IMA Journal of Complex Networks (2021) Page 1 of 13
doi:10.1093/comnet/xxx000

Pearson Correlations on Complex Networks

MICHELE COSCIA*
IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen, DK
*Corresponding author: mcos@itu.dk

[Received on 20 September 2021]

Complex networks are useful tools to understand propagation events like epidemics, word-of-mouth, adoption of habits, and innovations. Estimating the correlation between two processes happening on the same network is therefore an important problem with a number of applications. However, at present there is no way to do so: current methods either correlate a network with itself, a single process with the network structure, or calculate a network distance between two processes. In this paper, we propose to extend the Pearson correlation coefficient to work on complex networks. Given two vectors, we define a function that uses the topology of the network to return a correlation coefficient. We show that our formulation is intuitive and returns the expected values in a number of scenarios. We also demonstrate how the classical Pearson correlation coefficient is unable to do so. We conclude the paper with two case studies, showcasing how our network correlation can facilitate tasks in social network analysis and economics. We provide examples of how we could use our network correlation to infer user characteristics from their activities on social media; and relationships between industrial products, under some assumptions as to what should make two exporting countries similar.

Keywords: Network Correlation, Node vectors, Correlation, Pearson, Network Economics

1. Introduction

Complex networks have been widely used in the literature to study a number of propagation phenomena. Classical results include the description of how diseases move through relations in a social network [8], the maximization of the impact of a viral marketing campaign [28, 31], the adoption of behaviors in a population [19, 37], and the diffusion of new productive activities in economic regions [34, 38]. In all these cases, one could represent the spreading phenomenon as a node field of activation levels: each node has a value proportional to how much it is being affected by the event.

It is a natural question to ask oneself: if we have two events propagating on the same network, how related are they? Specifically: are they affected by the network structure in the same way? Do they obey the same propagation rules? One part of the answers to these questions is the estimation of the correlation between the two events. Clearly, if the two node activation vectors are strongly (anti)correlated, then we have a strong prior for the existence of a relation between them. Such correlation would behave like the classical Pearson linear correlation, but also taking into account the topology of the underlying network.

In practice, in this paper we want to explore methods to solve the network correlation problem, i.e. to calculate a vector-vector correlation in a non-Euclidean space defined by a network. Unfortunately, in the literature there currently is no way to calculate such a correlation. There are a number of senses in which "network correlation" is intended, but none of them captures the objective we have just sketched.

The first interpretation of "network correlation" is as a global property of a network [6]. The idea is that the static structural properties of a network are not the only important thing: nodes also activate

over time and their auto-correlations are an important property of the network. Thus, this is a network property, which does not take node vectors as an input. This approach has been used to study propagation of signals in a network [22], estimating the network's resilience to cascade failures [17], and applied to fields such as psychology [14].

Another network correlation sense takes two networks as an input and estimates the correlation between their topologies [27]: is an edge in one network more or less likely to appear if it appears in another network? The same formulation has been applied to layers of a multilayer network [35], or to different scales of the same network [18].

A closer formulation of the problem takes as input a network and a vector. Given the vector's values, the question is estimating the likelihood that the phenomenon described by the vector really propagated through the edges of the network [20]. One can see that this is not a network correlation in the sense we intend in the present paper, as that would require estimating the correlation between two vectors over a network, rather than between a vector and a network.

The node vector distance class of problems [9, 10] accepts the same input as we do: two node vectors and a network. However, it focuses on estimating the *distance* between two vectors on the network rather than their *correlation*. As such, they are as related to our network correlation problem as the Euclidean distance is related to the Pearson correlation. Moreover, all of those methods are only defined to work on a network with a single connected component, which is an unacceptable limitation if we want to estimate network correlations, as we will see in the paper.

The closest related work in network science is an attempt to define the concept of network variance [13]. In this work, the authors extend the notion of variance to include network relationships between elements of a vector in a similar fashion to what we do in this paper. Since they define network variance, they can also derive a measure of network co-variance, which they sketch in the paper without further investigation. Whether the network co-variance definition that the authors derive from their variance is compatible with our definition of network correlation is something to be investigated in future works.

Finally one should not confuse "network correlation" with "correlation networks": networks whose edges represent linear correlations between vectors – each vector being a node [5, 7, 16, 30].

In this paper, we propose to extend the Pearson correlation coefficient to work on complex networks. Given two vectors, we define a function that uses the topology of the network to return a correlation coefficient. Specifically, we assume that two nodes influence each other proportionally to the inverse of their network distance. We estimate the network distance as the length of the shortest paths in number of edges. We propose an exponential distance decay: nodes at each hop away from node v influence the value of v exponentially less.

Our correlation coefficient is closely related to Moran's I measure of spatial autocorrelation [33]. In fact, it is equivalent to Moran's I if one were to ignore a constant scaling coefficient, define a suitable spatial kernel to describe the network's topology – which is not compatible with how Moran's I is normally defined –, and use a proper measure of network variance rather than the simple variance used in Moran's I.

