Social Learning Strategies Tournament

Details of Stage I Results

Luke Rendell and Kevin Laland, University of St Andrews
November 26, 2008

This document provides a report on the first stage of analysis of submitted entries to the social learning strategies tournament. 104 entries were received in total. Here we describe the evaluation process adopted, and the results for those strategies that did not progress to Stage II.

Stage I details

The simulations each contained a single population of 100 individuals. Strategies initially took part in round-robin contests against all other strategies. A contest involved each strategy invading a population of the other strategy. In each simulation, a population consisting purely of one strategy was run for 100 rounds to establish behavioural repertoires. After 100 rounds, mutation was introduced, providing the second strategy the opportunity to invade. Simulations were then run for a further 10,000 rounds. Each pairwise contest was repeated 10 times with strategy A as the invader and 10 times with strategy B as the invader. The mean frequencies of each strategy in the last quarter of each run (i.e. the last 2,500 rounds) were then averaged over the 20 repetitions, and this average was recorded as the score of that strategy in that contest. Strategies were then ranked on their average score. Simulations were run using grid computing technology, taking approximately 64,000 hours of CPU time, and we gratefully acknowledge the provision of this resource by the UK National Grid Service (http://www.grid-support.ac.uk/).

Initial condition and results

We first ran a complete set of pairwise contests with all 104 strategies, as detailed above. This involved 5,356 individual contests, with 107,120 individual simulation runs. The parameter conditions chosen for these runs were p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05 and $\sigma_{copyPayoffError}$=1[a]. No strategies were disqualified because they took too long to run (rule 7). The computational resources available to us meant we had the opportunity to test further conditions at this stage. We elected to run further conditions on a subset of strategies selected on the basis of their scores in the complete pairwise procedure. We did this because our cut-off of the top ten strategies cut the payoff distribution with a series of strategies (ranked 6 to 24) with relatively small differences in their scores such that the initial ranking could have easily been dependent on the set of conditions we happened to choose. The rules we set out allowed for the running of further conditions in stage I (section 1.1.2 of the tournament rules). Table 1 gives the scores for those strategies ranked lower than 24 in these initial conditions.

[a] These were parameters unspecified in the tournament rules; p_c is the independent probability of the payoff of an act changing between rounds, $n_{observe}$ is the number of individuals observed when the move OBSERVE is played, $p_{copyActWrong}$ is the probability that an observed payoff is associated with an incorrect act (i.e. an error in social learning) and $\sigma_{copyPayoffError}$ is the standard deviation of the error distribution of observed payoffs.

Table 1: Stage I initial condition results for ranks 25 to 104

Rank	Strategy	Score
25	*oneThirdSocial*	0.698
26	*whoDoISee*	0.695
27	*whatYouSeeIsWhatYouDo*	0.683
28	*aHandfulOfSkills*	0.670
29	*lookahead*	0.666
30	*instancebased*	0.657
31	*gatherDataAndHillClimb*	0.652
32	*breakthroughInnovation*	0.651
33	*learnFromOthers*	0.646
34	*spyNWork*	0.641
35	*waitForSomethingBetter*	0.641
36	*followTheMeans*	0.637
37	*sabbath*	0.620
38	*copyMoreWhenYounger*	0.620
39	*divideAndConquer*	0.618
40	*rebelWithoutACause*	0.605
41	*adaptiveControl*	0.596
42	*cUDOS*	0.591
43	*hydra*	0.590
44	*copyAndSwitch*	0.566
45	*carefullyRecalcDude*	0.565
46	*stCoop*	0.562
47	*itTakesAVillage*	0.560
48	*lateAdopter*	0.558
49	*copyWhenConditionsAreStable*	0.554
50	*goldberg*	0.553
51	*bestMeanSelect*	0.549
52	*roman5*	0.549
53	*weightedMoves*	0.546
54	*herculesAtTheCrossroads*	0.544
55	*indexBasedLearningDecision*	0.543
56	*learnFirst*	0.527
57	*marmoset*	0.515
58	*infantJuvenileMature*	0.515
59	*julieDecstep25*	0.509
60	*lowVarianceObserver*	0.508
61	*estimatePcThenChooseStrategy*	0.498
62	*copyWhenPayoffsDecreasePlus*	0.492
63	*staticLearningReduction*	0.479
64	*wiseOrObserve*	0.473
65	*greatExpectations*	0.472
66	*kISAgent*	0.469
67	*updateWhatYouHave*	0.464
68	*learnUntilProfitable*	0.451
69	*observeEarlyInnovateLittle*	0.411
70	*infoScrounger*	0.405
71	*weightedSocialLuceChoice*	0.368
72	*optimalStopping*	0.363
73	*tangle7*	0.349
74	*twiceAsGood*	0.328
75	*goodacts*	0.319
76	*magpie*	0.315
77	*criticalSocialLearner*	0.310
78	*knowledgeWeighters*	0.307

Table 1 contd.

Rank	Strategy	Score
79	*adaptolution*	0.303
80	*innovateBeforeObserveInRadicalEnvironmentalChange*	0.298
81	*monkeyStudent*	0.288
82	*smartObserve*	0.287
83	*mainlyObservers*	0.279
84	*ratchet*	0.275
85	*genderedStrategy*	0.272
86	*innovateAndObserve*	0.264
87	*constantProbability*	0.224
88	*piRounds*	0.218
89	*kiss*	0.201
90	*econoSearch*	0.193
91	*copyIfBetter*	0.164
92	*weightedExploitation*	0.154
93	*noImitator*	0.151
94	*smashNgrab*	0.136
95	*higherLearning*	0.126
96	*prospector*	0.113
97	*aynRandGambit*	0.108
98	*unobservant*	0.104
99	*randomness*	0.087
100	*copyWhenPayoffsDecrease*	0.066
101	*balancedCopyWhenPayoffsDecrease*	0.063
102	*exploitOneInnovation*	0.051
103	*stateDefined*	0.044
104	*observeNoThanks*	0.023

Further conditions and results

We found it was computationally feasible to run a further 8 conditions (but only for the subset of strategies ranking in the top 24 of the first set of contests); these conditions are set out in Table 2 below.

