivalent as $P_{tichu} = 0.5$, and we define the mean equivalent Tichu probability $\langle \tilde{P}_{tichu}\rangle$ as the average of the histogram where the $P_{tichu} < 0.5$ is taken as 0.5. Given a 8 card hand with P_{grand} chance of winning Grand Tichu, we want to compare the expected return from calling Grand Tichu and waiting to call Tichu.

$$
\begin{aligned}
E[Score_{grand}|hand] &= 200 \times (P_{grand} - (1 - P_{grand})) \\
&= 200 \times (2P_{grand} - 1) \\
E[Score_{tichu}|hand] &= 100 \times (\langle \tilde{P}_{tichu}\rangle - (1 - \langle \tilde{P}_{tichu}\rangle)) \\
&= 100 \times (2\tilde{P}_{tichu} - 1)
\end{aligned}
$$

The hand here means a fixed 8 card hand. Since different 8 cards hand with the same P_{grand} can have different $\langle \tilde{P}_{tichu}\rangle$ distributions, we also run many realizations of 8 cards Grand Tichu hand to compute the expectation.

$$
\begin{aligned}
E[Score_{grand}|P_{grand} = P] &= 200 \times (2P - 1) \\
E[Score_{tichu}|P_{grand} = P] &= \langle 100 \times (2\langle \tilde{P}_{tichu}\rangle - 1)\rangle_{P_{grand}=p} \\
&= 100 \times (2\langle \tilde{P}_{tichu}|_{P_{grand}=P}\rangle - 1)
\end{aligned}
$$

Then we can compare $E[Score_{grand}|P_{grand} = P]$ and $E[Score_{tichu}|P_{grand} = P]$ to see whether calling Grand Tichu is better or waiting till 14 cards is better. We then get

$$
\begin{aligned}
\text{Grand Tichu if } P_{grand} &> 0.5\langle \tilde{P}_{tichu}|_{P_{grand}=P}\rangle + 0.25 \\
\text{Wait if } P_{grand} &\leq 0.5\langle \tilde{P}_{tichu}|_{P_{grand}=P}\rangle + 0.25
\end{aligned}
$$

We run 10000 realizations of 8 cards Grand Tichu hand, and each with 100 realizations of 14 cards Tichu hand. Fig.5 below shows P_{grand}, $\langle \tilde{P}_{tichu}\rangle$, and $0.5\langle \tilde{P}_{tichu}\rangle + 0.25$ at different Grand Tichu index I_g. If the blue dot is above the yellow dot ($P_{grand} > 0.5\langle \tilde{P}_{tichu}|_{P_{grand}=P}\rangle + 0.25$) than it is better to call Grand Tichu, whereas if the yellow dot is on top ($P_{grand} \leq 0.5\langle \tilde{P}_{tichu}|_{P_{grand}=P}\rangle + 0.25$), then it is better to wait to see the next 6 cards.

From Fig.5, we see that when $I_g \geq 2$, it is better to call Grand Tichu. This means that we should call Grand Tichu when our 8 cards hand has at least two Ace. Even though an one Ace hand has $P_{grand} > 0.5$, it is better to hold off on calling Grand Tichu and see the rest of the cards to decide whether to call Tichu or not.

6. CONCLUSIONS AND FUTURE WORK

For Grand Tichu predictions, it seems that the uncertainty is very large and the individual strength of the cards give pretty much all the information. For Tichu, adding the features of important connections between cards (identifying patterns such as straights, bombs etc) significantly increases the prediction accuracy. This suggests that with an even larger data set, the more complicated models might be able to learn the connection themselves. For the dataset we used, models that are more complicated than Logistic Regression tends to overfit. If we have access to more data, models such as Random Forest and Boosting should perform better than Logistic Regression.

In our simulation we evaluate randomly generated hands with our trained Logistic Regression model. However, The model is trained with data that human players would call Tichu or Grand Tichu, and the distribution might be different from randomly generated hands. Therefore, when running the simulation, the probability estimates for the randomly generated hands might be biased.

There are many game mechanics that we did not consider in our analysis. For example, there is card passing between all players which would on average increase the power of each hand and cause the distribution of cards to change. Also cards like 5, 10, K, dragon, and phoenix are worth points by themselves, and we did not take that into consideration. Another important game mechanics is called one-two, which is when both players in one team going out before anyone on the other team does. All of these would change the optimal strategy and should be considered in future analysis. Also, an alternative and maybe more straightforward approach is to train a regression model that predicts the score outcome instead of classification models that predict the betting outcome.

REFERENCES

2018, Tichu game description,
https://boardgamegeek.com/boardgame/215/tichu