Applied Data Analysis : Betrayal for the win

Arthur Bricq
Robotics
EPFL
Montagne, Suisse
arthur.bricq@epfl.ch

Taavet Kangur
Robotics
EPFL
Ecublens, Switzerland
taavet.kangur@epfl.com

Diane Marquette
Robotics
EPFL
Ecublens, Switzerland
diane.marquette@epfl.ch

***Abstract*—In this work, we focus on the orders sent in the tabletop strategy game Diplomacy in order to predict the event of a betrayal and the effect of betrayal on winning chances. The work includes the data wrangling and distillation until the needed features can be extracted for our prediction.**

***Index Terms*—data analysis, friendship, betrayal, human interaction**

I. Introduction

Data analysis is a multifaceted complex challenge depending on the data available, the cleanliness of this data, the questions asked and finally the interpretation of the results. In order to give us a full fledged example to apply our knowledge to, a new unseen dataset is chosen to force us to work on the nitty gritty and encounter the problems a data analyst would encounter during everyday work.

In this study, games of the tabletop and online game *Diplomacy* are analysed. As this is a social game, each order and message have their meaning and can effect the course of the game.

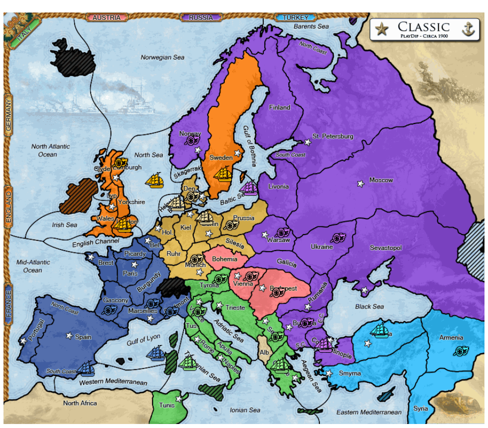

Fig. 1. The diplomacy game map [1]

The game is composed of turns inside of seasons in which the players can issue orders to their troops. Those orders are : Move, Hold, Convoy, Support, Build, Retreat and Destroy. This work is focused on the effects of betrayal on the player(s) in the game and their predictability using machine learning.

The map of the game *Diplomacy* is shown on Figure 1 and marks the supply center, which 18 of are needed to win the game, as stars.

A. Definitions

The definition of a betrayal, and also, of a friendship come from the paper ”Linguistic Harbinger of Betrayal” which is talked more in detail in the next section. Here they are :

- Act of friendship: a successful *SUPPORT* order from one player to another one.
- Act of hostility: a successful *MOVE* order from one player to another one.
- Friendship: a relationship between two players consisting of at least four acts of friendship emitted in a given time interval of four seasons, with at least two acts coming from each player. Once a friendship is created, it remains existing until it gets broken.
- Betrayal: a hostile act against a player with who you are engaged in a friendship.

B. Research questions

The research questions looking to be answered are :

- What’s the probability of winning if you betray your allies?
- How does it change depending on the number of allies betrayed?
- What’s the probability of winning if you have been betrayed?
- How many players did a winner betray on average?
- Can a classifier be trained to predict a betrayal based on orders ?

II. Related work

This work follows a first analysis done on the paper ”Linguistic Harbinger of Betrayal” [2] which focuses on the risk of being betrayed by another player with whom one is engaged in a stable friendship, depending on the messages sent between the player. Close attention to each word is given by evaluation of its politeness score and the mean over the seasons preceding the betrayal. Other aspects such as talkativeness, planning discourse and sentiment are also analysed to predict a betrayal. As the dataset used in this work does not contain messages, the focus is brought to each player's orders instead of their messages.

III. Data Collection

The used dataset is DiplomacyBoardGame - dataset by maxstrange [3] and contains just under 21200 games, with all the orders issued by each player to their troops but no messages between players. This has the added difficulty of needed to parse in a subset of the data is order to have processing times which are manageable.

IV. Methods

Our goal is to see if we can predict betrayals using elements from the game. The analysis is divided in several tasks. First an analysis is performed to determine whether it is possible to do such a study using it. Once concluded would be able to extract **game features** from the data, we could work on our data-processing pipeline. For each game, it is crucial to :

1) Detect **acts of friendship** and **acts of betrayal**.
2) Deduce **friendships** and **betrayals**.
3) For each friendship, extract the **features** for our classifier.

