SOCIAL SCIENCES

How much can we influence the rate of innovation?

T. M. A. Fink[1]* and M. Reeves[2]

Innovation is how organizations drive technological change, but the rate of innovation can vary considerably from one technological domain to another. To understand why some domains flourish more rapidly than others, we studied a model of innovation in which products are built out of components. We derived a conservation law for the average size of the product space as more components are acquired and tested our insights using historical data from language, gastronomy, mixed drinks, and technology. We find that the innovation rate is partly influenceable and partly predetermined, similar to how traits are partly set by nurture and partly set by nature. The predetermined aspect is fixed solely by the distribution of the complexity of products in each domain. Different distributions can produce markedly different innovation rates. This helps explain why some domains show faster innovation than others, despite similar efforts to accelerate them. Our insights also give a quantitative perspective on lean methodology, frugal innovation, and mechanisms to encourage tinkering.

INTRODUCTION

Innovation drives technological change (*1*–*4*), but the rate of innovation can vary markedly from one technology to another (*5*). For example, Moore's law implies an increase in computational speed of 40% per year, and the cost of photovoltaic modules has decreased 10% per year, but the price of coal has remained roughly constant (*6*). While some research has been done on the origin and propagation of innovations (*7*–*11*), it remains unclear what causes some domains to progress more rapidly than others (*12*, *13*). Is the pace of technological change set by human intervention, or does each particular domain have its own intrinsic rate?

On the one hand, modeling and the analysis of interventions suggest that different innovation rates are man-made: the result of good or bad strategic choices (*14*, *15*). Models of economic complexity (*16*–*21*) indicate that countries can influence the range and quality of products they produce by the capabilities they invest in. Component models of innovation (*22*–*25*) imply that firms can affect the space of products they can make by their choice of building blocks. At the level of individual products, lean methodology (*26*) aims to shorten product development cycles by iterative learning, and frugal innovation (*27*) makes technologies more accessible by simplifying them.

On the other hand, long-term historical data suggest that different innovation rates are intrinsic: the result of inherent properties specific to each domain. An analysis of record-breaking innovations implies that different domains have persistent and forecastable behavior (*28*). A study of 53 technologies from the Santa Fe Institute's Performance Curve Database and elsewhere suggests that the technologies follow a generalized version of Moore's law, but at different rates that depend on the technology (*6*).

In prior work (*22*, *23*), we studied how the innovation rate depends on the order of components adopted. Here, we study the other side of the coin: the extent to which the innovation rate is intrinsic to each particular domain. While the choices that organizations make play an important role in determining their success, we find that this is countered by an intrinsic innovation rate specific to each domain. These opposing forces are reminiscent of nurture versus nature in human traits.

Here, we do three things. First, we study data from four domains: language, gastronomy, mixed drinks, and technology. In each domain, we measure how the number of makeable products (words, recipes, cocktails, and software products) grows as we acquire new components (letters, ingredients, beverages, and development tools). We do this for an arbitrary order of component acquisition and the average over all possible component orderings. Second, we prove a conservation law for how innovation occurs through time: The average size of the product space times a complexity discount is constant over every stage of the innovation process. We use this law to forecast the size of the product space in the future based on the complexity of the products we can make now. Third, we show that the growth of the average product space depends only on the distribution of product complexity and not on details about which components make up which products. Front-loaded complexity distributions—those that have a lot of simple products, the average product complexity being equal—have much higher innovation rates. We apply our insights to lean methodology, frugal innovation, and tinkering.

RESULTS

Lego game

Let us illustrate the problem with Lego bricks. Consider two different Lego sets: a Star Wars set and a castle set. The Star Wars set can be used to make a variety of spaceship toys. All of these toys are equally complex, with the same number of bricks in each. The castle set, on the other hand, can be used to make some simple toys, such as Lego men with swords, and some complex toys, such as a knight's castle made from walls, windows, ramps, and other parts. In both sets, the average number of bricks per toy is the same. In the Star Wars set, all of the toys are moderately complex, whereas in the castle set, some toys are simple, many are moderate, and a few are complex.

Now, imagine that Carol is playing with the Star Wars set and that her friend Dan is playing with the castle set. Both have the same goal: to make as many different toys as possible. In the morning, Carol patiently collects wings, thrusters, and guns but is not able to actually complete any toys. Dan, on the other hand, makes many simple toys early on. By lunchtime, things are not much better for Carol. She can make a few toys, but Dan has further outpaced her. Only as the day ends and both players acquire all of their bricks does Carol's luck

[1]London Institute for Mathematical Sciences, 35a South Street, Mayfair, London W1K 2XF, UK. [2]BCG Henderson Institute, The Boston Consulting Group, New York, NY, USA.
*Corresponding author. Email: tmafink@gmail.com

change, and she finally catches up with Dan. Dan was able to make more and more toys throughout the day, whereas Carol could hardly make any until the end of the day. Dan enjoyed playing at every stage, while for Carol, play seemed like work because she saw little return on her efforts until the end. As we shall see, the contrast in their performances does not reflect the players themselves but rather is inherent to the Lego sets they used.

Components and products

In the same way that Lego toys are made up of distinct bricks, we take products to be made up of distinct components (*2*, *3*, *29*–*33*), that is, "a combination of components to some purpose" (*3*). A component can be a material object, such as a touch screen, or a skill, such as coding in Java, or a routine, such as a client survey. Once we have access to a component, we do not have to worry about running out; there are no capacity constraints. Any subset of our components can be combined, but a combination either is or is not a product, according to some universal recipe book of products. Suppose further that there are a total of N possible components in "God's own cupboard" but that, at any given stage n, we only have in our basket n of these N possible building blocks. At every stage, we pick a new component to add to our basket, increasing n by 1.

