An agent-based model for promoting modest technologies

Yuji Saikai

Abstract

Promotion of a technology often encounters a gap between private and public benefits if it has only modest appeal to individuals. Information is crucial at the early stages of the adoption process. At the late phase, however, what if the society is saturated with the required information for decision making, yet people have not made up their minds or are indifferent? The issue becomes complex if individual decisions involve social interaction. To provide a policy tool for the scenario, we build an agent-based model and emphasize the inadequacy of the gold standard, RCT, for studying complex systems.

Keywords: ABM, bounded rationality, dynamic social network, norm, RCT, technology adoption, tipping point

Introduction

An adoption process of novel technology takes stages. At the early phases, as it is new to everyone, information is evidently one of the key determinants, and therefore how it is transmitted is of great interest. In development economics, it is often modeled as learning (Bandiera and Rasul, 2006; Conley and Udry, 2010) using Bayesian approach (Aldana, Foltz, Barham, and Useche, 2011). With detailed network data, Banerjee, Chandrasekhar, Duflo, and Jackson (2013) claim that it is crucial to figure out effective 'injection points' in order to maximize the information diffusion. However, as the process matures and the technology becomes familiar, the adoption hinges on different factors. For example, Foster and Rosenzweig (2010) explain that in Kenya fertilizer was no longer new at the time of the study in early 2000, and therefore learning did not play a role. Nevertheless, the issue was clearly the adoption of fertilizer (Duflo, Kremer, and Robinson, 2008). So, it seems legitimate to ask: what if the society is fully saturated with the information, yet people have not made up their minds or are indifferent about the technology? At that stage, what diffuse are not ingredients for decision making but decisions themselves. This is one of the questions we try to answer by constructing an agent-based model.

Economists traditionally tend to think that, taking rationality as given, suboptimal decisions are caused by a lack of the right environment, e.g. resources, information and institutions. As behavioral economists point out, even in the 'right' environment, people may still fail to make decisions, let alone making optimal decisions, perhaps due to the subtlety in benefits or procrastination (Thaler and Sunstein, 2008). We think that many issues of technology adoption in developing countries can be seen and analyzed from this perspective.

Similar to public goods provision, the problem often involves a significant gap between private benefits and social benefits perceived by respectively individuals and policy makers (Duflo, Kremer, and Robinson, 2011; Kremer and Miguel, 2007). In other words, if a technology in question is beneficial enough for both stakeholders, the gap will not be significant and may be relatively simple to resolve. In contrast, if most people see it disadvantageous, then promoting such technology will likely be ill-conceived. It seems, therefore, situations that require careful analysis arise when most people do not find definite values and, as a result, end up essentially indifferent. It may be because of high adjustment costs, insufficient adaptation to local environments, or cultural incompatibility. In any case, policy makers are promoting a technology with modest appeal to individuals.

Within the context of social diffusion for technology adoption, we address a more general question: how adequate randomized controlled trial is for studying complex systems. Over the past decade, randomized controlled trial (RCT) has been the most celebrated approach to analyzing intervention effects. It tries to evaluate policies by conducting experiments and directly inferring causal relationship. However, the lack of theories has received much criticism (Deaton, 2010). In particular, without very strong assumptions (i.e. homogeneous data-generating process), RCTs are logically unable to establish external validity (Cartwright, 2007). The problem becomes evident in situations where the underlying process is so complex that it is almost impossible to defend such assumptions. Social diffusion is clearly one of the cases. Aside from the epistemological and technical issues, to us, the most disturbing aspect concerns estimated effects and their relevance for real-world policies.

Even if we dismiss the issue of external validity, the magnitude of effects that RCTs estimate may not provide relevant information for actual policies. By the very nature of randomization, the observed effects (if any) are averaged out and necessarily smaller than those of extreme cases. Given the scarcity of public resources, cost-effectiveness is often as important as causality for any magnitude. Therefore, besides average effects, it may be necessary to figure out extreme effects, which makes most of the resources. In complex systems, extreme effects do not necessarily require an extreme intervention because large difference may be caused by a qualitative shift in the internal mechanism—phase transition (Miller and Page, 2009). Comparing average effects and extreme effects generated by our model, we highlight the inadequacy of RCT for studying complex systems.

