01412D Program Description I

Program Title MATRIX GAME - CALCULATOR LEARNS TO LEARN

Contributor's Name MICHAEL SEGAL

Address COLLEGE OF PHYSICIANS + SURGEONS, 630 W 168th

City NY State NY Zip Code 10032

Program Description, Equations, Variables The calculator plays repeated rounds of a simple matrix game against the challenger. It watches different aspects of the challenger's behaviour to predict his subsequent behaviour and learn to outwit him. The calculator learns to preferentially use the more successful learning paradigm.

THE GAME

The human opponent tries to match the computer's choices of "0" or "1". Both human opponent and calculator choose "0" or "1"; if the human opponent matches choosing "0" he wins 2 points. If he fails to match choosing "0" he loses 2 points. The same holds for choosing "1" except that he wins one point on a match and loses one point when failing to match. (The payoffs to the computer are opposite to those of the human opponent. Since human and computer payoffs add up to zero, this is called a zero-sum game.)

Play many rounds of the game; if you like, call ±20 points what is needed to win.

The calculator learns what is the best learning strategy for playing the opponent. In trying to outwit you, it departs from the conservative strategy (see below). It is important to remember that the computer is no better than the person who programmed it. So when it leaves the conservative strategy to take advantage of you, you can try to take advantage of it.

Program Description II

Sketch(es)

CALCULATOR \ HUMAN	0	1
0	2	-1
1	-2	1

This is a summary of the payoffs to the human opponent.

Sample Problem(s) Suppose you think you can win by doing all zero's. Input card #1. Start off the program with a "random" integer [2] [E] and the calculator returns 0.0 and is ready to begin. [0] [A] → 0.1 indicating you chose "0" while it chose "1". [B] → -2. indicating that you lost two points on that round. [C] → -2. indicating that you are losing by two points so far. [D] → 0.1 displaying the last outcome in case you forgot it. [0] [A] → 0.0 indicating you chose "0" and it chose "0". [0] [A] → 0.1, [0] [A] → 0.1, [0] [A] → 0.1, [0] [A] → 0.1, [0] [A] → 0.1, [C] → -10.

This was handled quite quickly by the computer. More subtle strategies are handled more subtly. Good luck.

[redacted] NOTES:

1) Although you input your move before the computer displays its move, its move has in fact been calculated already and is stored in register A.
2) The payoff is tabulated into the score regardless of whether you ask to see them or not.
3) Those familiar with the first version of this program should note that outcomes are displayed here as human guess, computer guess, and not the reverse.

Reference(s) Original work of the author. For a clear introduction to two person game theory and analysis of repeated rounds of play, see Rapoport, Anatol. Two Person Game Theory: The Essential Ideas. The University of Michigan Press, Ann Arbor, 1973; Rapoport, Anatol, and Chammah, Albert M. Prisoner's Dilema: A Study in Conflict and Cooperation. The University of Michigan Press, Ann Arbor, 1970.

User Instructions

CARD #1

MATRIX GAME

1 ◀ 2 ▶

GUESS | PAYOFF | SCORE | LAST GUESS | n → START

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1	Read in both sides of the program			
2.	TO START a) for the standard game, input a low integer to start off the game differently each time	n	E	0.0
	b) for the generalized game, you have begun with card #2 and have read in this card during flashing zeros; go to step 3 once the display stops at 0.0			
3	Enter your guess (0 or 1). The display will return as your guess. computer guess. An error message indicates your guess was not zero or one. Press any key to lift the error condition and start at step 3 again with zero or one.	your guess	A	your guess. computer guess
4	Play another round by repeating step 3			
5	Optional: see your payoff on the previous round		B	payoff
6	Optional: see your total score		C	total score
7	Optional: display last outcome		D	your guess. computer guess

User Instructions

CARD #2

1 MATRIX GAME
GENERALIZED VERSION

INPUT GAME n→ START

2

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS	OUTPUT DATA/UNITS
1.	Read in both sides of the program			
2	Input the game. If you get "Error" this could mean: a) the game was unfair to the calculator b) "a" was not the highest of the four numbers. Press any key to lift the error condition and start at 2 again with a legal game.	a b c d	↑ ↑ ↑ D	0. or Error
3	Input a low integer to start off the game differently each time.	n	E	flashing 0.
4	Read in both sides of the main program and follow the instructions for it.			

Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	GTO 5	22 05	Branch for generalized program
	f LBL E	31 25 15	(n ⇒ START)
	f CL REG	31 43	
	h ABS	35 64	loop n times, n is an integer ≥ 1
	1	01	
	+	61	
	h ST I	35 33	
	.	83	a tested random seed
	5	05	
010	2	02	
	8	08	
	4	04	
	1	01	
	6	06	
	3	03	
	g LBL fe	32 25 15	random number generator
	9	09	
	9	09	
	7	07	
020	x	71	
	g FRAC	32 83	
	f OSZ	31 33	
	GTO fe	22 31 15	
	STO B	33 12	
	1	01	a = 2
	STO 3	33 03	b = -1
	CHS	42	c = -2
	STO 1	33 01	d = 1
	2	02	
030	STO 0	33 00	
	CHS	42	
	STO 2	33 02	
	1	01	initializing counter and state monitor
	0	00	
	h ST I	35 33	
	STO 8	33 08	
	.	83	computer tendancy = .5
	5	05	
	STO 7	33 07	
040	.	83	stochastic conservatism = .9
	9	09	
	STO C	33 13	
	2	02	human tendancy = human equilibrium = 2/3
	ENTER	41	
	3	03	
	÷	81	
	STO D	33 14	
	STO 5	33 05	
	f P ⇄ S	31 42	store human equilibrium in behaviouristic registers.
050	STO 0	33 00	
	STO 3	33 03	
	STO 4	33 04	
	STO 5	33 05	
	STO 6	33 06	
	STO 1	33 01	
	STO 2	33 02	
	STO 7	33 07	
	f P ⇄ S	31 42	
	GTO 5	22 05	DO FIRST GUESS
060	f LBL A	31 25 11	GUESS
	ENTER	41	
	f INT	31 83	Error if noninteger input or < 0 or > 1
	g x≠y	32 61	
	GTO 9	22 09	
f x<0		31 71	
GTO 9		22 09	
1		01	
h x⇄y		35 52	
	g x>y	32 81	
070	GTO 9	22 09	
	STO E	33 15	(new beh. reg.) = 1/2 (old beh. reg.) + 1/2 (new guess)
	2	02	
	STO ÷ (i)	33 81 24	
	÷	81	
	STO + (i)	33 61 24	
	RCL E	34 15	(new human tendancy) = (stoch. conservation)(old human tendancy) + (1 − stoch. cons.)(new guess)
	1	01	
	RCL C	34 13	
	STO x 5	33 71 05	
080	−	51	
	x	71	
	STO + 5	33 61 05	
	RCL D	34 14	computer goal = .75 if human tendancy < human equilibrium and = .25 otherwise
	RCL 5	34 05	
	−	51	
	1	01	
	+	61	
	f INT	31 83	
	2	02	
090	÷	81	
	4	04	
	h 1/x	35 62	
	+	61	
	2	02	(new comp. tend.) = 1/2 (old comp. tend.) + 1/2 (comp. goal)
	STO ÷ 7	33 81 07	
	÷	81	
	STO + 7	33 61 07	
	h Rc I	35 34	Extract the human opponent's last move from the behaviouristic state monitor. Save it temporarily in the stack.
	1	01	
100	0	00	
	−	51	
	4	04	
	÷	81	
	g FRAC	32 83	
	2	02	
	x	71	
	f INT	31 83	
	RCL E	34 15	code present outcome as a, b, c, or d and recall the appropriate payoff
	RCL A	34 11	
110	2	02	
	x	71	
	+	61	