We provide some validation examples to show that the measure behaves as expected in simple scenarios: perfect correlations for vectors that are linear combination of each other, and zero correlation for uniformly random node vectors. Additionally, we show how the classical Pearson correlation is blind to simple local network propagation vectors, which are instead captured by our network-aware correlation.

Our network correlation measure could be used in all analytic scenarios we mentioned in the first paragraph, and more. To show it, we apply it to a couple of application scenarios: relating various user characteristics on social media and estimating the similarity between the export baskets of countries. To estimate export basket similarities we should take into account the fact that some pairs of products

are more similar than others. Our network correlation can consider this information, while Pearson correlation cannot. Moreover, our network correlation could be used to improve the quality of the Product Space itself, a feat that cannot be performed with a plain Pearson correlation.

We provide the network correlation code as part of an open source library[1]. The data and the code necessary to replicate our results is also freely available[2].

2. Network Correlation

2.1 *Problem Definition*

Let $G = (V, E)$ be a graph, where V is the set of nodes and $E \subseteq V \times V$ the set of edges. For simplicity, we assume that G is unweighted, i.e. $(u, v) \in E$, with no weights. For convenience, we also define N_v as the set of v's neighbors, i.e. all nodes directly connected to v in G: $N_v = \{u : (u, v) \in E\}$.

We are given two vectors x and y. Both x and y are vectors of length $|V|$. One could interpret them as containing the activation values of each node $v \in V$. For this paper, we assume that x and y can take any real value, i.e. $x \subseteq V \times \mathbb{R}$, meaning that that each entry could be positive, zero, or negative.

We want to define a function $\rho_{x,y,G}$, telling us the correlation between x and y, while accounting for the structure in G. $\rho_{x,y,G}$ should have the same domain as the Pearson correlation coefficient, meaning that it is equal to -1 for perfectly anti-correlated vectors, 1 for perfectly correlated vectors, and 0 for non-correlated vectors.

Accounting for the structure in G means that each entry x_v in x relates not only to the corresponding y_v entry in y – as it would in the Pearson correlation – but also on all the entries in y in N_v, and then the neighbors' neighbors, and so on. There should be a distance decay in the network: the farther a node is from v in G, the less it matters for the correlation.

2.2 *Solution*

Intuitively, this problem can be solved by realizing that it is a generalization of the Pearson correlation coefficient – which is a measure of linear correlation between two vectors [24]. The sketch of the solution is that Pearson only tests for linear correlations in an Euclidean space where all dimensions contribute equally to the correlation, while our ρ_G function lives in a more complex space determined by the topology of G.

Thus, it is useful to start from the formula of the sample-based Pearson correlation coefficient:

$$\rho_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}. \tag{2.1}$$

Let us add some useful notation. Hereafter, $\hat{x} = (x - \bar{x})$ is the centered version of x, and $\sigma_x = \sqrt{\sum_{i=1}^{n}\hat{x}^2}$ is x's standard deviation. We can now rewrite the previous equation as:

$$\rho_{x,y} = \frac{\text{sum}(I \times (\hat{x} \otimes \hat{y}))}{\sigma_x \sigma_y}, \tag{2.2}$$

where I is the identity matrix, $\times$ is the elementwise product operation, $\otimes$ is the outer product operation, and sum is an operator that returns the sum of all cells of a matrix. The outer product of two

[1] https://www.michelecoscia.com/?page_id=1733

[2] https://www.michelecoscia.com/?page_id=1733#nvdcorr

vectors of the same length is a square matrix, whose i,j-th entry is the product of x's ith entry and y's jth. The sum of the main diagonal of the outer product is exactly equal to $\sum_{i=1}^{n} \hat{x}_i \hat{y}_i$, which is why we need to multiply elementwise the result of $\hat{x} \otimes \hat{y}$ by the identity matrix I, which keeps the main diagonal discarding the rest.

In the same spirit, we can also rewrite the formula for standard deviation as $\sigma_x = \sqrt{\text{sum}(I \times (\hat{x} \otimes \hat{x}))}$. The reason why we want to do so is to highlight how the Pearson coefficient is a special case of the network correlation coefficient. Specifically, it is a network correlation coefficient for a network composed by isolated nodes – represented by the identity matrix I. We can replace I with a suitably defined weight matrix W. Each entry of W tells us how related two nodes are in the network, with one being maximum relation and zero no relation at all.

Of course, there are many viable Ws, and in this paper we only focus on one. Specifically, we assume that two nodes are related proportionally to their shortest path distance in the network. We calculate L as the matrix telling us such pairwise distance. Then, we can have $W = e^{-L}$. We exponentiate L for several reasons. First, L's main diagonal is zeroes, which implies that W's main diagonal is of ones. Second, if two nodes are on different connected components then their distance is infinite, which means that the corresponding entry in W is equal to zero. This maps well on the Pearson correlation coefficient, because it reduces W to I in the case of a network exclusively composed by isolated nodes.

