Utrecht University

Economic diversification as a percolation process

Author
Alje van Dam

Supervisors
prof. dr. ir. Jason Frank*
prof. dr. Koen Frenken†

*Mathematical Institute, Utrecht University
†Copernicus Institute for Sustainable Development, Utrecht University

Master Thesis
in
Mathematical Sciences

October 27, 2016

Abstract

In this thesis, we will use network theory to model some of the mechanisms that influence the economic development of countries. In the Economic Complexity framework of Hausmann & Hidalgo [10], the production of any product requires a combination of specific capabilities, representing different skills and other non-tradable inputs. The relatedness between products in terms of their capability requirements can then be inferred from country-product export data [12], [2].
This leads to a network representation of the economy called the product space [13]. In this network, products are connected if they are produced by the same (type of) countries, suggesting they require similar capabilities for their production. It has been shown empirically that countries tend to develop products that are 'nearby' in the product space, building upon capabilities that are already present. This leads to the interesting perspective of viewing economical development as a diffusion process on a complex network.
Here we take a theoretical approach and instead of starting from the data, try to capture the mechanisms that govern this diffusion process in a simple theoretical model that describes how the accumulation of capabilities can lead to economic growth. We construct a network model of the product space as the one-mode projection of a randomly generated bipartite country-capability network that indicates the capability requirements of every product. We then describe economic development as a percolation process on this product network. We also incorporate in the model the effects of knowledge spillovers, in which exchange of capabilities between industries enables economic growth. The model can explain the sudden 'take-off' of some countries' economies and stagnant development of others, and provides insight in how the distribution of capabilities can affect economic growth.

Contents

1. Introduction

Despite great technological development enabling increased mobility and communication in the past century, the world's economies remain divided into poor and rich. Some countries like the Asian tigers have managed to make the transition from the poor to rich within a few decades by going through a phase of explosive economic growth and industrialization. Other countries lag behind and remain underdeveloped. What separates the rich, poor and developing countries and what drives this economic growth? Why do some countries' economies appear stagnant, while others experience explosive growth?
Classic economic theory uses productive factors such as capital an labour and technology to explain economic growth, and most policy is reliant on aggregate variables such as GDP or the level of education in a country. These theories however cannot explain what drives the divergence of countries' incomes and development. Recently, a new framework has been proposed that takes a new perspective and aims to elucidate the mechanisms of economic growth using non-aggregate data.

1.1. Economic Complexity

This *Economic Complexity* framework [2], [4], [10], [11], [12], [13], [21], [23] states that the economic development of countries is determined, or at least constrained, by the specific skills and knowhow that are present in a country rather than capital, labor and technology as in the classical theories of economic growth. Instead of using classical economic theory to explain economic growth, economic complexity uses a data-driven approach to reveal drivers of economic growth. In this approach, the products produced by a country serve as an indicator for the knowledge present in that country. Measures of economic complexity have been developed that aim to infer the knowledge present in a country and the knowledge needed for production of specific products from countries' export data.
The main subject of study in economic complexity is the country-product matrix M_{cp}, which is constructed using international trade data. The M_{cp} matrix tells whether a country c exports a certain product p with revealed comparative advantage (RCA) or not:

$$M_{cp} = \begin{cases} 1 & \text{if product } p \text{ is exported with RCA} \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

One way to interpret this data is by considering M_{cp} to define a bipartite network connecting countries and the products they produce. This leads to a network with

two classes of nodes, namely countries and products, where there is an edge between a country c and a product p if and only if $M_{cp} = 1$.
Two quantities of interest are product ubiquity k_p and country diversity k_c, which are given by the degree in the bipartite network or written in terms of the M_{cp} matrix as

$$k_p = \sum_c M_{cp}$$
$$k_c = \sum_p M_{cp}.$$

It has been shown that there is an inverse relation between these two quantities, i.e. highly diversified countries (countries that export many different products) tend to export on average more exclusive products that are less ubiquitous, arguably because these products require the combination of a greater variety of different inputs for production and hence a more developed economy. Less developed countries are seen to produce only ubiquitous products that are easier to produce, such as raw materials and agricultural products [12].
The key insight is that developed countries seem to be diversified rather than specialized, meaning that the countries that produce highly sophisticated products tend to produce a wide variety of products, including simple products that are also produced by less developed countries [10], [12]. In other words, it seems that countries produce all the products they *can* produce, given the productive inputs that are available in a country. This means that the country-product network has a nested structure, and the M_{cp} matrix has a triangular structure when sorted according to country diversity.
The relation between product ubiquity and country diversity suggests that the M_{cp} matrix holds information on the level of development of a countries' economy, and the level of sophistication of products. This has lead to development of measures of economic complexity, leading to the concepts of country fitness F_c and product complexity Q_p [4].

1.2. Measures of economic complexity

The fitness-complexity algorithm computes from the M_{cp} matrix metrics of countries' fitness (an estimate for the level of development of a countries' economy) and products' complexities (an estimate for the level of sophistication of a product) [4], [12]. The iterative scheme is given by

$$\tilde{F}_c^{(n+1)} = \sum_p M_{cp} Q_p^{(n)}, \qquad F_c^{(n+1)} = \frac{\tilde{F}_c^{(n+1)}}{\sum_c \tilde{F}_c^{(n+1)}}$$
$$\tilde{Q}_p^{(n+1)} = \frac{1}{\sum_c M_{cp} \frac{1}{F_c^{(n)}}}, \qquad Q_p^{(n+1)} = \frac{\tilde{Q}_p^{(n+1)}}{\sum_p \tilde{Q}_p^{(n+1)}}$$

Intuitively, a countries' fitness is determined by the mean complexity of the products it exports. This causes the fitness of a country to grow as it either diversifies into

production of more products and even more so if these products are more complex.
The complexity of a product is inversely proportional to a measure of how 'easy' it is to produce a product: a product is thought to be easily produced if many countries produce it, where countries with low fitness have a larger contribution. If a product is produced only by a few high-fitness countries, it will have high complexity. On the other hand, if a product is also produced by many countries with low fitness, this implies low complexity.
Each time step, both quantities are normalized. Starting from some initial condition the fitness-complexity algorithm converges to a measure of economic complexity of a country F_c and a measure for the complexity of a product Q_p. These measures produce rankings of the state of countries' economies, where the economies with highest complexities are the highly industrialized and diversified countries [4]. The fitness measure for the complexity of a countries' economy has been shown to correlate with GDP, and is even thought to be predictive of future growth: countries with high fitness but low GDP have been shown to be among the fastest growing economies [5].

1.3. The Product Space

The structure of the M_{cp} matrix also holds information about the relatedness between products in terms of their input requirements. If two products often co-occur in the export basket of a country, this suggest they require roughly the same inputs in terms of specific skills or technical knowledge needed, other products or resources involved in the value chain or quality of the institutions needed for production. Without specifying *how* product are related, co-occurrences indicate similarity of products in one or more of the above mentioned dimensions. This leads to a measure of proximity between products, which tells how close products are in terms of their input requirements.
Formally, the proximity $\phi(p, p')$ between two products p and p' is given by the minimum of the pairwise conditional probability that a country produces p or p' given it also produces the other product [13]:

$$\begin{aligned}\phi(p,p') &= \min\{(P(M_{cp}=1|M_{cp'}=1), P(M_{cp'}=1|M_{cp}=1)\} \\ &= \min\{\frac{\sum_c M_{cp}M_{cp'}}{\sum_c M_{cp'}}, \frac{\sum_c M_{cp}M_{cp'}}{\sum_c M_{cp}}\}\end{aligned}$$

One can use the proximity measure to construct a network of products in which products with high proximity are connected by edges, called the product space [13]. Products that are adjacent in the product space (i.e. have close proximity) are thus thought to require approximately the same inputs, meaning that production of one product could naturally lead to the other, since it indicates the presence of most required inputs in a country for the adjacent product.

Figure 1.1 shows a network representation of the product space as given in [13]. The network shows a complex structure, with a dense core of products that are related to many other products, and a more sparse periphery with less interrelated products such as oil, raw materials and agricultural products. Furthermore the network shows distinct

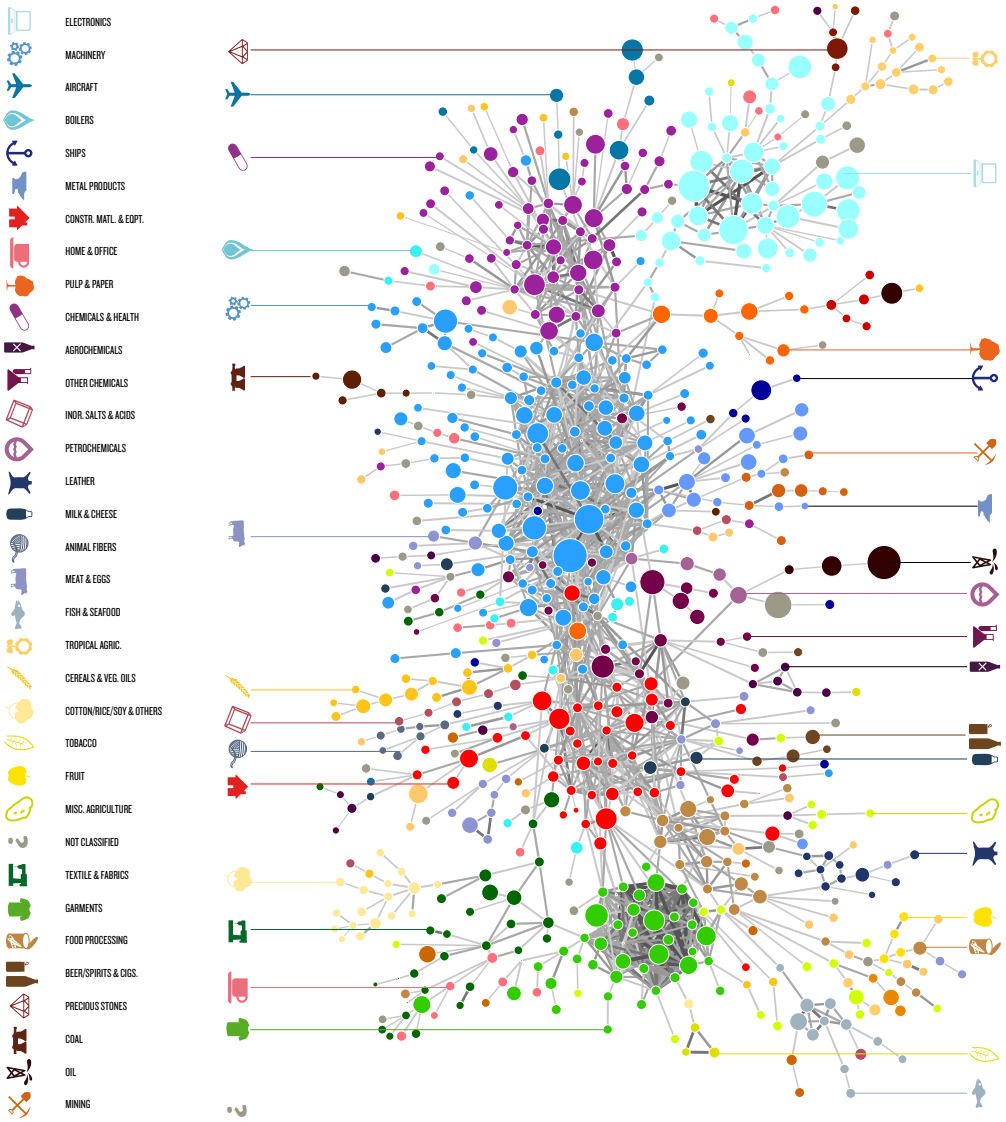


Figure 1.1.: A network representation of the product space by [11]. Nodes are colored according to product classification and node sizes are proportional to world trade.

clusters of related products that largely coincide with the used product classification (node color). For example, the bottom shows a clear garments cluster and on the top right one can find the electronics industry.
In [23], a variant of the product space was developed, called the taxonomy network. Using a different measure of product relatedness, the taxonomy network is a directed network which incorporates a notion of causality in the network, so that one product leads to the other in a specific order.
The product space and the taxonomy space reveal the pathways of economic development of countries in the sense that when countries diversify, they do so in related

industries. This way, the network shows which products a country is likely to diversify into, given its current position in the network. This process of related diversification has also been shown to operate at the level of regions looking at regional entry in new products [15].
In the framework of the product space and the taxonomy space, one can think of economic growth as a country 'spreading through' the product space, where development of adjacent industries may in turn lead to production of products that are adjacent to the newly developed ones. The position of a country in the product space shows the current productive structure of a country, and reveals opportunities for future growth. Countries that produce products that are in relatively isolated, sparse parts of the product space have little opportunity to diversify, whereas production of products in dense parts of the network, with many neighbors, give many opportunities for diversification [13]. This way the product space can also function as a policy tool to see which industries are feasible for a country to develop given their current position in the network [11].

1.4. Contribution

The methods discussed above are purely empirical: data on product exports is used to infer information about how products are related in terms of their input requirements and which of these are available in different countries. These methods have been proven a useful tool in policy making and economic analysis, and give insight in the development of countries' economies.
Less work has been done however on developing appropriate theoretical models that describe economic growth of individual countries within this framework. This thesis will aim for a theoretical approach and instead of starting from the data, try to capture economic development of individual countries as a diffusion process on the product space in a theoretical model. In particular, we are interested in the implications of seeing economic development as a diffusion process on a network.
We propose to model economic development as a percolation process on a complex network. Percolation was first introduced in Physics as a model to describe the flow of a liquid through a porous material. It has been applied in more settings, such as the modeling of forest fires and infectious diseases, but recently also in the modeling of diffusion of information through social networks [20] and the adoption of new technologies [24]. Here we propose to model economic development as a percolation process on the product space.
Depending on the structure of the network and the initial conditions, diffusion on a network may stop and remain restricted to a small fraction of the network, or propagate and lead to wide-spread diffusion. These two regimes are separated by a phase transition. The aim of this thesis is to use percolation theory to describe dynamics observed in the data, such as the stagnant growth of underdeveloped countries and the sudden take-off of countries that go through periods of rapid economic development [5]. This way we hope to capture some of the implications of economic growth as a network phenomenon

in a simple theoretical model, which will be extended to fit the economic complexity framework.
The rest of this thesis is structured as follows. Chapter 2 will describe the basic percolation model and shortly discuss its implications when applied to the setting of economic development. This chapter also contains a derivation of the percolation threshold and percolating cluster sizes for the case of random graphs. In Chapter 3 we will construct a network model of the product space starting from the theory of capabilities as introduced in [12]. We describe properties of the constructed network and derive the degree distributions and connectivity properties of the model product space. Chapters 4 and 5 discuss the percolation properties of the model product space for different initial conditions. Chapter 6 introduces a model in which economic growth is driven endogenously by individual industries acquiring new capabilities, and explores the the effect of knowledge spillovers within this framework. We use this model to investigate the time evolution of economic diversification. We conclude with a discussion.

2. Percolation

In this chapter we will study the dynamics of the basic percolation model on random networks with a given degree distribution. Firstly, the model is formalized and notation introduced. The mathematical methods to describe the basic properties of the model are shortly introduced. In Section 2.4 results for the case of infinite random graphs with given degree distributions are derived.

2.1. Economic diversification as a percolation process

In the model, a country is given a global parameter v that stands for the general level of development of its economy. One can think of this parameter as describing the general conditions in a country that are needed for the production of increasingly sophisticated products, such as the general level of education and quality of institutions.
The economic state of a country is represented by its position in a network that resembles the product space. The nodes in the network represent products, that are connected by edges that indicate that two products are related in terms of their required conditions for production. If a country produces a certain product, we say the node representing that product in the network is in an active state.
Furthermore, every product p in the network is assigned a certain level of sophistication x_p, which resembles the 'difficulty' of production of that particular product. Simple products like raw materials and agricultural products will have low x_p as they do not require a high general level of development. More sophisticated products like electronics and machinery however would require a higher basic level of coordination in the form of institution and knowledge for example, and would therefore have higher x_p. A country can only be active in production of a product if the general conditions have been met, i.e. $v \geq x_p$. If this condition has been met, a product will be activated when a related (neighboring) product is already being produced, indicating the presence of industry-specific knowledge. This models the mechanism of related diversification.
One can then ask the question how the number of products a country can activate in the network given a number of initial seed products depends on the level of development v. From percolation theory we know this is a highly nonlinear relationship, marked by a sharp increase in diffusion size when v passes some critical threshold value v_c. These properties depend strongly on the network structure and properties.

2.2. Model setup

Let us model the above mentioned product space as an unweighted, undirected graph consisting of $p = 1, \ldots, N_p$ vertices. The degree $d(p)$ of a randomly sampled vertex p is given by the degree distribution $P(k) = P(d(p) = k)$. Vertices also get assigned a parameter $x_p \in [0, 1]$ which represents the difficulty of an industry. The x_p are sampled from a probability distribution $f(y) = P(x_p = y)$, with $\int_0^1 f(y)\mathrm{d}y = 1$. The global parameter v determines which vertices are *operational*, meaning that they are accessible given the level of development of a country. A vertex p is operational if $x_p \leq v$. The operational network is the network that is left by only considering operational vertices. Every vertex can be in two states - *active* or *inactive*. In every time step, active vertices activate all their inactive neighbors in the operational network. Hence a vertex gets activated in time step t under two conditions:

- the vertex is in the operational network ($x_p \leq v$)
- at least one neighbor is active at time t.