Product space

The products we can make depend on the components we have in our basket. For example, from the letters a, b, c, and d, we can make 10 English words: a, ad, add, baa, bad, cab, cad, dab, dad, and dB. Adding e to the basket increases the number of words we can make to 28; however, some letters are better than others in expanding the word space. If we add f to the letters a to e, then the number of words we can make goes from 28 to 46, but if we add l instead of f, then the number jumps from 28 to 82. The order in which we acquire components plays an important role in how fast the space of products grows. In prior work (*22*, *23*), we developed strategies for optimally choosing the order of components. Here, we show that each domain is predisposed toward its own intrinsic innovation rate.

Size of the product space

To study how the number of makeable products grows as we acquire components, we gathered data from four domains: language, gastronomy (*34*, *35*), mixed drinks, and technology (see Methods). We then did the following experiment for each domain. Starting with an empty basket, we added to it, one component at a time, all of the N possible components. After adding each one, we measured the number of products $p(\boldsymbol{n})$ that we could make, where $\boldsymbol{n}$ is the set of n components in our basket. We differentiate between a specific set of components $\boldsymbol{n}$ and the number of components n to highlight the dependence of $p(\boldsymbol{n})$ on the particular basket of components and not just its magnitude. For example, more words can be made from the first five letters of the alphabet than the last five letters. The size of the product space is shown in Fig. 1 (points), where we acquired components in alphabetical order. Acquiring them in a different order would give a different rate of growth.

Average size of the product space

To sidestep this dependence of the size of the product space on the order of component acquisition, we need to take the average over all possible orders in which to acquire components. But it is not possible to do this numerically even for moderate values of n, since the number of orders grows as $n!$. To overcome this bottleneck, we derived a mathematical formula that analytically gives the exact average (see Methods). Using this technique, we can compute the growth of the average size of the product space, $\bar{p}(n)$, also shown in Fig. 1 (lines). Whereas p depends on the specific set of components $\boldsymbol{n}$, $\bar{p}$ depends only on the number of components n.

Complexity of products

To understand what determines the growth of the average size of the product space, let us take a look at product complexity. The complexity c of a product is the number of distinct components it is made of. Multiple occurrences of a component count once, so that the word "banana" has $c = 3$ letters, not 6. To be able to make a product of complexity c, we need to have in our basket all c of its components. We denote the number of makeable products of complexity c by $p(\boldsymbol{n}, c)$, so that summing $p(\boldsymbol{n}, c)$ over c gives $p(\boldsymbol{n})$. For example, of the 10 words we can make from the letters a, b, c, and d listed above, 1 word contains $c = 1$ different kinds of components, 5 words contain $c = 2$, and 4 words contain $c = 3$.

Conservation law for products

We discovered that there is a mathematical structure underlying how the average size of the product space grows over time, which we prove in Methods. It takes the form of a conservation law: $\bar{p}(n, c)/\binom{n}{c}$ is constant over all stages of the innovation process, where $\binom{n}{c}$ is the binomial coefficient. In other words, for two stages n and n',

$$\bar{p}(n', c)/\binom{n'}{c} = \bar{p}(n, c)/\binom{n}{c}. \quad (1)$$

When n and n' are much greater than c, we can approximate $\binom{n}{c}$ and $\binom{n'}{c}$ by n^c and n'^c, and we find $\bar{p}(n', c) \simeq \bar{p}(n, c)(n'/n)^c$. Let us try to understand this intuitively. It says that the number of makeable products at current stage n is not a good estimate of the number of makeable products at future stage n'. The current number discounts the future number by the factor $(n/n')^c$. The farther into the future we look, the greater the distortion, but not all products are discounted in the same way: Products with higher complexity c are discounted exponentially more. We call the factor $(n/n')^c$ the complexity discount to highlight this exponential dependence on the complexity. To correct for this discount, we must amplify the current number of products by its inverse, $(n'/n)^c$, to obtain the correct estimate for the future.

Forecasting growth

We can use Eq. 1 to forecast the size of the product space in the future from information we have about the present. Summing $\bar{p}(n', c) \simeq \bar{p}(n, c)(n'/n)^c$ over complexity c, with $x = n'/n$, and noting that the size of the product space is an unbiased estimate of its mean, we find

$$p(\boldsymbol{n}') \simeq p(\boldsymbol{n}, 1)x + p(\boldsymbol{n}, 2)x^2 + p(\boldsymbol{n}, 3)x^3 + \ldots \quad (2)$$

Equation 2 has the form of a polynomial in $x = n'/n$, where $x = 1$ is the present time and $x > 1$ is some time in the future. The coefficient in front of x^c is simply the number of products that we can make at current stage n that have complexity c.