REGISTERS

0	1	2	3	4	5	6	7	8	9
a	b	c	d	(Stoch.-beh.) human winnings	HUMAN TENDANCY	TOTAL HUMAN WINNINGS	COMPUTER TENDANCY	Counter for mix evaluator	Paradigm mix (0 = Stoch.)
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
a0	a1	b0	b1	c0	c1	d0	d1		

A	B	C	D	E	I
COMPUTER GUESS	RANDOM SEED	STOCHASTIC CONSERVATISM	a+d-b-c then human equilibrium	HUMAN GUESS	BEHAVIOURISTIC STATE MONITOR

Program Listing II

PROGRAM #1

STEP	KEY ENTRY	KEY CODE	COMMENTS
	h ST I	35 33	
	CLX	44	
	RCL(i)	34 24	
	STO + 6	33 61 06	Add payoff to winnings
	h F? 0	35 71 00	change (stoch - beh)
	CHS	42	human winnings counter.
	STO + 4	33 61 04	
120	h R↓	35 53	Recall last human move from stack
	h RC I	35 34	
	2	02	
	X	71	compute new
	+	61	behaviouristic
	1	01	state monitor
	0	00	
	+	61	
	h ST I	35 33	
	f LBL 5	31 25 05	
130	1	01	DSZ on R_8
	STO - 8	33 51 08	
	RCL 8	34 08	
	f x≠0	31 61	
	GTO 8	22 08	
	RCL 4	34 04	CHANGING PARADIGM
	f x=0	31 51	MIX: If stoch.
	GTO f a	22 31 11	human winnings
	f x<0	31 71	= behaviouristic,
	0	00	don't change.
140	f x>0	31 81	If Stoch. > beh.
	1	01	then use beh. more
	ENTER	41	If beh. > stoch.
	2	02	then use stoch. more
	STO ÷ 9	33 81 09	
	÷	81	
	STO + 9	33 61 09	
	g LBL f a	32 25 11	
	1	01	
	0	00	
150	STO 8	33 08	
	CLX	44	
	STO 4	33 04	
	f LBL 8	31 25 08	
	RCL B	34 12	random
	9	09	number
	9	09	generator
	7	07	
	X	71	
	g FRAC	32 83	
160	STO B	33 12	
	RCL 9	34 09	if paradigm mix >
	g x>y	32 81	random number play
	GTO 6	22 06	behaviouristically.
	h CF 0	35 61 00	STOCHASTIC PLAY
	RCL B	34 12	
	9	09	Random number
	9	09	generator
	7	07	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	X	71	
170	g FRAC	32 83	
	STO B	33 12	
	RCL 7	34 07	if human tendancy
	g x>y	32 81	> random number
	GTO 0	22 00	then choose "1"
	CLX	44	choose "0"
	GTO 7	22 07	
	f LBL 6	31 25 06	PLAY BEHAVIOURISTICALLY
	h SF 0	35 51 00	
	RCL (i)	34 24	if (reg)(b) + (1-reg)(a)
180	RCL D	34 14	> (reg)(d) + (1-reg) c
	g x>y	32 81	then choose "1"
	GTO 0	22 00	
	CLX	44	otherwise
	GTO 7	22 07	choose "0"
	f LBL 0	31 25 00	CHOOSE "1"
	1	01	
	f LBL 7	31 25 07	STORE GUESS
	STO A	33 11	
	f LBL D	31 25 14	DISPLAYS LAST GUESS
190	h RC I	35 34	
	1	01	
	0	00	
	−	51	(human guess) · (computer guess)
	4	04	
	÷	81	
	f INT	31 83	
	1	01	
	0	00	
	÷	81	
200	RCL E	34 15	
	+	61	
	DSP 1	23 01	
	h RTN	35 22	
	f LBL B	31 25 12	PAYOFF
	DSP 0	23 00	
	h RC I	35 34	determine last
	1	01	situation
	0	00	
	−	51	
210	2	02	
	÷	81	
	f INT	31 83	
	h X≠I	35 24	
	RCL (i)	34 24	display its payoff
	h R↓	35 53	
	h X≠I	35 24	
	h R↑	35 54	
	h RTN	35 22	
	f LBL C	31 25 13	SCORE
220	DSP 0	23 00	recall total
	RCL 6	34 06	human winnings
	h RTN	35 22	