A. Wrangling with the dataset

The dataset we use is massive . A **qualitative class diagram** is made to illustrate the different fields and the class association that matter. [1]

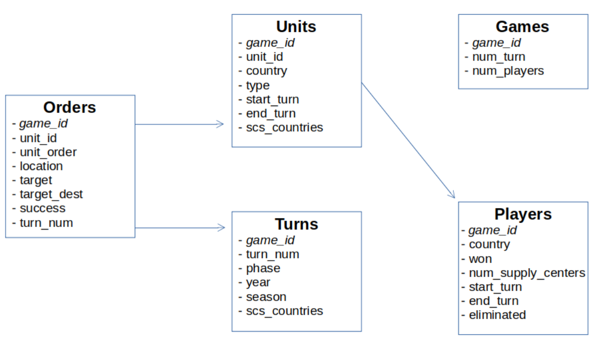

Fig. 2. Qualitative class diagram

A player aware of the rules would understand that the **structural element** of one game are the **orders**. Orders are at the very beginning of the entire data-wrangling step, since they represents all the **actions** done by the **units** of each **country**.

We'll give an illustration of how to use all those instances with associations. Given an MOVE order (it means a country displaced one of its unit from one **location** A to another location B), we want to know whether this order is aggressive (it is the case if there is at location B another unit from another player) or if it is just a simple motion. In order to know if there is a unit already present where current unit is moving, one must look at all the **successful previous orders passed by another players** during the same game **towards the same location**. However, orders don't include the *country* field and therefore it is not straightforward to know to which country commanded the move. To do so, one must first query several **unit id** over *orders* and then query a *country* over *units*. This illustrates the problem of finding the **targeted country** of a given order: the country - if there is one - that owns the target location of the order. Our approach to solve this is to pre-process all the orders of a game by computing in advance all the **emitting countries** and the **targeted countries**. Another pre-processing step consists of extracting the **season** of each move, which is contained inside the **turns** dataframe. This information is crucial for defining friendships.

Once pre-processing of a game is finished, the **orders** frame is much more human-readable: each order has become a clear action done by one player, at a given season, eventually targeting another player.

It is now easier to recognise **acts of friendship** and **acts of hostility**, that are themselves used to define **friendships** and **betrayals**.

B. Feature generation

Once the data processing is finished, each game has a **friendships matrix**. It's composed of as many rows as there are seasons in this game, and as many columns as there are possible friendships [2]. It is a binary matrix with 1s when 2 players are engaged in a friendships and 0s when there are not. Finally, when a friendship is broken, a letter signifies which country emitted the betrayal.

Betrayals must be detected using the current state of a game. The inputs of the classifier are seasons of one game characterized by some features describing the context of the friendship at this moment of the game. Computing and selecting the features is key in designing a good classifier. After several iterations, the list of features computed for each game is as follows :

- Length of the friendship at this season
- Number of acts of **hostility** of each of the two involved players towards all other players at this season
- Number of acts of **supports** of each of the two involved players towards all other players at this season
- Average number of mutual supports of the two involved players towards each other for the last three seasons

Season's features are computed for one player X (the betrayer) being friend with another Y (the victim). If the selected game doesn't have a betrayal, X and Y are sampled randomly [3].

C. Classification

The goal of the classifier is to predict if player X will betray in the next season its current friend given a season described

[1]This class diagram is just for illustration purpose and is not rigorous. For instance, we do not show all the associations towards the 'Games' class, where really each instance of the dataset points towards one game.

[2]7 players leads to 21 different possible friendships

[3]Though this situation doesn't happen in our training set.

by its 8 features. For this task, we trained a **logistic regression classifier** over a training set that we constructed from a random selection of 200 games of a certain length [4] where a betrayal happens at least once. Features are normalized before training and their means and standard deviations are saved to normalize the testing set later on. A testing set is generated separately from the training set and contains 20 games selected as previously explained.

