

【2021 中研院社會模擬工作坊】第二組「六月初四」結案報告

第一篇論文（編號 2）：*Mark, N. P. (2002). Cultural transmission, disproportionate prior exposure, and the evolution of cooperation. American Sociological Review, 323-344.*

論文介紹：

作者想要了解在何種條件下可以有效地讓所有行動者都與他人合作，他指出了文化傳遞機制的重要性，文化傳遞指的是行動者在與他人的互動中，會觀察他人行為結果進而改變自己的行為。作者想要證明正是此文化傳遞機制的存在，讓「合作」行為如此普遍。

文中討論的賽局，是在一個大的系統之內，每一次隨機地將兩個行動者配對，並進行一次性且無法離開的囚徒賽局，對於這種賽局的設定，過往的文獻都認為大家最終都會互相背叛（不合作）。而作者加入了文化傳遞的機制，使大家最終都成為合作者，他做出以下設定：每個行動者身上都帶有兩個屬性，第一個是策略（行為）：合作 C 或是背叛 D，第二個則為其 fitness（同時也是前一局獲得的報償 pay-off），以數字 0,1,2,3 表示。每一個時間點有五個步驟：第一是隨機配對行動者；二是行動者與對手比較 fitness 高低；第三步是根據 payoff-matrix 來決定各自的報償；第四步則是決定下一回合的行為，方法是在第二步驟比較中，fitness 較低者會轉換行為成與 fitness 較高者相同；第五步則是利用這一局所得到的 pay-off 來當作下回合的 fitness。以下面 Table1. 第一行第二列的組合舉例，當 C0 遇上 D1 時，因為 D1 的 fitness 較高，所以在「下一回合時」，C 會轉變至 D 策略。而在報償方面，C 與 D 的組合中，C 會得到 0 報償，D 會得到 3 報償，所以下一個時間點兩個行動者各自會變成 D0 與 D3。

Table 1. Consequences of Behaviors and Fitness Levels of Interactants at Time t for Behaviors and Fitness Levels of the Same Individuals at Time $t + 1$

	Defector		
	D0 at time t	D1 at time t	D3 at time t
Cooperator			
C0 at time t	C0, D3 at time $t + 1$	D0, D3 at time $t + 1$	D0, D3 at time $t + 1$
C2 at time t	C0, C3 at time $t + 1$	C0, C3 at time $t + 1$	D0, D3 at time $t + 1$
C3 at time t	C0, C3 at time $t + 1$	C0, C3 at time $t + 1$	C0, D3 at time $t + 1$

		ACTOR B	
		Cooperate	Defect
ACTOR A	Cooperate	2, 2	0, 3
	Defect	3, 0	1, 1

Figure 1. Example of a Prisoner's Dilemma Reward Structure

$$pC0_{t+1} = pC0_t(pD0_t) + pC2_t(pD0_t + pD1_t) + pC3_t(pD0_t + pD1_t + pD3_t).$$

由於我們可以確切知道每一組類別經過一次配對後的結果，所以我們可以根據上一個時間點六種行動者（C0、C2、C2、D0、D1、D3）的比例，來計算出下一個時間點六種行動者的比例。因此只要確保每一兩兩配對是隨機分配（獨立），則可以用解析解來分析系統的演變。

如果設定初始狀態的六種行動者佔比為（0.25, 0.25, 0, 0, 0.25, 0.25），則我們只要重複進行一百回合的各類別比例變化的計算後，即可以得到文中的 Table 2. (p. 335)。文中 Table 3. (p. 336) 則是看三種不同比例的初始狀態，檢視系統最終的穩定狀態，結果顯示，即便一開

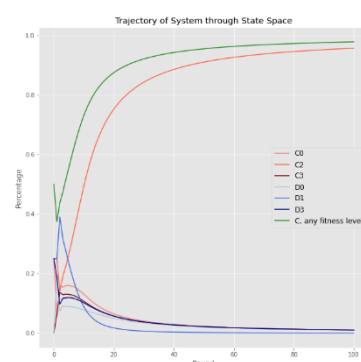
始背叛者占 99%，在經過九萬九千多次回合後，系統最終還是會達到合作者佔絕對多數的狀態。作者就以此顯示 Cultural transmission 對於解釋「合作」行為普及的重要性。