Suppose that we start with one active vertex p, which we call the seed node. Every time step all active nodes activate all their operational neighbors. The next time step these neighbors will in turn activate all their operational neighbors, eventually causing all operational vertices with an (indirect) connection to the seed node to be active at the end of the process.
This way a country will eventually develop all industries that are in the connected component of the operational network of an industry in which it was initially active, and the diffusion size is determined by the connectedness of the operational network. The probability for a randomly sampled vertex p to be in the operational network is given by

$$P(\text{vertex } p \text{ operational}) = P(x_p \leq v) = F(v),$$

where $F(v) = \int_0^v f(y)\mathrm{d}y$. This can also be interpreted as the fraction of vertices in the original network that are operational. The number of vertices N_v in the operational network is thus given by $N_v = F(v)N_p$. The parameter v determines the network's capability to spread the activation of nodes. For low values of v, the operational network will consist of multiple finite connected components, since nodes with $x_p > v$ fragment the operational network. As v increases, more nodes become operational and larger connected components will form. For some critical value v_c, a large enough fraction of the nodes is operational, causing all connected components in the operational network to 'merge' into a giant connected component, that allows for wide-spread diffusion through the network.
In the following sections we will determine analytically under which conditions the formation of a giant connected component takes place, and how this depends on the global parameter v in the case of infinite random graphs with given degree distribution. For real, finite networks, the percolation properties highly depend on the specific structure of the network.

2.3. Generating functions and the tree ansatz

In order to study percolation on random networks, we will make some extensive use of *generating functions* [17]. This Section provides a short introduction, following [17]. A generating function is a function that contains all information about the probability distributions we are dealing with, and provides a practical way of representing and deriving them. Some aspects are particularly useful, as we will discuss below.
Firstly, we introduce the probability generating function for the degree distribution $P(k)$, which gives the probability that a randomly chosen vertex has degree k. We define the generating function for the degree distribution as

$$G(x) = \sum_{k=0}^{\infty} P(k)x^k.$$

Since we assume that $P(k)$ is normalized to unity, we have that

$$G(1) = \sum_{k=0}^{\infty} P(k) = 1.$$

This definition allows extraction of information about de probability distribution $P(k)$ through simple manipulations of $G(x)$:

- The probabilities $P(k)$ are given through the kth derivative of $G_0(x)$ in $x = 0$, divided by $k!$, since

$$\left[\frac{\mathrm{d}^k G(x)}{\mathrm{d}x^k}\right]_{x=0} = k! \cdot P(k).$$

- The nth moment of the distribution $P(k)$ is given by

$$\left[\left(x\frac{\mathrm{d}}{\mathrm{d}x}\right)^n G(x)\right]_{x=1} = \sum_{k=0}^{\infty} k^n P(k).$$

- In particular, the expected degree $\langle k\rangle$ for a random vertex i is given by

$$\left[x\frac{\mathrm{d}G(x)}{\mathrm{d}x}\right]_{x=1} = \sum_{k=0}^{\infty} kP(k).$$

- The probability that m independently randomly sampled vertices have total degree k is given by

$$\frac{1}{k!}\left[\frac{\mathrm{d}^k}{\mathrm{d}x^k}(G(x)^m)\right]_{x=0}.$$

The last property follows from the fact that the mth power of $G(x)$ gives the probability generating function of the total degree of m randomly sampled vertices being k. For example, we have

$$\begin{aligned} G(x)^2 = \left[\sum_{k=0}^{\infty} P(k)x^k\right]^2 &= \sum_{j,l=0}^{\infty} p_l p_j x^{l+j} \\ &= p_0p_0 + (p_0p_1 + p_1p_0)x + (p_0p_2 + p_1p_1 + p_2p_0)x^3 \\ &\quad + (p_0p_3 + p_1p_2 + p_2p_1 + p_3p_0)x^3 + \ldots, \end{aligned}$$

from which we can see that this gives exactly the right probabilities for every $j + l = k$. We can use generating function to derive results on the structure of the network for given v under certain assumptions, called the tree ansatz [6]. Firstly, we assume that the only prescribed property of the network is its degree distribution. Hence we assume no degree-degree correlations or other structure. Second, we assume that the network does not contain finite loops, i.e. starting from a random vertex, one cannot return to that vertex through a finite number of edges. In particular, this means that we assume the network has no clustering. This has been shown to hold true for infinite sparse random networks without degree-degree correlations. In this case, we say that the network is *locally tree-like* [6].

2.4. Cluster sizes and the percolation threshold

This section follows the derivations in [3] and [17] for the distributions of finite cluster sizes, the percolation threshold v_c and the size of the percolating cluster S_v. Firstly, we define the generating functions of some quantities of interest. Recall that $P(k)$ gives the probability that a randomly chosen vertex p has degree k, and $F(v) = P(x_p \leq v)$ is the probability that a vertex is operational. Then $P(k)F(v)$ is the probability of a random vertex having degree k and being operational, and the probability generating function for this distribution is given by

$$G_0(x) = \sum_{k=0}^{\infty} P(k)F(v)x^k. \tag{2.1}$$

The overall fraction of operational vertices P_v is then given by $P_v = G_0(1) = \sum_k P(k)F(v) = F(v)$ since the x_p are distributed independent of degree k. The mean degree of an operational vertex $\langle k_v \rangle$ can also be retrieved by

$$\langle k_v \rangle = G_0'(1) = \sum_{k=0}^{\infty} kP(k)F(v).$$

In order to study diffusion, we are interested in the number connections a nearest neighbor of a random vertex has. This is given by the degree distribution of a vertex at the

end of a random edge. Following a random edge, one will more likely encounter a vertex with high degree than one with low degree. In fact, the probability of the end of an edge having degree k is proportional to $kP(k)$ [6]. Hence, after normalization the generating function for the probability of a random neighbor having degree k and being operational is given by

$$\frac{\sum_k kP(k)F(v)x^k}{\sum_k kP(k)}.$$

Since we are interested in the capability of neighbors to activate other vertices, we are interested in the number of outgoing edges, excluding the edge along which we arrived. Thus by subtracting one edge from the degree, the probability $Q(k)$ of having *remaining degree* k is given by [16]

$$Q(k) = \frac{(k+1)P(k+1)F(v)}{\sum_k kP(k)}. \tag{2.2}$$

We define $G_1(x)$ to be the probability generating function of the remaining degree

$$\begin{aligned} G_1(x) = \sum_{k=0}^{\infty} Q(k)x^k &= \frac{\sum_{k=0}^{\infty}(k+1)P(k+1)F(v)x^k}{\sum_k kP(k)} \\ &= \frac{\sum_{k=1}^{\infty} kP(k)F(v)x^{k-1}}{\sum_k kP(k)} = \frac{G_0'(x)}{\langle k\rangle}. \end{aligned} \tag{2.3}$$

Hence $G_1(x)$ gives the probability that a vertex on the end of a random edge is operational and has remaining degree k. G_0 gives the probability that a random vertex is operational and has degree k. Note that $G_1(x) \geq G_0(x)$ in general. These two generating functions allow us to determine how diffusion on the network will take place.

2.4.1. Subcritical regime

We will start by deriving the distribution for *finite size clusters* in the operational network. We define a finite cluster as a finite set of vertices that are all (indirectly) connected to each other.

Define $H_1(x)$ to be the generating function for the distribution of the sizes of components that can be reached by following the end of a randomly chosen edge. If all operational clusters in the graph are finite, we have $H_1(1) = 1$. If the end of the sampled edge is not operational, the cluster size is zero. This happens with probability $1 - G_1(1)$ since the probability of the end of an edge being operational is given by $G_1(1)$. The edge will lead to an operational vertex with k outgoing edges with probability $Q(k)$ given by (2.2). All operational vertices at the end of these k edges are part of the same cluster of size s. Hence the probability generating function of them being part of a cluster of size s is also given by $H_1(x)$. This leads to the relation [17]

$$H_1(x) = 1 - G_1(1) + xQ(1)H_1(x) + xQ(2)[H_1(x)]^2 + xQ(3)[H_1(x)]^3 \ldots,$$

where the first two terms account for the probability of a cluster of size 0 (not operational), and the power in the $Q(k)H_1(x)^k$ terms represent the probability generating functions for the probability that the sum of k ends of edges are part of a cluster of size s and the operational vertex has k remaining edges. Note that we can write this relation as

$$H_1(x) = 1 - G_1(1) + xG_1[H_1(x)]. \tag{2.4}$$

Doing the same but for a random vertex as opposed to the end of a random edge, we obtain the probability generating function for the size of an operational cluster of a random vertex:

$$H_0(x) = 1 - G_0(1) + xG_0[H_1(x)]. \tag{2.5}$$

Hence the mean operational cluster size $\langle s \rangle$ is given by

$$\langle s \rangle = H_0'(1) = G_0[H_1(1)] + G_0'[H_1(1)]H_1'(1) \tag{2.6}$$

$$= G_0(1) + G_0'(1)\frac{G_1(1)}{1 - G_1'(1)} = G_0(1) + \frac{G_0'(1)^2}{\langle k \rangle - G_0''(1)}, \tag{2.7}$$

where $\langle k \rangle$ is the average degree of all vertices (including non-operational vertices) in the network and we used that $H_1(1) = 1$, i.e. we assumed finite cluster sizes [3].

2.4.2. Percolation threshold

It can be seen that the average cluster size diverges for $G_0''(1) = \langle k \rangle$, or $G_1'(1) = 1$. In other words, the mean cluster size diverges if the expected number of second nearest neighbors of a random vertex in the operational network is greater or equal than 1, meaning every random vertex is expected to activate at least one more vertex that is not its direct neighbor. As this point is approached, finite clusters grow and eventually merge into one giant (infinite) connected component, indicating the percolation threshold. This transition indicates the formation of a giant connected operational network of infinite size. The condition for divergence can also be written as

$$G_0''(1) = \sum_k k(k-1)P(k)F(v) = \langle k \rangle.$$

This means that the percolation threshold is given by v_c such that

$$F(v_c) = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$$

In particular, for an Erdös-Renyi random graph we have $\langle k \rangle^2 = \langle k^2 \rangle - \langle k \rangle$ so the percolation threshold is given by $F(v_c) = 1/\langle k \rangle$ [17].

2.4.3. Supercritical regime

The percolation threshold gives us the the critical value $v = v_c$ for which a giant component forms. For $v > v_c$ we thus know that there exists a giant connected component of operational vertices, through which wide-spread diffusion can occur. The size of this giant component can be measured by the fraction of operational nodes that belong to the giant component, expressed by S_v.
In the derivation above we had that $H_0(1) = 1$ since the probabilities of being in a finite operational cluster summed to 1. Note however that we defined $H_0(x)$ and $H_1(x)$ to generate the probabilities for vertices and respectively ends of edges to belong to a *finite* operational cluster. Therefore vertices belonging to the giant component should be excluded from this quantity, and we have that $H_0(1) = 1 - S_v$ rather than 1. Thus we can express the size of the giant component (using (2.5)) as

$$S_v = 1 - H_0(1) = 1 - (1 - G_0(1) + G_0(H_1^*(1)) = G_0(1) - G_0(H_1^*(1)) = F(v) - G_0(H_1^*(1)),$$

where $H_1^*(1)$ satisfies (from (2.4))

$$H_1^*(1) = 1 - G_1(1) + G_1(H_1^*(1)),$$

which can now have a nontrivial solution. Below the percolation threshold the only solution is $H_1^*(1) = 1$, leading to $S_v = 0$.
Note that since S_v gives the fraction of operational nodes that are in the giant component, this also gives the probability that activating a random operational node leads to wide-spread diffusion in the network.

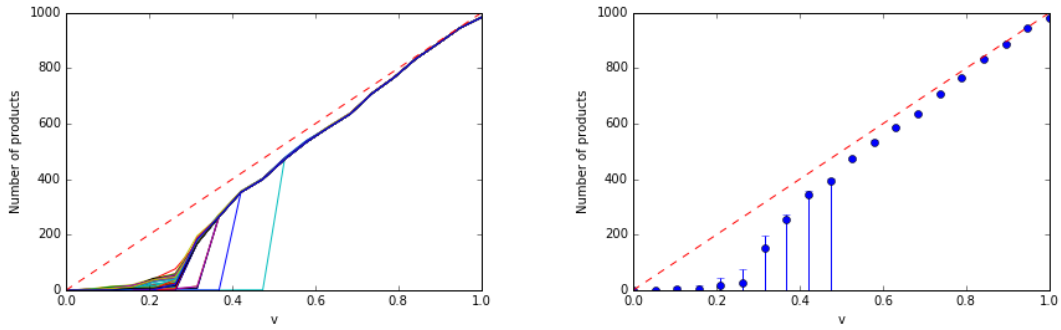
2.5. Percolation on a random graph

Figure 2.1 shows the percolation process in an Erdös-Renyi random graph with $N_p = 1000$ nodes and mean degree $\langle k \rangle = 4$ for different values of v. Here the x_p values are uniformly distributed between 0 and 1, and the diffusion starts with 10 randomly selected seed nodes that are initially active.
The red line denotes the expected number of activated products for a given v for a fully connected network, which equals the number of operational nodes for a given v:

$$\sum_{i=1}^{n} P(x_p < v) = N_p \cdot v.$$

The case of a fully connected network corresponds to a situation in which all products are related to each other and every product in the operational network will be activated. For networks that are not fully connected, operational nodes may remain unactivated if they are not connected to another active product. The operational nodes that are not a part of the same connected component as the seed products are not activated, even though they could be, given the level of development v.

In economic terms, a country in the subcritical regime may increase its level of development v but experience a minimal increase in the products it produces, since many operational nodes (products it is potentially capable of making) are disconnected from currently active industries. This results in a seemingly stagnant economy, with low returns on an increase in v. Figure 2.1 shows that for low values of v diffusion does not spread and the number of active products remains far below the number of operational nodes.
The critical value v_c equals $\frac{1}{\langle k \rangle} = 1/4$ for this particular network (see Section 2.4.2). For this value a phase transition occurs and there is a strong increase in the number of activated products as the operational network forms a giant connected component, enabling wide-spread diffusion. This could explain the sudden 'take-off' of some economies, in which a period of stagnant economic growth in the subcritical regime is followed by a period of rapid diversification and economic growth as v passes the percolation threshold and a giant connected component emerges in the operational network. Hence passing v_c would mean an escape from the poverty trap. This mechanism may also contribute to the divergence of rich and poor countries: underdeveloped, poor countries have limited growth options since their operational network is in a subcritical state. More developed countries have access to a large fraction of the network, and have many opportunities for diversification, enabling economic growth.



(a) 50 individual simulation runs. Most runs enter the supercritical regime around $v = 0.25$, with the exception of some outliers for which the seeds do not hit the giant connected component until v is larger.

(b) Averages over the 50 simulation runs. There is a clear sharp increase in diffusion size around the theoretical threshold value $v_c = .25$. Error bars indicate minimum and maximum values over the 50 simulations.

Figure 2.1.: Diffusion sizes in an ER random graph with mean degree 4, $N_p = 1000$ for different values of v. Diffusion was initiated with 10 random seed nodes.

2.6. Discussion

In the above we have discussed the properties of the basic percolation model and how it can be applied to model the diversification of a country as a diffusion process in the

product space. We have seen that the nonlinear dependence of diffusion size on parameter v can possibly explain some phenomena observed in the economic development of countries, where the development of underdeveloped countries seems to be stagnant, and more developed countries experience a sudden 'take-off' in the number of products they produce. The model shows that we can use the idea of the product space to explain these phenomena as a network effect.
We have modeled the product space as an Erdös-Renyi random graph. The product space however has a highly structured topology, with closely connected dense parts and more sparse parts [13]. We can then ask the question how the structure of the product space influences the diversification process. Different network topologies and network properties like clustering, degree-degree correlations and correlated values of the x_p can have a big effect on the diffusive properties of the network [24], [19], [6].
Furthermore we have drawn the x_p values for products from a uniform distribution. It seems likely that related industries will also be related in how difficult they are to produce, i.e. the x_p values of neighboring products will be correlated. This motivates the construction of a model product space that is based on a micro foundation, in which the network structure and the x_p values are related. This space will be constructed as a projection of the bipartite capability-product matrix, which was introduced as part of the binomial model in [10].

3. The model product space

3.1. The theory of capabilities

One attempt to quantify the knowledge and know-how needed to produce certain products as discussed in the previous chapter is given by the theory of capabilities [12]. Capabilities are defined as non-tradeable inputs that are required for the production of a product. The idea is that in order to engage in any economic activity, one has to have all the required complementary skills and inputs needed for production. In order to produce T-shirts for example, one would need the appropriate machinery and skilled labour, but also access to services like accountants, factory space and infrastructure. Of course, the capabilities needed for production of T-shirts will be very similar to those needed for the production of sweaters. Other more complex product like airplanes may require many more capabilities, including appropriate norms and regulations.
An analogy used to describe this mechanism is that of the game of Scrabble. The capabilities are represented by letters, and different words stand for different products that can be made by combining these letters. A country with many letters (capabilities), is able to combine these letters into many different words of different lengths. A country with few letters however can only make a few short words, representing simple products that require few inputs.
Hence economic growth is driven by the accumulation of capabilities (letters) in a country, and conditioned by the capability requirements of different products. Countries that have access to more capabilities will be able to make not only more different but also more complex products (longer words). Obtaining new capabilities will allow countries to diversify into producing new products that require a combination of the new and old capabilities. Since these products only differ in at most the newly obtained capability, these new products will be similar to the products that where already produced in terms of their input requirements. This way, economic development is explained as a path-dependent process, in which countries develop new products that are close to their current productive structure.
Seeing development of industries as a combinatorial process in which capabilities are recombined implies that there are increasing returns in the number of products that can be produced as extra capabilities are acquired [10]. Countries that already have many capabilities will be able to combine these with the newly acquired capability, enabling the production of many more products. This means more developed countries will diversify easily as acquisition of a new capability will lead to many more growth opportunities. A less developed country however will not have as many existing capabilities to combine the new capability with, therefore only being able to produce a few extra products if any.

