Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

Collusion Detection in Online Bridge

Jeff Yan
School of Computer Science
Newcastle University, UK
jeff.yan@ncl.ac.uk

Abstract

Collusion is a major unsolved security problem in online bridge: by illicitly exchanging card information over the telephone, instant messenger or the like, cheaters can gain huge advantages over honest players. It is very hard if not impossible to prevent collusion from happening. Instead, we motivate an AI-based detection approach and discuss its challenges. We challenge the AI community to create automated methods for detecting collusive traces left in game records with an accuracy that can be achieved by human masters.

Introduction

Contract bridge is a four-person card game played between two partnerships. Unlike chess, in which all pieces are on the board and known to each side, bridge is a game with hidden information. A player knows only a subset of 52 cards in the process of bidding and card play. She cannot tell any card or intention to her partner other than through predefined, publicly-known *conventions* to convey information. Information exchanged in this legitimate way is usually imperfect, and it may be plausible or wrong. However, collusive cheaters can grab huge advantages over honest players through exchanging *unauthorised information* such as cards held by each other to eliminate uncertainty caused by imperfect information.

Collusion in bridge occurs largely within a partnership, but a player can also collude with a kibitzer who observes the game, or with another player at a different table – e.g. in duplicate bridge, where a partnership is compared with other pair(s) playing the exact same hands under the same conditions, and final scores are calculated by comparing one pair's score to that of the other pair(s).

In face-to-face bridge, one has to stealthily pass card information to his or her collusive partner through spoken or body language. Online cheaters exploit out-of-band communication channels such as telephones and instant messengers, instead. The motives for people to cheat in face-to-face bridge more or less remain in its online counterpart. For example, there are regular and popular online tournaments (e.g. at `Okbridge.com`) that are sanctioned by American Contract Bridge League, meaning that these tournaments award official masterpoints, which are valid for climbing ACBL's rank ladder to achieve well-respected titles such as Life Master. On the other hand, playing bridge online makes it much easier to cheat than in face-to-face bridge. Cheaters collude for their personal gains (in whatever forms), but ruin the game experience of other people.

The common practice of detecting collusion in face-to-face tournaments is a committee review approach, which relies on a team of experienced human experts to analyse a complete record of game play after the fact. However, this approach is time-consuming and expensive, and cannot be scaled to handle thousands of games played online everyday in an effective but economic way. On the other hand, common security mechanisms for mitigating collusion do not work well in online bridge, either (Yan 2003a; 2003b). As such, although managers of online bridge communities have been aware of the problem for long, currently they still largely depend on tips from players whether somebody is cheating.

AI appears to be the last resort to this collusion problem, which to the best of our knowledge has not yet been studied in this community. In this paper, **we challenge AI researchers to create automated means that detects collusive play in bridge at human master level**. This challenge is highly related to, but not the same as, the problem of creating a bridge program that is consistently as good as human experts. We expect this challenge to stimulate some fundamental AI research. Potentially such research will also contribute to turning online bridge into a testbed for studying collusive human behaviours, which is otherwise difficult to observe in other contexts such as bid-rigging in procurement auctions (Bajari and Summers 2002).

The outline of this paper is as follows. We first compare honest and collusive play in bridge. Next, we discuss the feasibility of automated collusion detection in bridge – we will outline a possible way to do this, and discuss how to evaluate its performance and progress. We then discuss open problems in the form of a research agenda that groups the problems into near- and longer-term types. We also discuss how to make our approach more sophisticated. Following a review of related work, we conclude the paper.

How collusion matters

We compare an honest game with a collusive one as follows.

Honest play (based on incomplete information)

Bidding. Each player knows 25% of the cards (which is perfect information), and can deduce from each bid made by others some additional information, which is imperfect.

Card Play. There is no further information until the opening lead. When dummy's hand is revealed, the three others all know 50% perfect information, and each can also deduce additional imperfect information from bids made by others. For example, declarer can know precisely the high cards of each suit held by her opponent side, and estimate high card points (HCP)[1] and suit distribution[2] of each opponent. Moreover, each player may deduce from each played trick some information that is either imperfect or perfect.

Collusive play (based on complete information; we assume that North and South collude, and each knows all the cards held by the other.)