LABELS

A GUESS	B PAYOFF	C SCORE	D DISPLAYS LAST GUESS	E R/S START	0 0 if behaviouristic
a STRATEGIES EQUAL	b	c	d	e RANDOM LOOP	1
0 CHOOSE 1	1	2	3	4	2
5 NEXT GUESS	6 BEHAVIOURISTIC PLAY	7 STORES GUESS	8 Used	9 Error	3

FLAGS

	ON	OFF
0	☐	☑
1	☐	☑
2	☐	☑
3	☐	☑

SET STATUS

TRIG		DISP	
DEG	☑	FIX	☑
GRAD	☐	SCI	☐
RAD	☐	ENG	☐
		n	0

Program Listing I

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	f LBL D	31 25 14	INPUT GAME
	f CL REG	31 43	storing a, b, c, d
	STO 3	33 03	
	h R↓	35 53	
	STO 2	33 02	
	h R↓	35 53	
	STO 1	33 01	
	h R↓	35 53	
	STO 0	33 00	
010	+	61	computing a+d-c-b
	CHS	42	
	+	61	
	+	61	
	CHS	42	
	STO D	33 14	
	RCL 0	34 00	Error if "a" is not the largest of a, b, c, or d
	RCL 1	34 01	
	g X>Y	32 81	
	GTO 9	22 09	
020	CLX	44	
	RCL 2	34 02	
	g X>Y	32 81	
	GTO 9	22 09	
	CLX	44	
	RCL 3	34 03	
	g X>Y	32 81	
	GTO 9	22 09	
	.	83	stochastic conservatism = .9
	9	09	
030	STO C	33 13	
	RCL 2	34 02	if c≥d then saddle point is "c". If V>0 refuse to play. Otherwise use only the (deterministic) behaviouristic strategy
	RCL 3	34 03	
	g X>Y	32 81	
	GTO 1	22 01	
	RCL 2	34 02	
	f X>0	31 81	
	GTO 9	22 09	
	1	01	
	STO 9	33 09	
040	CLX	44	
	h RTN	35 22	
	f LBL 1	31 25 01	IF c<d
	RCL 1	34 01	if b≥d then saddle point is "d". If V>0 refuse to play. Otherwise play behaviouristically.
	RCL 3	34 03	
	g X>Y	32 81	
	GTO 3	22 03	
	f X>0	31 81	
	GTO 9	22 09	
	1	01	
050	STO 9	33 09	
	CLX	44	
	h RTN	35 22	
	f LBL 3	31 25 03	COMPUTES P, Q, V
	RCL 0	34 00	
	RCL 3	34 03	
	X	71	
	RCL 1	34 01	
	RCL 2	34 02	
	X	71	
060	−	51	$V = \frac{ad-bc}{a+d-b-c}$ if V>0 reject game
	RCL D	34 14	
	÷	81	
	f X>0	31 81	
	GTO 9	22 09	
	RCL 0	34 00	$P = \frac{a-b}{a+d-b-c}$ store as human tendancy
	RCL 1	34 01	
	−	51	
	RCL D	34 14	
	÷	81	
070	STO 7	33 07	
	RCL 0	34 00	$Q = \frac{a-c}{a+d-b-c}$ store as human equilibrium and as initial human tendancy.
	RCL 2	34 02	
	−	51	
	RCL D	34 14	
	÷	81	
	STO 5	33 05	
	STO D	33 14	
	f P≷S	31 42	does "0" the first time these situations occur.
	STO 0	33 00	
080	STO 3	33 03	
	STO 4	33 04	
	STO 5	33 05	
	STO 6	33 06	
	1	01	
	.	83	
	1	01	
	÷	81	
	STO 1	33 01	does "1" the first time these situations occur
	STO 2	33 02	
090	STO 7	33 07	
	f P≷S	31 42	
	CLX	44	
	h RTN	35 22	
	f LBL E	31 25 15	n → START
	h ABS	35 64	loop n times, n is an integer ≥1
	1	01	
	+	61	
	h ST I	35 33	
	.	83	a tested random seed
100	5	05	
	2	02	
	8	08	
	4	04	
	1	01	
	6	06	
	3	03	
	g LBL fe	32 25 15	random number generator
	9	09	
	9	09	
110	7	07	
	X	71	
	g FRAC	32 83	