Thus countries with few capabilities benefit less from the acquisition of new capabilities, leaving them trapped in a state of underdevelopment. This could explain the (growing) inequality in the development of nations, and the sudden take-off of some economies that reach a point in which acquiring capabilities starts paying off and leading to even more growth opportunities [10].
A theoretical model describing this mechanism is the Binomial model [10]. The model explicitly deals with capabilities and aims to explain the possible mechanisms through which the M_{cp} matrix obtains its structure. The idea is that the observable M_{cp} matrix is a product of two bipartite networks: the product-capability network defined by the matrix T_{ap}, and the country-capability network defined by the matrix C_{ca}.
We can describe the capabilities that are present in every country with the country-capability matrix C_{ca}, setting $C_{ca} = 1$ if capability a is present in country c and $C_{ca} = 0$ if it is not. Likewise, the product-capability matrix T_{ap} dictates which products require which capabilities. Thus we have

$$T_{ap} = \begin{cases} 1 & \text{if product } p \text{ requires capability } a \\ 0 & \text{otherwise.} \end{cases}$$

and

$$C_{ca} = \begin{cases} 1 & \text{if country } c \text{ has capability } a \\ 0 & \text{otherwise.} \end{cases}$$

The binomial model is based on three assumptions:

1. Every product requires a specific combination of capabilities given by the vector T_p
2. Every country has a certain set of capabilities given by the vector C_a
3. A country will produce a product if it has all the required capabilities.

Under the above assumptions, the M_{cp} matrix can be thought of as the outcome of the matrix operation [10], [21]

$$M_{cp} = C_{ca} \odot T_{ap} = \prod_a (1 - T_{ap}(1 - C_{ca})),$$

for which M_{cp} is exactly 1 if country c has all the necessary capabilities to produce product p. The operator $\odot$ is referred to as the Leontief operator, as it resembles a Leontief production function in binary form [10]. It states that the production of a product requires at least all necessary inputs as dictated by the T_{ap} matrix.
Note that the matrices T_{ap} and C_{ca} are not considered observable in practice since there is no clear way of measuring capabilities, and the only data is available on M_{cp} in the form of product exports. Extracting information on T_{ap} and C_{ca} has been the objective of the measures of economic complexity as discussed in Section 1.2 [12], [4].

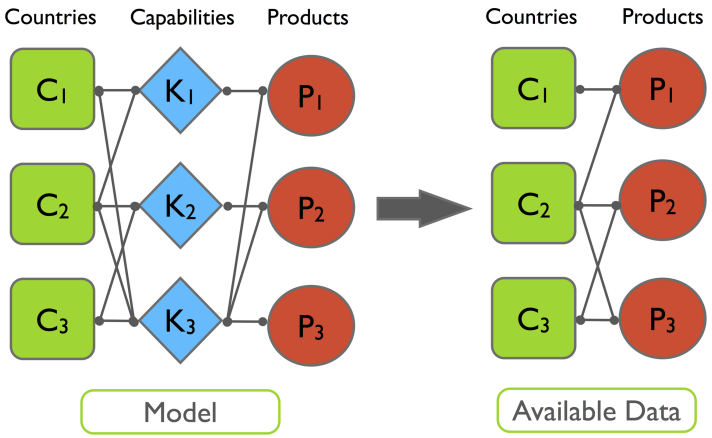


Figure 3.1.: Schematic representation of the bipartite country-product network as the outcome of a tripartite network, in which countries are connected to capabilities and capabilities are connected to products. Figure taken from [21].

Factorizing the M_{cp} matrix as a product of two bipartite networks in which we deal with capabilities explicitly, allows us to study how the T_{ap} network conditions the development of individual countries. Instead of inferring information from the M_{cp} matrix to construct an empirical product space based on the output of different countries, we will assume a fixed structure of the T_{ap} matrix, which we then use to define a theoretical product space. We then study the development of individual countries on this space.
In the following sections we will construct this model product space from the bipartite capability-product network. In particular, we investigate properties of this product space when we model the bipartite capability-product network as a random network, as in previous work [10]. We explore analytically and numerically the in- and out-degree distributions of products conditioned on the complexity of a product and derive the joint degree distribution. In Chapter 4 we will simulate the percolation process on this network and discuss the results.

3.2. The capability-product network and its projection

The model assumes a finite number of products N_p that each require at most N_a capabilities for production. Every product p requires a certain combination of specific capabilities to be produced, given by the binary vector T_p. The capability requirements of all products is given by the bipartite capability-product network which is defined by the N_a by N_p capability-product matrix T_{ap}, for which

$$T_{ap} = \begin{cases} 1 & \text{if product } p \text{ requires capability } a \\ 0 & \text{otherwise.} \end{cases}$$

We think of this network as the recipe book that defines for all products what their input requirements are. Every column T_p represents the capability requirements for a single product. These requirements need not be unique, since there can be multiple (different) products that require the same set of capabilities.
As an example, consider a world in which we only distinguish four different capabilities ($N_a = 4$), and there are 6 possible products ($N_p = 6$). Then one possible form of the T_{ap} matrix is

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{3.1}$$

meaning the product represented in the last column requires the third and fourth capabilities in order to be produced. Furthermore we note that the number of capabilities required by a product p is given $q(p) = \sum_a T_{ap}$, which gives us a measure for product complexity.
The total number of possible combinations of N_a capabilities is given by 2^{N_a}. Hence N_a can be seen as a measure of the complexity of the product space. The quantity $\frac{N_p}{2^{N_a}}$ gives the fraction of combinations of capabilities that leads to a viable product.
Since the bipartite network has a fixed number of edges and nodes, the number of edges coming from nodes in class P must equal the number of edges coming from class A. This means that we have the constraint

$$N_p \cdot \langle q \rangle = N_a \cdot \langle r \rangle,$$

where $\langle q \rangle$ and $\langle r \rangle$ denote the average degree of all products and capabilities respectively.
We construct a product space by projecting this bipartite capability-product network onto a product-product network, putting a link between product p and p' in the product-product network if they share at least one capability. The number of capabilities shared by p and p' is then given by the edge weight

$$W_{pp'} = \sum_a T_{ap} T_{ap'}.$$

The symmetric weight matrix $W = T^T T$ with entries $W_{pp'}$ counts the number of shared neighbors in the original bipartite network, i.e. the number of capability requirements that p and p' share. The diagonal elements $W_{pp} = \sum_a T_{ap} T_{ap} = \sum_a T_{ap} = q(p)$ are the product complexities, and are given by the original degree of nodes from class P in the bipartite network.
The matrix $W_{pp'}$ defines an undirected, weighted network. In the product space however we are interested in the *distance* between products p and p' in terms of the capabilities one would have to learn extra to be able to produce product p' given you can produce product p. We define the distance as

$$\delta(p, p') = q(p') - \sum_a T_{ap} T_{ap'} = W_{p'p'} - W_{pp'},$$

which gives explicitly how many capabilities have to be obtained to go from p to p'. This results in a directed, weighted network where the weights indicate the distances between products.

Lastly, we introduce a threshold parameter α which defines what the maximum distance

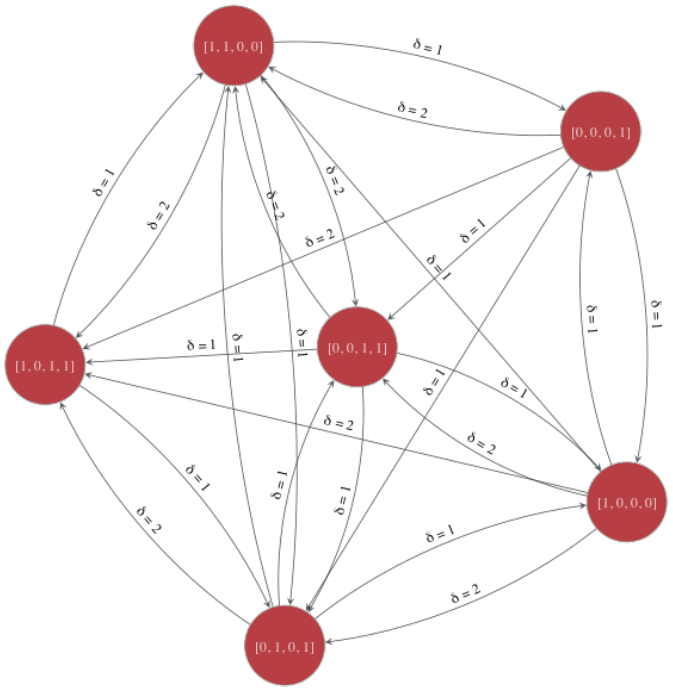


Figure 3.2.: Distance network constructed for example matrix given in (3.1)

is for two products to be connected. The model product space is the unweighted network in which there is a directed edge from p to p' if $\delta(p, p') \leq \alpha$. This way the parameter α defines how many capabilities can be learned in one step, so that a country can move from one product to the other. We denote the thresholded network with $\mathcal{P}(\alpha)$.

Here we are interested in the structure of $\mathcal{P}(\alpha)$ and how it depends on our choices of T_{ap}, N_a and N_p. In particular we are interested its degree distribution and connectivity properties, as these are the main determinants of diffusion properties of the network. In the following sections, we will investigate these properties for the simple case of a random T_{ap} matrix.

3.3. The Binomial case

To construct T_{ap} we sample N_p binary vectors of length N_a that represent the combinations of capabilities that are needed for the production of those products. How these vectors are sampled determines the structure of T_{ap} and hence the structure of the product network.

In the remainder of the thesis we will consider the case where entries of T_{ap} are i.i.d.

Binary random variables with parameter ρ, modeling the maximally random case. The capability-product matrix is then given by

$$T_{ap} = \begin{cases} 1 & \text{with probability } \rho \\ 0 & \text{with probability } 1-\rho \end{cases}$$

with $\rho \in [0,1]$. Parameter ρ equals the probability of an edge being present between a random pair (a,p) in the bipartite network.
This assumption is motivated by the fact that we have very limited information on capability requirements of products in the real world since they are not easily measured or even defined. We simply assume products are combinations of capabilities, without knowing anything about which capabilities are used more often, and how many capabilities are needed for a product. Hence we make no further assumptions and model the capability-product network as a random network. This binomial assumption is also used in related literature in similar ways [10], [9]. Furthermore, taking T_{ap} to be random does not lead to trivial dynamics since we project the network, leading to a nontrivial structure of the model product space. Lastly, the assumption of a binary capability-product matrix allows to describe the system with a single parameter ρ and allows the derivation of analytical results.
Under this assumption, the distribution of product complexities $q(p)$ is given by the distribution of the column sums:

$$P(q(p)=k) = P\left(\sum_a T_{ap} = k\right) = \binom{N_a}{k} \rho^k (1-\rho)^{N_a-k}, \tag{3.2}$$

which is the probability to have k successes out of N_a trials, with success probability ρ. Here a success corresponds to having an entry 1 in one of the N_a possible entries of column T_p, indicating product p requires a capability. Hence (3.2) gives the probability of a product requiring exactly k capabilities, and the product complexities are $\text{Bin}(N_a, \rho)$ distributed. As a consequence, the mean product complexity in the system is given by $N_a\rho$, and each capability is required by on average $N_p\rho$ products.

3.3.1. Deriving distributions

How do the in- and out-degree of a product in the product network relate to its complexity $q(p)$? The in-degree of a product will directly affect the chances of it being activated, since a product with high in-degree will have many possible activators. Likewise, a product with high out-degree has high potential of leading to further development of products. One expects high complexity products to be hard to make, hence having low in-degree. On the other hand, since being able to produce a high complexity product means one has many capabilities, high complexity products are expected to have high out-degree, with many links pointing to lower complexity products that require a subset of their capability requirements.
Denote with $d^{\alpha}_{in}(p)$ and $d^{\alpha}_{out}(p)$ the in- and out degree of a node p for a given threshold

α. The in-degree of a product p' is given by

$$d_{in}^{\alpha}(p') = \sum_{p \neq p'} \mathbf{1}_{\{\delta(p,p') \leq \alpha\}},$$

which simply counts the number of nodes $p \in \mathcal{P}$ that are at distance than less or equal α to p'.
The probability of a product p' with complexity k' having j in-edges is given by

$$P(d_{in}^{\alpha}(p') = j | q(p') = k') = \binom{N_p - 1}{j} \eta_{in}(k, \alpha))^j (1 - \eta_{in}(k, \alpha)^{N_p - 1 - j} \tag{3.3}$$

where

$$\eta_{in}(k', \alpha) = P(\delta(p, p') \leq \alpha | q(p') = k') \tag{3.4}$$

is the probability that node p' has an incoming edge given $q(p') = k'$.
Similarly, we have for a node p with $q(p) = k$ that the out-degree is given by

$$P(d_{out}^{\alpha}(p) = l | q(p) = k) = \binom{N_p - 1}{l} \eta_{out}(k, \alpha)^l (1 - \eta_{out}(k, \alpha))^{N_p - 1 - l} \tag{3.5}$$

with

$$\eta_{out}(k, \alpha) = P(\delta(p, p') \leq \alpha | q(p) = k).$$

From 3.4 we can also derive the probability of a random pair of nodes having a given distance $P(\delta(p, p') \leq \alpha)$, which is $\text{Bin}(N_a, \rho - \rho^2)$ distributed (see Section A.2). Furthermore we can find the *unconditional* degree distributions by summing over all possible complexities k':

$$P(d_{in}^{\alpha}(p') = j) = \sum_{k'=0}^{N_a} P(d_{in}^{\alpha}(p') = j | q(p') = k') P(q(p') = k'). \tag{3.6}$$

and

$$P(d_{out}^{\alpha}(p) = l) = \sum_{k=0}^{N_a} P(d_{out}^{\alpha}(p) = l | q(p) = k) P(q(p) = k). \tag{3.7}$$

These are the probabilities that a random node in the product network has in-degree j or out-degree l. In the following we will derive $\eta_{in}(k', \alpha)$ and $\eta_{out}(k, \alpha)$ for the case of a random bipartite network with edge probability ρ.

3.3.2. Deriving $\eta_{in}(k', \alpha)$ and $\eta_{out}(k, \alpha)$ for the Binomial case

To compute the probability of a product with given complexity k' having an incoming edge, we first derive the weight probabilities, conditioned on product complexity $q(p')$. This is the probability that a given node with complexity $q(p') = k'$ shares w capability requirements with another node p:

$$\begin{aligned} P(W_{pp'} = w | q(p') = k') &= P\left(\sum_{a \in A} T_{ap} T_{ap'} = w | \sum_{a \in A} T_{ap'} = k' \right) \\ &= P\left(\sum_{a \in A_{p'}} T_{ap} = w | q(p') = k' \right) = \binom{k'}{w} \rho^w (1-\rho)^{k'-w}, \end{aligned} \tag{3.8}$$

where $A_{p'}$ is the set of all a such that $T_{ap'} = 1$. Here $|A_{p'}| = q(p') = k'$ and $k' \geq w$. Since the weight is symmetric, i.e. $W_{pp'} = W_{p'p}$ it does not matter if we condition on the target node p' or the source node p. For further derivations of the weight distribution see Section A.1.

From the weight matrix we can infer the distance measure from the product complexities as $\delta(p, p') = q(p') - W_{pp'}$. The distribution of distances for given complexity of target node p' is easily found by using (3.8) as follows:

$$\begin{aligned} P(\delta(p, p') = d | q(p') = k') &= P\left(q(p') - W_{pp'} = d | q(p') = k' \right) \\ &= P\left(W_{pp'} = k' - d \right) \\ &= \binom{k'}{k'-d} \rho^{k'-d} (1-\rho)^d \\ &= \binom{k'}{d} (1-\rho)^d \rho^{k'-d}. \end{aligned} \tag{3.9}$$

We find that the distance of a random node p *towards* a target node p' with $q(p') = k'$ is given by a $\text{Bin}(k', 1-\rho)$ distribution. Intuitively this makes sense, since the probability of finding a node p that has distance d to node p' is exactly the probability of finding that $T_{ap} = 0$ in one of the k' entries where $T_{ap'} = 1$ for d capabilities. Also, the distance is in a way the complement of the weight distribution: the distance counts exactly how many of the capabilities that p' requires are *not* also required by p (compare (3.9) to (3.8)).