For example, imagine that, in language, we only have access to the first two-thirds of the alphabet, that is, we have in our basket the letters a to q. From these, we can make 2800 words. Using Eq. 2, we can forecast the number of makeable words when we have all

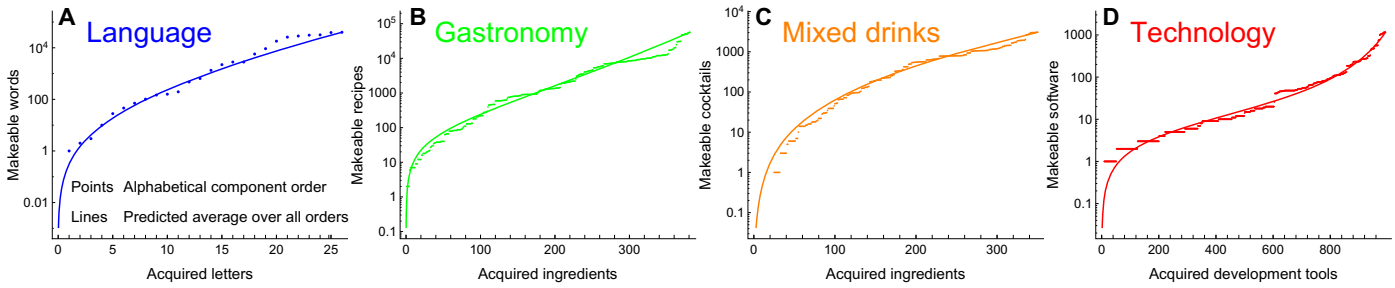

Fig. 1. We studied products and the components used to make them from four domains. (**A**) In language, the products are 39,915 English words and the components are the 26 letters. (**B**) In gastronomy, the products are 56,498 recipes and the components are 381 ingredients. (**C**) In mixed drinks, the products are 3053 cocktails and the components are 350 beverages. (**D**) In technology, the products are 1158 software products and the components are 993 development tools used to make them. For each domain, we show the number of products we can make when we acquire components in alphabetical order (points) and the average number of products we can make over all possible orders in which to acquire the components (lines).

26 letters, without knowing anything about what the new letters will be. Evaluating Eq. 2 at $x = 26/17$, we predict 29,809 makeable words, which, in log terms, is within 2.8% of the actual number. For gastronomy, mixed drinks, and technology, with the first two-thirds of the components (arranged in alphabetical order) in our basket, we predict the size of the product space to within 2.7, 1.4, and 0.4% of the actual number. The further ahead we forecast, of course, the less accurate our prediction becomes.

Specific complexity distributions

If we assume a specific distribution of product complexity, then we can calculate $\bar{p}(n')$ explicitly. We evaluate three common distributions, all with the same mean complexity $\bar{c}$: constant, binomial-distributed, and Poisson-distributed product complexity (Fig. 2, D to F). Products with constant complexity are like the toys in the Star Wars Lego set: They all have a similar number of components. Products with Poisson complexity are like the toys in the castle Lego set: Some are simple, many are moderate, and a few are complex. We find (see Methods) that the number of products we can make at stage n' can be expressed just in terms of $x = n'/n$ and the mean complexity $\bar{c}$,

$$\bar{p}(n') \simeq p(n) \cdot \begin{cases} x^{\bar{c}} & \text{constant complexity,} \\ \left(\frac{1+x}{2}\right)^{2\bar{c}} & \text{binomial complexity,} \\ e^{(x-1)\bar{c}} & \text{Poisson complexity.} \end{cases} \quad (3)$$

These three growth rates are plotted for $\bar{c} = 8$ in Fig. 2 (A to C). They are markedly different: Poisson complexity yields much faster innovation than binomial complexity, which, in turn, yields much faster innovation than constant.

To test our prediction that different distributions of product complexity lead to markedly different innovation rates, we did the following experiment. We gathered together all of the 56,498 gastronomy recipes and glued them together end to end to form a giant list of ingredients. We then cut this giant list into pieces with sizes different from before to make new recipes. This is similar to how we might tear up a long sentence, disregarding spaces, to make new imaginary words. We did this three times, choosing the sizes of the pieces to have one of the three distributions: constant, binomial, and Poisson. All three distributions have the same mean recipe complexity $\bar{c} = 8$. This experiment, further described in Methods, preserves the frequency of the different components in the original recipes. We compare the results (points) with Eq. 3 in Fig. 2 (A to C). It confirms our prediction that the average innovation rate is set by the complexity distribution of the products.

Particular versus average innovation rates

To understand the relationship between a particular innovation rate and the average innovation rate, let us look at component usefulness. The usefulness of a component is the number of products it appears in. The usefulness of different components varies a lot within a given domain. For instance, in the English language, e is used in 26,015 words, whereas j is only used in 540 words. In gastronomy, egg is used in 20,951 recipes; angelica is only in 1. In technology, Google Analytics is in 749 software products; GoDaddy is in 1. In Fig. 3 (E to H), we show the rank-frequency distributions for all of the components in each domain. Different domains have qualitatively different distributions.

The size of the product space for a particular basket of components depends on the usefulness of acquired components: Adding egg to our basket will boost the number of products we can make a lot more than adding angelica. On the other hand, we know from Eq. 2 that the average size of the product space is independent of component usefulness. The result is that the distribution of component usefulness affects the size of fluctuations around the average value—how jumpy the curve is—but only the distribution of product complexity affects the average innovation rate itself.

DISCUSSION

Nature versus nurture

In any given domain, the innovation rate depends partly on the components we choose to acquire and partly on properties of the domain itself. In other words, the innovation rate is partly influenceable and partly predetermined. This is similar to how traits are partly set by nurture and partly set by nature. For example, running fast depends on both training and genetics. In previous work (*22*, *23*), we studied the "nurture" aspect of innovation: how we can influence the innovation rate by strategically choosing the right components. Here, we studied the "nature" aspect of innovation: how each domain is fundamentally predisposed toward its own intrinsic innovation rate. This intrinsic rate—the average over all possible component orderings—is set solely by the distribution of product complexity, and it can vary markedly from domain to domain. The rates for specific component orderings fluctuate around the intrinsic rate.