The final reason to exponentiate is that the contribution of progressively far away nodes decays quickly as the distance grows, which is preferable over a linear decay. The number of nodes at an increasing number of hops away grows exponentially on most network topologies, thus the influence a node can exert over another greatly diminishes with distance. Of course, one could take $W = e^{-kL}$, adding a free parameter k regulating such decay. A reasonable alternative to our W would involve using the effective resistance between two nodes [12], which has a number advantages over L, for instance it is more stable (values in L can vary greatly with the addition/deletion of a single edge).

To sum up, the network correlation distance can be calculated – assuming $k = 1$ – as:

$$\rho_{x,y,G} = \frac{\text{sum}(W \times (\hat{x} \otimes \hat{y}))}{\sigma_{x,W}\sigma_{y,W}}, \tag{2.3}$$

in which also the standard deviation needs to be calculated over the network as $\sigma_{x,W} = \sqrt{\text{sum}(W \times (\hat{x} \otimes \hat{x}))}$. Without using our contracted notation and allowing for an arbitrary parameter k, Equation 2.3 expands to:

$$\rho_{x,y,G} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} e^{-kl_{ij}}(x_i - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} e^{-kl_{ij}}(x_i - \bar{x})(x_j - \bar{x})}\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} e^{-kl_{ij}}(y_i - \bar{y})(y_j - \bar{y})}}, \tag{2.4}$$

with l_{ij} being the i,j-th element of L, i.e. the length of the path between nodes i and j.

Given that $\rho_{x,y,G}$ is a special transformation of Moran's I, there is a way to derive it from Moran's formulation. We do so in Supporting Information (SI) Section 1.

3. Validation

3.1 *Basic*

The operation of calculating a correlation over a complex network is hard to picture. The first thing we need to do is to test in extremely simple scenarios whether the measure is doing what we think it is doing. The first thing to do is to test when ρ_G is equal to one when we think it should – for linear

combinations of a vector –; and to zero when it should – for independent and uniformly random vectors. In both of these cases ρ_G behaves as expected – SI Section 3 contains more information about the tests we ran.

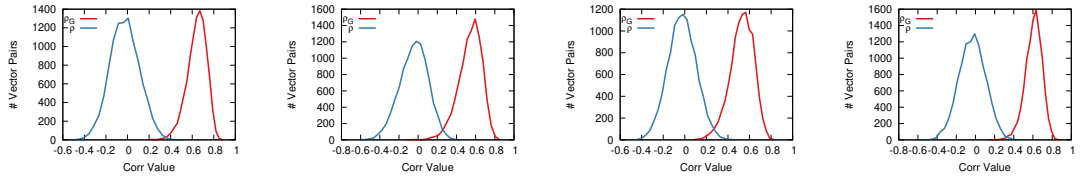
These two tests show that ρ_G is a proper extension of ρ: the network's topology does not affect extreme cases where the value of ρ_G is expected. However, they also fail to show any difference between ρ_G and ρ: in all cases tested so far, the two measures behave exactly the same.

There is an easy way to show the difference between ρ_G and ρ. Suppose that we start from a uniform random node vector x. Then, we generate each entry for node v in y as:

$$y_v = \sum_{u \in N_v} x_u / |N_v|, \tag{3.1}$$

where N_v is the set of neighbors of v. In practice, y is the network neighbor average of x.

Clearly, x and y are uncorrelated from the perspective of ρ, since N_v does not include v itself. On the other hand, x and y are strongly correlated in a network sense, because connected nodes have overlaps in their neighborhoods, while nodes at the opposite ends of a network do not.



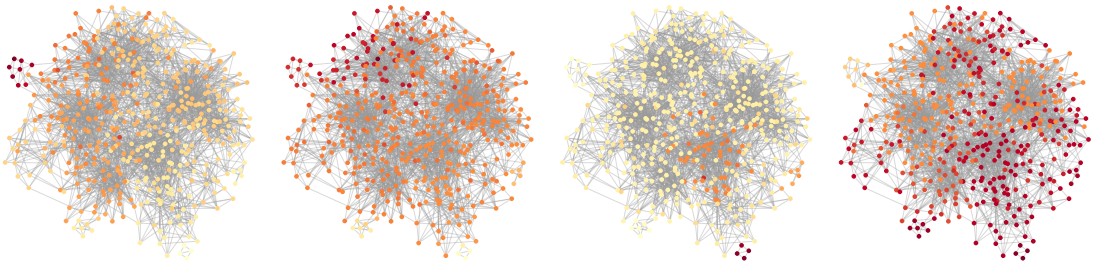
(a) $G_{n,m}$ (b) Small World. (c) Barabasi-Albert. (d) SBM.