Model

Assumptions

There is a new technology that policy makers value and want to promote in a society. People fully understand the technology, i.e. its benefits and how to use it, and nothing prevents them from adopting it. In the subpopulation we investigate, however, no one is currently using it because some of them have not made up their minds or the others are indifferent. The society is populated by three types of people: Good, Bad, and Indifferent, whose proportions are respectively α, β, and $1-\alpha-\beta$ where $\alpha, \beta \in (0,1)$ and $\alpha+\beta < 0.5$. Until the first trial, each type behaves identically but thereafter takes a distinct response: Good continues to use it, Bad never uses it again, and Indifferent remains indifferent. The type proportions are fixed throughout.

To specify Indifferent's behavior as well as pre-trial behavior of Good and Bad, we build on the model of 'thoughtless conformity' to norms (Epstein, 2001). The basic idea is that in reality people automatically follow many conventions, either professional or social, when they make decisions rather than optimizing them every single time. Choosing a default option as a norm has two consequences: self-enforcement and thoughtlessness. As Axelrod (1986) explains, a norm is often established through recursive reinforcement—the norm supports the mechanism that brings about its emergence. According to Epstein, the latter feature is even more remarkable. Once the norm is entrenched, people do not give much thought to whether to follow it partly because it helps reduce their cognitive loads, i.e. thoughtlessness may be optimal decisions with bounded rationality. In our model, all individuals are assumed to start in the 'indifferent state', in which Goods and Bads act like Indifferent until the first trial. The following describes the behavioral rules in the indifferent state based on the notion of thoughtless conformity.

Person i has some neighbors in her local network, which is dynamic. She is indifferent about whether to adopt the technology and therefore follows the norm—the majority choice in her neighborhood. To do that, she samples neighbors' choices. Suppose the current neighborhood size is r_i, i.e. the local network extends form her position by degree r_i. Upon examining the neighbors' choices, she first looks at as far as $r_i + 1$ degrees, wondering if any new change is happening. Then, she compares two samples between $r_i + 1$ and r_i, and if there is significant difference she becomes alarmed and updates the search size to $r_i + 1$ for the next period. Otherwise, she then looks at $r_i - 1$ neighborhood hoping to reduce the cognitive effort. If there is no significant difference, she shrinks it to $r_i - 1$. If the norm is entrenched, i.e. no neighbors are acting differently, she barely gives a thought to her decision each time, which is modeled as the minimum size $r_i = 1$. Following Epstein (2001), we use 0.05 as a threshold value to determine 'significant difference' between two samples. It is not a critical parameter, but instead values around 0.05 turns out admissible for modeling the phenomenon of interest. If too large, the objective (balancing alertness and cognitive loads) would not be adequately modeled. If too small, r_i would tend to diverge due to the order of examination (alertness first). Notice that, in contrast to many social network models, here each local network is endogenously dynamic, evolving according to the individual's objective.

Specification

In order to implement the dynamics above, for ease of presentation, we use a 2-dimensional grid space where every cell is occupied by a single individual. While in some cases, e.g. modeling farmers attached to physical plots, the grid may be a natural choice, it is by no means the only way for the current model. We may use a 1-dimensional ring where neighbors sit only side by side (Epstein, 2001) or an abstract network topology without any reference to Euclidean space. The following is a typical realization under $\alpha = \beta = 0.1$ in 30×30 space:

