Proceedings of the Fourteenth International AAAI Conference on Web and Social Media (ICWSM 2020)

Generalized Euclidean Measure to Estimate Network Distances

Michele Coscia[1]
[1]IT University of Copenhagen
Rued Langgaards Vej 7
Copenhagen, DK 2300
mcos@itu.dk

Abstract

Estimating the distance covered by a propagation phenomenon on a network is an important task: it can help us estimating the infectiousness of a disease or the effectiveness of an online viral marketing campaign. However, so far the only way to make such an estimate relies on solving the optimal transportation problem, or by adapting graph signal processing techniques. Such solutions are either inefficient, because they require solving a complex optimization problem; or fragile, because they were not designed with this problem in mind. In this paper, we propose a new generalized Euclidean approach to estimate distances between weighted groups of nodes in a network. We do so by adapting the Mahalanobis distance, incorporating the graph's topology via the pseudoinverse of its Laplacian. In experiments we see that this measure returns intuitive distances which agree with the ones a human would estimate. We also show that the measure is able to recover the infection parameter in an epidemic model, or the activation threshold in a cascade model. We conclude by showing that the measure can be used in online social media settings to identify fast-spreading behaviors. Our measure is also less computationally expensive.

Introduction

Network analysis has emerged as a versatile tool for analyzing complex phenomena in the real world. Applications include the detection of groups in social systems (Fortunato 2010), the prediction of future connections (Lü and Zhou 2011), and the description of emerging properties of complex systems (Barabási and Bonabeau 2003). Of particular interest for this paper, two complementary tasks in network analysis are the modeling of the diffusion of diseases in social networks (Colizza et al. 2006) and the planning of viral marketing campaigns on social media (Leskovec, Adamic, and Huberman 2007). Both cases can be represented with the same model: nodes in the network transition between two states, infected and not infected. In the former case we want to minimize the number of people infected by a disease, in the latter case we want to maximize the number of users converted into customers by the campaign.

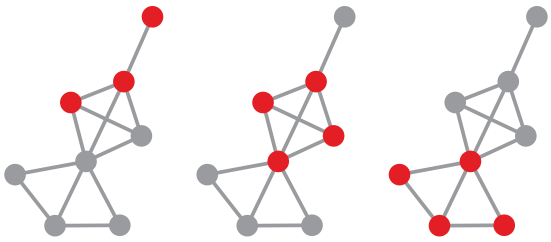


Figure 1: Different activation states of a network. Active nodes in red and inactive nodes in gray.

In this paper, we focus on an aspect of these problems that has hitherto not received much attention. In network epidemics, one is usually interested in modeling the evolution of the system at large: how many nodes are infected at which point in the disease's history? What are the best immunization strategies to prevent a global outbreak? In viral marketing, one defines complex contagion rules and tries to find the smallest possible seed set of initially infected nodes such that, once the campaign is over, the maximum possible number of users will be converted into customers.

Here, we take an outsider perspective. We take the epidemic / viral marketing as an unfolding event – or one that has already reached its final state –, with no possibility of intervention. We are interested in estimating how quickly the spreading event is passing through the network. In other words, given an initial and a final state of infected nodes, we want to estimate the distance between the two states.

Consider Figure 1. In Figure 1(a) we have a possible initial state of the network, with some nodes affected by a campaign and others which are not. Figure 1(b) and 1(c) are two possible outcomes of word of mouth. Which of the two results is the farthest from the initial condition?

In practice, we can consider the initial and final states as vectors. We want to calculate a spatial distance between these vectors. The trivial solution would be to estimate their Euclidean distance. However, these vectors do not live on an Euclidean space. The shape of the space is complex, and it is defined by the topology of the network. We need to ex-

tend the Euclidean distance to be applicable to a network topology. We do so by creating a new measure inspired by the Mahalanobis distance (Mahalanobis 1936), in which we use information coming from the graph Laplacian. We focus on the graph Laplacian due to its relationship with diffusion processes (Coifman and Lafon 2006).

To the best of our knowledge, this is the first time that the network state distance problem is presented in this specific framing. There are other approaches in the computer science literature that can be adapted to estimate some sort of network state distance. Two examples are the earth mover distance in computer vision (Rubner, Tomasi, and Guibas 2000) and the field of signal processing on graphs (Shuman et al. 2013). However, we show in the paper that our problem is a more general version of such alternatives, and thus calls for a different approach.