REGISTERS

0	1	2	3	4	5	6	7	8	9
a	b	c	d		Q (INITIAL HUMAN TENDANCY)		P (COMPUTER TENDANCY)	COUNTER FOR mix evaluator	
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9

A	B	C	D	E	I
	RANDOM SEED	STOCHASTIC CONSERVATISM	a+d-b-c then Q (HUMAN EQUILIBRIUM)		BEHAVIOURISTIC STATE MONITOR

01412 PROGRAM #2 — Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
	f DSZ	31 33	
	GTO fe	22 31 15	
	STO B	33 12	
	1	01	initializing counter and state monitor
	0	00	
	hST I	35 33	
	STO 8	33 08	
120	CLX	44	
	f LBL 5	31 25 05	flash "0."
	PAUSE	35 72	
	GTO 5	22 05	
130			
140			
150			
160			
170			
180			
190			
200			
210			
220			

LABELS

A	B	C	D INPUT GAME	E n→ START
a	b	c	d	e RANDOM LOOP
0	1 if c < d	2	3 CALCULATES P, Q, V	4
5	6	7	8	9

FLAGS

0
1
2
3

SET STATUS

FLAGS			TRIG	DISP
	ON	OFF		
0	☐	☒	DEG ☒	FIX ☒
1	☐	☒	GRAD ☐	SCI ☐
2	☐	☒	RAD ☐	ENG ☐
3	☐	☒		n 0

Program Explanation

The calculator observes the human opponent's play using two different learning strategies, and then learns which of these is more successful.

Stochastic learning

It turns out that the conservative strategies in this game are for the calculator to guess "0" or "1" with equal frequency, and the human opponent to guess "1" 2/3 of the time. (For a derivation, see the references.)

If the human opponent guesses "0" too often, it pays for the calculator to guess "1" more often. This is the crux of the stochastic strategy. A record is kept ("human tendancy") of the frequency of human "0" or "1" responses. This record is biased more toward recent rounds of the game. If this is less than 2/3, the computer tends to respond with "1" more often (the "computer tendancy" gets raised.)

The learning speed of the stochastic paradigm has been adjusted empirically. It shouldn't learn too fast or too far, because then the human opponent will catch on and retaliate. It shouldn't shift too slowly either, since this will impair the ability to react to what the human opponent is doing.

Behaviouristic learning

This paradigm looks for regularities in the human opponent's responses to reinforcements in particular situations. It looks at 8 situations, based on four general situations. These four situations — a, b, c, and d - are the four outcomes in the matrix illustrated at the right. These letters will also be used below to indicate the payoffs (to the human opponent) of the outcomes. These four situations are divided into two sub-situations each, on the basis of the human's move on the previous round. For example, situation c1 is human "0" calculator "1" preceeded by human "1". A record is kept of the liklihood of the human doing a "1" after each such situation. These records are biased more towards recent rounds of the game. Using this information, the calculator attempts to predict future behaviour on the basis of past behaviour. Consider the following game and assume that the calculator has decided

