

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 ~~also~~ 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~~ 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.

This program has been verified only with respect to the numerical example given in *Program Description II*. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP NOR THE CONTRIBUTOR MAKES ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND WITH REGARD TO THIS PROGRAM MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER HP NOR THE CONTRIBUTOR SHALL BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, USE OR PERFORMANCE OF THIS PROGRAM MATERIAL.

Sketch(es)

		HUMAN	
		0	1
CALCULATOR	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 $\boxed{2} \boxed{E}$ and the calculator returns 0.0 and is ready to begin. $\boxed{0} \boxed{A} \rightarrow 0.1$ indicating you chose "0" while it chose "1". $\boxed{B} \rightarrow -2.$ indicating that you lost two points on that round. $\boxed{C} \rightarrow -2.$ indicating that you are losing by two points so far. $\boxed{D} \rightarrow 0.1$ displaying the last outcome in case you forgot it. $\boxed{0} \boxed{A} \rightarrow 0.0$ indicating you chose "0" and it chose "0". $\boxed{0} \boxed{A} \rightarrow 0.1$, $\boxed{0} \boxed{A} \rightarrow 0.1$, $\boxed{0} \boxed{A} \rightarrow 0.1$, $\boxed{0} \boxed{A} \rightarrow 0.1$, $\boxed{C} \rightarrow -10.$

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

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 Dilemma: A Study in Conflict and Cooperation. The University of Michigan Press, Ann Arbor, 1970.

GUESS PAYOFF SCORE GUESS START

[illegible]

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	GTO 5	22 05	Branch for generalized program		STO 7	33 07	
	F LBL E	31 25 15	<u>n = START</u>		F P = S	31 42	
	F CL REG	31 43			GTO 5	22 05	DO FIRST GUESS
	h ABS	35 64	loop n times,	060	F LBL A	31 25 11	<u>GUESS</u>
	1	01	n is an		ENTER	41	
	+	61	integer ≥ 1		F INT	31 83	Error if
	h ST I	35 33			g X ≠ Y	32 61	noninteger input
	.	83			GTO 9	22 09	or < 0
	5	05			f x < 0	31 71	or > 1
010	2	02	a tested		GTO 9	22 09	
	8	08	random seed		1	01	
	4	04			h x ≠ y	35 52	
	1	01			g X > Y	32 81	
	6	06		070	GTO 9	22 09	
	3	03			STO E	33 15	
	g LBL fe	32 25 15			2	02	
	9	09			STO ÷ (i)	33 81 24	
	9	09			÷	81	
	7	07			STO + (i)	33 61 24	
020	x	71	random		RCL E	34 15	
	g FRAC	32 83	number		1	01	
	f OSZ	31 33	generator		RCL C	34 13	
	GTO fe	22 31 15			STO x 5	33 71 05	
	STO B	33 12		080	-	51	
	1	01			x	71	
	STO 3	33 03			STO + 5	33 61 05	
	CHS	42			RCL D	34 14	
	STO 1	33 01			RCL 5	34 05	
	2	02			-	51	
030	STO 0	33 00	a = 2		1	01	
	CHS	42	b = -1		+	61	
	STO 2	33 02	c = -2		F INT	31 83	
	1	01	d = 1		2	02	
	0	00		090	÷	81	
	h ST I	35 33	initializing counter		4	04	
	STO 8	33 08	and state monitor		h 1/x	35 62	
	.	83			+	61	
	5	05	computer		2	02	
	STO 7	33 07	tendency = .5		STO ÷ 7	33 81 07	
040	.	83			÷	81	
	9	09	stochastic		STO + 7	33 61 07	
	STO C	33 13	conservatism = .9		h RCL I	35 34	
	2	02			1	01	
	ENTER	41	human tendency =	100	0	00	
	3	03	human equilibrium = $\frac{2}{3}$		-	51	
	÷	81			4	04	
	STO D	33 14			÷	81	
	STO 5	33 05			g FRAC	32 83	
	F P = S	31 42			2	02	
050	STO 0	33 00	store human		x	71	
	STO 3	33 03	equilibrium		F INT	31 83	
	STO 4	33 04	in behaviouristic		RCL E	34 15	
	STO 5	33 05	registers.		RCL A	34 11	
	STO 6	33 06		110	2	02	
	STO 1	33 01			x	71	
	STO 2	33 02			+	61	

REGISTERS									
0	1	2	3	4 (Stoch-beh)	5 HUMAN	6 TOTAL	7 COMPUTER	8 Counter for	9 Paradigm mix
a	b	c	d	human winnings	TENDANCY	HUMAN WINNINGS	TENDANCY	evaluator	(0 = Stoch.)
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
a0	a1	b0	b1	c0	c1	d0	d1		
A COMPUTER GUESS		B RANDOM SEED		C STOCHASTIC CONSERVATISM		D a+d-b-c then human equilibrium		E HUMAN GUESS	
								I BEHAVIOURISTIC STATE MONITOR	