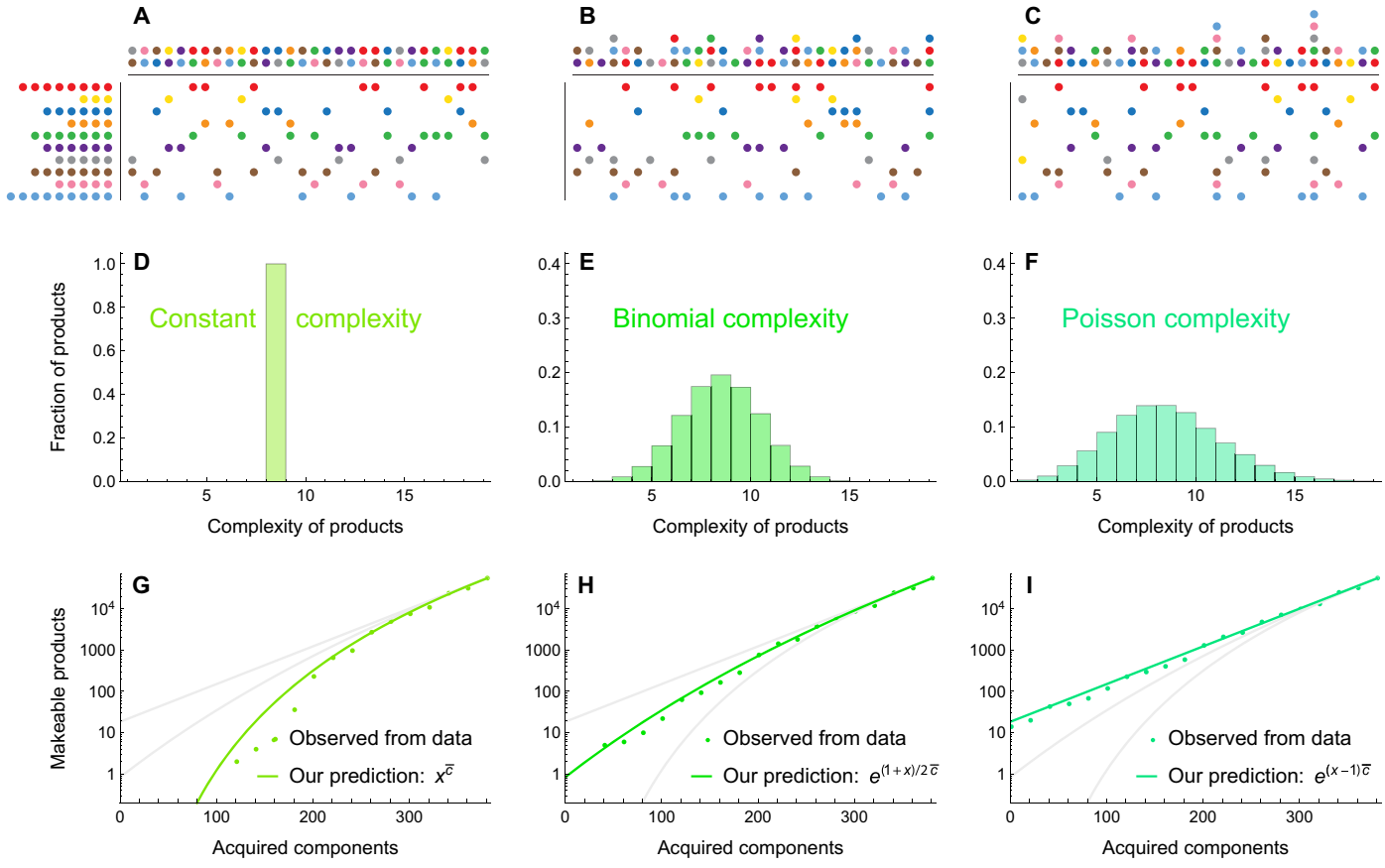

Fig. 2. We compared the innovation rates for different distributions of product complexity. (**A** to **C**) The same set of components can be used to make different sets of products. We show toy examples of 10 components used to make 30 products with (A) constant complexity, (B) binomial complexity, and (C) Poisson complexity. (**D** to **I**) To test our prediction that the average innovation rate depends only on the distribution of product complexity, we glued together all 56,498 gastronomy recipes end to end to form a giant strip of 464,405 components. We then cut these into pieces with different lengths to make new recipes. We chose the lengths to be constant (D), binomially distributed (E), and Poisson-distributed (F), all with mean length $\bar{c} = 8$. When we added components to our basket in a random order and measured the size of the product space (points) (G to I), we found that it follows our prediction in Eq. 3 (lines).

Complexity discount

When we can make some set of products $p(\boldsymbol{n})$ at stage n, we might think naively that these represent an unbiased draw across all possible products in the universal recipe book. In fact, the draw is not at all unbiased, but is strongly weighted toward simpler products, as we are much more likely to have hit upon the components in them. For example, if a child knows only half the letters in the alphabet, then he is a lot more likely to be capable of writing "banana," $\binom{n}{c}/\binom{n'}{c} = \binom{13}{3}/\binom{26}{3} = 11\%$, than "orange," $\binom{13}{6}/\binom{26}{6} = 0.7\%$: "Banana," made of three distinct components, is a simpler word than "orange," made of six. His vocabulary, far from falling uniformly over all words, is strongly weighted toward simpler words. In other words, the chance of drawing complex products is discounted, by the factor $\binom{n}{c}/\binom{n'}{c} \simeq (n/n')^c$, which grows exponentially with complexity c.