FIG. 1: The number of vector pairs (y axis) generating vectors with a given correlation value (x axis) for ρ_G (red) and ρ (blue) on different network topologies.

Figure 1 compares the ρ_G and ρ values over $10,000$ random vector initiations on a number of network topologies (all initialized to have 100 nodes): $G_{n,m}$ (GNM) random graphs [15] (Figure 1(a)), Watts-Strogatz (WS) small world model [36] (Figure 1(b)), Barabasi-Albert (BA) [4] (Figure 1(c)), and Stochastic BlockModel (SBM) [25] (Figure 1(d)). From the figure, we can see that indeed ρ is blind to these local correlations and centers on zero – plus/minus some expected random fluctuations. On the other hand, ρ_G successfully captures the local network correlations, being significantly shifted from zero.

3.2 *Advanced*

In this section we provide a more advanced validation with more complex settings. In this test, we assume that there is one source node s_1 emitting into the network. At time $t = 0$, its weight in the node vector x is equal to one, while all other entries are equal to zero. At each timestep, each node – except s_1 – updates its value in the vector with the average value of its neighbors. The source node's value is always equal to one. We stop after 30 timesteps. This procedure is related to the calculation of the electric potential on the nodes. The result is something similar to Figure 2(a) – on an LFR benchmark [29] with 500 nodes and $2,710$ edges.



(a) (b) (c) (d)

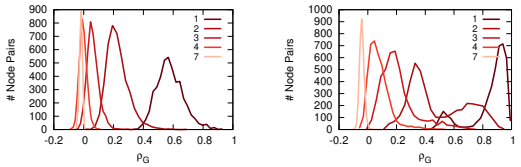
FIG. 2: The values of different node vectors x after the emission process described in the text, with different starting points. The node's color represents the value in x, from dark red (high) to bright yellow (low). (a) Origin in the top-left small cluster. (b) Positive correlation with (a). (c) No correlation with (a). (d) Negative correlation with (a).

Intuitively, we can get a strong positive network correlation by running the same process from a different source s_2 that is nearby s_1 in the network. Figure 2(b) shows the result of picking a node that is at two hops from the source in Figure 2(a). The two vectors have a $\rho_G = 0.295$.

We can also generate a correlation close to zero, by running the same process from a source s_3 that is far from s_1. Figure 2(c) shows the result of picking a node that is at seven hops from the source in Figure 2(a). The two vectors have a $\rho_G = 0.034$.

Following the same reasoning, we can also define an anti-correlation. This would require to run a different process: rather than having an emitting node, we have a sink node. All nodes have an entry of one in the vector, except the sink which has zero. We iteratively update each node's entry with the average of their neighbors, except the sink which stays at zero. To get a proper network anti-correlation, we need to again pick a sink s_4 that is close to s_1. Figure 2(d) shows the result of this process, bearing $\rho_G = -0.904$ with the vector from Figure 2(a).

From this example, it is clear that ρ_G contains information about the relative location of the source nodes. This means that, by looking at ρ_G for two sources s_1 and s_2, we can guess their shortest path distance. However, this is true only to a point: as the distance between s_1 and s_2 increases, the ρ_G of the vectors they generate tends to zero. We can show this by plotting the distribution of ρ_G for node pairs at increasing distances. For each distance, we sample 4,000 random node pairs at that distance in a GNM random graph with $n = 1500$ and $m = 4250$. Figure 3(a) shows such distribution.



(a) $G_{n,m}$ (b) Small World.

FIG. 3: The number of node pairs (y axis) generating vectors with a given ρ_G value (x axis) at different shortest path distance levels (line color).

From the figure we can see that it is easy to distinguish between pairs of nodes at one, two, and three hops away. From four hops to seven – close to be the diameter of the network –, there is little to no difference in the ρ_G observed. Note that the underlying topology of the network can greatly affect these distributions. To show it, we run the same test on a WS model with the same number of nodes and edges – and probability of rewiring set to 0.15.

Figure 3(b) shows the distribution, which departs from Figure 3(a) in many respects. First, due to the diameter being larger, it is somewhat possible to distinguish sources at four and seven hops away. Second, there are many more node pairs generating a correlation close to one. Finally, other classes of node pairs appear. Both the ρ_G distributions for direct neighbors and sources at two hops away are bimodal. This is because a certain fraction of these nodes have one of their connections rewired as a shortcut, dramatically affecting the result of the propagation process.

We now turn to a comparison with the non-network Pearson correlation ρ. Many of the results presented so far can be replicated without taking the network into account. To understand what ρ_G adds to ρ we perform the same single-source-emission process on a variety of networks. To the GNM, LFR, WS, BA, and SBM network models used before we add the Power-Cluster (PC) [26] model. We compare ρ_G and ρ in their predictions of the shortest path distance between the source nodes. To do so, we sample an equal number of node pairs at each distance level.