PROGRAM #1

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
	hST I	35 33			X	71	
	CLX	44		170	gFRAC	32 83	
	RCL(i)	34 24			STO B	33 12	
	STO + 6	33 61 06	Add payoff to winnings		RCL 7	34 07	if human tendency
	h F? 0	35 71 00	change (stoch - beh)		g X > Y	32 81	> random number
	CHS	42	human winnings counter.		GTO 0	22 00	then choose "1"
	STO + 4	33 61 04			CLX	44	choose "0"
120	h R↓	25 53	Recall last human move		GTO 7	22 07	
	h RC I	35 34	from stack		F LBL 6	31 25 06	PLAY BEHAVIOURISTICALLY
	2	02			h SF 0	35 51 00	
	X	71	compute new		RCL(i)	34 24	if (reg)(b) + (1-reg)(a)
	+	61	behaviouristic	180	RCL D	34 14	> (reg)(d) + (1-reg) c
	1	01	state monitor		g X > Y	32 81	then choose "1"
	0	00			GTO 0	22 00	
	+	61			CLX	44	otherwise
	hST I	35 33			GTO 7	22 07	choose "0"
	F LBL 5	31 25 05			F LBL 0	31 25 00	CHOOSE "1"
130	1	01			1	01	
	STO - 8	33 51 08	DSZ on R8		F LBL 7	31 25 07	STORE GUESS
	RCL 8	34 08			STO A	33 11	
	Fx ≠ 0	31 61			F LBL D	31 25 14	DISPLAYS LAST GUESS
	GTO 8	22 08		190	h RC I	35 34	
	RCL 4	34 04	CHANGING PARADIGM		1	01	
	Fx = 0	31 51	MIX: IF stoch.		0	00	
	GTO fa	22 31 11	human winnings		-	51	
	Fx < 0	31 71	= behaviouristic,		4	04	
	0	00	don't change.		÷	81	
140	Fx > 0	31 81	IF stoch. > beh.		F INT	31 83	
	1	01	then use beh. more		1	01	
	ENTER	41	If beh. > stoch.		0	00	
	2	02	then use stoch. more		÷	81	
	STO ÷ 9	33 81 09		200	RCL E	34 15	
	÷	81			+	61	
	STO + 9	33 61 09			DSP 1	23 01	
	g LBL fa	32 25 11			h RTN	35 22	
	1	01			F LBL B	31 25 12	PAYOFF
	0	00			DSP 0	23 00	
150	STO 8	33 08			h RC I	35 34	
	CLX	44			1	01	
	STO 4	33 04			0	00	
	F LBL 8	31 25 08			-	51	
	RCL 8	34 12	random	210	2	02	
	9	09	number		÷	81	
	9	09	generator		F INT	31 83	
	7	07			h X ≠ I	35 24	
	X	71			RCL(i)	34 24	
	gFRAC	32 83			h R↓	35 53	
160	STO B	33 12			h X ≠ I	35 24	
	RCL 9	34 09	if paradigm mix >		h R↑	35 54	
	g X > Y	32 81	random number play		h RTN	35 22	
	GTO 6	22 06	behaviouristically.		F LBL C	31 25 13	SCORE
	h CF 0	35 61 00	STOCHASTIC PLAY	220	DSP 0	23 00	
	RCL B	34 12			RCL 6	34 06	
	9	09	Random number		h RTN	35 22	
	9	09	generator				
	7	07					

LABELS

FLAGS

SET STATUS

A GUESS	B PAYOFF	C SCORE	D DISPLAYS LAST GUESS	E n → START	0 02 if behavioural	FLAGS	TRIG	DISP
a STRATEGIES EQUAL	b	c	d	e RANDOM LOOP		ON OFF		
0 CHOOSE 1	1	2	3	4	2	0 ☐ ☒	DEG ☒	FIX ☒
5 NEXT GUESS	6 BEHAVIOURIST PLAY	7 STORES GUESS	8 Used	9 Error	3	1 ☐ ☒	GRAD ☐	SCI ☐
						2 ☐ ☒	RAD ☐	ENG ☐
						3 ☐ ☒		n 0

