Supplement for "Self-correcting Information Cascades"

1 Experimental Instructions

The file cas_instruc.ppt contains the slide presentation used for the instructions in our experiments. The file is for our $T = 20$, $q = 5/9$ treatment; files for the other treatments are modified slightly in the obvious ways. These instructions were read aloud publicly once all participants were seated in the laboratory.

2 Experimental Software

The experiments were run with java-based software created at SSEL (www.ssel.caltech.edu). The central program is called the server and each subject connects to the server through a client. The executables for both the server and client are included and named jury-server.bat and jury-client.bat, respectively. The required input for the programs is stored in three separate files: a parameter file, and sample file, and a match file. A set of these files is included, named test.par, test.smp, and test.mat, respectively.

3 Data

Included are thirteen csv files, one for each experimental session listed in Table 3 of the paper. The rows list the decisions from each session in chronological order. The columns are ordered as follows: subject ID number, match number, an irrelevant column of zeros, decision, signal, Jar, payoff, time (number of seconds elapsed since the start of the match when the decision was made). The coding scheme records a zero for Jar 1 choices and states and red signals (respectively, A choices and states and a signals in the paper's notation), and a one for Jar 2 choices and states and blue signals (respectively, B choices and states and b signals in the paper's notation).

4 Computer Programs

Most of the data analysis was conducted in Matlab and GAUSS. The following GAUSS routine uses experimental data stored in an $MT \times 2$ matrix called "data;" every T rows correspond to a single match, with a total of M matches. The first column contains subjects' signals and the second column subjects' choices. Note that it is trivial to put the data in this format from the included data files. The outcome of the procedure is the log-likelihood under QRE for a single treatment (i.e. with a fixed precision, q, and fixed length, T) although it is easy to adapt the procedure to deal with pooled data. It is also straightforward to adapt the program for estimating the other models in the paper. Note that the procedure is simple because information cascade experiments concern individual decision-making environments, not games, so there is no need to solve fixed-point equations to compute the QRE.

```
PROC loglikelihood(λ);
LOCAL logL,signal,choice,m,t,p,π^a,π^b,P(A|a),P(A|b),P(B|a),P(B|b),p^+,p^-;
  logL=0; m=1;
  DO WHILE m<=M;
   p=1/2; t=1;
   DO WHILE t<=T;
     π^a=qp/(qp+(1-q)(1-p));
     π^b=(1-q)p/((1-q)p+q(1-p));
     P(A|a)=1/(1+exp(λ(1-2π^a))); P(B|a)=1-P(A|a);
     P(A|b)=1/(1+exp(λ(1-2π^b))); P(B|b)=1-P(A|b);
     p^+=(pqP(A|a)+p(1-q)P(A|b))/((pq+(1-p)(1-q))P(A|a)+(p(1-q)+(1-p)q)P(A|b));
     p^-=(pqP(B|a)+p(1-q)P(B|b))/((pq+(1-p)(1-q))P(B|a)+(p(1-q)+(1-p)q)P(B|b));
     signal=data[(m-1)T+t,1]; choice=data[(m-1)T+t,2];
     IF signal==1 AND choice==1; p=p^+; logL=logL+ln(P(A|a)); ENDIF;
     IF signal==0 AND choice==1; p=p^+; logL=logL+ln(P(A|b)); ENDIF;
     IF signal==1 AND choice==0; p=p^-; logL=logL+ln(P(B|a)); ENDIF;
     IF signal==0 AND choice==0; p=p^-; logL=logL+ln(P(B|b)); ENDIF;
     t=t+1;
   ENDO;
   m=m+1;
  ENDO;
  RETP(logL);
ENDP;
```