We use two measures of quality: accuracy and mean absolute error. We use a generalized linear model with a Poisson link to translate ρ_G and ρ into a shortest path length estimate. We use such model because it is more appropriate than an OLS for discrete positive outcomes – like a path length. As accuracy, we count the number of times the model returns the correct path length over all attempts. As mean absolute error, we average the absolute difference between the model's guess and the actual path length. To contextualize these numbers, we also show the result of a random predictor, whose accuracy is by definition one over the diameter of the network.

Topology	ρ_G Acc	ρ Acc	Rnd Acc	ρ_G Err	ρ Err	Rnd Err
GNM	**0.449**	0.312	0.143	**0.197**	0.258	0.646
WS	**0.423**	0.313	0.143	**0.164**	0.226	0.574
BA	**0.409**	0.338	0.167	**0.221**	0.266	0.581
PC	**0.406**	0.299	0.167	**0.226**	0.279	0.561
SBM	**0.387**	0.328	0.200	**0.258**	0.264	0.530
LFR	**0.371**	0.354	0.167	**0.254**	0.265	0.579

Table 1: The accuracy (left panel) and mean absolute error (right panel) of ρ_G, ρ, and a random guess when predicting the shortest path length between two sources of a propagation event.

Table 1 shows the result. As we can see, ρ_G is a more accurate and precise estimate of the path length than ρ. This shows that taking the network's topology into account when calculating the correlation is useful. Moreover, this is true across all the different network models we test. ρ_G has an advantage over ρ of around ten percentage point on average.

From Figure 3 we know that, in this specific experiment setup, ρ_G finds it difficult to distinguish sources at four or more hops. Thus, we should expect to see a performance increase if we were to treat all source pairs at four or more hops as a single class. We see this performance improvement in the result table in SI Section 4.

4. Applications

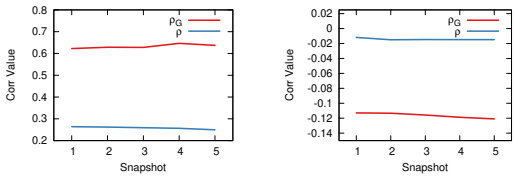
We look at two possible applications of our network correlation measure in two fields: social network analysis (Section 4.1) and economics (Sections 4.2 and 4.3).

4.1 *Social Network Analysis*

Many social networks have quantitative attributes on nodes. Evaluating correlations on such attributes might lead to interesting insights. To make an example, we focus on the Anobii social network, which is focused on reading. Each user in Anobii has a bookshelf with the books they read, a wishlist with the books they want to read, and can tag books. Among the information they provide the platform, they can state their gender and age.

We access five temporal snapshots of the social network, using data from a study of Anobii [1]. The snapshots are taken at 15 days intervals, starting from Sept 11 2009 until Dec 24 2009. To make sure we only look at engaged users, we select users that have specified their gender, age, have at least one book in their shelf and wishlist, and are part of the largest connected component of the 2-core of the network (i.e. they have at least two friends who also have at least two friends). This results in a network that goes from 10k to 12k nodes and from 30k to 34k edges, from the first to the last snapshot, respectively.

We then test network and non-network correlations among all the metadata attached to the nodes. Figure 4 shows the evolution of correlations in two interesting cases.



(a) (b)

FIG. 4: Correlations and network correlations (y axis) across snapshots (x axis) between: (a) Size of bookshelf and wishlist, and (b) Age and number of tags attached to a book.

Figure 4(a) is the correlation between the number of books read and the number of books in the wishlist of each user. We expect strong readers to also have longer wishlists. In fact, the regular correlation between these two variables is positive and significant at around 0.25. However, we would also expect "power users" with high activity to cluster up in the core of the network. Thus, we expect the network location of a user to be indicative of their level of activity. In fact, the network correlation ρ_G of shelf and wishlist size is much stronger, around 0.63.

Figure 4(b) is the correlation between the age of a user and the number of tags they attached to books on the platform. One reasonable hypothesis could be that older users are less inclined to use many tags, as this level of online engagement is more typical of younger people. The Pearson correlation would reject this hypothesis: there is no significant correlation between age and number of tags ($\rho = -0.015$, $p = 0.12$). However, there is a significant negative correlation if we take into account the network structure ($\rho_G = -0.12$) confirming the hypothesis. What this means is that it is not possible to predict the age of a user by looking the number of tags they use, but it is possible if we also look at the number of tags used by their friends – and, to a lesser extent, the friends of their friends.

This simplified analysis shows that the network correlation can be used to gain interesting insights about a social network, either boosting an already present signal or by disentangling an interesting correlation that would otherwise go unnoticed.

