IJCAI–13 Formatting Instructions*

Francesca Rossi
University of Padova
Italy
pcchair13@ijcai.org

Abstract

The *IJCAI–13 Proceedings* will be printed from electronic manuscripts submitted by the authors. The electronic manuscript will also be included in the online version of the proceedings. This paper provides the style instructions.

1 Introduction

2 General Test Game

A *test game* $G = (A, \mathbb{B}, p, v, m, t)$ is a 2-player Bayes game between the tester and the test taker. The tester has a set of potential problems $A = \{a_1, a_2, \ldots, a_n\}$ to test and the test taker has an uncertain type (thus Bayes). The possible types are characterized by $\mathbb{B} = \{B_1, B_2, \ldots, B_k\}$ and functions p, v, m. Set $B_l \subseteq A$ is the unsolvable problem set of the l-th type test taker. Function $p : \mathbb{B} \to [0, 1]$ characterizes the probability that a particular type test taker occurs. The memorize capacity function $m : \mathbb{B} \to \mathbb{N}$ denotes how many problems a particular test taker can memorize and the value function $v : \mathbb{B} \to \mathbb{R}^+$ denotes how much utility the tester gets if that test taker failed the test (recall that the tester cannot fail test takers who can solve all problems and he want to fail all others). For simplicity, we write $p(B_l), m(B_l), v(B_l)$ as p_l, m_l, v_l when the context is clear. During the test, t out of n problems will be tested and a particular test taker will pass the test if each of those tested problems is either not in the unsolvable problem set or memorized.

The game only has two outcomes, pass or fail. The tester gets 0 utility if a test taker passes and he gets v_l utility if a type-l test taker fails. On the type-l test taker's side, he gets 0 utility if he passes and $-v_l$ if he fails. From the test taker's perspective, any utility function that has a higher pass utility than fail utility will give him the same incentive to pass the test at his best. We define them as $0, -v_l$ specifically for the zero-sum property of the game.

We want to find the optimal strategy for the tester to maximize his utility under Stackelberg settings. That is, the tester firstly reveals his strategy about how t problems are picked to test, then the test taker chooses the best memorizing strategy. This corresponds to our confidential-free assumption where test takers know what problems might be tested, their answers, and how likely they are tested. The only constraint for test takers is that they cannot memorize too many (greater than m_l) problem answers.

2.1 General LP Formulation

Though general

2.2 Hardness on Test Size

In order to formally characterize *test game*'s hardness, define

Definition 1 (OPTIMAL TEST STRATEGY Problem)**.** *Given a test game G and a value u, the* OPTIMAL TEST STRATEGY *problem is to decide whether the tester has a Stackelberg strategy with at least u utility.*

TO BE CONTINUED WITH Vertex Cover

2.3 Hardness on Memorize Capacity

Theorem 1. *Even if test size is* 2, OPTIMAL TEST STRATEGY *is coNP-hard when memorize capacity is non-constant.*

Proof. We reduce INDEPENDENT SET instances to test games with test size 2 and ask the complementary question: is u the highest utility that the tester can get. That question has a yes answer if and only if the dual minimax LP [reference] has a feasible solution with objective at most u. For a graph $G = (V, E)$, let V be our test problem set. Add $|E| + |V| + 2$ types of test takers as follows:

- For each edge $e = (v_i, v_j) \in E$, add one type of test takers l_e with tester utility equals 1 if they fail and 0 if they pass. This type of test takers cannot solve only two problems, v_i, v_j. The number of problems they can memorize is 0.
- For each vertex v, we add one type of test takers l_v with tester utility equals $d_G - d(v)$ if they fail and 0 if they pass. Here d_G is the maximum vertex degree in G and $d(v)$ is the degree of vertex v. This type of test takers cannot solve only one problem v and the number of problems they can memorize is 0.
- Add one auxiliary type of test takers l_A with tester utility equals $a \cdot \varepsilon$ if they fail and 0 if they pass where $a = \binom{|V|}{2} - |E| - \binom{k}{2}$. This type of test takers cannot solve any problems but they can memorize 2.

*These match the formatting instructions of IJCAI-07. The support of IJCAI, Inc. is acknowledged.

- Finally, add one type of test takers l_K with tester utility equals $u^{l_K} = \varepsilon$ if they fail and 0 if they pass. This type of test takers cannot solve any problems but they can memorize k of them.

Then we show that deciding whether $u = (2d_G + a \cdot \varepsilon)/L$ is an upper-bound for the tester's utility, or equivalently whether we can find a dual solution with objective as low as $u = (2d_G + a \cdot \varepsilon)/L$ in our dual minimax LP, is equivalent to finding a size-k independent set in G. Recall that a dual solution is a strategy for test takers to memorize problems. If no one memorizes, the tester's utility U_r to test a problem set r is $U_r = (2d_G + a \cdot \varepsilon + \varepsilon)/L$ if $r \notin E$ and $U_r = (2d_G - 1 + a \cdot \varepsilon + \varepsilon)/L$ if $r \in E$. To lower the objective, test takers l_A, l_K should memorize problem sets that bring the maximum of U_r down. No matter what the strategy is (as long as they only memorize unsolvable problems), $\sum_r U_r$ is a constant. And because U_r for $r \notin E$ is much larger than U_r for $r \in E$ in terms of ε, it's better to only decrease U_r for $r \notin E$. The only way to do that is to let l_A test takers only memorize pairs of problems that are not edges and let l_K test takers only memorize size-k indepedent sets of problems. If there's a size-k independent set, let l_K test takers memorize one size-k independent set all the time and let l_A test takers memorize all pairs of problems that are not covered by that independent set with uniform probability. That will make $U_r = (2d_G + a \cdot \varepsilon)/L$ for $r \notin E$ and $U_r = (2d_G - 1 + a \cdot \varepsilon + \varepsilon)/L$ for $r \in E$. This is the best they can achieve and this can only be achieved when a size-k independent set exists in G.

$\square$

3 One-Problem Test Game

4 Experiment

5 Conclusion

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, and Daniel Pagenstecher. These instructions are the same as the ones for IJCAI–05, prepared by Kurt Steinkraus, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab.

A LaTeX and Word Style Files

The LaTeX and Word style files are available on the IJCAI–13 website, `http://www.ijcai-13.org/`. These style files implement the formatting instructions in this document.

The LaTeX files are `ijcai13.sty` and `ijcai13.tex`, and the BibTeX files are `named.bst` and `ijcai13.bib`. The LaTeX style file is for version 2e of LaTeX, and the BibTeX style file is for version 0.99c of BibTeX (*not* version 0.98i). The `ijcai13.sty` file is the same as the `ijcai07.sty` file used for IJCAI–07.

The Microsoft Word style file consists of a single file, `ijcai13.doc`. This template is the same as the one used for IJCAI–07.

These Microsoft Word and LaTeX files contain the source of the present document and may serve as a formatting sample.

Further information on using these styles for the preparation of papers for IJCAI–13 can be obtained by contacting `pcchair13@ijcai.org`.

References