On forecasting

Conservation laws allow us to make predictions. The conservation law that we derived governs the average size of the product space. However, we cannot measure the average size of the space, just the size of a particular instance of the space. For the size of the product space to be an unbiased estimate of its mean, the sequence of new components at each stage must be independent and identically distributed. In practice, however, the distribution for new components can vary over time. When this happens, we can only make meaningful forecasts over time scales that are short compared to this variation. This is analogous to how selection pressures in evolution are meaningful as long as the environmental change is slow compared to the reproduction time scale.

The benefit of being front-loaded

Our conservation law for the average innovation rate provides a surprising insight into how innovation occurs through time. Even when the mean product complexity $\bar{c}$ is the same across different domains, a domain with a front-loaded distribution of complexity yields much faster innovation. A front-loaded distribution, such as Poisson, has many simple products, whereas a distribution that is not front-loaded, such as constant complexity, has no simple products. A binomial distribution lies between these two. The more front-loaded the distribution, the faster the innovation rate tends to be.

A small difference in the fraction of simple products makes a big difference to the innovation rate, for two reasons. First, simpler products are exponentially more likely to be makeable. Second, the product space is exponentially smaller early on, so these simpler products make up a large fraction of the space. In our Lego example, Dan's castle set is front-loaded, whereas Carol's Star Wars set is not; it is this difference that led to their contrasting performances.

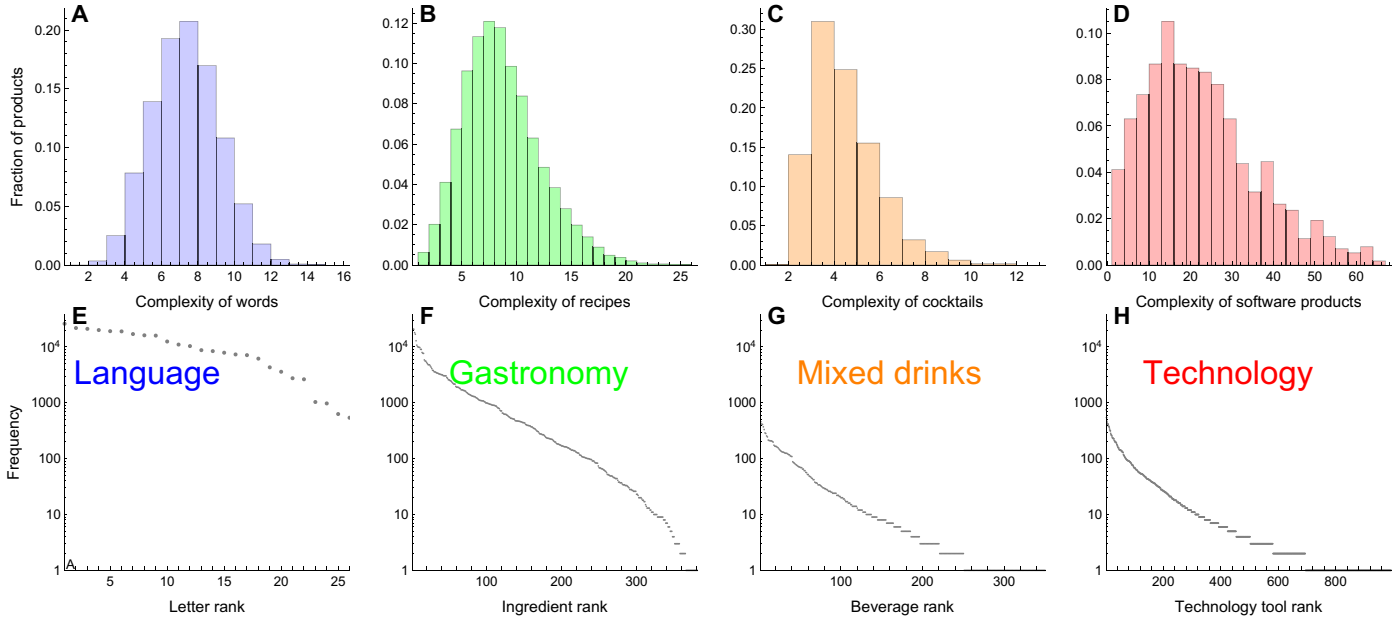

Fig. 3. We computed the distribution of product complexity and the rank frequency of components for our four domains. (**A** to **D**) The distribution of product complexity is shown for each of our four domains. (**E** to **H**) The number of products that each component appears in, where the components are in rank order from most to least frequent, is shown for each of our four domains.

When to go lean

Lean methodology (*26*) accelerates the search for product-market fit by quickly bringing a simple product to market. With early access to user feedback, this minimum viable product can be rapidly adapted to form the basis of a feasible business plan. Lean methodology has been practiced by software start-ups, government agencies, and health care corporations (*36*), but is a lean approach equally suited to all innovation domains?

Our work suggests that the scope for applying lean methodology depends on how front-loaded the distribution of product complexity is. In domains that are not front-loaded, there will be a scarcity of products that can be made at the start of the innovation process. Such domains are best suited to firms with the resources to weather sustained investment with little return early on. On the other hand, a front-loaded distribution of products will enable the rapid expansion of the product space straightaway. Resource-poor start-ups and developing communities are more likely to thrive in such domains. Many organizations are confronted with a choice about which domain to enter, and anticipating these differences ahead of time can help them choose the right one.