4.2 *Finding Related Exporters*

We now look at an application of network correlation to economics. A researcher could be interested in knowing which countries are similar because they export similar products. Simply correlating their export baskets would not work, because different products might share few or many capabilities to be produced, and thus contribute differently to the correlation. This is the principle of relatedness [23], which has been applied to develop the Product Space [21]: a network of products connected together if a significant number of countries can co-export them.

We use data from Comtrade in the 2011-2014 period, only including countries whose export and import traffic was larger than eight billion dollars, resulting in 135 countries. For this analysis, we use product codes under the SITC classification with four digits. We select products whose trade traffic was at least five billion dollars, resulting in 631 products.

We build a Product Space by counting the number of co-exporting countries for each pair of products, and then selecting only the edges that pass a statistical significance test according to the noise-corrected backboning process [11]. We set our threshold to 2.2, which can be interpreted as the z-score of the edge weight – roughly equivalent to a one-tailed p-value of 0.014.

We use the Product Space as our network space G, which provides our weights W. If products i and j are very similar, they are directly connected in the Product Spaces and thus $w_{ij} = e^{-1}$. Vice versa, if they are very dissimilar they might be n hops away and thus $w_{ij} = e^{-n}$.

We now calculate the pairwise correlation among all countries in the dataset. Each exporter provides a node vector whose entries are the Revealed Comparative Advantage [3] (RCA) value for each product. If y_{cp} is the amount country c exported of product p, then:

$$RCA_{cp} = \frac{y_{cp}/y_{c\cdot}}{y_{\cdot p}/y_{\cdot\cdot}}, \tag{4.1}$$

with $y_{c\cdot} = \sum_p y_{cp}$, $y_{\cdot p} = \sum_c y_{cp}$, and $y_{\cdot\cdot} = \sum_{c,p} y_{cp}$. We use RCA rather than export value because products have different inherent values which might create spurious correlations if not normalized via RCA.

We evaluate how well we estimate the actual country-country correlation by using the geographical distance and the trade traffic between the countries. Economics literature tells us that neighbors should be more correlated with each other [2], thus we expect to find a strong negative correlation between ρ_G and geographical distance. Common sense tells us that, if two countries can both export p, then they do not need to import it from each other – or at all. Thus we also expect a negative correlation between ρ_G and the trade volume between two countries.

Table 2 shows the result of a linear multivariate OLS model between the correlation value and the distance and trade volume between each pair of countries. We find the expected significant negative correlations: the export baskets of far away countries and of countries with strong trade relationships are indeed correlated less strongly than neighbors and countries with weak trade links.

The R^2 tells us how accurate ρ_G and ρ are in identifying neighbors and trade partners. ρ_G (column 1) has a higher R^2 than ρ (column 2). This implies that the information about product relationships in the Product Space is informative if we want to approximate relations between exporters. ρ_G allows us to use it, while ρ does not.

	Dependent variable:	
	ρ_G	ρ
	(1)	(2)
Geo Dist	-0.066^{***}	-0.032^{***}
	(0.004)	(0.001)
Trade	-0.008^{***}	-0.002^{***}
	(0.0002)	(0.0001)
Constant	0.885^{***}	0.332^{***}
	(0.034)	(0.012)
Observations	8,128	8,128
R^2	0.124	0.091
Adjusted R^2	0.123	0.090
Residual Std. Error	0.257	0.087
F Statistic	572.514^{***}	404.890^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01	

Table 2: The relation between ρ_G, ρ, and the geographical distance and the trade volume between each pair of countries. For each variable we report the regression coefficient (top row) and the standard error (bottom row between parenthesis).

SI Section 5 explores the level of agreement between ρ_G and ρ, highlighting how ρ_G is better able to take values in the whole -1 to $+1$ range, which is a possible reason for its ability of returning more precise predictions.

4.3 *Discovering the Latent Network*

The quality of a network correlation is only as good as the network itself. If the Product Space we build does not match the real relations among products, then any found correlation would not be meaningful. However, we could use ρ_G to construct the Product Space directly – under a certain set of assumptions. If we assume that our expectation of a negative relationship between ρ_G and geographical distance and trade volume is correct, we can rewire the Product Space to maximize the R^2. In other words, now we are not interested any more in finding correlations between countries, but we want to discover the latent network connecting products.

We do so via a simple simulated annealing process. We start from the Product Space built with the same procedure we outlined in the previous section. The only difference is that we use the SITC classification at the two digit level, to reduce the number of nodes in the network to 66. This is necessary to reduce the size of the search space, given that our simple simulated annealing is quite inefficient and used for demonstrating purposes only.

At each annealing step, we calculate the same OLS regression we show in Table 2. Then, we pick up a random pair of products. If the two products are connected, we remove the connection. If they are not, we connect them. If the edge addition/deletion increases the R^2 of the regression we are likely to keep it. At the end of the process, we obtain a G', which is an optimized version of G. Note that G' and G will have a different number of edges thus, to keep the comparison fair, we add/remove edges from G until it has the same density as G'. The edges are added/removed according to the same noise corrected backbone generating the original G.