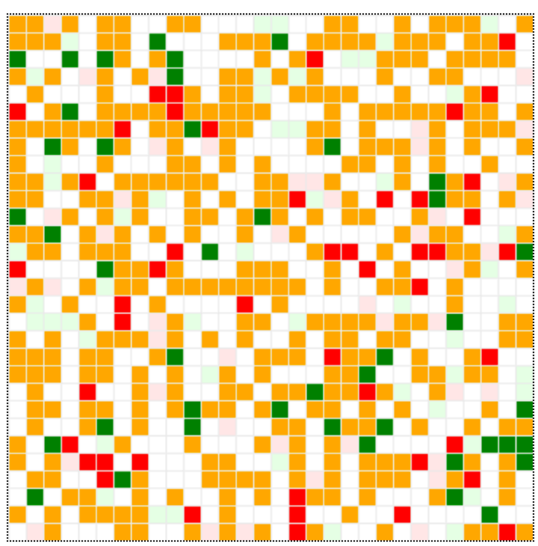

Figure 1: Grid space

where adoptions of Good, Bad, and Indifferent are denoted by ■, ■, and ■ respectively. In order to avoid boundary effects, the space is configured as toroidal. Regarding neighborhood, person j sitting at x_j is in i's r_i-neighborhood if $\|x_j - x_i\| \leq r_i$ where $\|\cdot\|$ is Euclidean norm. For example, suppose person i is Bad (■) examining $r_i = 2$ neighborhood.

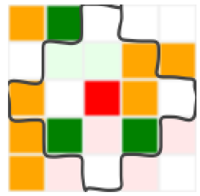

Figure 2: Neighborhood

As enclosed with the dark line, she finds 12 neighbors: 2 Goods (■) and 3 Indifferents (■) who are adopters, as well as 2 Goods (■), 1 Bad (■), and 4 Indifferents (□) who do not adopt it.

To reflect our ignorance about the 'true' decision process, in addition to the behavioral rules described in the previous section, people in the indifferent state randomly switch the adoption choice by 5% chance regardless of their neighbors. This also helps escape from unstable equilibria that could happen as an artifact of particular realizations.

The baseline specification is designed to implement a RCT equivalent. First, we set $\alpha = \beta = 0.1$ (i.e. 80% of the population is Indifferent). We start each run by randomly giving to half the population an adoption state—a treatment. Each run goes for 200 periods, which heuristically turned out a reasonable number to approximate a limit state because in most cases clear patterns emerged well before the 100th period. Then, by using the sample averages among the treated and controlled, we estimate the average treatment effect among the treated (Deaton, 2010)

$$\mathbf{E}(Y_{i1} - Y_{i0} \mid T_i = 1) = \mathbf{E}(Y_{i1} \mid T_i = 1) - \mathbf{E}(Y_{i0} \mid T_i = 0), \tag{1}$$

where Y is an adoption choice and T is a treatment state. After running the baseline, we modify it to implement a targeted intervention, where the equal-sized treatment group consists of all Goods and some randomly chosen Indifferents.

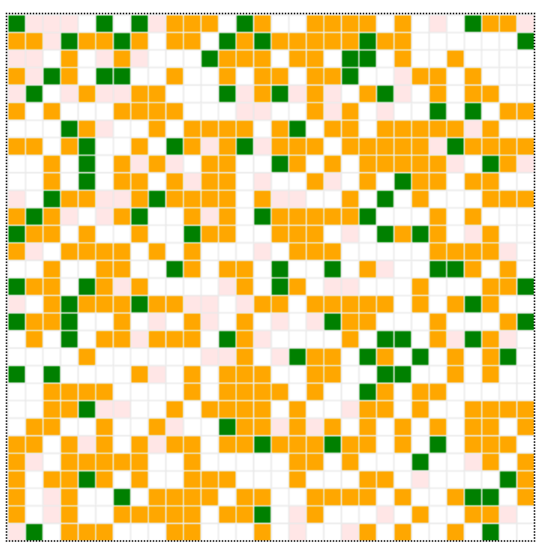

Figure 3: Targeted intervention

Notice that every Good receives a treatment and no Bad dose so (i.e. no ▪ or ▪). ABM allows us to experiment multiple interventions in the identical setting, which is usually impossible in the real world. Specifically, we fix a random seed, which is used in generating random numbers, so that we can retain the realization of all the stochastic processes involved. We iterate the process 500 times and report two distributions: adoption rate and average treatment effect.

Results & Discussion

Three classes of results

Starting from a random initial condition (Fig.1), some runs ended up with a medium adoption rate illustrated in the following figure. (Recall that the space is toroidal and therefore no division between 'north' and 'south').