Bidding. Either East or West knows from her own hand 25% perfect information, while both North and South know 50%. Both North and South precisely know all their suit combinations[3], which is crucial for both bidding and card play. Moreover, either North or South may get more information from each bid made by opponents than what opponents can deduce from bids made by North or South. For example, since both North and South know high cards held by the opponents together in each suit, it is easier for them to deduce the card or suit distribution in each opponent's hand.

Card Play. There is no further information until the opening lead. When North and South are declarer and dummy, they might appear to get no further advantage from collusion after the opening lead. However, the truth is that they can get more information than they (or any honest opponent) should have, either directly or by converting some information from imperfect to perfect with the aid of each played trick. When North and South are defenders, their information advantage is tremendous. For example, one of them can easily draw a perfect opening lead to an *entry*[4] held by the other. Additionally, once dummy's hand is revealed, both North and South know 100% perfect information.

Clearly, colluding players have an unfair information advantage. Capable cheaters can almost always achieve a best possible result for each deal if they choose to do so. Moreover, a cheater need not know all cards held by his or her colluding partner; it is often sufficient for them to exchange only the "mission-critical" information.

Automated Detection of Suspicious Play

We consider each action (a bid or card play) in a game as a decision made based on collectable information. Thus, each player has a decision-making sequence left in the game record. Denote by S_w the decision-making sequence of a player who has access to unauthorised information, and by S_o the decision-making sequence when she plays the same hand but without access to this information. If $S_w = S_o$ is true for all the games she has played, then there are no grounds to accuse her game play. Otherwise, collusive play theoretically could be detected by comparing these two sequences. That is, human decisions based on partial information are unlikely to be always the same as those based on more complete information. This echoes the rationale behind the committee review approach, namely, if players persistently gain through collusion, it is very likely for them to leave traces in their play.

All this appears to suggest that it is feasible to design an automated approach that utilises inference techniques developed in AI (in particular the computer bridge) community to detect collusive traces in bridge play. The core of such a design can be an inference engine (IE), which takes a complete record of each game as its input and analyses the bidding and play of each partnership. Actions that are based on a decision deemed too good to be drawn from partial information will be detected as suspicious play.

Collusive cheaters might leave traces in almost any part of the play, and it would be ideal to detect all the traces. However, we propose to initially focus on the following critical places:

- *Contract* bid: the higher contract a partnership have successfully made, the higher reward they get.
- *Penalty double* bid, which squeezes maximum reward from defeating an unrealistic contract of opponents.
- *Opening lead*: which is the first and sometimes a unique chance to defeat a contract.

The reasons for such an initial focus are as follows. First, these three scenarios are representative, and we have observed that many online cheaters often cash their collusive advantages at these critical places. Second, this reduces the problem's complexity so that useful progress can be made within a manageable timescale. It is technically more complicated to detect other collusive play than these three scenarios.

We outline as follows how to detect a suspicious contract bid, opening lead, or penalty double.

Contract or opening lead oriented detection

An algorithm for detecting a suspicious contract bid or opening lead can be as follows - for the latter, additional expert knowledge about opening leads is needed for inference.

- The IE first identifies in the game record the real action α, either a contract bid or opening lead. It then generates A_h, a set of action candidates for α as in honest play. Note: a computer bridge program just has to work out *only* a single best candidate for each action; however, A_h may include multiple candidates, since for example all the candidates might be of equal quality.
- The engine also generates A_c, a set of action candidates for α as in collusive play. Often, A_c can be decided

[1] The sum of A, K, Q and J each calculated with a predefined weight.

[2] Number of cards in each suit. For example, one may have a 4432 suit distribution.

[3] A suit combination means a partnership's combined holding in one suit.

[4] A card that can win a trick and thereby gain the lead for its holder.

largely by the combined hands of a partnership that is suspected to have colluded. Although cheaters exchanging only mission-critical information do not need to know all cards held by each other, the set of their collusive actions is typically a subset of A_c.

- If $(\alpha \in A_c) \wedge (\alpha \notin A_h)$, then a suspicious signal will be triggered. That is, if information that one can collect from his or her own hand and from others' bids cannot justify the contract bid or opening lead, the suspicion of collusion will be raised.