V. Results

A. Probability of winning

As seen on Figure 3, betrayers have a much higher chance of winning but also that betrayed players win more than player that are in a stable friendship over the game. This means that a betrayed player gets more during the seasons with the future betrayer than a player in a friendship.

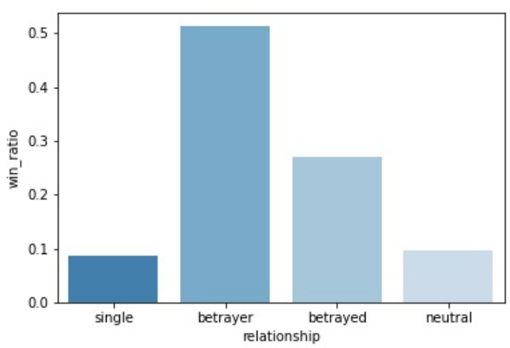

Fig. 3. Barplot of win ratio depending on actions during the game

B. Serial betrayers and betrayed betrayers

Out of a subset of 500 games, 56 contained betrayers and out of those, nine contained players who betrayed twice in a game. This is a rare occurrence and even though in four out of those nine games the serial betrayers won, this sample size is far too small to make any conclusion on the win rate of a serial betrayer.

In the same subset, only one betrayed betrayer appears but the player is betrayed before betraying and wins the game with one of his betrayers.

C. Classifier

The best classifier obtains a pseudo R^2 score of 0.07 and an accuracy of 85%. Though this result seems good, it is to be contrasted with the class imbalance of the training set. As our inputs correspond to seasons of a game, since a betrayal only happen at the end of a friendship there are many more data points where a betrayal will not happen the next season. Actually, 87.5% of the seasons have as outcome '0'. Hence, our classifier performs worse than a dummy classifier which would always classify the season as 'no betrayal'[5]. Purely speaking in term of accuracy isn't well suited when it comes to such a high imbalance. It is worth looking at the **confusion matrix**(Figure 4) of our classification.

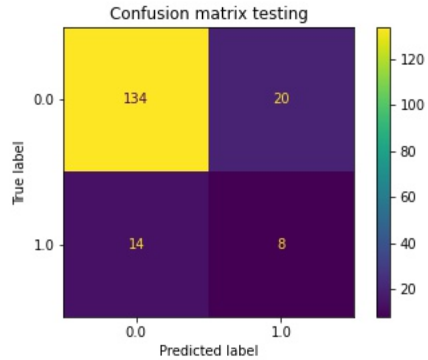

Fig. 4. Testing confusion matrix

We see that only 8 betrayals are correctly classified, that 20 normal seasons are misclassified as betrayals and that most seasons are classified as no betrayals even if there is actually a betrayal. Hence, it is safe to say that this classifier has really poor performances.

VI. Conclusion

Betrayal plays an important role in *Diplomacy* and as Figure 3 shows, being betrayed often leads to a better chance of winning than being in a friendship or being in none.

Having worked from start to finish on a large unexplored dataset has shown us the messiness that could be generated and how to clean it up for data analysis.

Despite the poor results obtained by our classifier, we believe that this work has been really interesting and that given more time, we would have tried to obtain better classification results. Working on the data-processing is very time-consuming and detecting friendships turned out to be harder than expected. The enormous size of the dataset, containing millions of rows from 5 different databases with for each many relationships to others, made the work longer than planned and we didn't have as much time as we wanted for **better feature design**, or for a proper **feature selection algorithm**. It was necessary to invest time assessing our current results and we had to stop before making them a *better than dummy*. There are also several observations that we made from our predictions such as that one same friendship has many betrayals classified, which is impossible, that we couldn't use in an iterative design procedure ; but that for sure would help a lot in designing a better classifier.

References

[1] Map of diplomacy. https://www.pinterest.ch/pin/123356477273107481/.
[2] Linguistic harbinger of betrayal. https://arxiv.org/pdf/1506.04744.pdf.
[3] Diplomacyboardgame. https://data.world/maxstrange/diplomacyboardgame.

[4] We ask to have more than 50 turns in the games, in order to increase the average number of friendships per selected game

[5] It is also worth noting that the accuracy of our classifier over the training set is of 86%, which is also not above the dummy classification