Contract Bridge Bidding by Learning

Chun-Yen Ho and **Hsuan-Tien Lin**
(based on the M.S. Thesis of Chun-Yen Ho)

Dept. of Computer Science and Information Engineering, National Taiwan University

January 27, 2015

Rules of Contract Bridge

North
♠KQJ64
♡Q953
♢K
♣Q65

West
♠T985
♡62
♢A864
♣AK9

East
♠A2
♡874
♢JT32
♣J742

South
♠73
♡AKJT
♢Q975
♣T83

- 52 cards, 4 players
- two teams: N-S and E-W
- cooperative within a team
- competitive between teams
- incomplete information game
  —most cards hidden to other players

goal: highest score in a zero-sum scenario

Stages of Bridge

bidding stage

- an auction for determining the contract

playing stage

- 13 rounds of a card-strength competition

bidding stage

West	North	East	South
	1♠	PASS	1NT
PASS	2♡	PASS	3♡
PASS	PASS	PASS	

playing stage

West	North	East	South
♡2	♡3	♡4	♡A
♢A	♢K	♢2	♢5
♡6	♡5	♡7	♡T
♢6	♣5	♢3	♢Q
⋮	⋮	⋮	⋮

score: 3♡ North +0 140

score: calculated by contract & winning rounds

Rules of Bridge Bidding

West	North	East	South
	1♠	PASS	1NT
PASS	2♡	PASS	3♡
PASS	PASS	PASS	

- each player calls one of
  1. PASS
  2. increase value of bid from an ordered set of calls $\{1♣, 1♢, 1♡, 1♠, 1NT, 2♣, \cdots, 7NT\}$
  3. other sophisticated calls to compete with the other team
- terminated by three consecutive PASS calls

goal: most profitable final bid (contract)

Our Contributions

	playing stage	bidding stage
rule-based (human-mimicking)	less popular	most existing works
data-based (learning)	competitive to human	**our work**

an algorithmic study of learning to bid, which ...

- formalizes the non-competitive bidding task as a proper **machine learning problem**
- merges **several machine learning techniques** to design a promising model for the problem
- reaches **competitive experimental results** to current computer bridge bidding program and sheds lights on **future studies**

will focus on key ideas behind the techniques

Challenges in Learning to Bid and Solutions

- teams may interfere with each other through competition
  - assumption: focus on the sub-problem of bidding without competition
- different bids ⇒ different scores as feedback
  - take cost-sensitive classification classifiers for making prediction
- need to use bidding sequence for exchanging incomplete information
  - consider upper-confidence-bound algorithm for exploring informative bidding sequences
- sophisticated rules to be satisfied
  - design tree-based model to properly represent the rules

let's now formalize the **sub-problem**!

Notations

North $\mathbf{x}_n$	South $\mathbf{x}_s$
♠KQJ64	♠73
♡Q953	♡AKJT
♢K	♢Q975
♣Q65	♣T83

West	North	East	South
	1♠ $\mathbf{b}[1]$	PASS	1NT $\mathbf{b}[2]$
PASS	2♡ $\mathbf{b}[3]$	PASS	3♡ $\mathbf{b}[4]$
PASS	PASS	PASS	

Length $\ell = 4$

- without loss of generality, assume only North-South bid, starting North
- suit $\mathbf{x}$ represented by suit length and high card points
- goal: learn $g(\mathbf{x}, \mathbf{b})$ to decides next bid with suit $\mathbf{x}$ and history $\mathbf{b}$
    - $(\mathbf{b}[1] = g(\mathbf{x}_n, []))$
      $< (\mathbf{b}[2] = g(\mathbf{x}_s, \mathbf{b}[1]))$
      $< \cdots < (\mathbf{b}[\ell] = g(\mathbf{x}_\ell, \mathbf{b}[1, 2, \ldots, \ell - 1]))$
    - $g(\mathbf{x}_{\ell+1}, \mathbf{b}[1, 2, \ldots, \ell]) = \text{PASS}$ (length at most ℓ)

how to evaluate goodness of g?