We find that

$$\begin{aligned} \eta_{in}(k', \alpha) &= P(\delta(p, p') \leq \alpha | q(p') = k') \\ &= \sum_{d=0}^{\alpha} \binom{k'}{d} (1-\rho)^d \rho^{k'-d}. \end{aligned}$$

The probability of a random node p' having distance d *from* a source node p with $q(p) = k$. This is the probability that the target node p' has d entries $T_{ap'} = 1$ on the

$N_a - k$ available remaining entries where $T_{ap} \neq 1$, giving

$$P(\delta(p, p') = d | q(p) = k) = \binom{N_a - k}{d} \rho^d (1-\rho)^{N_a - k - d}. \tag{3.10}$$

Then

$$\begin{aligned} \eta_{out}(k, \alpha) &= P(\delta(p, p') \leq \alpha | q(p) = k) \\ &= \sum_{d=0}^{\alpha} \binom{N_a - k}{d} \rho^d (1-\rho)^{N_a - k - d}. \end{aligned}$$

This gives us the in- and out-degree distributions of random products, conditional on their complexities for the binomial case.
In the remainder we will set $\alpha = 1$. This entails the assumption one can only 'learn' a single capability at a time, and diffusion does not occur between products at a distance greater than 1. This can be interpreted as a 'local search' constraint. Economically, this assumption is justified by the fact that in order to make a 'jump' in the network over a distance greater than 1, one would first have to acquire a capability which would be of no use until combined with another (yet to learn) capability, so there would be no incentive to learn any of these capabilities individually. Hence we assume there are only edges in the model product space between products that are at most one capability away. Setting $\alpha = 1$, we obtain

$$\eta_{in}(k', 1) = \rho^{k'} + k'(1-\rho)\rho^{k'-1} \tag{3.11}$$

$$\eta_{out}(k, 1) = (1-\rho)^{N_a - k} + (N_a - k)\rho(1-\rho)^{N_a - k - 1}. \tag{3.12}$$

The observed and expected in- and out degrees for products with given complexity are shown in Figure 3.6 for different values of ρ. Here the dots represent the observed degree of a product for one instance of T_{ap}. The lines are the expected values following from conditional distributions (3.3) and (3.5) derived above. The green line shows

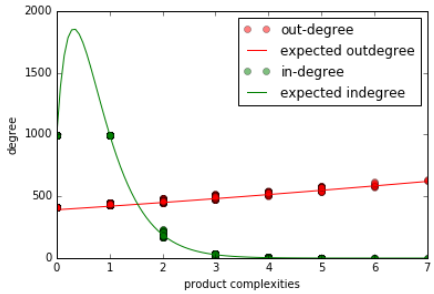
$$\begin{aligned} \mathbb{E}[d_{in} | q(p') = k'] &= (N_p - 1)\eta_{in}(k', 1) \\ &= (N_p - 1)(\rho^{k'} + k'(1-\rho)\rho^{k'-1}) \end{aligned}$$

and the red line shows

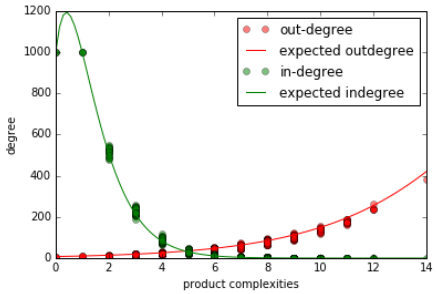
$$\begin{aligned} \mathbb{E}[d_{out} | q(p) = k] &= (N_p - 1)\eta_{out}(k, 1) \\ &= (N_p - 1)((1-\rho)^{N_a - k} + (N_a - k)\rho(1-\rho)^{N_a - k - 1}). \end{aligned}$$

Figure 3.6 shows the expected monotonic behavior of product degrees depending on their complexity. Furthermore it shows that our derivations are correct.
By setting $\alpha = 1$, the remaining parameters of the model are N_p, N_a and ρ. The connectivity of the product network can be studied by looking at how the distance between a random pair of nodes depends on ρ. The distribution of the distances is given by a $\text{Bin}(N_a, \rho - \rho^2)$ distribution (see Section A.2), so the mean distance between

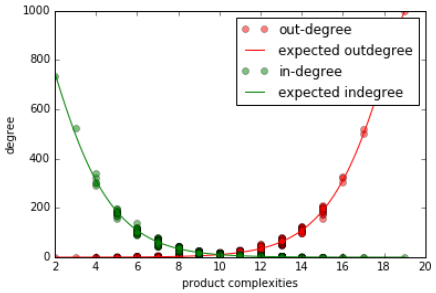
two products is given by $N_a(\rho - \rho^2)$, which is symmetric in ρ, takes its maximum at $\rho = .5$ and approaches 0 as ρ approaches 0 or 1. Notice also the dependence of the mean distance on N_a. As N_a grows, products are on average further apart. This means that the network becomes more poorly connected for larger values of N_a and moderate values of ρ.



(a) $\rho = 0.1$



(b) $\rho = 0.3$



(c) $\rho = 0.5$

Figure 3.3.: In- and out-degree distributions conditional on product complexity for $N_p = 1000$, $N_a = 20$ and $\alpha = 1$.

3.3.3. Joint degree distribution and percolation condition

From the economic perspective, we are interested in for which parameter values the product network allows wide-spread diffusion through the existence of a giant component, and for which parameter values the network consists of multiple disconnected components. A disconnected network would imply that poorly diversified countries could be 'stuck' in one of the components, and have minimal development opportunities as other products are unrelated in terms of their capability requirements.
In this section we will derive the conditions for which a significant fraction of the network can be reached from a given product, allowing wide-spread diffusion from a random seed product. We can measure this by looking at the expected out-component for a random node p, which is defined as all the nodes in the network that can be reached starting from node p. Likewise, one defines the in-component by all nodes from which one can

reach product p [17].
Using similar techniques as in Chapter 2, one can derive the condition for the emergence of a giant connected out-component for uncorrelated infinite random directed networks. The percolation condition for a giant out-component to arise is given by [17]

$$\sum_{jl}(2jl - j - l)P(d_{in} = j, d_{out} = l) \geq 0, \tag{3.13}$$

where $P(d_{in} = j, d_{out} = l)$ is the joint degree distribution of the network. If the percolation condition is met, the expected size of the out-component of a random node diverges and covers a fraction of the infinite network. Although the product network does not satisfy the tree ansatz, we will derive the percolation condition for a random network with the same degree distribution.
To derive the percolation condition we must first derive the joint degree distribution in the product network. The joint degree distribution gives the probability that a given node has in-degree j and out-degree l. In the product network, these are not independent. Products with high out-degree are likely to have a low in-degree and vice versa, since these are determined by the product complexities. High complexity products will have an edge towards all products that require a subset of their own capability requirements, and only have incoming edges from higher complexity products or products that require one less capability. We will derive the joint degree distribution by assuming that the in- and out-degree of a node are independent conditioned on their complexity. Assuming this conditional independence, we obtain

$$P(d_{in} = j, d_{out} = l|q(p) = k) = P(d_{in} = j|q(p) = k)P(d_{out} = l|q(p) = k).$$

The unconditional joint degree distribution is then given by

$$P(d_{in} = j, d_{out} = l) = \sum_{k=0}^{N_a} P(d_{in} = j|q(p) = k)P(d_{out} = l|q(p) = k)P(q(p) = k),$$

which we can compute using equations (3.3), (3.5) and (3.2).
The percolation condition (3.13) reduces to (see Appendix A)

$$(N_p - 1)\sum_{k=0}^{N_a} \eta_{in}(k, \alpha)\eta_{out}(k, \alpha)P(q(p) = k) \geq P(\delta(p, p') \leq \alpha).$$

Figure 3.4 shows for which parameter values the percolation condition is satisfied. For values of ρ above and below the area demarcated by the lines corresponding to a value of N_p, the percolation condition is satisfied and we can expect a giant out-component. It is clear that as N_a increases, a larger variety of products is possible and distances grow, causing the space to be poorly connected for low values of N_p or moderate values of ρ. Parameter ρ decreases the variety in products as it approaches 0 or 1. For $N_p = 1000$ and $N_a = 20$, a giant component exists for any value of ρ. For larger value of N_a however, the number of possible combinations increases and the distances between products increases.

This means that ρ would have to be increased or decreased, thereby changing the mean product complexity and decreasing the variety in capability requirements. This reduces distances between products and allows a giant out-component to form.

Figure 3.4 tells us for which parameters we can expect a giant out-component for

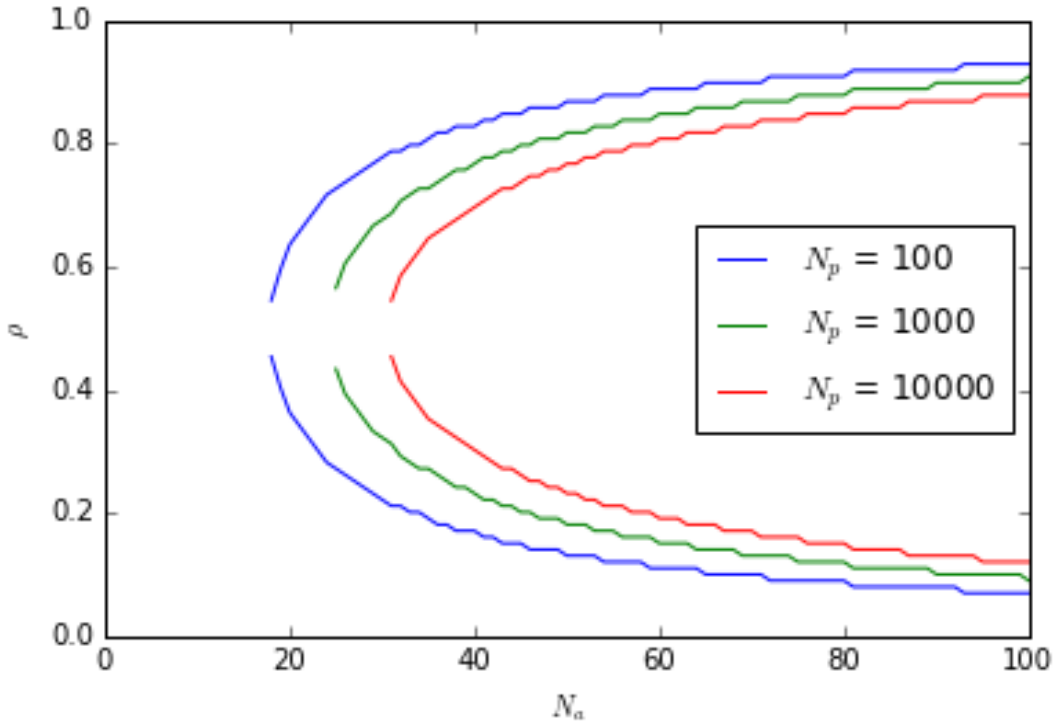


Figure 3.4.: The percolation condition for different values of N_a, N_p and ρ. The center right area in between each two lines is the part of parameter space for which the percolation condition is not satisfied.

a random product. As we have seen however, high complexity nodes have a high out-degree, whereas low complexity nodes have few outgoing links. This means that although on average we might expect a giant out-component, we expect this to happen only for high complexity products, meaning wide-spread diffusion is still not possible starting from low complexity products. We will further investigate this in the next section.

3.3.4. Up-edges

Thinking of the diffusion process, we are specifically interested in diffusion that goes from low to high complexity products. This 'upward' diffusion can only happen through edges that point in the direction of increasing complexity. We will call such edges 'up-edges' and we call ζ the probability that we find an up-edge (an out-edge pointing to a

higher complexity node) between a random pair of nodes:

$$\begin{aligned}\zeta &= P(\delta(p,p')=1, q(p')>q(p)) \\ &= \sum_{k=0}^{N_a} k\binom{N_a}{k'}\rho^{2k-1}(1-\rho)^{2N_a-2k+1}. \end{aligned} \tag{3.14}$$

This equals the probability of finding a product p' such that it has exactly one nonzero entry $T_{ap'}=1$ in the N_a-k entries where $T_{ap}=0$ and furthermore has k nonzero entries $T_{ap'}=1$ for all a where $T_{ap}=1$, so that $q(p)<q(p')$. For a derivation of (3.14) see Section A.3.

Figure 3.5 shows how the fraction of up-edges in the network depends on ρ for different values of N_a. We see that up-edges only occur for low or high values of ρ, i.e. if products require on average either very few or almost all of the available N_a capabilities. For $N_a=20$, we only expect up-edges for $\rho \leq 0.2$. As ρ approaches 0.5, the probability of finding up-edges in the network decreases.

The absence of up-edges for moderate values of ρ means that it is very hard to have

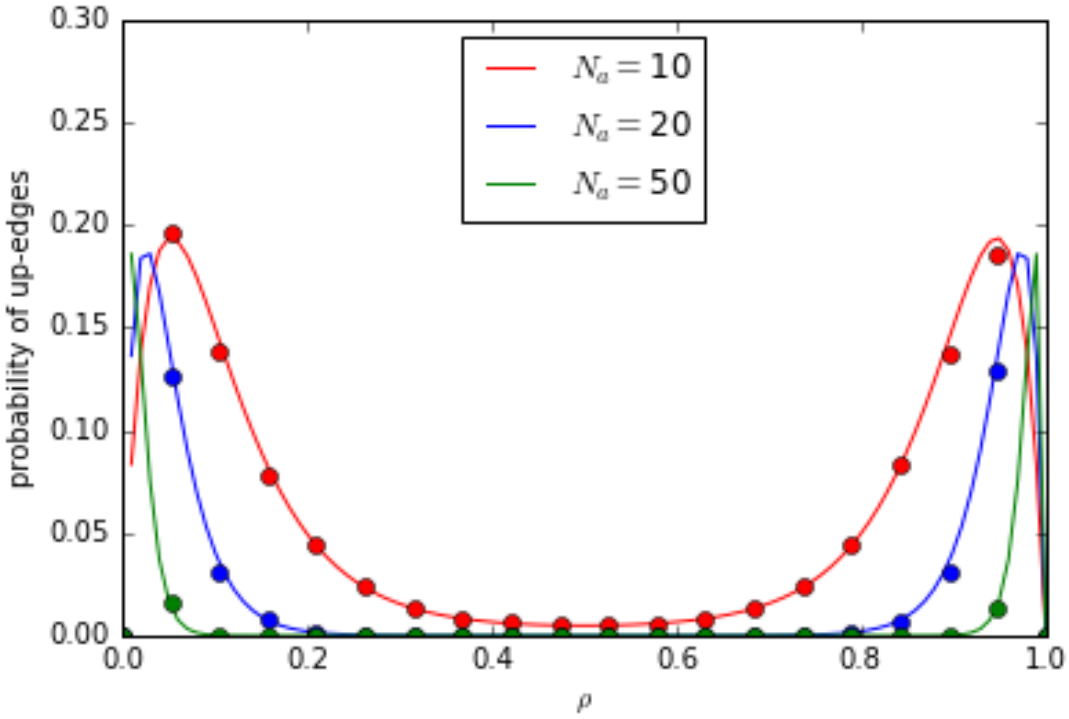
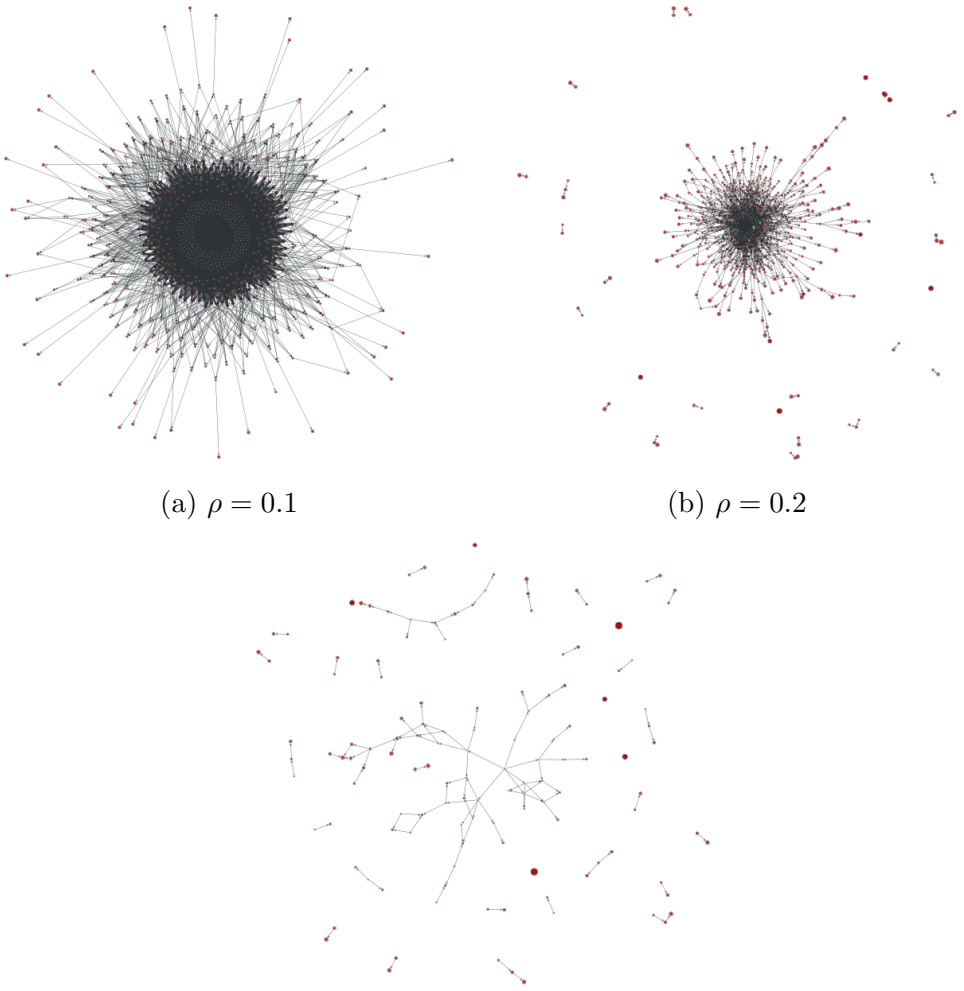


Figure 3.5.: Dependence of up-edges on N_a and ρ. The dots are the observed fraction of up-edges and the lines are the analytical prediction.

development from low to high complexity products since product of approximately the same complexity are expected to differ greatly in terms of *which combinations* of capabilities they require. As a consequence, we expect a small diffusion size when starting

with low complexity seeds in the binomial case with $\alpha = 1$ for moderate values of ρ. For $N_a = 20$, $N_p = 1000$ and $\rho = .5$, the expected number of up-edges from a random product is given by $(N_p - 1) \cdot \zeta = 9.53 \cdot 10^{-3}$, which means the expected number of up-edges in the whole network is given by $N_p \cdot 9.53 \cdot 10^{-3} = 9.53$. For $\rho = 0.3$ the expected number of up-edges in the network is 134, thus allowing some spread towards higher complexity products, but the number of up-edges is still very moderate compared to the total number of possible edges in the network (given by $N_p(N_p - 1)$). This tells thus that although we might expect a giant out-component for a random node, it is unlikely this will be the out-component of a low complexity product, making wide-spread diffusion starting from low complexity products impossible.