研究模擬、重製方法與結果：

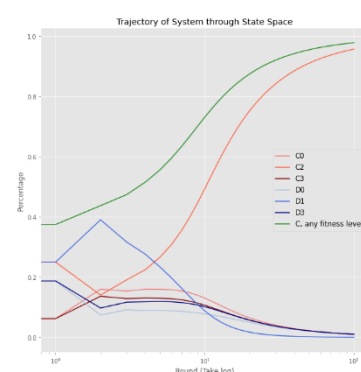
我們使用 Pythonx 來進行文中 Table 2.與 Table 3.兩個表格的數字重現。利用迴圈的方式，來計算指定的回合次數中，各類別行動者的比例。我們所重置得出來的數字結果，與文章中兩個表格的完全相同（由於方式裡沒有隨機性的存在，因此相當合理）。另外，我們也額外將 Tabel 2.的結果視覺化呈現，可以更清楚看出各類別變化軌跡的樣態。

關於重製過程中，有一些小細節可以提出。首先是，Tabel 3. 的部分，我們實作方式的邏輯其實跟作者是反向的，作者原本是想看在不同初始狀態的設定下，需要多少輪才會「接近穩定」狀態，但由於作者文中沒有提到對於何謂「接近穩定」的定義與確切數值，所以我們達到完全重現表格中的數字，反向指定回合數目，看是否經過作者所報告的回合數之後，各類別比例的結果會相同。另外，如果要重置此篇文章，在程式實作方面要特別注意浮點數的「精度（percision）」問題，尤其是經過的回合數多時，可能會因為誤差的持續累積而達到錯誤數字結果，需要特別使用 Python 中 Demial 的套件來處理。

	Initial State (t = 0)	Round 1	Round 2	Round 3	Round 4	Round 5	Round 10	Round 20	Round 100
Behavior-Fitness Category									
C0		0.25	0.06	0.16	0.15	0.16	0.13	0.06	0.01
C2		0.25	0.25	0.14	0.19	0.23	0.27	0.49	0.96
C3		0.00	0.06	0.14	0.13	0.13	0.11	0.06	0.01
D0		0.00	0.19	0.07	0.09	0.09	0.08	0.05	0.01
D1		0.25	0.25	0.39	0.32	0.28	0.23	0.09	0.02
D3		0.25	0.19	0.10	0.12	0.12	0.10	0.06	0.01
C, any fitness level		0.50	0.38	0.44	0.47	0.52	0.56	0.73	0.88
Mean fitness of cooperators		1.00	1.83	1.58	1.62	1.63	1.66	1.79	1.92
Mean fitness of defectors		2.00	1.30	1.22	1.27	1.31	1.34	1.47	1.51



	Initial State (t = 0)	State Close to Attractor Occupied by Indicated Time
Behavior-Fitness Category		
C0	0.2500000000	0.0001118158
C2	0.2500000000	0.9995527867
C3	0.0000000000	0.0001117908
D0	0.0000000000	0.0001117658
D1	0.2500000000	0.0000000500
D3	0.2500000000	0.0001117908
Initial State (t = 0) State Close to Attractor Occupied by Indicated Time		
Behavior-Fitness Category		
C0	0.0900000000	0.0001118136
C2	0.0100000000	0.9995527958
C3	0.0000000000	0.0001117886
D0	0.0000000000	0.0001117636
D1	0.8100000000	0.0000000500
D3	0.0900000000	0.0001117886
Initial State (t = 0) State Close to Attractor Occupied by Indicated Time		
Behavior-Fitness Category		
C0	0.0100000000	0.0000111804
C2	0.0000000000	0.9999552790
C3	0.0000000000	0.0000111801
D0	0.0000000000	0.0000111799
D1	0.0000000000	0.0000000005
D3	0.9900000000	0.0000111801



第二篇論文（編號 10）：*Fonoberova, M., Fonoberov, V. A., Mezic, I., Mezic, J., & Brantingham, P. J. (2012). Nonlinear dynamics of crime and violence in urban settings. Journal of Artificial Societies and Social Simulation, 15(1), 2.*

論文介紹：

這篇作品主要想要建構模型，來討論在不同人口規模的城市中，警察、公民與犯罪者數量上的互動關係。作者藉由 **agent-based** 所模擬出的模型，發現了諸多現象：首先，城市中執法人員的數量必須隨人口規模增加，才得以將犯罪活動維持在穩定的低程度。第二，平均每千人的犯罪事件數與人口數的關係為非單調（**non-monotonic**）關係。另外，暴力事件在小型城市中以全面爆發的方式擴散，而在大型城市中卻轉變成特定時空的、去中心化的分布。此外，作者也發現此模型中存在閾值，一旦執法人員數量被減少到某個臨界值，犯罪活動便會急遽地增加。