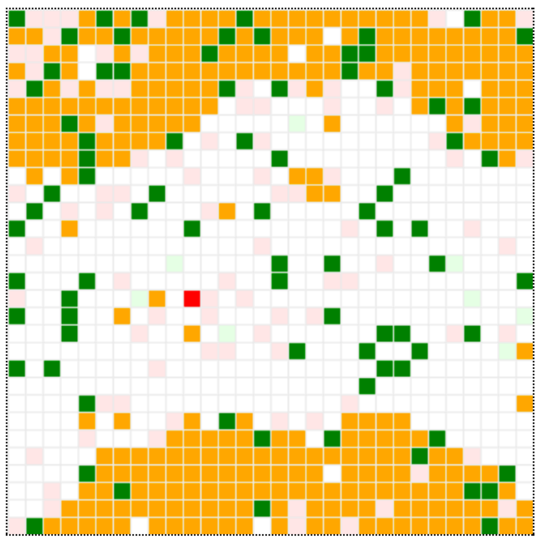

Figure 4: Medium adoption rate

Notice that some Goods (▪) are untapped despite the fact that they are the champions of the policy. It may be interpreted as a loss of efficiency. For much more details, we recorded three typical runs for low, medium, and high adoption rates.

Low: https://youtu.be/pD7sCw36_fc
Mid: https://youtu.be/5Ihb4MFc3NQ
High: https://youtu.be/N-mOrue5NQE

Adoption rate

Fig.5 shows the distribution of adoption rate under the randomized intervention (mean = 0.48 and std = 0.28). Since 10% of the population is Goods, who continue to adopt the technology regardless of the circumstances, the lowest decile is necessarily excluded from the support of the asymptotic distribution, which is realized here. Similarly, another 10% is made up of Bads, who never adopt it again, and therefore we do not expect to see more than 90% adoption rate.

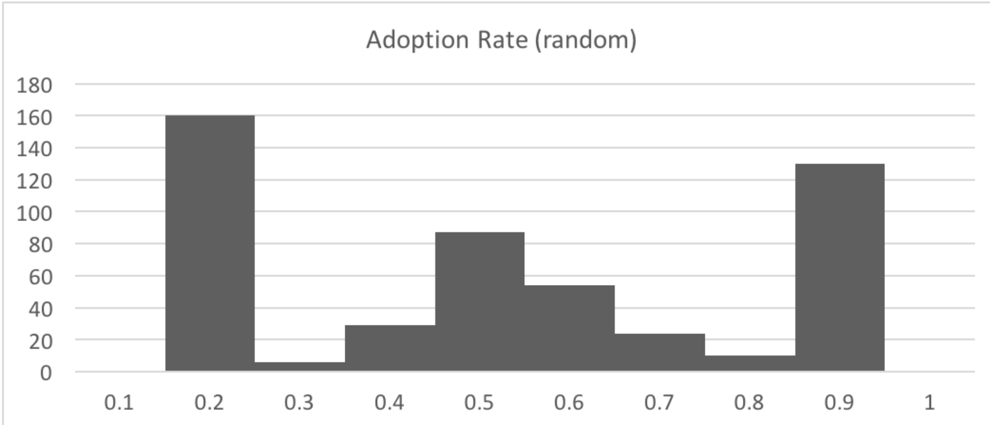

Figure 5: Distribution of adoption rate (random intervention)

Two polar masses are intuitive because as the self-enforcement property of norm emerges, Indifferents are likely to cascade into either extreme. However, there are a non-trivial number of realizations around 50% (medium adoption rate). This came as a surprise. Since we thought that balancing equilibria would be unstable and broken by the random noise in decision rules, the result is interesting to us. Even though the vast majority has no strong preference but rather follows norms, a sharp division can emerge as an equilibrium at the macro level, a similar phenomenon to the famous segregation model (Schelling, 1971).

Another aspect of interest is a clear indication that a central limit theorem will not apply to situations characterized by strong social interaction, i.e. each sample point is not independently generated. As opposed to the treatment effect discussed below, here the dependent variable is an individual decision itself, which is by the nature of the setting very dependent on each other. Although there are a number of variants of central limit theorem, some of which only require weak independence, the dependence in this case appears to be strong and likely violates the conditions.