(a) $\rho = 0.1$ (b) $\rho = 0.2$

(c) $\rho = 0.3$

Figure 3.6.: Product networks with only the up-edges and nodes connected by them, showing paths of increasing complexity in the product network for $N_p = 1000, N_a = 20$ and $\alpha = 1$. As ρ approaches 0.5, the network becomes increasingly sparse and disconnected. Node size represents product complexity.

3.4. Conclusions

In this chapter we have constructed a product network as the one-mode projection of the country-capability network T_{ap}. By defining a distance measure and setting a threshold $\alpha = 1$, this results in a directed network in which there is a link from p to p' if either p' requires a subset of the capabilities that p requires, or p' requires at most one capability p does not require. This construction is based on the assumption that countries only learn a single capability at a time, and that capabilities are only acquired if they lead to a product. For example, one will only make an effort in acquiring the skill of shoemaking

if this enables the production of shoes. On the other hand, all products for which the capability requirements have been met will be produced, meaning that production of a product p will enable a country to produce all products that require a subset of the capability requirements of p.
As a consequence, high complexity nodes have high out-degree since there is an outgoing edge to every product that requires subset of its capability requirements. This means most of the outgoing edges of a high complexity product point towards a low complexity product. Low complexity products have low out-degree, as there are not many products that have exactly one extra capability requirement. This leads to a network in which a majority of the edges point from high to low complexity products.
The percolation condition and distribution of up-edges shows that in order for the network to be connected and allow for upward diffusion, we require low values of N_a, either low or high values of ρ, and high values of N_p. This can be understood through the binomial distribution of product complexities. The number of possible products with capability requirements (i.e. strings with 1's and 0's) with a complexity k is given by $\binom{N_a}{k}$, which attains its maximum for $k \approx N_a/2$ and decreases fast as k approaches 0 or N_a. The mean product complexity is given by $\mathbb{E}[q(p)] = N_a\rho$, so for values of ρ around 0.5 we expect many products of complexity $q(p) \approx N_a/2$. These products are not necessarily close to each other in terms of distance as they may require many different combinations of capabilities, and thus the probability of two products being identical except for a single capability (which would lead to an up-edge between them) is very low. For low and high values of ρ, products will mainly have 0 or 1 entries, making it more likely they are similar and thus close to each other in terms of distance. Hence the observed properties of the distribution of up-edges is a direct consequence of our assumption that the entries T_{ap} are binary random variables with equal probability ρ.
This leads to the question whether a random T_{ap} matrix is a reasonable assumption. As discussed, the binomial assumption implies a binomial distribution of product complexities but also of capability 'usefulness', i.e. how often a capability is used for a product. The mean usefulness of a capability is given by $N_p\rho$. The binomial assumption implies a relatively low diversity of product complexity and product usefulness. When one thinks of capability usefulness however, one would expect high heterogeneity as some capabilities will be used much more often than others. For example, almost every industry will require the use of electricity, whereas very few industries will require the industry-specific knowledge of a watchmaker. What the structure is of the 'recipe book' is an interesting question in itself and is inherently difficult as capabilities cannot be measured.
Nevertheless, the network we created gives a micro-founded model of a theoretical product space, allowing to model mechanisms of economic growth for individual countries. We wish to emphasize the difference between the model product network and the empirical product space from [13]. As discussed in Section 1.3, the empirical product space is constructed by using export data to obtain information on the relatedness of products in terms of their capability requirements, and this can be mapped into the product space. The approach taken here is reversed: we assume an underlying structure of all products (the T_{ap} matrix) and see what the resulting structure of the product network is. Hence, the empirical product space can be considered as an attempt to proxy the network that

we assume here. With the current model we try to reconcile the product space logic of preferred paths of economic development and the binomial model in which each product requires several specific inputs, and these inputs are modeled by a random matrix. In the following chapter we will run the percolation model on the created product network and investigate its properties for different parameter values.

4. Percolation on the model product space

In this section we will simulate the percolation process on the model product space constructed in Section 3. Note that there are some key differences with the case of the Erdös-Renyi random graph in Section 2.5. Firstly, the constructed network is directed, and has a given structure that is dictated by the shape of the T_{ap} matrix. Every product p has a directed edge to any product p' requiring a subset of the capabilities required by p plus at most one extra capability. Second, we now have information on the number of capabilities required by each product.
Recall from Section 2.5 that x_p denotes the difficulty of making a product, and a product can only be produced if the level of development of a country is high enough, i.e. $v > x_p$. In the standard percolation model we assumed the x_p values where uniformly distributed between 0 and 1. Here, following our explicit representation of products and capabilities, we propose that the difficulty of making a product corresponds to the number of capabilities it requires, thus setting

$$x_p = \frac{q(p)}{N_a} = \frac{1}{N_a} \sum_a T_{ap}.$$

This means that the level of development of a country can now be interpreted as how many capabilities a country is able to combine, assuming that combining many different capabilities in order to produce a given product requires a higher level of development. We thus assume all capabilities are available to a country, but the number of capabilities that can be combined is restricted by the level of development v.
The fraction of nodes that are in the operational network for given v is now given by the cumulative distribution function of a binomial distribution, since $q(p)$ is $\mathrm{Bin}(N_a, \rho)$ distributed (see (3.2)):

$$P(x_p \leq v) = \frac{1}{N_a} \sum_{k=0}^{\lfloor N_a v \rfloor} \binom{N_a}{k} \rho^k (1-\rho)^{N_a - k}.$$

This brings an additional effect to the diffusion size, as the fraction of nodes in the operational network does not depend linearly on v as is the case in the standard percolation model in Section 2.5. Since most products will have a complexity close to the mean complexity $N_a\rho$, there is a sharp increase in the fraction of operational nodes as v approaches ρ. Thus even if the product network is fully connected, the fraction of operational nodes does not increase until v comes close to ρ, as there are not many low

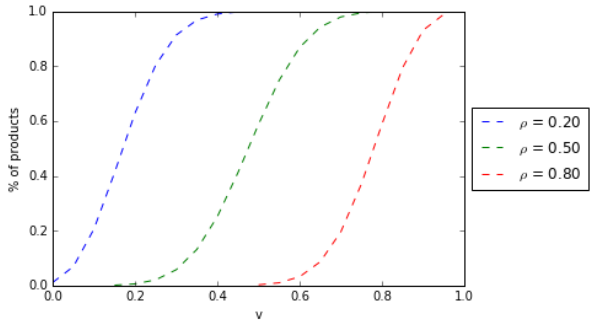


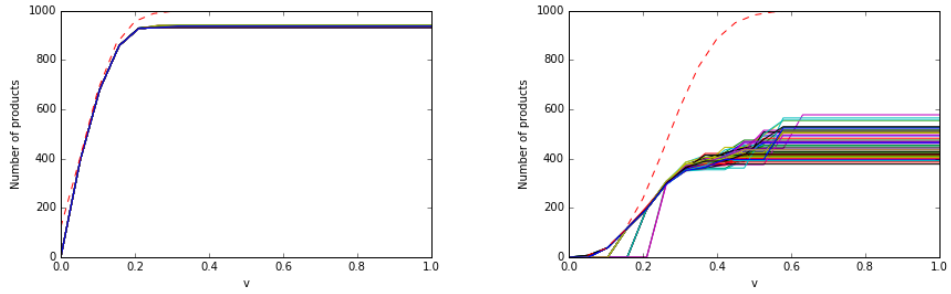
Figure 4.1.: Fraction of operational nodes for $N_a = 20$ and different values of ρ and v, with x_p given by products complexities.

complexity products. The same holds for values of v that are close to one: since there are not many high complexity products, all nodes are operational before $v = 1$.
Figure 4.1 shows the fraction of operational nodes for given different values of ρ. The dashed lines indicate how the fraction of operational nodes depends on parameter v. From the green line corresponding to $\rho = 0.5$, one can see that there are no products with complexity lower than 0.2 or higher than 0.8. This also means that parameter v only affects diffusion size for $v \in [0.2, 0.8]$. For other values of ρ, we see that the mean complexity $N_a\rho$ shifts, and the dependence of v shifts accordingly. In effect, parameter v can be seen to eliminate all columns from the T_{ap} matrix for which $\sum_a T_{ap} > v$. The operational network is the network resulting from the projection of the T_{ap} matrix without those columns.
In the following we will run every simulation with $N_p = 1000$ products, as this is of the order of magnitude of the number of products in datasets used in [13], [23]. Furthermore, we set $N_a = 20$ so that we expect a giant out-component (see Figure 3.4) for any value of ρ. For higher values of N_a, the distances between products grow as their capability requirements become more diverse, leading to a disconnected network unless many more products are sampled (N_p is increased).

4.1. Percolation on the model product space

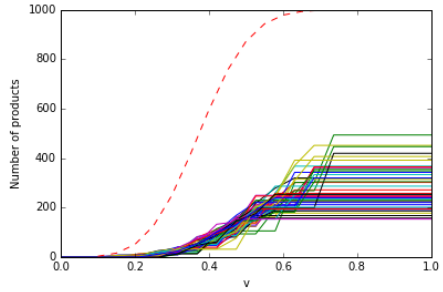
Figure 4.2 shows the results of the percolation process on the model product space with the x_p set equal to the product complexities. The diffusion is initialized by selecting 10 random seeds products. The red dashed line indicates the fraction of operational nodes in the network. As expected, the diffusion size decreases as ρ is increased. For $\rho = 0.1$, almost all nodes are activated as the network consists only of relatively simple products, causing distances to be small and the network being connected. As ρ increases, the diffusion size decreases as operational nodes are disconnected due to larger distances between products. For $\rho = 0.3$ and $\rho = 0.7$ there is a clear percolation threshold visible, for which a giant component emerges in the operational network as higher complexity nodes enter the operational network.

Varying ρ has two effects: it changes the number of operational nodes for given v (as

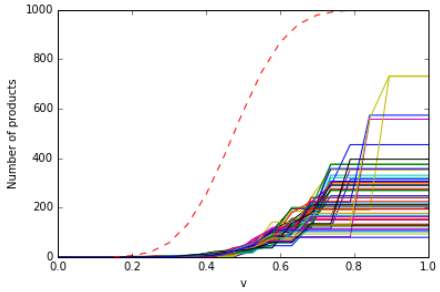


(a) $\rho = 0.1$

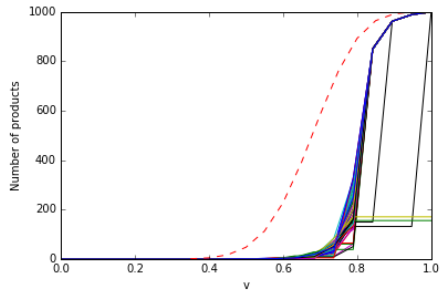
(b) $\rho = 0.3$



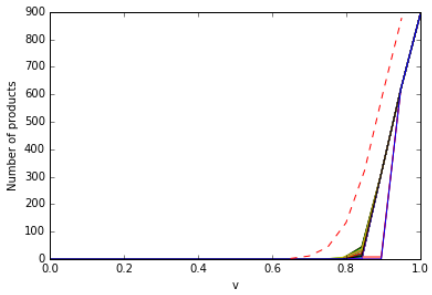
(c) $\rho = 0.4$



(d) $\rho = 0.5$



(e) $\rho = 0.7$

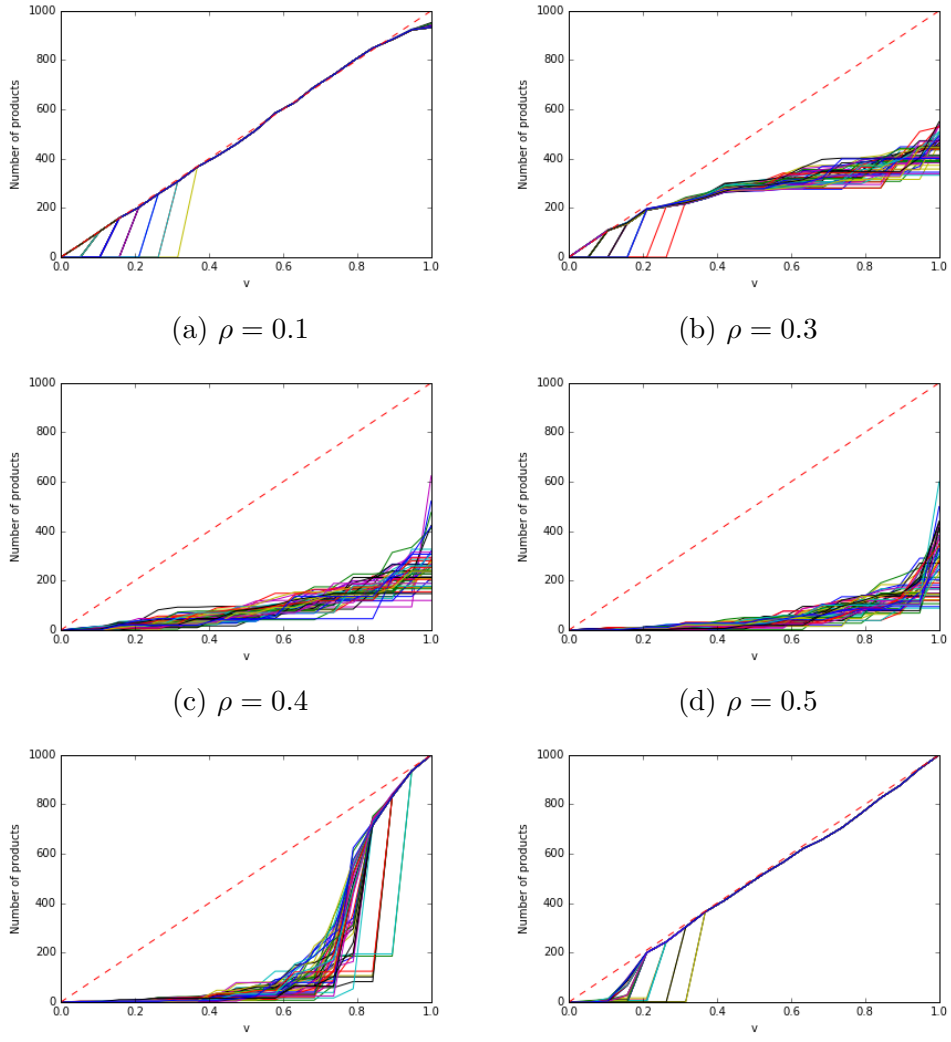


(f) $\rho = 0.9$

Figure 4.2.: Diffusion sizes over 50 simulation runs on the model product space for $N_p = 1000$, $N_a = 20$ for different values of v and ρ with x_p values given by the product complexities. Diffusion is initiated with 10 random seed products.

shown in Figure 4.1), and alters the structure of the product network, influencing how many operational nodes can actually be activated. Here we are interested in studying how ρ affects the network structure without the additional effect of the non-uniform distribution of product complexities. In other words, we wish to isolate the effect of the nonlinear dependence of the fraction of operational nodes on v caused by the binomial distribution of product complexities.
In Figure 4.3 the x_p values are sampled from a uniform distribution, but ranked according to product complexities, so the product with largest complexity $q(p)$ gets assigned the

largest x_p, etc. This allows us to only see the effect of the network structure on the relationship between diffusion size and v, since the fraction of operational nodes now depends linearly on v, but the order in which products of given complexity enter the network as v is increases is maintained. In this case, v can be interpreted as the fraction of total products that a country could potentially produce and we have $P(x_p < v) = v$. Products of equal complexity enter the operational network in random order.



(a) $\rho = 0.1$ (b) $\rho = 0.3$

(c) $\rho = 0.4$ (d) $\rho = 0.5$

(e) $\rho = 0.7$ (f) $\rho = 0.9$

Figure 4.3.: Diffusion sizes over 50 simulation runs on the model product space for $N_p = 1000$, $N_a = 20$ for different values of v and ρ with x_p values drawn from a uniform distribution and ranked according to product complexities. Diffusion is initiated with 10 random seed products.

From Figure 4.3 we can see no percolation threshold for $\rho = 0.4$ and $\rho = 0.5$, and there seems to be no transition in the network for moderate values of ρ. For $\rho = 0.1$,

a percolation threshold appears as simple products form a connected component as v in increased. Note that in Figure 4.2a, all products of complexity 1 become operational at once as v becomes larger than $1/N_a$, so that there is no percolation effect visible. In Figure 4.3a, these nodes are added to the network one by one, gradually building a network that percolates as soon as it connects to a seed node.
For $\rho = 0.3$, we expect the relatively simple products to connect to each other, and hence we observe an early percolation threshold. For $\rho = 0.7$ a connected component is formed by high complexity products, so the percolation threshold is observed for higher values of v since only then higher complexity products enter the operational network. For $\rho = 0.7$, the transition is stronger and the whole network becomes active, whereas for $\rho = 0.3$ the maximal diffusion size is around 500. This is because low complexity nodes are in the out-component of high complexity nodes, high complexity nodes are not in the out-component of low complexity nodes. Thus when a high complexity connected component forms (e.g. in the case of $\rho = 0.7$), the low complexity nodes are also activated since they are in the out-components of high complexity products. When a low complexity connected component forms however (e.g. for $\rho = 0.3$), high complexity nodes remain unactivated.
For $\rho = 0.9$ we again observe an early percolation threshold which may seem contradictory, but one must keep in mind that for this parameter value all products require almost all capabilities and are thus connect to each other. Also each of these nodes has a high probability of activating the few lower complexity nodes that enter the operational network for low values of v.