在文獻回顧的部分，作者利用不同的主流犯罪學理論，來當作其 **agent-based** 模型建立的基礎，第一大類是 **routine activities theory** (Cohen, 1979) 以及 **situational crime prevention perspectives** (Clarke, 1995)。Routine activities theory 認為在缺乏護衛者的情況下，有犯罪動機的加害者遇上適合的受害者，只有在此情形犯罪才會發生；此外，此理論認為犯罪者會在日常活動的地區犯案。第二大類則是 **general strain theory** (Agnew, 1992; Angew, 2001)，此理論認為在外部社會環境，對於阻擋個人或群體的社會、經濟等層面的機會，犯罪作為一種可行的回應，例如警察偏袒某族群人民的不公現象被認為是造成社會動盪的原因之一。

本文 **agent-based model** 的設定相當複雜，概括來說，模型中有 2 類行動者：分別是 **citizens**（公民）與 **LEOs**（執法人員）。公民的狀態有二種，分別是 **active**（犯罪者）或 **quiescent**（未進行犯罪者），公民的狀態取決於自身與周遭環境等因素。有固定比例的公民會總是犯罪、總不犯罪、隨周遭環境決定犯罪；執法人員職責為偵測周遭犯罪行為，並且逮捕犯罪者，全數事件發生在範圍的方格之中，Figure1 為示意圖。

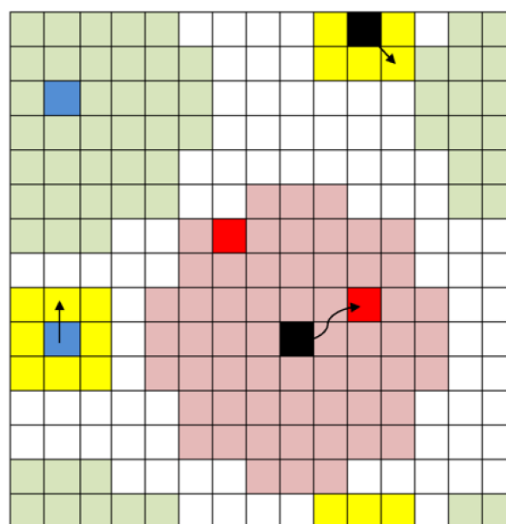


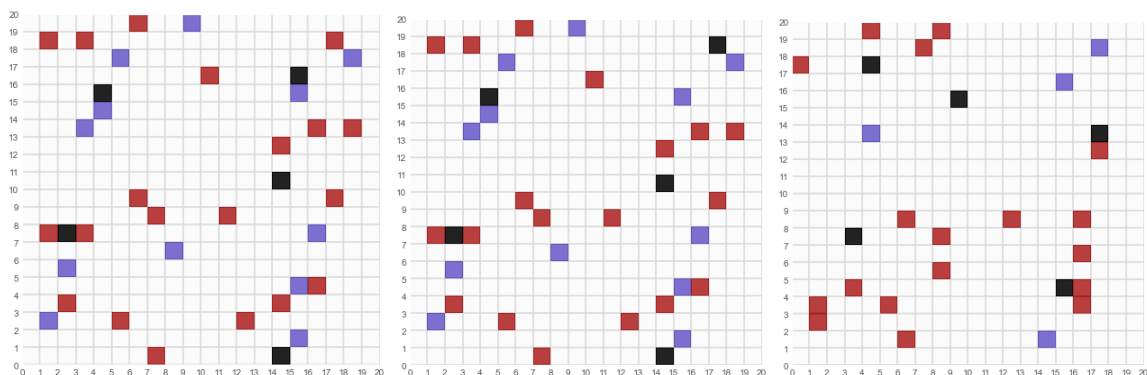
Figure 1. Sketch of the model. Currently quiescent (active) citizens are represented by blue (red) cells, LEOs are represented by black cells. Yellow cells show Moore neighborhood for an agent. During a move, an agent picks a random cell in his/her neighborhood and if the selected cell is unoccupied, he/she moves there, otherwise the agent stays put. Pink cells show a vision of radius 4 for the LEO (black cell). The LEO arrests the nearest active citizen in its field of vision. Green cells show vision of radius 4 for the citizen (blue cell). The citizens calculate their state depending on conditions on the lattice within their field of vision.