Regarding the targeted intervention, as clearly seen in the following histogram, we have a qualitatively different distribution (mean = 0.76 and std = 0.18).

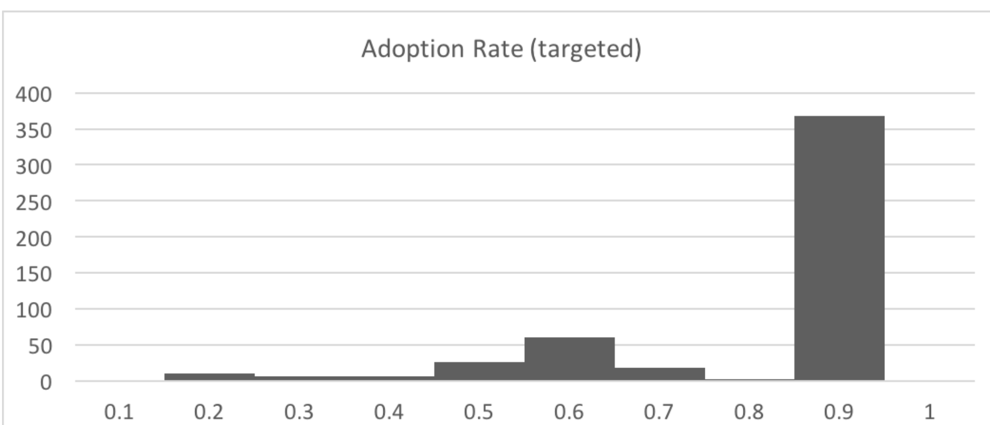

Figure 6: Distribution of adoption rate (targeted intervention)

It does not exhibit the three classes of results, but instead 74% mass is concentrated in the highest decile. On the way from the random to the targeted in the parameter subspace, there seems to be a tipping point (Jackson, 2008; Miller and Page, 2009) leading to the phase transition. The difference is significant: 28 percentage points higher in mean and 10 points lower in standard deviation. While we must take into account the feasibility and costs of locating Goods, the result may have strong policy implication.

Average treatment effect

In stark contrast to adoption rate, average treatment effect is distributed very nicely. It seems that the conditions for the central limit theorem are somehow restored.

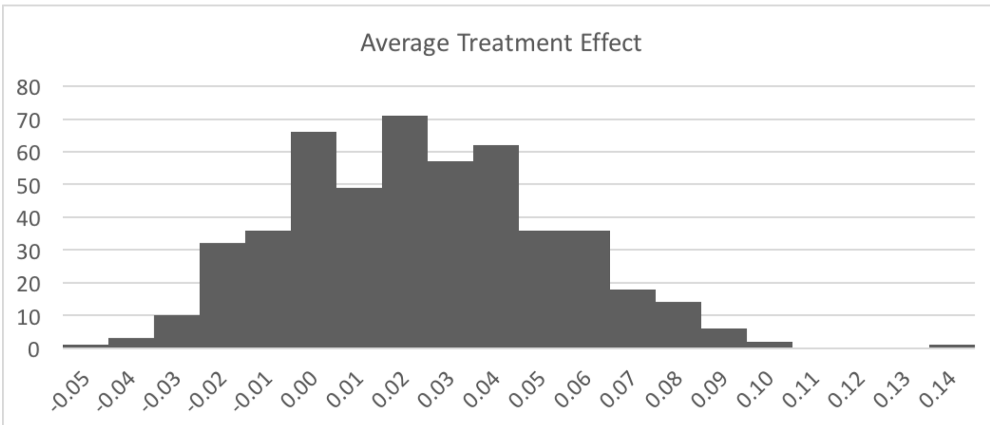

Figure 7: Distribution of average treatment effect among the treated

While it is difficult to isolate particular causes (which is the whole point of using an ABM), we should try to obtain some intuition. Average treatment effect on the treated, as shown on the left-hand side of Eq.1, means that the same set of individuals are placed in the identical social process twice: one for treated and the other for non-treated. Then, we difference two dependent variables within each individual in generating each sample point. It seems that, due to the identical social process, the differencing somehow reduces the complex interdependency.