Table 3 shows the performance comparison between ρ_G, ρ, and $\rho_{G'}$ in columns 1, 2, and 3, respectively. From column 1 we see that ρ_G has an R^2 of 0.093 – which is lower than the one it has in Table

	Dependent variable:		
	ρ_G	ρ	$\rho_{G'}$
	(1)	(2)	(3)
Geo Dist	-0.072^{***}	-0.052^{***}	-0.116^{***}
	(0.004)	(0.003)	(0.003)
Trade	-0.008^{***}	-0.003^{***}	-0.010^{***}
	(0.0003)	(0.0002)	(0.0002)
Constant	0.933^{***}	0.557^{***}	1.378^{***}
	(0.040)	(0.024)	(0.031)
Observations	8,128	8,128	8,128
R^2	0.093	0.061	0.239
Adjusted R^2	0.093	0.060	0.239
Residual Std. Error	0.303	0.182	0.232
F Statistic	417.018^{***}	261.620^{***}	$1{,}275.107^{***}$
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 3: The relation between ρ_G, ρ, $\rho_{G'}$, and the geographical distance and the trade volume between each pair of countries. For each variable we report the regression coefficient (top row) and the standard error (bottom row between parenthesis).

2, because the Product Space at the two digit level is more coarse (and, thus, less performant) than the Product Space at the four digit level. Nevertheless, ρ_G still outperforms ρ, which has an R^2 of only 0.061 (column 2). As expected, our optimization procedure was able to generate a much higher R^2 than either alternative: column 3 shows that $\rho_{G'}$ has an R^2 of 0.239, more than double performance compared to ρ_G.

From this analysis we can conclude that G' includes product-product relationships that are more accurate than the ones in G, assuming that they should be used to identify as related pairs of exporters which are neighbors and do not trade as much as expected. Thus, our network correlation could be used to inform the construction of more accurate Product Spaces.

5. Conclusion

In this paper we have extended the Pearson linear correlation to the case of node activation states in a network. In practice, we have considered the case in which the dimensions of two vectors are not all independent from each other and equally important, but they are coupled in a network topology. This is useful because it allows us to understand whether two events unfolding on a network are related to each other via the network's connections. Our case studies show that possible application scenarios involve describing user characteristics on social media and unveiling interesting patterns in the network of global trade. Such tasks cannot be performed by a normal Pearson correlation, as our experiments show: Pearson is blind to local propagation effects.

This network correlation could be used in a number of future developments. For instance, inspired by the classical correlation distance, it could be at the basis for a new Node Vector Distance. Its advantage over the alternatives is that, differently from all methods proposed so far in the literature, our network correlation is well-defined for networks with more than one connected component. Other potential future works involve the investigation of the distance decay parameter, which here we fixed as $k = 1$, but this is by no means the only reasonable solution. Moreover, one could investigate a whole new distance decay function, moving away from our proposed exponential decay – we think effective resistance [12]

is the most promising option, although it might have issues when the graph's number of nodes/edges is high [32]. We should also involve integrating edge weights into the measure, which for simplicity we have ignored in this paper.

Acknowledgment

The author wishes to thank Clara Vandeweerdt and Andres Gomez-Lievano for insightful conversations and advice.

REFERENCES

1. Aiello, L. M., Barrat, A., Cattuto, C., Ruffo, G. & Schifanella, R. (2010) Link creation and profile alignment in the aNobii social network. In *2010 IEEE Second International Conference on Social Computing*, pages 249–256. IEEE.

2. Bahar, D., Hausmann, R. & Hidalgo, C. A. (2014) Neighbors and the evolution of the comparative advantage of nations: Evidence of international knowledge diffusion?. *Journal of International Economics*, **92**(1), 111–123.

3. Balassa, B. (1965) Trade liberalisation and "revealed" comparative advantage 1. *The manchester school*, **33**(2), 99–123.

4. Barabási, A.-L. & Albert, R. (1999) Emergence of scaling in random networks. *science*, **286**(5439), 509–512.

5. Barnett, I. & Onnela, J.-P. (2016) Change point detection in correlation networks. *Scientific reports*, **6**(1), 1–11.

6. Barzel, B. & Biham, O. (2009) Quantifying the connectivity of a network: the network correlation function method. *Physical Review E*, **80**(4), 046104.

7. Bazzi, M., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J. & Howison, S. D. (2016) Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, **14**(1), 1–41.

8. Colizza, V., Barrat, A., Barthélemy, M. & Vespignani, A. (2006) The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences*, **103**(7), 2015–2020.

9. Coscia, M. (2020) Generalized Euclidean Measure to Estimate Network Distances. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 119–129.