CALCULATOR \ HUMAN	0	1
0	a	b
1	c	d

CALCULATOR \ HUMAN	0	1
0	2	-1
1	-2	1

that results of previous rounds indicate a liklihood of $\frac{1}{2}$ that the human will choose "1" (and thus the same liklihood of $\frac{1}{2}$ of choosing "0"). If the computer guesses "0" its expectation of payoff is $\frac{1}{2}(2) + \frac{1}{2}(-1)$, ie $\frac{1}{2}$ point lost to the human opponent. If the computer guesses "1", the expectation is $\frac{1}{2}(-2) + \frac{1}{2}(1)$, ie $\frac{1}{2}$ point gain to the calculator. The calculator maximizes its own expectation and chooses "1". In this way it watches the eight behavioural states and predicts what the human opponent will do based on his past behaviour.

In general, if the calculator expects to lose more on choosing "0" ie if (human prob. of 1)(b) + (human prob. of 0)(a) > (human prob. of 1)(d) + (human prob. of 0)(c), ie if (human prob. of 1) $< \frac{a-c}{a+d-b-c}$, then the calculator chooses "1" next time.

Learning to learn

The calculator always compiles information for both strategies. It begins by playing stochastically, but periodically a check is made as to which of the two learning paradigms has been doing better recently. The paradigm that is doing better is given an increased probability of being used (the "paradigm mix" is changed.)

THE GENERALIZED VERSION

The calculator will play any 2x2 zero-sum matrix game. The general matrix game is illustrated at the right. In the standard game, the values of a, b, c, and d were 2, -1, -2 and 1 respectively. Games other than the standard game have to be input using card #2. There is an important restriction: "a" must be greater than "b" "c", or "d". Notice that this is the case in the standard game. If you want, for example, the game on the left below, then transform it as indicated:

CALCULATOR \ HUMAN	0	1
0	a	b
1	c	d

CALCULATOR \ HUMAN	0	1
0	1	-2
1	-1	2

EXCHANGE ROWS →

CALCULATOR \ HUMAN	0	1
0	-1	2
1	1	-2

EXCHANGE COLUMNS →

CALCULATOR \ HUMAN	0	1
0	2	-1
1	-2	1

Note that nothing is really changed, only the labels "0" or "1" for each player have changed.

01412D Program Explanation (cont)

The computer will accept games which have "saddle points." A saddle point is an outcome at which rational players will stick. Consider the example on the right. The computer will always choose "1" since it loses less regardless of what the human does: loses -1 vs 2, or 0 vs 1. Knowing this, the human will choose "1" since he would rather have 0 than -1. It doesn't pay for either to deviate from this outcome. In such cases the computer can't be induced to learn; it will keep playing the saddle point. In this example the saddle point is 0, so the "value" of the game, ie the long run expected winnings per turn, is zero. Consequently the game is fair to both sides. The computer will not accept any games that are unfair to it. It responds to these by flashing "Error". It accept, of course, games that are fair or favourable to itself.

		HUMAN 0	HUMAN 1
CALCULATOR	0	2	1
	1	-1	0

Solving the game

The conservative strategy for 2x2 zero-sum saddle point games is to keep choosing the saddle point. For non saddle point games, the conservative strategy is to randomly vary your guesses. We will indicate by the letter "p" the proper frequency for the computer to guess "1" for the conservative strategy. The frequency for guessing "0" is of course $1-p$. Similarly, "Q" is the human opponent's conservative strategy. It turns out that:

$$p = \frac{a-b}{a+d-b-c} \qquad Q = \frac{a-c}{a+d-b-c}$$

The "value" of such a game is the long run expected winnings per turn of conservative play. The value of the game to the human opponent is:

$$V = \frac{ad-bc}{a+d-b-c}$$

For a derivation, see any book on game theory. Particularly useful is Rapoport (see references).

After making initial computations of the game, the calculator flashes "0." and you read in the first card and are ready to begin play.