4.2. Conclusions

We conclude that reaching the whole product space starting from few random seeds is only possible for extremely small or high values of ρ, such that products are close together and the space in densely connected. For values of ρ close to 0.5, the network is poorly connected and no wide-spread diffusion takes place. For $\rho = 0.3$ and $\rho = 0.7$ giant connected components form but there is an asymmetry in percolation effects because the network is directed: low complexity products are in the out-component of high complexity products, but high complexity products are not in the out-component of low complexity products. This explains the difference in diffusion sizes for low and high values of ρ. One should note however that the connected component for $\rho = 0.7$ exists of high complexity products, so diffusion can only take place with a high complexity seed. We will discuss this in more detail in Chapter 5.
Here we have initiated the diffusion by taking random seeds. If these seeds are of high complexity, they will activate all products that require a subset of their capability requirements, which can be a lot of products. This is why the diffusion sizes in Figure 4.3 are still reasonable for $\rho = 0.5$. From an economic perspective however it seems unrealistic to initiate diffusion this way; product exports have been shown to go from simple to increasingly complex products as a country develops its economy [12]. Therefore we

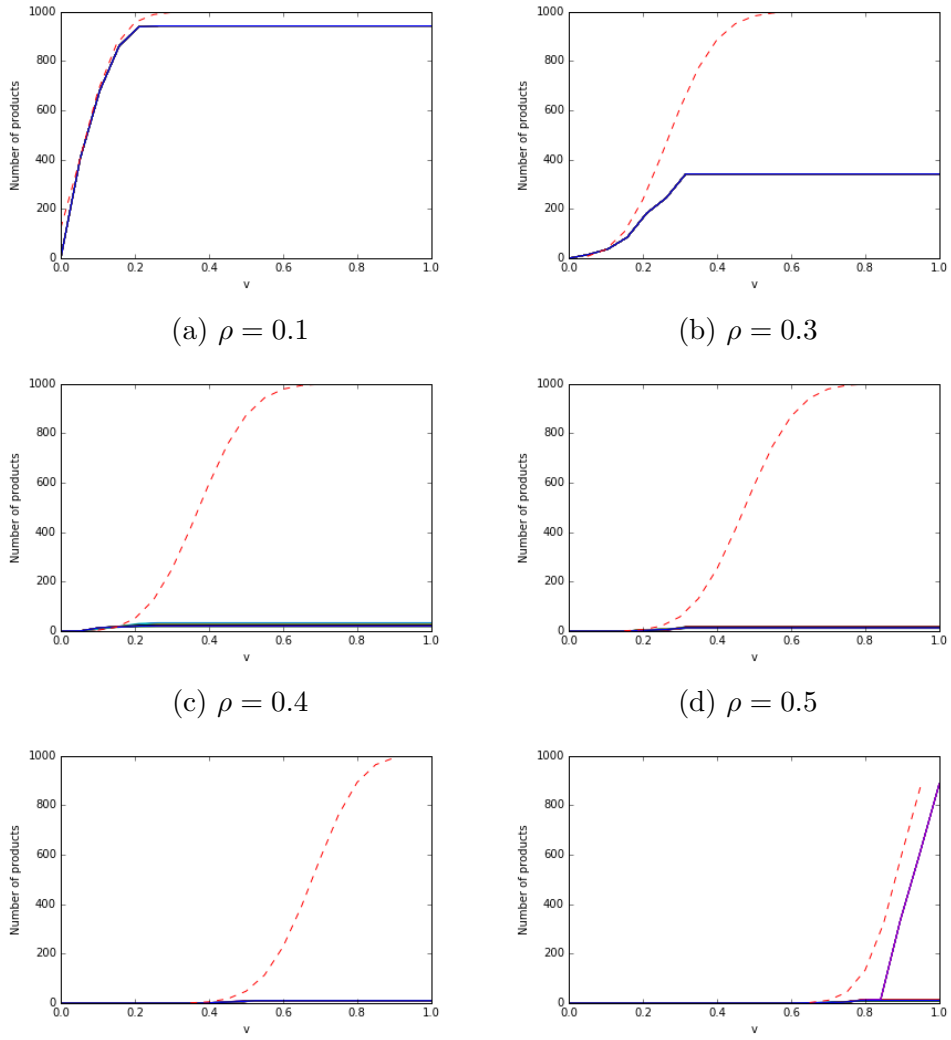
will consider in the next section the diffusion sizes when seeds are chosen to be of low complexity, modeling a country that starts with few capabilities.

5. Percolation as economic development

In this chapter, we initiate diffusion on the product network with low complexity products, modeling a country which only produces few low complexity products. We select the 10 seed products randomly from the 30 lowest complexity products. This way, widespread diffusion can only take place by developing products of increasing complexity. We again use values of $N_p = 1000$ and $N_a = 20$. From Section 3.3.4 we expect very limited diffusion for moderate values of ρ as almost no up-edges are present in the network, whereas for $\rho = 0.3$ diffusion can take place, although of moderate size. For $\rho = 0.7$ we expect as many up-edges, but between higher complexity products since the mean product complexity is higher.

5.1. Percolation with low complexity seeds

Indeed, Figure 5.1 shows that for $\rho = 0.7$ no diffusion takes place as the high complexity nodes are not in the out-component of the low complexity seed nodes. For low complexity seeds we see no diffusion for $\rho \in [0.4, 0.5, 0.7]$. For $\rho = 0.1$, diffusion takes place as there is a connected component of low complexity nodes. For $\rho = 0.9$ we also find diffusion, since even the lowest complexity nodes are of high complexity, and thus connected to the connected component. For $\rho = 0.3$ we observe an increase in the number of active products up to $v \approx 0.35$ in Figure 5.1, showing that the highest complexity product that can be made is of complexity $q(p) = N_a x_p = 7$. Apparently, products of higher complexity cannot be reached from lower complexity products. In this case, diffusion size is not bounded by v but by the structure of the product network.
We also observe that all 50 simulations where identical. This can be explained by the fact that since all seeds are sampled from the 30 lowest complexity products, they have high probability of being connected and thus being part of the same connected component. Hence for this value the product network is such that starting from random low complexity products, every country would be able to diversify into the same products, independent of initial conditions.

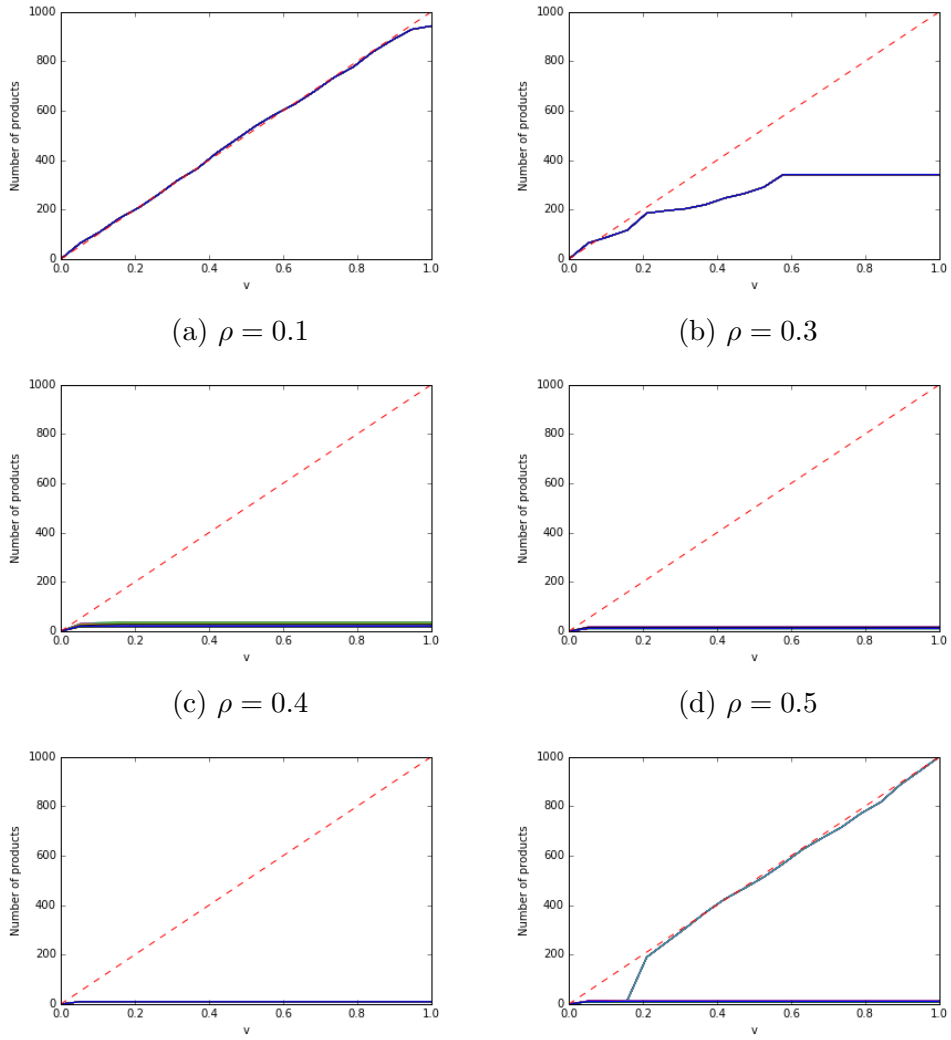


(a) $\rho = 0.1$ (b) $\rho = 0.3$

(c) $\rho = 0.4$ (d) $\rho = 0.5$

(e) $\rho = 0.7$ (f) $\rho = 0.9$

Figure 5.1.: Diffusion sizes over 50 simulation runs on the model product space for $N_p = 1000$, $N_a = 20$ for different values of v and ρ with x_p values given by the product complexities. Diffusion is initiated with 10 seed products selected randomly from the 30 lowest complexity products.



(a) $\rho = 0.1$ (b) $\rho = 0.3$

(c) $\rho = 0.4$ (d) $\rho = 0.5$

(e) $\rho = 0.7$ (f) $\rho = 0.9$

Figure 5.2.: Diffusion sizes over 50 simulation runs on the model product space for $N_p = 1000$, $N_a = 20$ for different values of v and ρ with x_p values drawn from a uniform distribution and ranked according to product complexities. Diffusion is initiated with 10 seed products selected randomly from the 30 lowest complexity products.

5.2. Conclusions

In Chapter 4 and 5 we applied the percolation model to the constructed product network. We considered if the network structure influences the development opportunities for countries through the network, in particular if we can find a sudden growth in diffusion size for small changes of parameter v.
Setting the required level of development equal to the complexity of a product, i.e. $x_p = q(p)$, we first notice that the distribution of product complexity in itself determines a nonlinear relationship between the number of products that can be made and parameter v.
Correcting for this by sampling x_p uniform and assigning them in order of product complexity allowed us to study only the network effect of parameter v. This showed that for values of ρ that away from 0.5, there is a percolation effect in the case of random seed products. The differences in diffusion sizes for ρ larger and ρ smaller than 0.5 can be explained through the directedness of the network.
To investigate if the network allows for development from low to high complexity products, we simulated diffusion starting from low complexity nodes. This showed that only for small or very large values of ρ diffusion can take place, since for moderate to high values the low complexity products are not in the in-component of the rest of the network.
We conclude that for binomially distributed product complexities, the assumption of 'local search', i.e. that countries only develop products that are at most one capability away from the products they currently produce, is very restrictive. The probability of being able to diversify into increasingly complex products when starting from simple ones is very low, which can be explained through the number of up-edges in the network.
Here we applied percolation to model the development properties of a country having access to all capabilities but being restricted in the number of capabilities it can combine. We have seen that if the network is poorly connected, parameter v is no longer a restriction of diffusion size, but rather the structure of the network itself restricts diffusion. The parameter v removes all the products from the network that have difficulty $x_p > v$, but if these are not in the out-component of the seed products, there is no effect on diffusion size.
In the following chapter we extend the model to study the time evolution of the diversification process, and diffusion is no longer restricted by a global parameter v, but driven by acquisition of new capabilities. We also implement the effect of spillovers, relaxing the assumption of local search. This also allows us to connect the model to existing literature.

6. Capability-driven diffusion and spillovers

In this chapter we consider the time evolution of the diversification process. In the previous chapters we assumed all capabilities that lead to an adjacent product in the network could be learned, and growth was constrained by the number of capabilities a country can combine, given by parameter v. The diffusion size then gives an indication of how many products can be reached for a given level of development v once all capabilities have been learned.
Here we model the diversification process in time, and see how the diversification process unfold in time as capabilities are learned. We suppose that every active product is being produced by an industry that has the necessary capabilities (given by T_p) to produce that product. Every time step a single industry can learn one additional capability, that in combination with the already present capabilities can enable production of new products, leading to a new industry in the following time step.
We will first introduce the model, and relate it to the percolation model discussed in Chapter 4. Then we relax the assumption of local search, and implement a mechanism of knowledge spillovers, allowing industries to recombine capabilities in order to make products that are not adjacent in the product network (i.e. are at distance more than one). This mechanism also relates the model to the Binomial model presented in [10] and other related work in the dynamics of innovations [7].

6.1. Model setup

Consider again the product network defined by the projection of the T_{ap} matrix. Every active product p is thought of to be produced by an industry. Since p is being produced, we assume all capabilities needed for production of p are available to that specific industry. We say that the 'capability basket' A_p of the industry producing p is given by T_p. Industries can add capabilities to their baskets through learning. One can imagine a new capability becoming available to an industry through internal processes such as research and development.
In the model, every time step a random industry learns a random new capability a which it does not already have. If adding this new random capability to the currently used ones leads to a new product, i.e. there is a 'recipe' for a product p' that requires a combination of a subset of the capabilities given by A_p, product p' is activated. This means that in the following time step, there is a new industry that is producing p'.
We keep track of active products through the vector M_p^t, where $M_p^t = 1$ if product p

is produced at time t and zero otherwise. Summarizing, the model in implemented as follows:

Algorithm 1 Diversification process

```
Select (low complexity) seed node
for every time step t do
    Select a random active product p
    Set capability basket of the industry producing p to A_p = T_p
    Add to A_p a random new capability: choose an a for which A_ap = 0 and set A_ap = 1
    Activate all products that can be produced given A_p:
    for all inactive products p' do
        if Σ_a T_ap' A_ap = q(p') then
            Set M^t_p' = 1
        end if
    end for
end for
```

Note that industries can only activate other industries at distance 1, just like in the percolation model. The difference here is that diversification occurs through trial and error: not all neighbors of an active product are activated every time step , but only the ones that become accessible through learning a random new capability. The outcome of the percolation model for $v = 1$ and the current model however are the same: as t goes to infinity, every active industry will have tried combining every possible new capability with its capability basket and will have activated all its neighbors. The process stops when the number of active products equals the size of the connected components of the seed product. It may take a very long time to reach this limit however, as many activation attempts fail as they do not lead to a new product.

As in the percolation model, products that are close to many other products will have a high probability of activating new products when a random capability is learned, leading to new industries. For more isolated products however, exactly the right capabilities must be learned in order to activate a nearby product, which happens with low probability since new capabilities are learned at random.

Figure 6.1 shows 20 simulation runs of the diversification process in time. It is clear that the diversification rate goes down as time progresses. This can be explained by the fact that as diversification increases, the probability of selecting a random product that does not lead to a new product increases. As the process unfolds, more products will have activated all their neighbors and will be unable to activate new products. Furthermore, as more neighbors become active, the probability of acquiring exactly the right capability to activate the remaining neighbors becomes smaller.

From the previous chapter, we know that the diversification process will continue until up to 400 products are active for $\rho = 0.3$ (see Figure 5.1). In the following we will extend the model and ask the question what happens to diffusion size and speed if we relax the assumption of local search, and allow industries to learn from each other and combine their capabilities, enabling them to activate products at distances greater than 1.

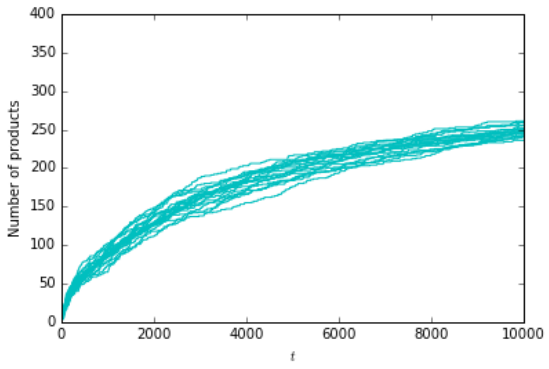


Figure 6.1.: 20 simulation runs of the diversification process for $N_p = 1000$, $N_a = 20$ and $\rho = 0.3$. Diffusion was initialized by a single seed product selected randomly from the 30 lowest complexity products.