The following example shows how a suspicious contract bid could be detected using the above approach. In this example, each player followed the Goren bidding system (Goren 1986). However, It is *not* a prerequisite for all players to use the same convention to make collusion detection work.

North: ♠AKJ643 ♡92 ♢75 ♣KQ3

SOUTH	WEST	NORTH	EAST
Pass	1♡	Double	2♡
2♠	Pass	Pass	Pass

South: ♠Q52 ♡864 ♢AJT86 ♣54

- The IE reads the bid sequence $b_1 = Pass$, ..., $b_8 = Pass$ and recognises South having bid b_5 =2♠ as the contract.
- Generates A_h, a set of candidate contract bids for b_5 as in honest play, by following the steps described below.

  IE does a bidding inference from the view of South. For example, it may generate an inference set from each bid b_i $(i = 2, 3, 4)$: one for b_2, interpreting that West is with 13 or more HCPs and has 5 or more hearts; one for b_3, interpreting that North has 13 or more HCPs, 3 or less hearts but other longer suits; and one for b_4, interpreting that East has 6 or more HCPs, and 3 or more hearts.

  By combining the above inferences with general bridge knowledge and cards held by South, IE recognises that North has 2 or less hearts, and b_3 meant to force South to bid his best suit. Thus, it generates a candidate set $A_h = \{3\diamondsuit\}$, having only one recommendation.
- Does another bidding inference that is similar to the one given above but with all cards held by North also as part of input, and then generates A_c =$\{2\spadesuit\}$, a set of candidate bids for collusive play.
- Since $b_5 \notin A_h$ but $b_5 \in B_c$, a suspicious bid is detected.

North's *double* in this example was not a good bid, but it is unlikely for South to recognise this mistake in honest play. (This example was taken from a real, online session. Some changes could have been made to make it better, but I decided to keep it as is to present a real-life example. A better example could also be created through imagination.)

Penalty double oriented detection

A typical collusion scenario involving with a penalty double is like the following. There is no clue, from the bidding sequence and cards held in one hand, that a too high contract is bid by opponents, but because of collusion, a cheater is sure that his or her side has enough tricks to defeat the contract, and thus a penalty double is bid.

A double following a contract bid is not necessarily a penalty double, but instead requests a partner to make an opening lead in a particular suit. So there is a need to differentiate a penalty double from a lead-directing one. We assume that a penalty double has been recognised – for example, the opening leader's double bid made after the contract bid must be a penalty bid – or alerted[5] by its bidder as required by the game rules. Thus, an automated detection of collusive penalty double can work as follows.

- Locates the contract bid, and calculates n_b, the number of tricks required to defeat the contract. Denoting by nM the contract, we have $n_b = 8 - n$.
- Does a bidding inference from the view of the player p who has made the penalty double bid, and uses these inferences and her own hand to calculate the number of her winning tricks, n_h.
- Does another bidding inference from the view of p, but with both hands of cards held by her partnership as input. Then combines these results and both hands to calculate n_c, the number of winning tricks of this partnership.
- If $n_h < n_b \leq n_c$, then collusion suspicion is raised.

Evaluation

It is unlikely that cheaters will voluntarily admit that they are cheating, and therefore we do not expect them to help with the evaluation of a cheat detection system. Instead, we can rely on wisdoms drawn from the committee review. Namely, by comparing the results of an automated detection system with conclusions drawn by human experts about the same set of game records, we can evaluate the performance of a detection scheme, and tune its implementation to enhance its efficiency. To this end, a benchmark data set collecting a set of representative cases of collusive play will be valuable.

The same bid or card play may have a different explanation under different conventions. However, to analyse the bidding and play sequence with the very convention that players have used is essential to the detection accuracy, and it can simplify the design of the AI engine. Therefore, it is necessary to record each deal together with the exact conventions that it is played. A simple way to do this is to annotate each alerted bid or card play with its corresponding convention. For example, a double bid can be recorded as *Dble/Lightner*, meaning a lead-directing double by following the Lightner convention.

A common format for benchmark data agreed by the community will facilitate data collection and exchange, and a shared benchmark data set will make it possible to soundly compare results achieved by different research teams, and to measure future progress.