模型中有相當多參數的設定， H 為每位公民被隨機分配到的艱辛程度（hardship）； L 則是對執法當局的合法性，每位公民擁有的值皆相同，介於 0 到 1 之間， H 與 L 決定了公民的委屈程度（grievance）， $E=H(1-L)$ 。另一方面，每位公民否敢於犯罪的冒險程度（risk aversion）為 K ； V 為圓半徑，是每位公民可以觀察到周遭的視野範圍距離；每位公民的感知風險（citizen's perceived risk） N 被定義為 $N=KP$ ， P 為執法人員與公民比經過特定函數所得到的值。

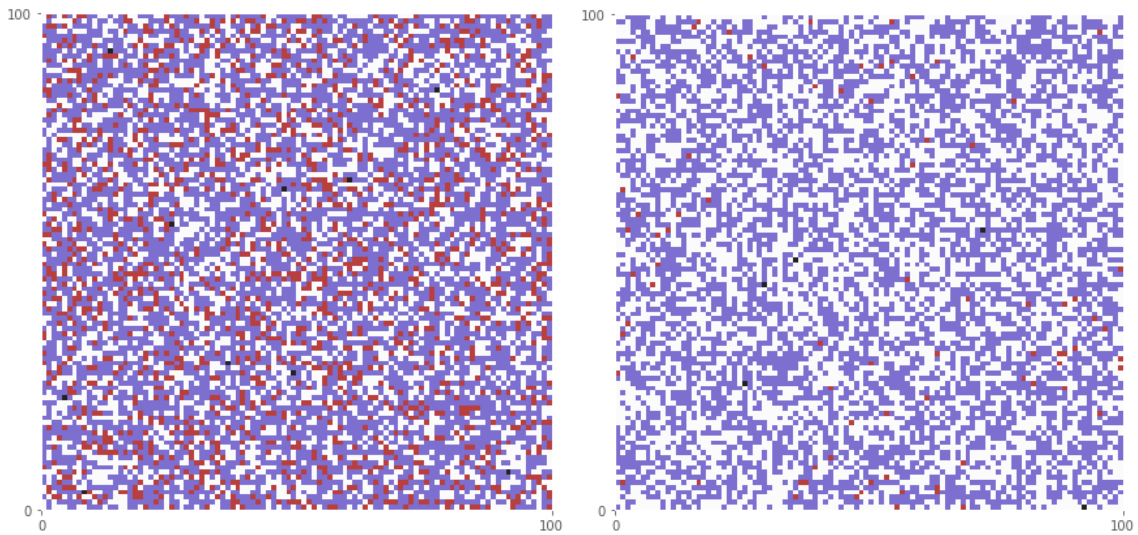
研究模擬、重製方法與結果：

模型步驟大致如下：第一步隨機放置行動者到方格上，第二步是對所有公民設定參數（如：風險感知等），第三步是讓總是犯罪的公民先關進監牢。第四等設置好環境後，就隨著 agent-based Model 的設定，進行每日的每人行動（agent 移動與警察逮捕犯罪者），過程中可以記錄犯罪者比例、監獄人數筆記、反罪高峰地區與時間等，並討論這些值與城市規模、視野大小等等參數之間的關係。

要重製這篇研究，最重要的就是要寫出程式碼，來執行文章所描述的 agent-based model，我們使用 Python 語言，利用物件導向的方法設計出 citizen, LEO, lattice，依照規則讓他們行動與互動。從結果來看，我們算是一半成功一半失敗，成功在於我們有實作出符合規則的 agent-based model（見下圖示意）。但我們面臨到的困難是，時間與運算資源不足，文章中的每一格圖表設定的天數都是 1000 天、尺寸最大到 600 x 600，而雖然我們的程式在天數不多或地圖尺寸不大的情況下可以運行，但要在如此有限的期限內重製出此研究的圖表與數據，是相當難以企及的。未來如果要繼續完成的話，可以奠基在此份程式碼之上（基本架構已完成，只是需要花費大量時間），並且增進程式的運行效率（或利用運行速度更快的 C++ 等語言）。順帶一題，我們在重製此篇研究的過程中，發現有兩處問題，一是文中的 function P of “sigmoidal” type 的公式有誤 (p.4)，在 Sigma 內的分子中， k' 前不應該加負號（沒有此負號才與 Figure2. 圖形相同）；二是文中對於 vision 的判定沒有清楚的定義，並不知道作者是如何判斷方格是在半徑之內，我們最後推測，是以格子中心點來計算距離，距離大於 ($vision + 0.25$) 則判定在視野範圍內。



（左：初始狀態；中：右上 LEO（黑點）執行 arrest 行動；右：整個 lattice 經過 1 天）



（左：根據文中 Figure 8. 參數，定義出 100x100 的初始 lattice；右：左圖在 agent-based model 的設定下，經過 100 天的狀態）