6.2. Spillovers

Here we implement knowledge spillovers in the system by assuming industries may learn from other industries to the extent that they use the same combinations of capabilities. The idea here is that knowledge is easily shared between industries that produce related products, as they combine capabilities in similar ways. One might even think of big firms producing multiple products, and hence having access to all capabilities required for production of those products.
How similar industries must be for knowledge transfer to take place in the model is determined by the parameter β. For $\beta = 1$, industries only share knowledge if their capability requirements are identical and learning new capabilities from other industries is not possible. For $\beta = 0$, industries share their knowledge with any other industry. For intermediate values, an industry producing product p can learn the capabilities from all other industries producing products p' for which $s(p, p') \geq \beta$, where $s(p, p')$ is the *similarity* between two products in terms of their capability requirements.
We define the similarity between two products as [2]

$$s(p, p') = 2\frac{\langle T_p, T_{p'} \rangle}{q(p) + q(p')}.$$

This quantity equals 1 if products require the same capabilities, and 0 if they share none. The numerator counts the number of capability requirements that both products share. The denominator is the sum of the product complexities, and causes the similarity to have small values if the two products only share a small fraction of their total capability requirements. As an example, consider the two products

$$p = (00011)$$
$$p' = (10011).$$

Then $s(p, p') = s(p', p) = 4/5$. For

$$p = (0000011)$$
$$p' = (1110011)$$

we have $s(p, p') = 4/7$. Note that although in the second example the two products share the same number of capability requirements, the similarity goes down as p' becomes more complex. The similarity measure thus takes into account the number of shared capabilities and the number of capabilities that are *not* shared.
Introducing spillovers in the model, the scheme looks as follows:

Algorithm 2 Diversification process with spillovers

```
Select (low complexity) seed nodes
for every time step t: do
    Select a random active product p
    Set capability basket of the industry producing p to A_p = T_p
    Add to A_p all capabilities from similar industries:
    for all p' such that s(p, p') ≥ β do
        If A'_p = 1 and A_p = 0, set A_p = 0
    end for
    Add to A_p a random new capability: choose an a for which A_ap = 0 and set A_ap = 1
    Activate all products that can be produced given A:
    for all inactive products p' do
        if Σ_a T_ap' A_ap = q(p') then
            Set M^t_p' = 1
        end if
    end for
end for
```

6.2.1. The $\beta = 0$ case

Let us first consider the case where $\beta = 0$. In that case, all industries are considered similar and thus share their capabilities with each other. Every time step, the selected industry learns the capabilities of every other active industry. As a consequence, it is no longer relevant which industry is chosen per time step, since the capability basket of every industry will be identical after they learn capabilities from each other. This means we can consider in this case the capability basket of a *country*, as in the binomial model [10].
If we now consider the diversification process, two things can happen every time step: the newly added capability leads to the activation of one or more new products, or there is no product that can now be activated and we move on to the next time step without any changes. Let us assume for the moment that in every time step a new product activation takes place. This means that every time step, a capability is added to the countries' capability basket. Starting with no capabilities, a country will have exactly t

capabilities at time step t. The expected number of active products for a given number of capabilities is derived in [10]. The probability of producing a product of complexity k given a country has l capabilities is given by $(\frac{l}{N_a})^k$. Assuming that at time t we have t capabilities, the expected number of active products at time t is given by

$$\begin{aligned}\langle d_c^t \rangle &\approx \sum_{k=0}^{N_a} P(M_p^t = 1 | q(p) = k) N_p P(q(p) = k) \\ &= \sum_{k=0}^{N_a} \left(\frac{t}{N_a}\right)^k N_p \binom{N_a}{k} \rho^k (1-\rho)^{N_a - k} \\ &= N_p \left(\rho \frac{t}{N_a} + (1-\rho)\right)^{N_a}. \end{aligned} \tag{6.1}$$

This equation is increasing in t and convex, hence the rate of diversification is increasing in the number of capabilities a country has. This means there are increasing returns in the number of active products to the number of capabilities obtained. Hence countries with many capabilities will diversify faster than countries with few capabilities. Countries with few capabilities will experience low returns on acquiring a capability, leading to the equivalent of a poverty trap for countries. Furthermore it is shown in [10] that this effect increases when N_a or ρ are increased. Also, this result is independent of the assumption that the product complexities are binomially distributed, since $P(q(p) = k)$ is by definition independent of the number of capabilities a country has.
Figure 6.2 shows 20 simulation runs of the diversification process with $\beta = 0$, and shows the expected number of active products according to equation (6.1). Since in the present model there are some time steps in which no capability is learned (which happens especially when not many products are active yet), we observe that on average diversification occurs slower than dicatated by equation (6.1). Nevertheless, we see that the simulations produce the same type of curves as the expected diversification, showing a clear increasing rate of diversification in time. In the case of no spillovers, where $\beta = 1$, the process is equivalent to the algorithm introduced in Section 6.1, where we observed that the rate of diversification decreases in time (see Figure 6.1).

In the general model we propose, a country does not possess a number of capabilities that enables it to produce products, but rather the individual industries learn new capabilities that may lead to production of a similar product. This means that although some industries may have access to certain capabilities, these capabilities are not necessarily available to other industries. This weakens the assumption of the Binomial model [10] that capabilities are either present or not to all industries in a country. Parameter β allows us to transition between two models extreme cases of the model.

6.2.2. Intermediate cases

In this section we will compare simulations for different values of β and see how spillovers affect diversification rates and final diffusion size. We have already seen that for $\beta = 0$, all products can be produced at the end of the process for most simulations.

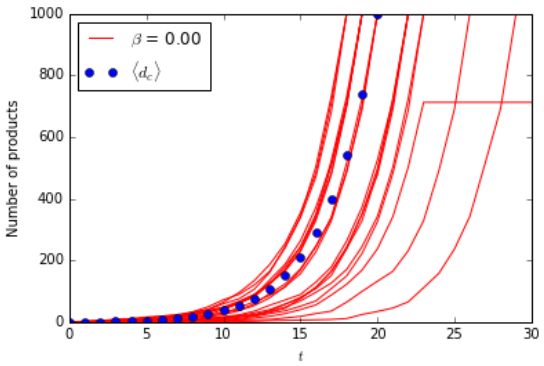


Figure 6.2.: 20 simulation runs of the diversification process for $N_p = 1000$, $N_a = 20$, $\rho = 0.3$ and $\beta = 0$. Diffusion was initialized by a single seed product selected randomly form the 30 lowest complexity products. The blue dots show the expected diversification for a given time step.

Figure 6.3 shows the diversification process for different values of β, where time is plotted on a logarithmic scale. We observe that also for values of β greater than 0, there is a point in which a state of stagnant growth is overcome and all products are activated at the end of the process for all simulations. As β increases and there is less sharing of capabilities, the time until 'take-off' is increased. For $\beta = 1$, this take-off does not occur at all, since the results in Chapter 5 tell us that the number of active product is around 400 at the end of the process for $\rho = 0.3$. Hence there is a value of β for which the qualitative behavior of the diversification process changes from decreasing innovation rates and no wide-spread diffusion to a situation in which every product can be reached and there is an increasing rate of diversification.

Figure 6.4a shows the diversification process for values of β between 0.7 and 0.8, where the transition from a decreasing to an increasing rate takes place. For $\beta = 0.7$ we find explosive growth within the first 2000 time steps. For $\beta = 0.73$ however, we find that first there is a regime where the diversification rate is decreasing, until a point is reached where the effect of spillovers takes over and the diversification rate increases in time. In Figure 6.4b the same simulations are shown, but time is plotted on a logarithmic scale. This makes clear that at the start of the process, diversification rates are the same for different values of β, indicating that spillovers have a minimal effect in when not many products are active and their complexity is low. Only when there are many active products with approximately the same complexity such that their similarity is high, spillovers start having an effect. This point is reached earlier for lower values of β.

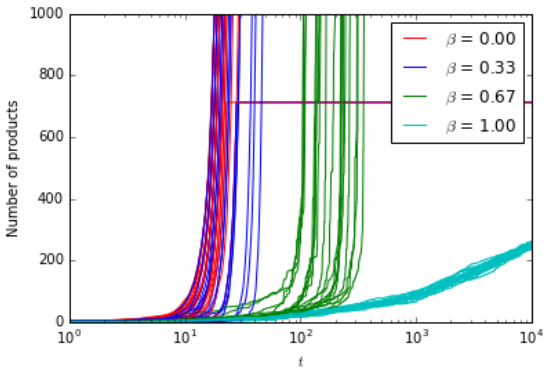
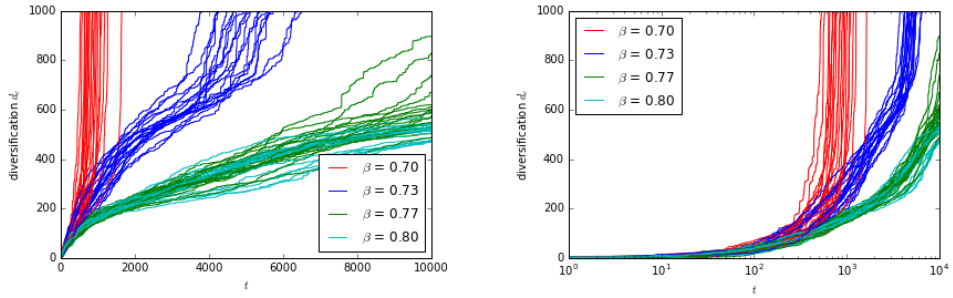


Figure 6.3.: 20 simulation runs of the diversification process for $N_p = 1000$, $N_a = 20$, $\rho = 0.3$ different values of β. Time steps are given on a logarithmic scale. Diffusion was initialized by a single seed product selected randomly from the 30 lowest complexity products.



(a) (b)

Figure 6.4.: 20 simulation runs of the diversification process for $N_p = 1000$, $N_a = 20$, $\rho = 0.3$ different values of β, on a normal and on a logarithmic time scale. Diffusion was initialized by a single seed product selected randomly from the 30 lowest complexity products.

6.3. Discussion

From the current model, we have seen that spillovers increase the opportunity of activating nodes by enabling product activations over distances greater than 1. In the absence of spillovers, products can only be activated through neighbors in the product network. The rate of diversification is decreasing in time as it becomes increasingly hard to activate new products every time step. Diffusion size is bounded by the size of the out-component of the seeds, which we have seen in Chapter 5 is very limited.
For $\beta = 0$ capabilities are exchanged between any industry, and we see explosive growth that can be explained through a combinatorial process as previously shown in [10]. As a consequence, countries with many capabilities can more easily engage in production of new products by acquiring a new capability than countries which have only few ca-

pabilities, which leads to a poverty trap.
The model here relaxes the assumptions of the binomial model in two ways: capabilities are only learnt if they lead to new products, and capabilities are introduced into the economy by individual industries, and may not spread to all other industries. The spillover parameter β determines how easily capabilities are transferred between industries. For intermediate values of β, the 'poverty trap' grows as β approaches 1 (less knowledge spillovers). We can identify two regimes: at first, there is a decreasing rate of diversification and spillovers have minimal effect. When there are enough products active that are approximately of equal complexity, spillovers occur and the rate of diversification increases in time, leading to an explosive growth of the number of active products, and enabling countries to activate all products in the network. For the binomial distribution of product complexities we expect the transition between these two regimes to be particularly sudden, as most products have complexity $N_a\rho$. Since products of equal complexity are likely to be similar, the effect of spillovers will suddenly be big once a country starts producing products of average complexity. For more dispersed product complexity distributions the effect of spillover may be less prominent. Note however that the result of increasing returns for $\beta = 0$ holds for any distribution of product complexities, so we always expect to end up in a regime of increasing return for $\beta = 0$.
We interpret parameter β as a measure of how easy know-how spreads in the economy. Not surprisingly, the model shows that economies in which capabilities are shared between industries are more effective of engaging in production of new products. One could argue that parameter β takes different values over time and space. In big cities for example, capabilities may be more easily transferred between industries as they are closer together and have more interactions. Historically, β may have decreases through globalization and urbanization. Also the codification of knowledge through for example patents, increased mobility of people and accessibility of knowledge through internet can contribute to the exchangeability of capabilities between industries.

7. Discussion

In this thesis, we have attempted to model the mechanisms by which economic growth is driven by accumulation of knowledge, and how this growth is restricted by the concept of related diversification. We have proposed to model these dynamics as a percolation process on a complex network. In this, we built on the framework of economic complexity to create a theoretical product space that countries explore as they develop economically. We have modeled the product space as the one-mode projection of the bipartite capability-product network T_{ap}, that dictates which products use which specific capabilities. We assumed the T_{ap} matrix to be random, with the probability of a product using a certain capability given by parameter ρ. As a consequence, the distribution of product complexities is given by a binomial distribution. In Chapter 3 we investigated the connectivity properties of this network for different parameter values, and concluded that connectivity decreases as N_a increases and ρ approaches 0.5.
In Chapter 4 and 5 we modeled economic growth as a percolation process in which growth is restricted by the general level of development of a country. Here we were interested in how the total number of reachable products depends on the level of development v. Modeling development by initiating diffusion with low complexity seeds, we conclude that diffusion is only possible for relatively small values of ρ, well below $\rho = 0.5$. For higher values of ρ, most products are not accessible by low complexity products, and for values of ρ close to 0.5 the network is not well connected, hindering diffusion. We find that the structure of the network and the assumption of local search is in many cases more restrictive than the level of development v. When diffusion is initiated with low complexity seeds, there is no percolation threshold due to the structure of the network.
In Chapter 6 we considered how the diversification process unfolds in time. We propose that diffusion is driven by accumulation of capabilities by individual industries. After infinitely many time steps, the number of active nodes is given by the diffusion size in the percolation model. In the model, the rate of diversification is decreasing in time, as it becomes less likely that exactly the right industries add exactly the right capabilities to their capability baskets as time unfolds. By implementing spillovers, we relaxed the assumption of local search and allowed industries that use similar combinations of capabilities to recombine their capabilities in order to make new product at further distances. How similar industries must be for them to exchange capabilities is given by parameter β. For $\beta = 0$, the case of full spillovers, we can related the model to the binomial model in [10] and find increasing returns in the accumulation of capabilities, leading to a poverty trap for countries with few capabilities. As β is increased, the poverty trap grows, and a regime in which the diversification rate is decreasing is clearly visible for early stages of the process. Once products reach a level of complexity for which spillovers take effect, the system shows an increasing rate of diversification where

the whole network is accessible.
The modeling approach taken in this thesis has severe limitations - it assumes a finite product space, does not model de-activation of products and does not consider interactions between countries. Hence we do not expect the model to have any predictive or even descriptive value of dynamics of economic development. Nevertheless, the model can help to understand the dynamics of the diversification process as proposed by the economic complexity framework. In this thesis we have tried to connect concepts of related diversification, accumulation of capabilities and knowledge spillovers into a simple model that shows how they can affect the diversification process.
How does the model relate to the real world? Considering the capability-product network T_{ap}, we could ask what would be a reasonable value for ρ. In the product network, ρ determines on average how many capability are used per product. For values of $\rho > 0.5$ products would use on average more than half of the available capabilities which seems unrealistic as we consider products in the real world to require very specific combinations of capabilities, and most large combinations of capabilities do not lead to a viable product. Also, the fact that for large values of ρ the distances in the product network decrease seems artificial, as it is a consequence of assuming a finite number of capabilities N_a. We therefore think of ρ taking values below 0.5, but not too small as in general products do require multiple capabilities.
Whereas ρ is considered a fixed value in the model, parameter v and β are considered controllable to some extent. Parameter β is a measure of how easily capabilities are shared between industries. It is clear from the model that if capabilities are easily shared, this can have a huge effect on economic development, overcoming disconnectedness from the product network a country is exploring. The most interesting question perhaps is what the structure of the product-capability network is in the real world, and how it evolves in time under the influence of innovation.
In general, the assumption that T_{ap} is a random matrix is a crucial one: it determines the product complexity distribution and the distances between products. The fact that the capability requirements of products are independent of each other is in contradiction with the idea that production of one product leads to the discovery of how to make another. This would suggest that the capability requirements of products are very dependent of each other. In the current model, we assume a fixed recipe book T_{ap} and the network of viable products is 'discovered' as diffusion unfolds. The resulting network of *active* products at the end of the process are related to each other in a hierarchical way.
A different approach to model the product network, that also overcomes the limitations of the assumption of a finite product network is one in which the network is generated as diversification takes place. This way, the product network is *created* by industries as they innovate and create new products, extending the network (see [18] for an example). Adding some hierarchical structure to the product network T_{ap} may give very different results and a more realistic model, more appropriate for modeling the time evolution of the diversification process. Other product complexity distributions are considered in e.g. [1] and [7] in a slightly different context. Note that the increasing returns result of equation (6.1) holds for any product complexity distribution, and thus any product complexity distribution.

An alternative way to think about the structure of the T_{ap} matrix in given in [22], where the distribution of the usefulness of capabilities is considered as opposed to the product complexity distribution. In the current model, the number of products a capability uses (the degree of the capabilities in the bipartite T_{ap} network) is binomially distributed with mean value $N_p\rho$. Hence all capabilities are considered to be approximately equally useful. Thinking of capabilities however it seems obvious that some general capabilities such as access to electricity are used in many more products than very specific capabilities such as the skill to process leather. Hence it seems reasonable that the distribution of capability usefulness is given by a distribution with high variance such as a power-law. Implementing this leads to results that captures the structure of export data better than the binomial model [22].
Another limitation is that the model only describes whether a product is being produced or not, but does not quantify products nor capabilities. For an extension of the binomial model in this direction see [8]. Furthermore the model proposed here does not incorporate interactions between countries and the disappearance of products from a countries' export basket. For an example of a model incorporating creative destruction see [14]. The approach taken here however can be generalized to other systems that deal with creation of new entities that are built of components. See for example [7] for an application to gastronomy, technology and language. In [7], the conclusion is that front-loaded product complexity distributions bring forth higher innovation rates, and innovation rates are fully determined by the product complexity distribution. Using the network approach proposed here on other systems than capabilities and products has the advantage that in some cases the building blocks (which where capabilities here) become tangible: in gastronomy for example, the recipe book T_{ap} is literally available, connecting ingredients to dishes. This allows to directly assess the product complexity and capability usefulness distributions for different systems. Understanding the structure of the product network may then shed light on the dynamics of innovation for specific systems. Other examples to bring the model to data include datasets that describe which skills are used for which occupations, or which patents are used in other new technologies.
What the current model does incorporate which is not present in other models mentioned above, and in particular in the binomial model [10], is the notion of local search, thereby addressing a central issue in the dynamics of economic development or innovation: how are new capabilities (components) introduced to the system? We proposed here that acquisition of capabilities occurs through invention of new products - and thus capabilities only enter the system when they lead to a new product. Why would one learn a capability if not all its complementary capabilities were already present? And once a capability is learned in order to produce a specific product, how does it diffuse to other industries? In [10] it is suggested that the challenge in economic development is to minimize this coordination problem, and related diversification is a result of this minimization. Here we have combined the combinatorial aspects of the theory of capabilities, and the restrictive aspect of related diversification into a single model, that may provide grounds for further theoretical and empirical investigation to how these forces are balanced in the economy.