Goodness of g

- score of final bid only known after playing stage
  —time consuming to compute
- approximation: double dummy analysis
  —compute goodness of g from audience view, **fast and usually good**
- store the difference of the best score and (score of each possible final bid) as cost vector **c**

contract	PASS	1♣	1♢	···	3♡	···	7♡	7♠	7NT
score	0	-50	-50	···	140	···	-200	-250	-250
cost	140	190	190		0		340	390	390
IMP	4	5	5		0		8	9	9

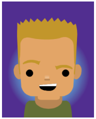
a cost-sensitive, sequence prediction problem with specialized constraints given data

$$\mathcal{D} = \{(\mathbf{x}_{ni}, \mathbf{x}_{si}, \mathbf{c}_i)\}_{i=1}^{N}$$

Key Idea in Simplified Scenario

- let $\ell = 1$,
  - Alice: $g(\mathbf{x}_n, []) = \mathbf{b}[1]$
  - Bob: $g(\mathbf{x}_s, \mathbf{b}[1]) = \mathbf{b}[2]$
  - Alice again: $g(\mathbf{x}_n, \mathbf{b}[1, 2]) = \text{PASS}$
- how to they practice?

- use current g_n on $\mathbf{x}_n$ to predict $\mathbf{b}[1]$
- receive c from Bob
- improve g_n with $((\mathbf{x}_n, []), \mathbf{b}[1], c)$ with cost-sensitive classifiers



- receive $\mathbf{b}[1] = g(\mathbf{x}_n, [])$ from Alice
- use g_s on $\mathbf{x}_s$ and $\mathbf{b}[1]$ to make the final bid $\mathbf{b}[2]$
- evaluate cost $c = \mathbf{c}[\mathbf{b}[2]]$
- improve g_s with $((\mathbf{x}_s, \mathbf{b}[1]), \mathbf{b}[2], c)$ with cost-sensitive classifiers

what if Alice is poor and always calls PASS?

Exploration and Exploitation

- Alice always poor $\Longrightarrow$ Bob always poor
- fixed poor calls: no help, should perhaps explore for helping calls
- uniform random calls: no information, should perhaps exploit previously good calls
- analogy in casino:
  - pulling the same machine: no help, should explore other machines
  - uniform random pulling: no help, should exploit lucky machine

analogy in machine learning: bandit model (online learning) and upper-confidence bound (UCB) algorithms

Respecting Bridge Rules

now we have

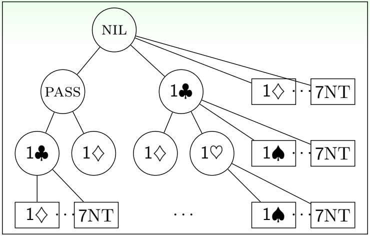
- cost-sensitive classifiers for 'improving'
- UCB algorithms for 'practicing'

the only remaining task

- respecting bridge rules

proposed model: tree-structured
with edges ⇔ valid calls

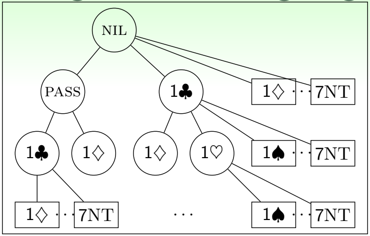
Model Structure



- layers of 'bidding nodes' to model the decision making process
  - internal (circle): a cost-sensitive classifier g to be learned
  - leaf (rectangle): always PASS afterwards
- $\ell + 1$ layers
  - layer 1: entry point of North
  - layer $\ell + 1$: all leaves
- tree 'pruning': only predicting $(\leq M)$-bids higher than current

bidding sequence ⇔ path from root to leaf

Introducing the Learning Algorithm



Snapshots of Multi-Layer Bandit Model

- tree for satisfying the bridge rules
- every 'practicing' decision made by standard UCB algorithms to balance **exploring** new bids and **exploiting** good bids
- every 'outcome' of bidding used to update cost-sensitive classifiers on **internal nodes**

check the paper for details and **more techniques to improve learning**

Experiment Settings

- 100,000 random deals as data
  - training: 80,000
  - validation: 10,000
  - testing: 10,000
- **x**: 2nd order combinations of (length of suits, high card points)
  —'similar' to what human bidding systems often use

our 'competitor':
Wbridge5, computer bridge champion,
with Standard American human bidding system

