Data Mining Poker Hand Histories

Aron Andersson, Johan Karlsteen and Rickard Andersson

{c02arn,c01jkn,c01ran}@cs.umu.se

Abstract

The goal of this project was to gain experience in data mining using different techniques and tools on a large database with poker hand histories. We found that the existing data mining programs were not able to do the things we wanted and we had to write our own SQL queries. We developed a way of analyzing a player's style and guess what holecards he has based on his actions. The whole process made us realize that data mining is not the answer to everything, but with the right knowledge and resources it can help you discover information hidden deep inside your data.

1 Introduction

Data mining[3] is a widely used technology today, and the aim of this project was to find out what techniques it consists of and to apply these techniques on poker hand histories. The game we wanted to examine data from was "Texas Hold 'em, Fixed Limit". The reason we chose fixed limit was because we thought it would give a more predictable playing style than "No Limit", where people can bet all of their money at any time.

We have tried to perform some basic data mining on a database containing poker hands (events taking place at a poker table, one for each player and playing session). Our goal is to discover some winning strategies for a successful poker player and ultimately be able to foresee what cards a player has based on his action during the game. There are a lot of commercial applications on the market capable of data mining, but is there really a one stop solution? We have tested some applications trying to answer this question. We have also created a simple web interface so that a user may ask questions to the database.

2 Approach

2.1 Gathering data

Before data mining can be performed, data is needed. A program called Poker Tracker[1] gathered information from the online poker site PartyPoker.com for several days and stored the information in an Access database. Information about 170000 hands were collected from different players (showing information about their cards only if shown in the game), and 10000 hands from one single player (always showing information about cards on hand).

The most interesting relation (the `hands` relation) had 90 different attributes. Since this data was not our own, we had to analyze what the different attributes meant and classify what values they could take on. This was done by examining different scenarios, drawing our conclusions from the combined attribute values. Most of the attributes were represented as integers, even if they just represented boolean values, posing another problem during this work.

To allow for easier access and multiple users, we converted the database to PostgreSQL. Unnecessary and redundant attributes were removed and new ones created. For example, an attribute `flop_top_pair` was added, indicating whether the the player got a pair with the highest card laid on the table at the flop. To create these new attributes several C functions were compiled as .so-files and loaded into the database. We chose to use C instead in favor of PL/PgSQL, mostly because of speed issues. Some hands had missing values, so these also had to be fixed to be able to execute the queries later on.

2.2 See5

We tried to use the commercial data mining application See5 [2](C5.0 under *nix). According to the web page, See5 is used in over 40 countries, helping users transform data into knowledge. See5 uses a textfile to describe the data and another comma-separated textfile for the input. Supposedly, it could create decision trees and rules, and the goal was to build this decision tree into our user interface to be able to advise the user. However, no matter how narrow we described the data, See5 never managed to produce anything more interesting than what could be seen as obvious. Moreover, the C source code provided to read and interpret the classifiers and models wouldn't even compile, so we had to look elsewhere for a solution. The manual for the system is far from complete and quite old too. However, there are surely other areas where See5 can be used more efficiently than in this case (some examples are given in the binary package).

2.3 Writing or own SQL queries

As the See5 approach failed we started to look for other data mining tools available but found none that fit our task. So the option left was to build our own SQL queries and try to extract information that way. The first thing we wanted to do was to describe different playing styles with continuous variables. From that we would be able to see in what interval a player should be in to have the highest chance of being a winning player. The two factors we chose to consider was how loose and how aggressive a player is (see the section 3.1 below for further explanation). The next thing we wanted to do was to be able to analyze a specific player and predict the player's actions based on how he or she had played before. It soon became obvious that we couldn't analyze other people as well as we wanted, because we never knew what cards they folded. We therefore had to change our approach and only analyze hands we had played ourselves, by that way we knew what the cards where even though the player folded. A web interface was developed to easily create different scenarios (see section 3.2). The next two sections will describe exactly how the player's hole cards are ”guesstimated”.

2.3.1 Pre-flop hand analysis

The algorithm to calculate the probabilities for a player's hand based on his actions is as follows: For each possible hand H_n:
A_n = The actions the person has done preflop (input from web interface)
P_n = The probability that the hand H_n will be dealt of all possible hands
X_n = Percentage of times the player does actions A with hand H_n
$Y_n = P_n \times X_n$ (Probability player is dealt hand H_n & performs action A_n)
$W_n = Y_n / \sum_{i=0}^{n} Y_i$
Wn is the probability the player has hand H_n.

2.3.2 Flop hand analysis

The flop analysis is solely based on the player's actions on the flop and doesn't consider the cards on the flop nor the player's preflop actions. When the player's actions on the flop have been specified an SQL query is built that looks at all hands in the database where the player has played in the same way. For each hand that fit the requirements it is determined if the player in that hand had one pair, two pair or overcards and so on. From this information it is then easy to calculate the probability that the player has a pair and so on.