That said, the very small and even negative effects make a case against RCT as a research approach to evaluating policies designed for situations where the underlying mechanism involves a significant amount of social interaction. As Cartwright (2007) points out, without unrealistically strong assumptions (i.e. homogeneous data-generating process), RCTs are unable to establish external validity. Using an ABM, it is quite simple indeed to create a different social process by changing parameters and get a dramatically different estimate for average treatment effect. Even if the results were applied to almost identical cases and therefore those assumptions are defendable, it would be at best pointless and likely harmful to expect zero or negative effects of public spending.

Conclusion

We have built an agent-based model to provide a analytical tool for promoting technologies that have modest appeal to most individuals. The setting involves a significant degree of social interaction, which makes the decision process complex. Given the 'gold standard' position of RCT in development economics, we have cast specific doubts on the adequacy of RCT for studying complex systems.

Even aside from the issue of external validity, knowing a cause of the average outcome is hardly useful for policy making in most circumstances. This is especially true if the underlying mechanism is not incremental throughout but rather involves a phase transition. In our model, RCT-based average treatment effect is trivial or even negative, leading to 0.48 mean adoption rate with 0.28 standard deviation. In contrast, we can achieve at 74% chance the highest level of adoption rate by locating and targeting the champions, while the overall mean adoption rate is 0.76 with 0.18 standard deviation. Without modeling the underlying complexity, policy evaluation seems almost guaranteed to fail (Deaton, 2010).

Our model is general and provides much flexibility in order to adapt to local specificities. With local data, many parameters can be separately estimated, e.g. the type proportions, α, β, while the others are calibrated to fit the model. Interventions need be neither completely randomly nor completely targeted. Even below the tipping point, partial targeting may be optimal depending on constraints. The decision rules should incorporate more sophisticated theories on human behavior so that policy options can be broadened further. Last but not least, the model must be tested by both observational and experimental data. Thus, instead of directly inferring policy effects, RCTs have many roles to play.

Many social phenomena can be modeled as complex systems, where heterogeneous agents dynamically interact with each other in an evolving environment. In order to analyze such systems, one useful abstraction is to see society as distributed computational devices (Arthur, 2014; Epstein, 2006). In fact, many approaches in the existing literature are compatible with this view (e.g. Bayesian learning and social network analysis). ABM explicitly adopts the perspective, specifies individual properties, and literally compute each single sub-process. It is a bottom-up approach to explaining macroscopic phenomena, i.e. providing microfoundation, which is the predominant philosophy in modern economics. We believe that use of ABM forms a response to Deaton's call, theory-driven experiment, yet conducted in silico.

References

Aldana, U., Foltz, J. D., Barham, B. L., & Useche, P. (2011, January 1). Sequential Adoption of Package Technologies: The Dynamics of Stacked Trait Corn Adoption. *American Journal of Agricultural Economics*, *93*(1), 130–143.

Arthur, B. W. (2014, October 30). Complexity economics: A different framework for economic thought. In *Complexity and the Economy*. Oxford ; New York: Oxford University Press.

Axelrod, R. (1986). An Evolutionary Approach to Norms. *The American Political Science Review*, *80*(4), 1095–1111.

Bandiera, O., & Rasul, I. (2006). Social networks and technology adoption in northern Mozambique. *The Economic Journal*, *116*(514), 869–902.

Banerjee, A., Chandrasekhar, A. G., Duflo, E., & Jackson, M. O. (2013, July 26). The Diffusion of Microfinance. *Science*, *341*(6144), 1236498.

Cartwright, N. (2007, March 1). Are RCTs the Gold Standard? *BioSocieties*, *2*(1), 11–20.

Conley, T. G., & Udry, C. R. (2010, March). Learning about a New Technology: Pineapple in Ghana. *American Economic Review*, *100*(1), 35–69.

Deaton, A. (2010, June). Instruments, Randomization, and Learning about Development. *Journal of Economic Literature*, *48*(2), 424–455.