10. Coscia, M., Gomez-Lievano, A., Mcnerney, J. & Neffke, F. (2020) The Node Vector Distance Problem in Complex Networks. *ACM Computing Surveys (CSUR)*, **53**(6), 1–27.

11. Coscia, M. & Neffke, F. M. (2017) Network backboning with noisy data. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 425–436. IEEE.

12. Devriendt, K. (2020) Effective resistance is more than distance: Laplacians, Simplices and the Schur complement. *arXiv preprint arXiv:2010.04521*.

13. Devriendt, K., Martin-Gutierrez, S. & Lambiotte, R. (2020) Variance and covariance of distributions on graphs. *arXiv preprint arXiv:2008.09155*.

14. Epskamp, S., Borsboom, D. & Fried, E. I. (2018) Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, **50**(1), 195–212.

15. Erdős, P. & Rényi, A. (1960) On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, **5**(1), 17–60.

16. Friedman, J. & Alm, E. J. (2012) Inferring correlation networks from genomic survey data. *PLoS Comput Biol*, **8**(9), e1002687.

17. Gao, J., Barzel, B. & Barabási, A.-L. (2016) Universal resilience patterns in complex networks. *Nature*, **530**(7590), 307–312.

18. Gosak, M., Markovič, R., Dolenšek, J., Rupnik, M. S., Marhl, M., Stožer, A. & Perc, M. (2018) Network science of biological systems at different scales: A review. *Physics of life reviews*, **24**, 118–135.

19. Granovetter, M. (1978) Threshold models of collective behavior. *American journal of sociology*, **83**(6), 1420–1443.

20. Guan, Z., Wu, J., Zhang, Q., Singh, A. & Yan, X. (2011) Assessing and ranking structural correlations in graphs. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 937–948.

21. Hausmann, R., Hidalgo, C. A., Bustos, S., Coscia, M. & Simoes, A. (2014) *The atlas of economic complexity: Mapping paths to prosperity*. Mit Press.

22. Hens, C., Harush, U., Haber, S., Cohen, R. & Barzel, B. (2019) Spatiotemporal signal propagation in complex networks. *Nature Physics*, **15**(4), 403–412.

23. Hidalgo, C. A., Balland, P.-A., Boschma, R., Delgado, M., Feldman, M., Frenken, K., Glaeser, E., He, C., Kogler, D. F., Morrison, A. et al. (2018) The principle of relatedness. In *International conference on complex systems*, pages 451–457. Springer.

24. Hogg, R. V., McKean, J. & Craig, A. T. (2005) *Introduction to mathematical statistics*. Pearson Education.

25. Holland, P. W., Laskey, K. B. & Leinhardt, S. (1983) Stochastic blockmodels: First steps. *Social networks*, **5**(2), 109–137.

26. Holme, P. & Kim, B. J. (2002) Growing scale-free networks with tunable clustering. *Physical review E*, **65**(2), 026107.

27. Huang, S., Zhang, J., Wang, L. & Hua, X.-S. (2015) Social friend recommendation based on multiple network correlation. *IEEE transactions on multimedia*, **18**(2), 287–299.

28. Kempe, D., Kleinberg, J. & Tardos, É. (2003) Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146.

29. Lancichinetti, A., Fortunato, S. & Radicchi, F. (2008) Benchmark graphs for testing community detection algorithms. *Physical review E*, **78**(4), 046110.

30. Langfelder, P. & Horvath, S. (2008) WGCNA: an R package for weighted correlation network analysis. *BMC bioinformatics*, **9**(1), 1–13.

31. Leskovec, J., Adamic, L. A. & Huberman, B. A. (2007) The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, **1**(1), 39.

32. Luxburg, U., Radl, A. & Hein, M. (2010) Getting lost in space: Large sample analysis of the resistance distance. *Advances in Neural Information Processing Systems*, **23**, 2622–2630.

33. Moran, P. A. (1950) Notes on continuous stochastic phenomena. *Biometrika*, **37**(1/2), 17–23.

34. Neffke, F., Hartog, M., Boschma, R. & Henning, M. (2018) Agents of structural change: The role of firms and entrepreneurs in regional diversification. *Economic Geography*, **94**(1), 23–48.

35. Nicosia, V. & Latora, V. (2015) Measuring and modeling correlations in multiplex networks. *Physical Review E*, **92**(3), 032805.

36. Watts, D. J. & Strogatz, S. H. (1998) Collective dynamics of 'small-world'networks. *nature*, **393**(6684), 440–442.

37. Weeks, M. R., Zhan, W., Li, J., Hilario, H., Abbott, M. & Medina, Z. (2015) Female condom use and adoption among men and women in a general low-income urban US population. *AIDS and Behavior*, **19**(9), 1642–1654.

38. Youn, H., Strumsky, D., Bettencourt, L. M. & Lobo, J. (2015) Invention as a combinatorial process: evidence from US patents. *Journal of the Royal Society interface*, **12**(106), 20150272.