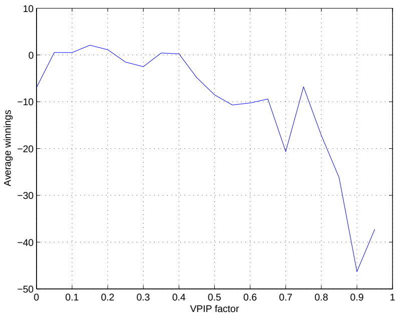
3 Results

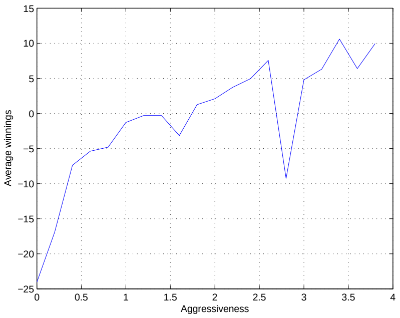
3.1 Winning factors

A player's tightness or looseness can be measured by how many times he voluntarily puts money in the pot (without being forced to pay the blind fee). Someone who does this for 20% of his dealt hands is considered a tight player, whereas 50% or higher is considered loose. The following graph depicts the relation between the VPIP factor (how often one voluntarily puts money in put) and average winnings for the observed players.

Another interesting factor is the aggressiveness of a player, which is determined by a player's raise percentage plus his bet percentage divided by his call percentage: $\frac{P_{raise}+P_{bet}}{P_{call}}$ If a player bets and raises a lot compared to the number of times he just calls other people's bets, he is considered an aggressive player.

We examined the VPIP factor and aggressiveness for all 4000 examined players, disregarding the ones who had played fewer than 25 hands. The following graphs depict the relations between the average winnings and the previously mentioned factors. One can see that a VPIP factor at around $0.15 - 0.20$ combined with a high aggressiveness yields the best winnings.





3.2 Predicting cards

The result is a web interface written in PHP that queries the database and presents the results. The interface calls a command line program that executes large SQL queries. The web interface can be found at:
`http://www.cs.umu.se/ c02arn/db/poker/`. To use it, select whether to predict actions during pre-flop or flop, select the actions your opponent has taken and then click the submit button to get the results. For the pre-flop choice, possible cards are displayed along with the percentage they have occurred, given the actions selected. The flop choice calculates the possibilities for different card combination the user has had while performing the selected actions (note that some combinations overlap).

4 Discussion

The results showing that the best strategy is to voluntarily put money in the pot between 15% and 20% of the time together with being very aggressive is close to exactly what the experts advocate on different internet poker forums. This means that our work produced good results. It also means that we discovered facts that others already knew without any data mining (or perhaps that is how they figured it out).

The other part of the project where we tried to calculate the probabilities that a player had a specific hand based on his previous actions is harder to draw any real conclusions from. We can analyze ourselves as players and get information like ”when I raise on the flop I have top pair 39% of the time”. But to know this for other people is what would be really interesting. To actually have access to the database of a real poker site and be able to analyze all the hands players are folding would be a real goldmine.

The next step to get a more accurate estimation of the player’s hand would be to combine the cards on the flop with the player’s actions on the flop with the preflop probabilities for each hand. But doing this is not trivial and some quite advanced combinatorics and statistics would be required. It may also be possible to analyze the player’s actions on the turn and river (the fourth and fifth card) combined with the previous actions, but to do this accurately would be even harder (and probably yield huge SQL queries).

5 Conclusions

The first thing we realized when we started the project was how important it is to know your data from the inside out. It is obvious that you should know the exact structure of the databases you are going to analyze, but it is also important to know what data is redundant information and what data that should correlate with each other. With this information it is much easier to determine what you want to get out of the data mining process and how to approach the problems that will arise.

The second thing that became apparent was that the data mining tools available were tailored to analyze a special kind of data and give a special kind of facts over that data. All the programs we tested lacked the customization options we needed to analyze our database in a meaningful way. The solution was to create our own SQL queries that gave statistics over different correlations in the database. And creating your own tools and techniques is probably what it comes down to in most projects which are not similar to the most common areas of data mining (sales, insurances, etc).

Our third and final issue with our project was the sample size. As previously mentioned, we had a fairly large database with 170000 hands stored in it. The enormous number of possible situations that can arise in poker made it impossible to analyze very specific situations as we most often only had two or three instances of that exact situation in the database. Our data was also skewed; when we collected it we were only able to retrieve the hands where the player went to showdown with the hand. In other words, we couldn't know what cards the player had when he folded. This made the whole process a lot harder and we had to abandon some of the ideas we had.

So to summarize what he have learned in this project it should be said that data mining is not some kind of magic wand you point at your data and get all these fantastic revelations. Although, if you have the right knowledge and resources and know what you are looking for, data mining can be a very effective tool in discovering statistics and information hidden deep inside your data.

References

[1] PJI, Inc. Poker tracker. Webpage, 2005. `http://www.pokertracker.com/`.

[2] RuleQuest Research. See5. Webpage, 2004. `http://www.rulequest.com/see5-info.html`.

[3] Two Crows Corporation. Introduction to Data Mining and Knowledge Discovery. Webpage, 1999. `http://www.twocrows.com/intro-dm.pdf`.