Frugal innovation

Frugal (*27*, *37*) and reverse innovation (*38*, *39*) make a technology more widely available by reducing the number of necessary components. In doing so, the modified technology will only approximate the original technology, but this is outweighed by the significant boost in reach. Our model of innovation offers a quantitative explanation of this phenomenon. Using Eq. 1, we see that the probability of being able to make a given product with an arbitrary basket of components decreases exponentially with the complexity of the product. A small reduction in product complexity increases many times over the probability that it can be made. This increase is stronger for developing communities with fewer resources (components n) than it is for advanced communities with many resources. In this way, a greater fraction of developing communities will gain access to the simplified technology than advanced ones.

How to encourage tinkering

Tinkering is improving something in an experimental manner. It tends to be process-driven rather than goal-driven; the journey is the reward rather than the end result. Tinkering is important because it can make innovation feel like play instead of work. Our model provides a basis for promoting tinkering in domains that can be reverse-engineered to have different product complexity distributions.

Consider, for example, software. In drawing programs, spreadsheets, and word processors, commands can be combined in different ways to perform tasks. Think of the commands as the components in our model, and the tasks as the products in our model. The user's innovation rate is the growth in size of his task space as he learns new commands. Learning a new command requires effort, and the user's return on that effort is the number of new tasks he can perform. A user remains motivated when his return exceeds his effort at each stage of the learning process; otherwise, he is liable to give up. Software and mobile app designers can reverse-engineer an optimal user experience journey by building a Poisson distribution of task complexity, so that the number of tasks that can be performed rises steadily as new commands are learned.

A familiar example of tinkering is building with toy constructions sets, such as Lego, Meccano, and Zometool. Just like the ability to perform more tasks compensates for having to learn a command, the thrill of making new inventions motivates a tinkerer to select new bricks and explore new combinations. To encourage tinkering, the distribution of product complexity should be front-loaded so that new products can be made throughout the innovation process, not just at the end. This makes innovating feel more like play, as was the case with Dan, and less like work, as was the case with Carol.

METHODS

Data

Our four datasets were obtained as follows. In language, our list of common English words is from the built-in WordList library in Mathematica 10.4. Of the 40,127 words in WordList, we only considered the 39,919 made from the 26 letters a to z, ignoring case: We excluded words containing a hyphen, space, and so on. In mixed drinks, the 3053 cocktails were curated by us from the website www.thecocktaildb.com. In gastronomy, the 56,498 recipes can be found in the supplementary materials in (*34*). In technology, the 1158 software products and the development tools used to make them can be found at www.stackshare.io.

Proof or product invariant

A product of complexity c contains c distinct components. Let $\boldsymbol{N}$ be the set of N possible components, $\boldsymbol{n}$ be our basket of n components chosen from $\boldsymbol{N}$, and $\boldsymbol{c}$ be some combination of c components selected from our basket $\boldsymbol{n}$. The number of products of complexity c that we can make from our basket can be found by considering all possible combinations and adding up the number that are products

$$p(\boldsymbol{n}, c) = \sum_{\boldsymbol{c}\subseteq\boldsymbol{n}} \mathrm{prod}(\boldsymbol{c}),$$

where $\mathrm{prod}(\boldsymbol{c})$ takes the value 0 if the combination of components $\boldsymbol{c}$ forms no product and 1 if it forms one product. [Occasionally, the same combination of components $\boldsymbol{c}$ forms multiple products: for example, beef, butter, and onion form two distinct recipes of complexity 3. In such cases, $\mathrm{prod}(\boldsymbol{c})$ takes the value 2 if $\boldsymbol{c}$ forms two products of complexity c, and so on.] The average number of products we can make, $\bar{p}(n, c)$, is the average of $p(\boldsymbol{n}, c)$ over all subsets $\boldsymbol{n} \subseteq \boldsymbol{N}$; there are $\binom{N}{n}$ such subsets. Therefore

$$\begin{aligned}\bar{p}(n, c) &= 1/\binom{N}{n} \sum_{\boldsymbol{n}\subseteq\boldsymbol{N}} p(\boldsymbol{n}, c) \\ &= 1/\binom{N}{n} \sum_{\boldsymbol{n}\subseteq\boldsymbol{N}} \sum_{\boldsymbol{c}\subseteq\boldsymbol{n}} \mathrm{prod}(\boldsymbol{c}).\end{aligned} \tag{4}$$

Consider some particular combination of components $\boldsymbol{c}'$. The double sum above will count $\boldsymbol{c}'$ once if $c = n$ but multiple times if $c < n$, because $\boldsymbol{c}'$ will belong to multiple sets $\boldsymbol{n}$. How many? In any set $\boldsymbol{n}$ that contains $\boldsymbol{c}'$, there are $n - c$ free elements to choose, from $N - c$ other components. Therefore, Eq. 4 will count every combination $\boldsymbol{c}$ a total of $\binom{N-c}{n-c}$ times, and

$$\begin{aligned}\bar{p}(n, c) &= 1/\binom{N}{n}\binom{N-c}{n-c} \sum_{\boldsymbol{c}\subseteq\boldsymbol{N}} \mathrm{prod}(\boldsymbol{c}) \\ &= \binom{n}{c}/\binom{N}{c}\bar{p}(\boldsymbol{N}, c).\end{aligned}$$