Table 2: Details of further conditions run for top 24 strategies

Condition	p_c	$n_{observe}$	$p_{copyActWrong}$	$\sigma_{copyPayoffError}$
1	0.001	1	0.01	1
2	0.1	1	0.01	1
3	0.001	1	0.1	1
4	0.1	1	0.1	1
5	0.001	6	0.01	1
6	0.1	6	0.01	1
7	0.001	6	0.1	1
8	0.1	6	0.1	1

Note that we did not vary the parameter $\sigma_{copyPayoffError}$ in these conditions, as we reasoned that the parameter $p_{copyActWrong}$ would affect the accuracy of social learning to a stronger degree and so to also vary $\sigma_{copyPayoffError}$ orthogonally would unnecessarily duplicate effort in exploring the effect of the accuracy of social learning as well as doubling the computation required[b]. For these further conditions we reduced the number of repetitions per contest to 3 symmetric repetitions (i.e. 3 runs with strategy A as invader and 3 runs with strategy B as invader) as opposed to the 10 such repetitions run for the initial pairwise contest. We selected the value 3 based on Figure 1, which shows that the distribution of CV values for each pairwise contest does not change for more than 3 repetitions.

The top 10 strategies once this procedure was complete were, in alphabetical order, *copyWhenYoungThenLearnWhenPayoffsDrop*, *discountmachine*, *dynamicAspirationLevel*, *intergeneration*, *livingdog*, *prospero*, *rummer*, *valueVariance*, *wePreyClan*, and *whenTheGoingGetsToughGetScrounging*. The final score of a strategy was calculated as the average of its scores in the eight extra conditions and its score against the other 23 strategies under the initial condition. The overall results for strategies outside the top 10 in the further conditions round are given in Table 3. Note that scores here are lower than those in Table 1 because they are from contests involving only the top 24 strategies, so high scores gained against poorer performing strategies are not included. No strategy switched from the original pairwise results by more than 11 places, and the average change in rank was 2.5 places; this means it was unlikely in the extreme that any strategies outside the top 24 would have been elevated to the top 10 and thus stage II by this procedure. We are reassured that the most effective strategies are included in the 10 selected by the fact that the scores in the top 10 range from ~0.75 down to 0.5, thus incorporating a reasonably large variation in scores.

[b] In practice, we ran a single pairwise run with two extra conditions varying $\sigma_{copyPayoffError}$ from the initial condition above (p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05, $\underline{\boldsymbol{\sigma_{copyPayoffError}}\mathbf{=5}}$ and p_c=0.01, $n_{observe}$=1, $p_{copyActWrong}$=0.05, $\underline{\boldsymbol{\sigma_{copyPayoffError}}\mathbf{=10}}$) and found that including these conditions made no difference at all to the strategies that were eventually selected.

Figure 1: Boxplot of pairwise CV distributions by number of repetitions.

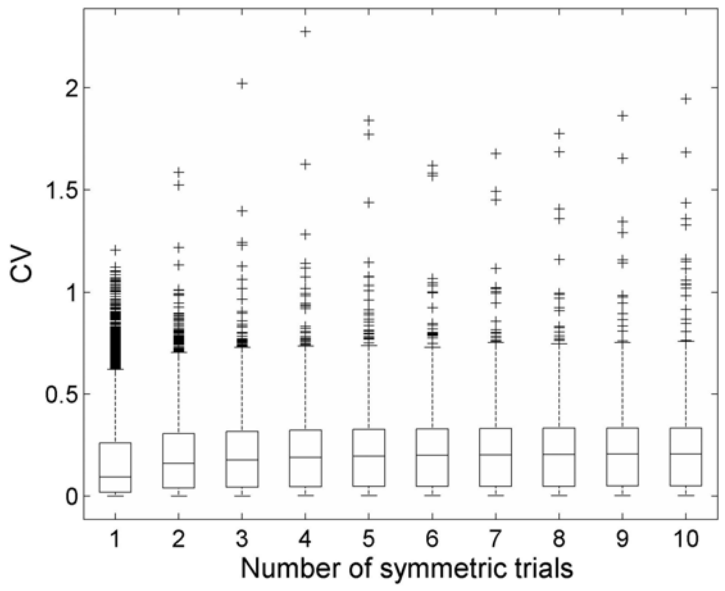

Table 3: Overall stage I results for top 24 strategies. Note that scores here are lower than those in Table 1 because they are from contests involving only the top 24 strategies, so high scores gained against poorer performing strategies are not included.

Rank	Strategy	Score
11	*stabilityObserver*	0.499
12	*w00t*	0.498
13	*senescence*	0.473
14	*progressivepeakseeker*	0.461
15	*evchooser*	0.456
16	*keepUp*	0.434
17	*improvedCwpd*	0.384
18	*indecisiveJDK*	0.384
19	*halfmax*	0.345
20	*observe3ThenExploit*	0.339
21	*learnAtTheBeginningThenExploit*	0.337
22	*weightedContextAware*	0.336
23	*startExploitRecover*	0.293
24	*firstLookThenExploit*	0.280