Bibliography

[1] S. Bustos, C. Gomez, R. Hausmann, and C. A. Hidalgo. The Dynamics of Nestedness Predicts the Evolution of Industrial Ecosystems. *PLoS One*, 7(11):e49393, 2012.

[2] G. Caldarelli, M. Cristelli, A. Gabrielli, L. Pietronero, A. Scala, and A. Tacchella. A Network Analysis of Countries' Export Flows: Firm Grounds for the Building Blocks of the Economy. *PLoS ONE*, 7(10):1–11, 2012.

[3] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: percolation on random graphs. *Physical Review Letters*, 85(25):5468–5471, 2000.

[4] M. Cristelli, A. Gabrielli, A. Tacchella, G. Caldarelli, and L. Pietronero. Measuring the Intangibles: A Metrics for the Economic Complexity of Countries and Products. *PLoS ONE*, 8(8), 2013.

[5] M. Cristelli, A. Tacchella, and L. Pietronero. The Heterogeneous Dynamics of Economic Complexity. *Plos One*, 10(2):e0117174, 2015.

[6] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(1275):1–79, 2008.

[7] T. M. A. Fink, M. Reeves, R. Palma, and R. S. Farr. Dynamics of rapid innovation. (2):6, 2016.

[8] A. Gomez-lievano. Applying Distributional Approaches to Understand Patterns of Urban Diversification. *Dissertation*, 2014.

[9] A. Gomez-lievano, O. Patterson-lomba, and R. Hausmann. Explaining the Prevalence , Scaling and Variance of Urban Phenomena. *arXiv*, pages 1–39.

[10] R. Hausmann and C. a. Hidalgo. The network structure of economic output. *Journal of Economic Growth*, 16(4):309–342, 2011.

[11] R. Hausmann, C. A. Hidalgo, S. Bustos, M. Coscia, S. Chung, J. Jimenez, A. Simoes, and M. A. Yildirim. *The Atlas of Economic Complexity*. 2011.

[12] C. a. Hidalgo and R. Hausmann. *The building blocks of economic complexity.*, volume 106. 2009.

[13] C. A. Hidalgo, B. Klinger, A.-L. Barabasi, and R. Hausmann. The Product Space Conditions the Development of Nations. *Science*, 317(5837):482–487, 2007.

[14] P. Klimek, R. Hausmann, and S. Thurner. Empirical confirmation of creative destruction from world trade data. *PLoS ONE*, 7(6):1–9, 2012.

[15] F. Neffke, M. Henning, and R. Boschma. How Do Regions Diversify over Time? Industry Relatedness and the Development of New Growth Paths in Regions. *Economic Geography*, 87(3):237–265, 2011.

[16] M. E. J. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[17] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. 2000.

[18] F. Saracco, R. Di Clemente, A. Gabrielli, and L. Pietronero. From innovation to diversification: A simple competitive model. *PLoS ONE*, 10(11):1–19, 2015.

[19] M. Á. Serrano and M. Boguná. Clustering in complex networks. II. Percolation properties. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 74(5):1–8, 2006.

[20] S. Solomon, G. Weisbuch, L. De Arcangelis, N. Jan, and D. Stauffer. Social percolation models. *Physica A: Statistical Mechanics and its Applications*, 277:239–247, 2000.

[21] A. Tacchella, M. Cristelli, G. Caldarelli, A. Gabrielli, and L. Pietronero. A New Metrics for Countries' Fitness and Products' Complexity. *Scientific Reports*, 2:1–4, 2012.

[22] A. Tacchella, R. Di Clemente, A. Gabrielli, and L. Pietronero. The Build-Up of Diversity in Complex Ecosystems. 2016.

[23] A. Zaccaria, M. Cristelli, A. Tacchella, and L. Pietronero. How the taxonomy of products drives the economic development of countries. *PLoS ONE*, 9(12):1–17, 2014.

[24] P. Zeppini and K. Frenken. Networks , Percolation , and Demand. 2015.

A. Derivations

A.1. Derivation of distribution of weights

The probability of two products sharing a given number of capabilities, i.e. that an entry of the weight matrix attains a certain value,

$$P(W_{pp'} = w)$$

is given by a $\text{Bin}(N_a, \rho^2)$ distribution. Here we will derive this result in two different ways, to verify the result and also check the methodology. We will first derive the results directly from the T_{ap} matrix, and then use conditional probabilities and the law of total probability to get the same results.
Since the T_{ap} are independent for all a, p,

$$T_{ap}T_{ap'} = \begin{cases} 1 & \text{with probability } \rho^2 \\ 0 & \text{with probability } 1-\rho^2 \end{cases}$$

which indicates whether two products share a capability or not. The random variable

$$W_{pp'} = \sum_{a \in A} T_{ap}T_{ap'}$$

gives the probability of finding an edge weight in the one-mode projection and is given by

$$P(W_{pp'} = w) = \binom{N_a}{w} (\rho^2)^w (1-\rho^2)^{N_a - w}. \tag{A.1}$$

This gives the distribution of the weight matrix $W_{pp'}$.
We can check (A.1) by deriving the unconditional probability that two random products

share w capability requirements using (3.8):

$$\begin{aligned}
P(W_{pp'} = w) &= \sum_{k'=0}^{N_a} P\left(W_{pp'} = w | q(p') = k'\right) P(q(p') = k') \\
&= \sum_{k'=0}^{N_a} \binom{k'}{w} \rho^w (1-\rho)^{k'-w} \binom{N_a}{k'} \rho^{k'} (1-\rho)^{N_a - k'} \\
&= \sum_{k'=0}^{N_a} \frac{k'! N_a!}{w!(k'-w)! k'! (N_a - w)!} \rho^{w+k'} (1 - \rho^{N_a - w}) \\
&= \sum_{k'=0}^{N_a} \frac{N_a}{w!(N_a - w)!} \cdot \frac{(N_a - w)!}{(k'-w)!(N_a - k')!} \rho^{w+k'} (1-\rho)^{N_a - w} \\
&= \sum_{k'=0}^{N_a} \binom{N_a}{w} \binom{N_a - w}{k' - w} \rho^{w+k'} (1-\rho)^{N_a - w}.
\end{aligned}$$

We now set $l = k - w \geq 0$ and obtain

$$\begin{aligned}
P(W_{pp'} = w) &= \sum_{l=0}^{N_a - w} \binom{N_a - w}{l} \rho^l \binom{N_a}{w} \rho^{2w} (1-\rho)^{N_a - w} \\
&= (1+\rho)^{N_a - w} \binom{N_a}{w} \rho^{2w} (1-\rho)^{N_a - w} \\
&= \binom{N_a}{w} \rho^{2w} (1-\rho^2)^{N_a - w},
\end{aligned}$$

where we used the binomial theorem in the second equality. This tells us that the $W_{pp'}$ are Binomially distributed with parameters (N_a, ρ^2), consistent with (A.1).

A.2. Derivation of the distribution of distances

We can use these conditional probabilities (3.9) and (3.10) to compute the probability of finding two nodes at distance $\delta(p, p')$ from each other, or in other words the probability that an entry in the weight matrix takes a given value. We find that this quantity is $\mathrm{Bin}(N_a, \rho - \rho^2)$ distributed.
Using (3.9) we obtain the probability of finding a random node with distance d to a

given target node p' as

$$\begin{aligned}
P(\delta(p,p')=d) &= \sum_{k'=0}^{N_a} P\left(\delta(p,p')=d|q(p')=k'\right) P(q(p')=k') \\
&= \sum_{k'=0}^{N_a} \binom{k'}{d}(1-\rho)^d \rho^{k'-d} \binom{N_a}{k'} \rho^{k'}(1-\rho)^{N_a-k'} \\
&= \sum_{k'=0}^{N_a} \frac{k'!}{d!(k'-d)!} \cdot \frac{N_a!}{k'!(N_a-k')!} \rho^{2k'-d}(1-\rho)^{N_a-k'+d} \\
&= \sum_{k'=0}^{N_a} \frac{N_a!}{d!(N_a-d)!} \cdot \frac{N_a-d!}{(k'-d)!(N_a-k')!} \rho^{2k'-d}(1-\rho)^{N_a-k'+d} \\
&= \binom{N_a}{d} \rho^d (1-\rho)^d \sum_{l=0}^{N_a-d} \binom{N_a-d}{l} \rho^{2l}(1-\rho)^{N_a-d-l} \\
&= \binom{N_a}{d} \rho^d (1-\rho)^d \cdot (\rho^2+1-\rho)^{N_a-d} \\
&= \binom{N_a}{d} (\rho-\rho^2)^d (1-(\rho-\rho^2))^{N_a-d}
\end{aligned}$$

where we used $l = k - d \geq 0$ and the binomial theorem. Thus we find a $\text{Bin}(N_a, \rho - \rho^2)$ distribution, which gives the probability that a random entry in the distance matrix equals d. We verify this result by computing it using (3.10), finding

$$\begin{aligned}
P(\delta(p,p')=d) &= \sum_{k=0}^{N_a} P\left(\delta(p,p')=d|q(p)=k\right) P(q(p)=k) \qquad \text{(A.2)} \\
&= \sum_{k=0}^{N_a} \binom{N_a-k}{d} \rho^d (1-\rho)^{N_a-k-d} \binom{N_a}{k} \rho^k (1-\rho)^{N_a-k} \\
&= \binom{N_a}{d} \rho^d (1-\rho)^d \sum_{k=0}^{N_a-d} \binom{N_a-d}{k} \rho^k (1-\rho)^{2N_a-2k-2d} \\
&= \binom{N_a}{d} (\rho-\rho^2)^d \cdot (\rho+(1-\rho)^2)^{N_a-d} \\
&= \binom{N_a}{d} (\rho-\rho^2)^d (1-(\rho-\rho^2))^{N_a-d}.
\end{aligned}$$

Here we used that $k \leq N_a - d$ and the binomial theorem.

A.3. Derivation of distribution of up-edges

Here we derive the probability of finding an up-edge in the network. Furthermore we assume $\alpha = 1$, such that products are only connected if $\delta(p,p') \leq 1$.

Firstly, note that an up-edge never occurs between nodes with distance 0 since the target node can have at most one capability requirement that the source node does not have, which means that it must also have all capabilities the source node has in order to have a higher complexity. In other words, in order to go to higher complexity products one must learn a new capability.
So an up-edge only occurs for $\delta(p,p')=1$. We call $\zeta(k)$ the probability that a random node p with complexity $q(p)=k$ has an out-edge pointing to a higher complexity node:

$$\zeta_{out}(k) = P(\delta(p,p')=1, q(p')>q(p)|q(p)=k).$$

This equals the probability of finding a product p' such that it has exactly one nonzero entry $T_{ap'}=1$ in the N_a-k entries where $T_{ap}=0$ and furthermore has k nonzero entries $T_{ap'}=1$ for all a where $T_{ap}=1$, so that $q(p)<q(p')$. So the probability of an up-edge given $q(p)=k$ is

$$\begin{aligned}\zeta_{out}(k) &= P(\delta(p,p')=1, q(p')=k+1|q(p)=k)\\ &= \rho^k\binom{N_a-k}{1}\rho(1-\rho)^{N_a-k-1}\\ &= (N_a-k)\rho^{k+1}(1-\rho)^{N_a-k-1}.\end{aligned}$$

Thus the probability that a random node has an outgoing up-edge is given by

$$\begin{aligned}\zeta &= \sum_{k=0}^{N_a}\zeta_{out}(k)P(q(p)=k)\\ &= \sum_{k=0}^{N_a}(N_a-k)\rho^{k+1}(1-\rho)^{N_a-k-1}\binom{N_a}{k}\rho^k(1-\rho)^{N_a-k}\\ &= \sum_{k=0}^{N_a}(N_a-k)\binom{N_a-1}{k}\rho^{2k+1}(1-\rho)^{2N_a-2k-1}\\ &= \sum_{k+1=0}^{N_a}(k+1)\binom{N_a}{k+1}\rho^{2k+1}(1-\rho)^{2N_a-2k-1}\\ &= \sum_{l=0}^{N_a}l\binom{N_a}{l}\rho^{2l-1}(1-\rho)^{2N_a-2l+1}.\end{aligned}$$

To check, we also compute the probability that a random node has an incoming up-edge (coming from a lower complexity node). This is given by

$$\begin{aligned}\zeta_{in}(k') &= P(\delta(p,p')=1, q(p)=k'-1|q(p')=k')\\ &= \binom{k'}{k'-1}\rho^{k'-1}(1-\rho)(1-\rho)^{N_a-k'}\\ &= k'\rho^{k'-1}(1-\rho)^{N_a-k'+1}.\end{aligned}$$

Again summing over all possible k' we get

$$\begin{aligned}\sum_{k'=0}^{N_a} \zeta_{in}(k') P(q(p') = k') &= \sum_{k=0}^{N_a} k' \rho^{k'-1} (1-\rho)^{N_a - k' + 1} \binom{N_a}{k'} \rho^{k'} (1-\rho)^{N_a - k'} \\ &= \sum_{k'=0}^{N_a} k' \binom{N_a}{k'} \rho^{2k'-1} (1-\rho)^{2N_a - 2k' + 1} \\ &= \zeta\end{aligned}$$

which gives the desired result.
The quantity ζ probability that a random node has an up-edge, or the fraction of all possible edges that will have an up-edge. This distribution is shown in Figure 3.5. The quantity $(N_p - 1) \cdot \zeta$ gives the expected number of up-edges from a random node. The quantity $N_p \cdot (N_p - 1) \cdot \zeta$ gives the expected number of up-edges in the network.

A.4. Derivation of percolation condition

The mean in-degree in given by

$$\begin{aligned}\langle j \rangle = \mathbb{E}[d_{in}^{\alpha}(p')] &= \sum_{p \neq p'} \mathbb{E}[\mathbf{1}_{\{\delta(p,p') \leq \alpha\}}] \\ &= (N_p - 1) P(\delta(p, p') \leq \alpha).\end{aligned}$$

We know the distribution of $\delta(p, p')$ is $\text{Bin}(N_a, \rho - \rho^2)$. So

$$\langle j \rangle = (N_p - 1)((1 - (\rho - \rho^2))^{N_a} + N_a(\rho - \rho^2)(1 - (\rho - \rho^2))^{N_a - 1}).$$

Likewise, for the mean out-degree $\langle l \rangle$ we have

$$\begin{aligned}\langle l \rangle = \mathbb{E}[d_{out}^{\alpha}(p)] &= (N_p - 1) P(\delta(p, p') \leq \alpha) \\ &= (N_p - 1)((1 - (\rho - \rho^2))^{N_a} + N_a(\rho - \rho^2)(1 - (\rho - \rho^2))^{N_a - 1})\end{aligned}$$

For the expectation of the product we use conditional independence:

$$\begin{aligned}
\langle jl \rangle &= \sum_{j=0}^{N_p-1} \sum_{l=0}^{N_p-1} jl P(d_{in} = j, d_{out} = l) \\
&= \sum_{j=0}^{N_p-1} \sum_{l=0}^{N_p-1} jl \sum_{k=0}^{N_a} P(d_{in}^{\alpha} = j | q(p) = k) P(d_{out}^{\alpha} = l | q(p) = k) P(q(p) = k) \\
&= \sum_{k=0}^{N_a} \sum_{l=0}^{N_p-1} l P(d_{out}^{\alpha} = l | q(p) = k) \sum_{j=0}^{N_p-1} j P(d_{in}^{\alpha} = j | q(p) = k) P(q(p) = k) \\
&= \sum_{k=0}^{N_a} \sum_{l=0}^{N_p-1} l P(d_{out}^{\alpha} = l | q(p) = k) \mathbb{E}[d_{in}^{\alpha}(p) | q(p) = k] P(q(p) = k) \\
&= \sum_{k=0}^{N_a} \sum_{l=0}^{N_p-1} l P(d_{out}^{\alpha} = l | q(p) = k) (N_p - 1) \eta_{in}(k, \alpha) P(q(p) = k) \\
&= (N_p - 1)^2 \sum_{k=0}^{N_a} \eta_{in}(k, \alpha) \eta_{out}(k, \alpha) P(q(p) = k).
\end{aligned}$$

The condition for a giant in- or out- component to exist is given by [17]

$$2\langle jl \rangle - \langle j \rangle \cdot \langle l \rangle \geq 0,$$

leading to the model-specific condition

$$(N_p - 1) \sum_{k=0}^{N_a} \eta_{in}(k, \alpha) \eta_{out}(k, \alpha) P(q(p) = k) \geq P(\delta(p, p') \leq \alpha).$$