Duflo, E., Kremer, M., & Robinson, J. (2008, April). How High Are Rates of Return to Fertilizer? Evidence from Field Experiments in Kenya. *American Economic Review*, *98*(2), 482–488.

Duflo, E., Kremer, M., & Robinson, J. (2011). Nudging farmers to use fertilizer: Theory and experimental evidence from Kenya. *The American Economic Review*, *101*(6), 2350–2390.
Epstein, J. M. (2001, August 1). Learning to Be Thoughtless: Social Norms and Individual Computation. *Computational Economics*, *18*(1), 9–24.
Epstein, J. M. (2006). Agent-based computational models and generative social science. In *Generative social science: Studies in agent-based computational modeling*. Princeton University Press.
Foster, A. D., & Rosenzweig, M. R. (2010, September 4). Microeconomics of Technology Adoption. *Annual Review of Economics*, *2*(1), 395–424.
Jackson, M. O. (2008). *Social and economic networks*. Princeton university press Princeton.
Kremer, M., & Miguel, E. (2007, January 8). The Illusion of Sustainability. *The Quarterly Journal of Economics*, *122*(3), 1007–1065.
Miller, J. H., & Page, S. E. (2009). *Complex adaptive systems: An introduction to computational models of social life*. Princeton university press.
Schelling, T. C. (1971). Dynamic models of segregation. *Journal of mathematical sociology*, *1*(2), 143–186.
Thaler, R. H., & Sunstein, C. R. (2008). *Nudge: Improving Decisions About Health, Wealth, and Happiness*. Yale University Press.

Appendix

Initialization

900 individuals are randomly placed over the grid space. Their initial neighborhood sizes are uniformly drawn from the set $\{1, 2, 3, 4, 5\}$, whose expected value is 3. This particular set is justified by both the context and heuristics. Since each run starts under the condition where the new technology has not been introduced, i.e. there is the established norm of using the existing one. So, it makes sense to begin with low alertness. In addition, heuristically, we learned that no matter what initial values simulation started with, the sample average converged to values between 2 and 4. For robustness, we can use even a lower expected value or for that matter indeed any initial value (e.g. homogeneous one) without changing the qualitative characteristics of the results.

Activation

The model uses discrete time and lets agents behave each period. Specifically, 20% of the population is randomly selected and activated at each period. This is another precaution of avoiding spurious artifacts that could arise if all the agents are systematically activated every time.

Since the indifferent decision may involve a tie, we need a tie-breaking rule. We follow a conservative rule that requires the strict majority if something is to be changed. For example, if the neighbors' choices are split exactly in half, the agent will not change her adoption choice into either alternative. It is also applied to updating the neighborhood size—the threshold must be strictly exceeded.

Randomization

A random seed is generated by the time stamp at which each run is executed. Since NumPy accepts as a seed a non-negative integer between 0 and 4294967295, we truncate each time stamp, containing Hour-Minute-Second-Microsecond, down to the 9 leftmost digits.

```
seed = '{:%H%M%S%f}'.format(datetime.datetime.now())
seed = int(seed[:9])
```

Main codes

The model is built on Mesa, an agent-based modeling framework in Python. The following are the codes responsible for the model's algorithm. All the codes, including the auxiliary ones, are available on our GitHub page (https://github.com/ysaikai/TechAdoption).