To simplify the research probem to make initial progress, it is sensible to stick to one bidding convention at first, rather than try to deal with them all. For the same reason, we do not need to use genuine online games at first. We can set

[5]The game rules require players to draw opponents' attention to bids and play that follow unusual conventions.

up our own double-blind experiment in which some partnerships are asked to cheat and others to play honestly and then see if we can detect the cheats. Using genuine online game records could cause much time and technical overheads that are unnecessary for the initial stage of research.

It is also useful to set a realistic goal for the research of automated collusion detection. Some collusion might remain undetectable, e.g. when cheaters are strong players who know what could cause suspicion, and what could be explained away.

Research Problems and Agenda

Bid inference is key to the collusion detection approach outlined above. The following bidding inference methods have been explored for computer bridge.

- *Explicit inference*, which simulates human thinking behaviours by explicitly representing inferences. This approach has been widely used, e.g. in (Green 1995; Hutton-squire 1997; MacLeod 1989; Stanier 1975; Quinlan 1979), but achieved only mixed success (Frank 1997).
- *Quantitative reasoning*, which handles inferencing by using quantitative adjustments to hand evaluations. In this approach, each hand is assigned a number of points P representing its strength. P will be adjusted according to other players bids. A next bid for a player is based on rules associated with his or her current P value. This method was innovated by the COBRA system (Lindelof 1983), and claimed impressive performance.
- *Borel simulation*, which is a Monte Carlo-like simulation and first adopted by GIB (Ginsberg 1999; 2001), one of the strongest bridge programs at the time. The method works as follows. At each point in the bidding, the program constructs a set D of deals consistent with the bidding so far, and it queries a large database of bidding rules to find all plausible candidate bids. For each candidate bid, the rest of the bidding process is projected using the same database, and then the play of the resulting contract is simulated for each deal $d \in D$ in a double-dummy way. The candidate bid that leads to the best averaged result will be chosen by the program.
- *Hypothetical reasoning* was explored in (Ando and Uehara 2001; Ando, Kobayashi, and Uehara 2003).
- *Machine learning* methods were introduced in the recent years. For example, the Monte Carlo sampling method was combined with a learning algorithm to achieve a promising result (Amit and Markovitch 2006). A neural network approach was showed to be able to effectively bid no trump hands (DeLooze and Downey 2007).

However, it is unclear yet which of the above approaches can achieve the best collusion detection, whether they can be combined to achieve an even better result, or a new approach is called for. A study of each approach, together with a reasonablely sized proof-of-concept implementation of a few selected approaches, can be an immediate next step. Other near-term research can include:

- It is unknown but interesting which, among contract, opening lead and penalty double oriented cheat detection, is most effective, and which most reliable.

  Static lead rules (Granovetter 1999) compiled by Matthew Granovetter, a well-known bridge expert and author, have proved to work very well in practice. It is a good first step to use these rules to detect suspect opening leads.
- At a high level, collusion detection in bridge is similar to intrusion detection that has been extensively studied in computer security. The main difference is the expert knwoledge, but the basic idea of detecting the difference between normal and abnormal behavior is similar. Therefore, it is interesting to explore whether security researchers working on intrusion detection can find in collusion detection another application for their techniques. Similarly, how will the study of collusion detection in bridge inform intrusion detection researchers?

The detection approach outlined in the previous section is intuitive, but not sophisticated enough. For example, there are at least the following additional aspects to the problem of collusion detection.

- *The skill level of a player should be taken into consideration.* When a beginning player, a medium-level player or a top player plays the same hand, their play can be significantly different. However, their play can all be honest, and reasonable if judged according to their skill level.

  Therefore, modeling human bridge players can be important to collusion detection. Furthermore, we will need an accurate ranking system for contract bridge. The masterpoint system used by the American Contract Bridge League does not accurately rate the current skills of a player, and somehow measures how long and how often she or he has played. The rating systems used by some online bridge are better, but still can be improved.
- *There is a statistical property in collusion detection.* It is imprudent to label a player as cheater once one suspicious play or two is detected, since such a suspicion is only an indication of illogical play, and it could be the result of collusion, lucky play (e.g., a lucky guess or gamble, or a mistake like misclicking a bid or card), or even honest but imaginative play of a genius. When an isolated suspicious play is examined, there is no precise way to conclude that it is exclusively the result of collusion.

