APPENDIX B. Computational Methods and Issues

To perform the integration required by eq(15), we impose a finite grid on the \((v, \theta)\) space. To ensure that the results are robust to the specification of the grid, we experimented with a variety of finite grids. We have settled on a computationally efficient grid consisting of \(5 \times 35 = 175\) points generated as follows. First, for \(\theta\), we specify 5 uniformly spaced points: \((0, 0.25, 0.5, 0.75, 1.0)\).

The \(v\)-subspace is represented by 35 points. To describe these 35 points first recall eq(20) which defines the probabilistic choice function for each rule, and consider a rule with one and only one \(v_k > 0\). In games with binary lottery payoffs (0 to 100%), how responsive is the choice behavior of such a rule to, say, a 10 percentage point difference in payoff? In other words, suppose the evidence for one action, say \(j\), is 10 points higher than any other action. How much more likely is action \(j\) to be chosen? Using eq(20), the probability of choosing action \(j\) out of \(J\) actions would be

\[
 r = \frac{\exp(10v_k)}{[J-1 + \exp(10v_k)]}. \tag{B1}
\]

Clearly, for large (small) values of \(v_k\), this probability \(r\) is close to \(1/(1/J)\), and is a non-linear function of \(v_k\). For the many symmetric normal form games we have used, payoff differences of 10% are typical. A smaller difference, say 5%, is on the margin of what we generally consider statistically significant, while a larger difference, say 20%, seems
too crude relative to human discriminating abilities. Thus, a 10 percentage point payoff
difference is a reasonable standard by which to assess the behavioral impact of $v_k$
weights.

We would like the range of $v_k$ weights in our grid to span fairly the range of
choice behaviors. In other words, we would like our grid points to correspond to equally
space behaviors (probabilities). Solving eq(B1) for $v_k$ in terms of $r$ gives $v_k$ equal to

$$(0.1) \ln[(J-1)r/(1-r)] \equiv f(r,J).$$

(B2)

For any probability $r \in [1/J, 1)$, $f(r,J)$ gives the magnitude of the $v_k$ weight such that an
action with a 10 point payoff advantage would have a choice probability of $r$. We note,
however, that $f(1,J) = \infty$ and from eq(B1), we can see that the choice of probability an
action that has a 10 point higher payoff than any other action will be quite close to one
for all values of $v_k > 1$. Thus, bounding the $v_k$ weights from above by 1 would not
induce any significant loss in the behavior that could be represented. We choose $\bar{v} \equiv 0.1
\ln(49996) \approx 1.082$ as our upper bound.\(^1\) This corresponds to a choice probability of
49996/(49995+J), which when $J = 5$ is 0.99992, and is clearly quite insensitive to $J$ for all
practical values of $J$ that would be used in experimental games. Now, take four equally
spaced points from $[1/J, (J-1)/J]$: $r_h([1 + 0.25h(J-1)]/J$, for $h=0,\ldots,3$. This partition yields
5 values for $(k: \{f(r_h, J) for h=0,\ldots,3\}$, plus $\bar{v}$. While these values are unevenly space in

---

\(^1\) This admittedly ad hoc value came about by considering the case of $J = 5$, and an upper bound on $r$
corresponding to 0.9999 of the interval $[1/J, 1]$. Since the results are robust to this number and to $J$, this
value became “grandfathered” in our code.
\(\nu\)-space, they induce evenly spaced behaviors (probabilistic choices) for the reference 10 point payoff difference.

Next we require that the sum of the weights over the three dimensions of \(\nu\)-space, \(\nu_k\) not exceed \(\overline{\nu}\). This restriction effectively creates competition among the 3 kinds of evidence (leve-1, level-2 and Nash): when some \(\nu_k\) is increased, the weight on some other kind of evidence has to be decreased. Without this competition, the MLE procedure will produce the following spurious results. Suppose that after 7 or 8 periods, choice behavior has converged to the extent that the level-1 and level-2 rules put high probability on the best-response. The MLE procedure is likely to drive both \(\nu_1\) and \(\nu_2\) as high as possible, creating only a slight increase in the log-likelihood value buy obscuring the relative importance of level-1 and level-2 rules evidence. To explore the effect of this restriction, we experimented with a variety of values for the upper bound (some higher and some lower than \(\overline{\nu}\)). Typically, an increase in the upper bound will increase the likelihood function only slightly, and have no significant effect on the parameter estimates (other than better identifying the relative \(\nu\) weights). Thus, the results are robust to the specification of the upper bound.

Given this restriction on the sum of the weights, it is natural to specify the other points of the grid so the sum of the weights equals a value in \(\{f(r_n,J), h=0,\ldots,3\}\). To do this, let define the 35 point “triangular” grid:

\[
T \equiv \{i \equiv (i_1, i_2, i_3) \in \{0, 1, 2, 3, 4\}^3 | S(i) \equiv \sum_{i=1}^{3} i_k \leq 4 \}.
\]  

(B3)
Then for any $i \in T$, define the weight for dimension $k$ as

$$v_k (i) \equiv f(r_{S(i)}, J)[i_k / S(j)]$$

The “distance”, $\|\|$ used in eq(9a and 10a) is the Euclidean distance on this index grid.

The entire grid is then the Cartesian product of the $5 \theta$ points and these $35 \nu$ points. In addition to this fixed grid of 175 points, we add the mean $((\nu, \overline{\theta}))$ as a variable grid point, making a total of 176 points in all.

To find a $\xi$ vector that maximizes $LL(\xi)$, eq(28), we use a simulated annealing algorithm [Goffe (1994)] for high (but declining) temperatures, and then feed the result into the Nelder and Mead (1965) algorithm. We find the simulated annealing algorithm to be effective in exploring the parameter space, but very slow to converge once it settled in on a local maximum, while the latter algorithm converges much faster locally.