[model.py]

```
import datetime
import numpy as np
from mesa import Model
from schedule import RandomSingleActivation
from mesa.space import MultiGrid
from mesa.datacollection import DataCollector
from agents import Person

class TechAdopt(Model):
  types = (-1, 0, 1)
  alpha = 0.1
  beta = 0.1
  proportion_type = (beta, 1-alpha-beta, alpha)
  size_intervention = 0.5
  size_target = 1 # Targeted proportion of Goods
  treated = list() # Agents treated
  controlled = list() # Agents controlled

  def __init__(self, width, height, seed=None, mode=0):
    '''Truncate down to 9 digits as a seed must be between 0 and 4294967295'''
    if seed is None:
      seed = '{:%H%M%S%f}'.format(datetime.datetime.now())
      seed = int(seed[:9])
    np.random.seed(seed)
    self.seed = seed

    self.mode = mode # 0: random, 1: targeted
    self.width = width
    self.height = height
    self.N = width*height
    self.schedule = RandomSingleActivation(self)
    self.grid = MultiGrid(width, height, torus=True)

    '''The exact numbers of each agent'''
    self.size = [int(self.N*p) for p in self.proportion_type]
    self.size[1] = self.N - self.size[0] - self.size[2]

    '''Create agents'''
    aid = -1
    self.type0 = list() # Instances of type 0
    tmp = list()
    for i in range(len(self.types)):
      tmp.extend( [self.types[i]]*self.size[i] )
    np.random.shuffle(tmp)
    for contents, x, y in self.grid.coord_iter():
```

```
      aid += 1
      person = Person(aid, (x,y), self)
      person.type = tmp.pop()
      self.schedule.add(person)
      self.grid.place_agent(person, (x,y))
      if person.type == 0:
        self.type0.append(person)

    '''DataCollector'''
    self.datacollector = DataCollector(
      model_reporters = {
        "AvgRate": lambda m: sum(a.adoption for a in m.schedule.agents) / m.N},
      agent_reporters = {
        "Radius": lambda a: a.radius} )

    self.running = True


  def step(self):
    self.schedule.step()
    '''Intervention'''
    if self.schedule.steps == 1:
      num = int( self.size_intervention*self.N )
      if self.mode == 0:
        self.treated = self.intervene_random(num)
      elif self.mode == 1:
        self.treated = self.intervene_target(num)
      for a in self.treated:
        a.adoption = True
        a.experience = True
      self.controlled = list( set(self.schedule.agents) - set(self.treated) )

    self.datacollector.collect(self)


  def intervene_random(self, num):
    treated = np.random.choice(self.schedule.agents, num, replace=False)
    return treated


  def intervene_target(self, num):
    pool = [a for a in self.schedule.agents if a.type == 1]
    num_Goods = int(self.size[2]*self.size_target)
    treated = list(np.random.choice(pool, num_Goods, replace=False))
    diff = num - len(treated)
    if diff <= 0:
      treated = np.random.choice(treated, num, replace=False)
    else:
      pool = list(set(self.schedule.agents) - set(treated))
      # pool = self.type0
      treated.extend( np.random.choice(pool, diff, replace=False) )
    return treated


def compute_ATT(model):
  treated = model.treated
  controlled = model.controlled
  avg_treat = np.mean([a.adoption for a in treated])
  avg_control = np.mean([a.adoption for a in controlled])
```

```
  return avg_treat - avg_control
```

[agents.py]

```
import numpy as np
from mesa import Agent

class Person(Agent):
  threshold = 0.05
  ub_radius = 5
  noise = 0.05

  def __init__(self, aid, pos, model):
    super().__init__(aid, model)
    self.pos = pos
    self.type = 0
    self.radius = np.random.randint(1, self.ub_radius+1)
    self.adoption = False
    self.experience = False


  def step(self):
    if self.experience==False or self.type==0:
      if (np.random.rand() < self.noise):
        self.adoption = not self.adoption
      else:
        self.radius = self.update_radius()
        rate = self.get_adoption_rate(self.radius)
        '''Requiring the strict majority'''
        if rate != 0.5:
          self.adoption = (rate > 0.5)
      if self.adoption:
        self.experience = True
    else:
      if self.type==-1:
        self.adoption = False


  def update_radius(self):
    r = self.radius
    rate = self.get_adoption_rate(r)
    rate_plus = self.get_adoption_rate(r+1)
    '''Requiring being strictly greater than the threshold'''
    if abs(rate_plus - rate) > self.threshold:
      r += 1
    elif r >= 2:
      rate_minus = self.get_adoption_rate(r-1)
      if abs(rate_minus - rate) < self.threshold:
        r -= 1
    return r


  def get_adoption_rate(self, r):
    neighbors = self.model.grid.get_neighbors(self.pos, moore=True, radius=r)
    rate = np.mean([a.adoption for a in neighbors])
    return rate
```