The same must be true when we replace n by n'. Solving both equations for $\bar{p}(\boldsymbol{N}, c)$ and equating them, we find

$$\bar{p}(n', c)/\binom{n'}{c} = \bar{p}(n, c)/\binom{n}{c}.$$

Summing over c, we find

$$\bar{p}(n') = \sum_c \bar{p}(n, c)\binom{n'}{c}/\binom{n}{c}.$$

When the number of components is big compared to the product size ($n, n' \gg c$), using Stirling's approximation, we can approximate $\binom{n}{c}$ and $\binom{n'}{c}$ by n^c and n'^c, and thus

$$\bar{p}(n', c)/n'^c \simeq \bar{p}(n, c)/n^c.$$

Again, summing over c, we find

$$\bar{p}(n') \simeq \sum_c \bar{p}(n, c)(n'/n)^c. \tag{5}$$

Since the size of the product space is an unbiased estimate of its mean,

$$p(\boldsymbol{n}') \simeq \sum_c p(\boldsymbol{n}, c)(n'/n)^c.$$

Specific complexity distributions

When we assume a specific distribution for the complexity of products, we can explicitly write the relation between the future average size of the product space and the current average size. For a Poisson distribution of product complexity with mean $\bar{c}$, the average number of products with complexity c is

$$\bar{p}(n, c) = \bar{p}(n)\bar{c}^c e^{-\bar{c}}/c!.$$

Substituting this into Eq. 5, with $x = n'/n$, yields

$$\begin{aligned}\bar{p}(n') &\simeq \sum_{c=0}^{\infty} \bar{p}(n)\bar{c}^c e^{-\bar{c}}/c!(n'/n)^c \\ &= \bar{p}(n)e^{-\bar{c}} \sum_{c=0}^{\infty} (\bar{c}n'/n)^c/c! \\ &= \bar{p}(n)e^{(x-1)\bar{c}}.\end{aligned}$$

Similarly, for a binomial distribution of product complexity with event probability $^1/_2$, the average number of products with complexity c is

$$\bar{p}(n, c) = \bar{p}(n)\binom{2\bar{c}}{c}\left(\frac{1}{2}\right)^{2\bar{c}}.$$

Substituting this into Eq. 5, with $x = n'/n$, yields

$$\begin{aligned}\bar{p}(n') &\simeq \sum_{c=0}^{2\bar{c}} \bar{p}(n)\binom{2\bar{c}}{c}\left(\frac{1}{2}\right)^{2\bar{c}}(n'/n)^c \\ &= \bar{p}(n)\left(\frac{1}{2}\right)^{2\bar{c}} \sum_{c=0}^{2\bar{c}} \binom{2\bar{c}}{c}(n'/n)^c \\ &= \bar{p}(n)\left(\frac{1+x}{2}\right)^{2\bar{c}}.\end{aligned}$$

Glue and break

Our glue and break experiment in Fig. 2 (D to I) was done as follows. First, we glued together all 56,498 of the gastronomy recipes back to front to form one long strip of 464,405 components. Second, we sampled 55,000 numbers from each of three different distributions

to use as the new recipe lengths (that is, the number of ingredients in each recipe). In distribution one, the complexity was fixed at 8 and the numbers were 8, 8, 8, 8,…. In distribution two, the complexity was binomially distributed with mean 8 and event probability $^1/_2$ and the numbers were 7, 8, 6, 6, 11,…. In distribution three, the complexity was Poisson-distributed with mean 8 and the numbers were 5, 12, 20, 9, 5,…. Third, we broke up the giant strip of ingredients three different times, according to the three sequences of integers. This produced in each instance a new set of 55,000 recipes, all with a mean complexity of 8. (For comparison, the mean complexity of the actual recipes was 8.22.) We note that some of the new recipes might contain the same ingredient twice, since we did not take measures to ensure against this. Because we only count distinct components in a product, the complexity of some recipes will be less than the total number of ingredients in each. However, this effect is negligible and has virtually no effect on the histograms in Fig. 2 (D to F).