  But when we also consider the dimension of time – it is highly unlikely that one will be always lucky; anyway, being lucky means a small probabilistic event – and the skill level of players, the problem will become more tractable. Some further discussions can be found in (Yan 2003a).

Some longer-term open problems include:

- How to effectively detect suspicious signals hidden in other bids, or further card play beyond the opening lead? Or they simply do not matter (much)?
- Computer bridge, in particular its bidding techniques, has not yet matured enough to compete with human masters (Ginsberg 1999; 2001). It appears that there is also a gap between state-of-the-art AI techniques and the level

of AI maturity sufficient to meet the expectation of collusion detection at the human master level. However, how big is this gap? Will this AI-based cheat detection approach be able to come close to and eventually outperform the capability of human experts?

Quantifying the effectiveness of this AI-based approach is important. Even if it turns out to be unlikely to produce an automated solution at the human master level, a modest system could still serve as an effective screening mechanism, probably significantly reducing the number of cases requiring the attention of human experts.

It is also of academic interest to study how the techniques required for collusion detection will differ from those that eventually enable an expert-level bridge program.

- Is there an automated collusion detection approach that is radically different from the one discussed in this paper?
- When a detecting method is publicly known, it would be probable for skilled cheaters to evade detection, e.g. by adjusting their bids or card play to disguise their collusion. This appears to be an inherent problem for all collusion detectors, but could lead to the following research: it would be interesting to study the co-evolution dynamics of detection algorithms and collusive behaviour in online bridge, and study whether these dynamics give insights to understand collusion and its mitigation in other contexts.

To answer all the above questions, we suggest to focus on duplicate bridge first. Rubber bridge is another type of competitive play of contract Bridge, where a rubber ends when a side has won two games – one *game* in this context means 100 or more trick points scored on one deal. Rubber bridge has its own rules and the running score would be needed to detect cheating – placing a deal out of context is dangerous. For example, when a partnership executes a tactic of "losing a battle to win the war", their bidding and play in a single deal may appear to be ridiculously illogical. However, these can be entirely reasonable when the deal is examined in a rubber. As such, automated collusion detection in rubber bridge might be more complicated than in duplicates.

Related Work

Computer bridge has been an area of a rich history, with the aim of designing an expert-level bridge-playing program.

Bidding in computer bridge, which is most relevant to this paper, was reviewed in the previous section. Other components of a bridge program such as declarer play, defence, and double-dummy solver were reviewed in (Frank 1997; Ginsberg 2001; Amit and Markovitch 2006).

Strong bridge-playing programs on the market include Jack Bridge[6], Wbridge5[7], Bridge Baron – its key techniques were reported in (Smith, Nau, and Throop 1998) – and GIB (Ginsberg 1999; 2001). The first two programs are rumoured to use similar techniques as in GIB (Norvig and Russel 2009). However, none of these four programs is yet a consistent expert-level player.

[6] http://www.jackbridge.com

[7] http://www.wbridge5.com

Modelling human players can be central to the problem of collusion detection. But it appears that modelling a human player is more difficult than developing a good computer player, and the skill level of current best programs is not good enough yet to enable them being directly used to model strong human players.

To the best of our knowledge, collusion detection in bridge was first discussed in my PhD thesis (Yan 2003a). The present paper is the first public discussion of this topic, but due to space limit, some details remain in (Yan 2003a).

The most relevant topics in computer security include intrusion detection (Denning 1987), which has grown into a big body of literature since 1986, and online games security (Yan and Choi 2002; Yan and Randell 2005).

Conclusion

The problem of collusion detection in online bridge is both interesting and difficult. It is representative of a more general class of problems – detecting the use of prohibited information in decision making. For example, detecting bid-rigging in procurement auctions, detecting insider stock trading, detecting employment, loan, or housing discrimination, detecting racial profiling, and even detecting jury tampering are all real societal problems that have characteristics in common with the game challenge proposed here. As such, real progress on this challenge would likely have a broader impact on a class of useful and important problems.