Comparison on Different Model Structures

model	UCB	train	validation	test
Baseline ($\ell = 1$)	N/A	3.8814	3.9754	3.8465
Tree, $\ell = 2$	UCB1	3.1197 ± 0.0177	3.1981 ± 0.0268	3.0755 ± 0.0173
	LinUCB	3.1242 ± 0.0089	3.2190 ± 0.0121	3.0933 ± 0.0112
Tree, $\ell = 4$	UCB1	**2.9013 ± 0.0079**	3.0769 ± 0.0118	**2.9672 ± 0.0096**
	LinUCB	3.0918 ± 0.0344	3.1804 ± 0.0298	3.0672 ± 0.0379
Tree, $\ell = 6$	UCB1	**2.9025 ± 0.0210**	**3.0484 ± 0.0226**	**2.9616 ± 0.0234**
	LinUCB	3.0124 ± 0.0249	3.1301 ± 0.0264	3.0477 ± 0.0243
Wbridge5	N/A	N/A	**3.0527**	**2.9550**

- better than the baseline with tree
- competitive to Wbridge5

promising 'first try' to learn a bidding system automatically

Comparison with Wbridge5

Type	Difference	Number of Deals
PASS	-12	2116
PARTIAL	4205	4779
GAME	-1607	2670
SLAM	-1612	406
GRAND SLAM	-294	29

- majority of deals: PARTIAL
- strength of human system (Wbridge5): GAME and beyond
- strength of proposed model: PARTIAL

reasons:

- proposed model 'data driven', hence focusing on PARTIAL
- proposed model 'under non-competitive setting', but human system under competitive setting

Conclusions

- formalizes the non-competitive bidding task as a proper **machine learning problem**
- studied **machine learning approaches** for the task
- proposed a novel model with **cost-sensitive classifiers, UCB algorithms, and tree structure**
- reached **promising results** and demonstrated the **potential of machine learning** for the problem

Thank you! Any Questions?

Appendix: Opening Table

Bid	Tree model, $\ell = 4$	Tree model, $\ell = 6$	SAYC
PASS	0-11 HCP	0-12 HCP	0-11 HCP
1♣	10-19 HCP, no many ♡	9-19 HCP, 4-6 ♡	12+ HCP, 3+♣
1♢	Not Used	8-18 HCP, short ♠ and 4-6 ♣	12+ HCP, 3+♢
1♡	9-19 HCP, 4-6 ♡	12-23 HCP, w/o long suit	12+ HCP, 5+♡
1♠	16-23 HCP, near balanced	10-19 HCP, 4-6 ♠	12+ HCP, 5+♠
1NT	Not used	Not used	15-17 HCP, Balanced
2♣	0-17 HCP, long ♣	0-17 HCP, long ♣	22+ HCP
2♢	0-17 HCP, long ♢	0-17 HCP, long ♢	5-11 HCP, 6+♢
2♡	0-13 HCP, long ♡	0-13 HCP, long ♡	5-11 HCP, 6+♡
2♠	0-13 HCP, long ♠	0-13 HCP, long ♠	5-11 HCP, 6+♠
2NT	Not used	Not used	20-21 HCP, balanced
3♣	14-19 HCP, long ♣	15-19 HCP, long ♣	5-11 HCP, 7+♣
3♢	14-19 HCP, long ♢	15-19 HCP, long ♢	5-11 HCP, 7+♢
3♡	Not used	Not used	5-11 HCP, 7+♡
3♠	Not used	Not used	5-11 HCP, 7+♠
3NT	19-29 HCP, w/o a long suit	19-29 HCP, w/o a long suit	25-27 HCP, balanced
4♣	Not used	Not used	5-11 HCP, 8+♣
4♢	Not used	Not used	5-11 HCP, 8+♢
4♡	10-29 HCP, long ♡	11-29 HCP, long ♡	8+♡
4♠	10-29 HCP, long ♠	11-29 HCP, long ♠	8+♠
4NT	27-29 HCP, near balanced	27-29 HCP, near balanced	Not used
5♣	16-27 HCP, long ♣	16-27 HCP, long ♣	very long ♣
5♢	17-25 HCP, long ♢	17-25 HCP, long ♢	very long ♢