REFERENCES AND NOTES

1. D. H. Erwin, D. C. Krakauer, Insights into innovation. *Science* **304**, 1117–1119 (2004).
2. W. B. Arthur, The structure of invention. *Res. Policy* **36**, 274–287 (2007).
3. W. B. Arthur, *The Nature of Technology: What It Is and How It Evolves* (Penguin, 2010).
4. W. Janeway, *Doing Capitalism in the Innovation Economy* (Cambridge Univ. Press, 2018).
5. J. McNerney, J. D. Farmer, S. Redner, J. E. Trancik, Role of design complexity in technology improvement. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 9008–9013 (2011).
6. J. D. Farmer, F. Lafond, How predictable is technological progress? *Res. Policy* **45**, 647–665 (2016).
7. V. Sood, M. Mathieu, A. Shreim, P. Grassberger, M. Paczuski, Interacting branching process as a simple model of innovation. *Phys. Rev. Lett.* **105**, 178701 (2010).
8. I. Iacopini, S. Milojević, V. Latora, Network dynamics of innovation processes. *Phys. Rev. Lett.* **120**, 048301 (2018).
9. F. Tria, V. Loreto, V. D. P. Servedio, S. H. Strogatz, The dynamics of correlated novelties. *Sci. Rep.* **4**, 5890 (2014).
10. V. Loreto, V. D. P. Servedio, S. H. Strogatz, F. Tria, Dynamics on expanding spaces: Modeling the emergence of novelties, in *Creativity and Universality in Language* (Springer, 2016).
11. D. Rotolo, D. Hicks, B. R. Martin, What is an emerging technology? *Res. Policy* **44**, 1827–1843 (2015).
12. F. Lafond, A. G. Bailey, J. D. Bakker, D. Rebois, R. Zadourian, P. McSharry, J. D. Farmer, How well do experience curves predict technological progress? *Technol. Forecast. Soc. Change* **128**, 104–117 (2018).
13. T. Felin, S. Kauffman, R. Koppl, G. Longo, Economic opportunity and evolution: Beyond landscapes and bounded rationality. *Strateg. Entrep. J.* **8**, 269–282 (2014).
14. M. Reeves, K. Haanaes, J. Sinha, *Your Strategy Needs a Strategy* (Harvard Business Review Press, 2015).
15. P. Ghemawat, Evolving ideas about business strategy. *Bus. Hist. Rev.* **90**, 727–749 (2016).
16. C. A. Hidalgo, B. Klinger, A.-L. Barabási, R. Hausmann, The product space conditions the development of nations. *Science* **317**, 482–487 (2007).
17. C. Hidalgo, R. Hausmann, The buildings blocks of economic complexity. *Proc. Natl. Acad. Sci. U.S.A.* **106**, 10570–10575 (2009).
18. R. Hausmann, C. A. Hidalgo, The network structure of economic output. *J. Econ. Growth* **16**, 309–342 (2011).
19. A. Tacchella, M. Cristelli, G. Caldarelli, A. Gabrielli, L. Pietronero, A new metrics for countries' fitness and products' complexity. *Sci. Rep.* **2**, 723 (2012).
20. M. Cristelli, A. Gabrielli, A. Tacchella, G. Caldarelli, L. Pietronero, Measuring the intangibles: A metrics for the economic complexity of countries and products. *PLOS ONE* **8**, e70726 (2013).
21. R. Van Noorden, Physicists make 'weather forecasts' for economies. *Nature* **1038**, 16963 (2015).
22. T. M. A. Fink, M. Reeves, R. Palma, R. S. Farr, Serendipity and strategy in rapid innovation. *Nat. Commun.* **8**, 2002 (2017).
23. M. Reeves, T. Fink, R. Palma, J. Harnoss, Harnessing the secret structure of innovation. *MIT Sloan Manag. Rev.* **37**, 59 (2017).
24. T. Fink, P. Ghemawat, M. Reeves, Searching for great strategies. *Strategy Science* **2**, 272–281 (2017).
25. C. A. Hidalgo, From useless to keystone. *Nat. Phys.* **14**, 9–10 (2018).
26. E. Ries, *The Lean Startup* (Portfolio Penguin, 2011).
27. J. Prabhu, Frugal innovation: Doing more with less for more. *Philos Trans. A Math. Phys. Eng. Sci.* **375**, 20160372 (2017).
28. B. Nagy, J. D. Farmer, J. E. Trancik, J. P. Gonzales, Superexponential long-term trends in information technology. *Technol. Forecast. Soc. Change* **78**, 1356–1364 (2011).
29. A. Tacchella, "Economic complexity," thesis, Universit Roma Sapienza, 2014.
30. J. Schumpeter, *Business Cycles* (McGraw-Hill Book Company, 1939).
31. H. Youn, D. Strumsky, L. M. A. Bettencourt, J. P. Lobo, Invention as a combinatorial process: Evidence from US patents. *J. Roy. Soc. Interface* **44**, 0272 (2015).
32. T. E. Stuart, J. M. Podolny, Local search and the evolution of technological capabilities. *Strateg. Manage. J.* **17**, 21–38 (1996).
33. L. Fleming, Recombinant uncertainty in technological search. *Management Sci.* **47**, 117 (2001).
34. Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási, Flavor network and the principles of food pairing. *Sci. Rep.* **1**, 196 (2011).
35. A. M. Petruzzelli, T. Savino, Search, recombination, and innovation: Lessons from haute cuisine. *Long Range Plann.* **47**, 224–238 (2014).
36. S. Blank, Why the Lean start-up changes everything. *Harvard Bus. Rev.* **91**, 224–238 (2013).
37. V.-T. Tran, P. Ravaud, Frugal innovation in medicine for low resource settings. *BMC Med.* **14**, 102 (2016).
38. V. Govindarajan, R. Ramamurti, Reverse innovation, emerging markets, and global strategy. *Glob. Strateg. J.* **1**, 191–205 (2011).
39. C. K. Prahalad, R. A. Mashelkar, Innovation's Holy Grail. *Harvard Bus. Rev.* **88**, 132 (2010).

Acknowledgments

Funding: We acknowledge support from the European Commission FP7 grant 611272 "Growth and innovation policy modelling (GROWTHCOM)" and the BCG Henderson Institute. **Author contributions:** The methodology was planned by T.M.A.F. and M.R. Mathematical derivations were done by T.M.A.F. Data collection and analysis were done by T.M.A.F. Figure images were made by T.M.A.F. The manuscript was written by T.M.A.F. and M.R. Implications of the results were conceived by T.M.A.F. and M.R. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or at www.lims.ac.uk/innovation/data/. Additional data related to this paper may be requested from the authors.

Submitted 16 May 2018
Accepted 29 November 2018
Published 9 January 2019
10.1126/sciadv.aat6107

Citation: T. M. A. Fink, M. Reeves, How much can we influence the rate of innovation? *Sci. Adv.* **5**, eaat6107 (2019).

Science Advances

How much can we influence the rate of innovation?

T. M. A. Fink and M. Reeves

Sci Adv **5** (1), eaat6107.
DOI: 10.1126/sciadv.aat6107