In general, determining whether collusion has taken place in bridge requires determining whether bids/plays are rational given the available information, or whether they are unusually "lucky". This is a statistical property, and rational play is also a function of the skill levels and strategies of the players. As a result, there is no single right answer for what constitutes "correct" bidding/play, making it much more difficult to recognize "incorrect" bidding/play. The collusion detection approach outlined in this paper might not be able to completely solve the problem. However, it is a reasonable first step and should serve to stimulate useful discussion and research, even if a full solution to the challenge is currently beyond reach.

Acknowledgement

It is a pleasure to acknowledge the insightful and constructive comments from Professor Alan Bundy at Edinburgh University and all the five anonymous reviewers of AAAI-10. Also, I thank Isi Mitrani for his comments on an early draft of this paper and thank Cliff Jones for his help.

References

Amit, A., and Markovitch, S. 2006. Learning to bid in bridge. *Machine Learning* 63(3):287–327.

Ando, T., and Uehara, T. 2001. Reasoning by agents in computer bridge bidding. In *Computers and Games: Second International Conference (CG 2000)*, 346–364. Hamamatsu, Japan: LNCS 2063, Springer-Verlag.

Ando, T.; Kobayashi, N.; and Uehara, T. 2003. Cooperation and competition of agents in the auction of computer

bridge. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 86(12):76–86.

Bajari, P., and Summers, G. 2002. Detecting collusion in procurement auctions. *Antitrust Law Journal* 70(1):143–170.

DeLooze, L., and Downey, J. 2007. Bridge bidding with imperfect information. In *IEEE Symposium on Computational Intelligence and Games*, 368–373. Hawaii, USA: IEEE Computer Society.

Denning, D. 1987. An intrusion-detection model. *IEEE transaction on Software Engineering* 13(2):222–232.

Frank, I. 1997. Computer Bridge Survey. Technical Report TR-97-3, Electrotechnical Laboratory, Japan.

Ginsberg, M. 1999. GIB: Steps Towards an Expert Level Bridge-Playing Program. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'99)*, 584–589.

Ginsberg, M. 2001. GIB: Imperfect Information in a Computationally Challenging Game. *Journal of Artificial Intelligence Research* 14(1):303–358.

Goren, C. 1986. *Goren's New Bridge Complete*. USA: Century Hutchinson.

Granovetter, M. 1999. *Murder at the Bridge Table*. USA: Master Point Press.

Green, S. 1995. Drawing inferences from bridge bidding. the 4th Year Project Report, DAI, University of Edinburgh.

Hutton-squire, R. 1997. Drawing inferences about defender hands from bidding and play in bridge. AICS-4 dissertation, DAI, Univ of Edinburgh.

Lindelof, E. 1983. *Cobra: The Computer-Designed Bidding System*. London: Victor Gollanc.

MacLeod, J. 1989. Microbridge – a computer developed approach to bidding. In Levy, D., and Beal, D., eds., *Heuristic Programming in Artificial Intelligence - The First Computer Olympiad*, 81–87. Ellis Horwood.

Norvig, P., and Russel, S. 2009. *Artificial intelligence: a moder approach (3rd ed)*. USA: Prentice Hall.

Quinlan, J. 1979. A knowledge-based system for locating missing high cards in bridge. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'79)*, 705–710.

Smith, S. J.; Nau, D.; and Throop, T. 1998. Success in spades: Using ai planning techniques to win the world championship of computer bridge. *AI Magazine* 19(2):93–106.

Stanier, A. 1975. BRIBIP: A Bridge-Bidding Program. In *Proceeding of the 4th International Joint Conferences on Artificial Intelligence (IJCAI'75)*.

Yan, J., and Choi, H.-J. 2002. Security issues in online games. *The Electronic Library* 20(2).

Yan, J., and Randell, B. 2005. A systematic classification of cheating in online games. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games (NetGames'05)*, 1–9. Hawthorne, NY: ACM Press.

Yan, J. 2003a. *Security for Online Games. (Chap 6. Collusion Prevention and Detection)*. Ph.D. Dissertation, Cambridge University, Computer Laboratory, England.

Yan, J. 2003b. Security design in online games. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC'03)*, 286–297. Las Vegas, USA: IEEE Computer Society.