STEP	KEY ENTRY	KEY CODE	COMMENTS	STEP	KEY ENTRY	KEY CODE	COMMENTS
001	F LBL D	31 25 14	INPUT GAME		RCL 1	34 01	
	F CL REG	31 43			RCL 2	34 02	$V = \frac{ad-bc}{a+d-b-c}$
	STO 3	33 03			X	71	
	h RJ	35 53	storing	060	-	51	if $V > 0$
	STO 2	33 02	a, b, c, d		RCL D	34 14	reject game
	h RJ	35 53			÷	81	
	STO 1	33 01			fx > 0	31 81	
	h RJ	35 53			GTO 9	22 09	
	STO 0	33 00			RCL 80	34 00 34 03	
010	+	61			RCL 81	34 01 34 02	$P = \frac{a-b}{a+d-b-c}$
	CHS	42	computing		-	51	
	+	61	a+d-c-b		RCL D	34 14	store as human
	+	61			÷	81	tendency
	CHS	42		070	STO 7	33 07	
	STO D	33 14			RCL 0	34 00	
	RCL 0	34 00			RCL 2	34 02	$Q = \frac{a-c}{a+d-b-c}$
	RCL 1	34 01	Error if		-	51	
	gx > y	32 81	"a" is not		RCL D	34 14	store as human
	GTO 9	22 09	the largest		÷	81	equilibrium and as
020	CLX	44	of		STO 5	33 05	initial human tendency.
	RCL 2	34 02	a, b, c, or d		STO D	33 14	
	gx > y	32 81			F P = S	31 42	
	GTO 9	22 09			STO 0	33 00	does "0"
	CLX	44		080	STO 3	33 03	the first time
	RCL 3	34 03			STO 4	33 04	these situations
	gx > y	32 81			STO 5	33 05	occur.
	GTO 9	22 09			STO 6	33 06	
	.	83	stochastic		1	01	
	9	09	conservatism		.	83	
030	STO C	33 13	= .9		1	01	
	RCL 2	34 02	if $c \geq d$ then		÷	81	
	RCL 3	34 03	saddle point is "c".		STO 1	33 01	does "1" the
	gx > y	32 81			STO 2	33 02	first time these
	GTO 1	22 01	If $V > 0$ refuse	090	STO 7	33 07	situations occur
	RCL 2	34 02	to play.		F P = S	31 42	
	fx > 0	31 81	Otherwise use		CLX	44	
	GTO 9	22 09	only the		h RTN	35 22	
	1	01	(deterministic)		F LBL E	31 25 15	n → START
	STO 9	33 09	behaviouristic		h ABS	35 64	loop n times,
040	CLX	44	strategy		1	01	n is an integer
	h RTN	35 22			+	61	≥ 1
	F LBL 1	31 25 01	IF $c < d$		h ST I	35 33	
	RCL 1	34 01	if $b \geq d$ then		.	83	
	RCL 3	34 03	saddle point is "d".	100	5	05	
	gx > y	32 81			2	02	
	GTO 3	22 03	If $V > 0$ refuse		8	08	
	fx > 0	31 81	to play.		4	04	
	GTO 9	22 09	Otherwise play		1	01	
	1	01	behaviouristically.		6	06	
050	STO 9	33 09			3	03	
	CLX	44			g LBL fe	32 25 15	
	h RTN	35 22			9	09	
	F LBL 3	31 25 03	COMPUTES P, Q, V		9	09	
	RCL 0	34 00		110	7	07	
	RCL 3	34 03			X	71	
	X	71			g FRAC	32 83	

REGISTERS

0	1	2	3	4	5	6	7	8	9
a	b	c	d		Q (HUMAN TENDANCY)		P (COMPUTER TENDANCY)	COUNTER FOR MIX EVALUATOR	
S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
A		B RANDOM SEED		C STOCHASTIC CONSERVATISM		D a+d-b-c then Q (HUMAN EQUILIBRIUM)		E BEHAVIOURISTIC STATE MONITOR	

COMMENTS

LABELS					FLAGS	SET STATUS			
A	B	C	D	E		FLAGS		TRIG	DISP
			INPUT GAME	$n \rightarrow$ START	0				
a	b	c	d	e	1	ON OFF		DEG ☒	FIX ☒
				RANDOM LOOP		0	☐		
0	1	2	3	4	2	1	☐	GRAD ☐	SCI ☐
	if $c < d$		calculates P, Q, V			2	☐	RAD ☐	ENG ☐
5	6	7	8	9	3	3	☐		n <u>0</u>

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" $\frac{2}{3}$ 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 tendency") 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 $\frac{2}{3}$, the computer tends to respond with "1" more often (the "computer tendency" 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" preceded by human "1". A record is kept of the likelihood 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 ~~behavior~~ behaviour. Consider the following game and assume that the calculator has decided

	HUMAN	
	0	1
CALCULATOR	0	a b
	1	c d

that results of previous rounds indicate a likelihood of $\frac{1}{2}$ that the human will choose "1" (and thus the same likelihood 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 manner~~ way it watches the eight behavioural states and predicts what the human opponent will do based on his past behaviour.

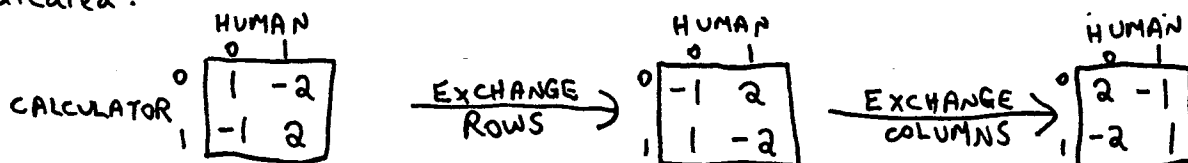
In general, if the calculator expects to lose more on choosing "0" ie if $(\text{human prob. of 1})(b) + (\text{human prob. of 0})(a) > (\text{human prob. of 1})(d) + (\text{human prob. of 0})(c)$, ie if $(\text{human prob. of 1}) < \frac{a-c}{a+d-b-c}$, then the calculator chooses "1" next time.

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 2×2 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:



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, i.e. 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 accepts, of course, games that are fair or favourable to itself.

	HUMAN	
CALCULATOR	0	2
	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:

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

$$q = \frac{a-c}{a+d-b-c}$$

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.