Solving this problem has a number of potential applications. In network epidemics, estimating the distance between two time steps in the infection propagation allows us to compute the speed of infection. This can be used to estimate the infectiousness parameters of a previously unknown disease, with fewer a priori assumptions. In online viral marketing, knowing the distance covered by previous campaigns allows us to identify which one was more successful at spreading on larger distances on the network, which could inform future campaigns. If one is reconstructing a network from indirect observations – e.g. projecting a bipartite network (Coscia and Rossi 2019) – with our measure they could benchmark the quality of their inferred topology: since the spreading should follow the topology, shorter distances imply a better alignment between the topology and the actual spreading phenomenon. Finally, this measure could be used to evaluate the temporal granularity with which we are observing a spreading phenomenon. If the disease jumps over large distances between temporal snapshots, this could mean that the temporal granularity of the observation should be increased.

In the experimental section, we validate our choice of measure by showing intuitive spreading events, in which the measure's behavior matches the distance that a human would estimate. Then, we show how our measure is able to inform us on each of the analytic scenarios we presented in the previous paragraph, by means of synthetic testing. We also show the behavior of the measure in a real world scenario using an online social media. Finally, we show that estimating spreading distances on a network with our method is not computationally demanding, allowing us to process networks of moderate size. In all cases, we show how our measure compares favorably with alternatives defined for related tasks and adapted to fit our problem definition.

The main contributions of the paper are the following: (i) We introduce the problem of estimating the distance of vectors on a network topology, a generalization of the earth mover distance problem in computer science; (ii) We connect such problem with applications in network epidemics and viral marketing; (iii) We propose a scalable solution, which matches the human intuition of distances on simple spaces; (iv) We perform extensive experiments showing the usefulness of such measure in different applications.

We release our code as a public open source library that anyone can use to solve the vector distance problem on networks[1]. The archive also contains the code necessary for the replication of our experiments.

Related Work

In this paper we focus on the problem of establishing the distance between two node occupancy vectors on a network. Note that here we look at changes of vectors in an unchanging network topology, thus approaches estimating the distance between two networks (Galas et al. 2017) are not applicable. Node vectors have been used to describe epidemics (Colizza et al. 2006), viral marketing (Kempe, Kleinberg, and Tardos 2003) (Leskovec, Adamic, and Huberman 2007) (Pennacchioli et al. 2013), infrastructure loads (Barrat, Barthelemy, and Vespignani 2008), transportation (Banavar et al. 2000), and more. However, in the data science and physics network literature, the problem of estimating the distance between two such vectors has rarely been tackled.

Estimating the distance between two vectors is a well studied and understood problem. There exist many solutions to it, ranging from simple linear correlations to more sophisticated distance measures like cosine or Mahalanobis distances (Mahalanobis 1936). The problem with these approaches is that they do not account for an underlying network structure. A large element-wise difference between portions of these vectors might be a small change, because the nodes they represent are clustered in the network. Vice versa, small differences should be amplified if they refer to nodes that are far from each other in the graph topology.

The closest related literature to this paper is the one on the optimal transportation problem (OTP). In its original formulation (Monge 1781), it still focuses on the distance between two probability distributions without an underlying network. However, it has been observed how this problem can be applied to transportation through an infrastructure, known as the multi-commodity network flow (Hitchcock 1941). In its most general form, the assumption is that we have a distribution of weights on the network's nodes, and we want to estimate the minimal number of edge crossings we have to perform to transform the origin distribution in the destination one. This is a complex problem, which has lead to an extensive search for efficient approximations (Erbar et al. 2017) (Pele and Werman 2009). All these approaches are interchangeable here, because they aim to more efficiently calculate the same measure, and we are only interested in the measure itself.

To the best of our knowledge, OTP on graphs has been mostly studied in the context of computer vision (Rubner, Tomasi, and Guibas 2000). OTP is similar to, but not the same as, the problem in this paper. OTP, as the name says, is an optimization problem. Optimality implies that the entire network structure is used to determine the most efficient way to transport the node weights. Here we reject such constraint. By doing so, we can estimate the distances in more computationally efficient ways.

Another closely related literature is the one on signal processing in graphs (Shuman et al. 2013). In this scenario,

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

the nodes of a network are assumed to be sensors capturing an underlying signal. The structure of the network is used to represent interdependences and/or correlations between these sensors. Given the observed data, represented as node weights, one wants to design localized transformation methods that account for the structure of the data domain. After the proper transformation is applied, we can represent graph signals as independent from the graph's topology, i.e. embedded in the "true" space. A popular approach to the problem is to perform graph spectroscopy (Hammond, Vandergheynst, and Gribonval 2011) by calculating the eigenvectors of the graph Laplacian, also known as the "graph Fourier transform". These approaches can be used to establish the distance between two different signals on a graph, as we show later.

Problem Definition

Let $G = (V, E)$ be a graph, where V represents the set of nodes and $E \subseteq V \times V$ the set of edges. In this paper, we consider an undirected unweighted graph. Undirected means that, if $(u, v) \in E$ is an edge, then $(u, v) = (v, u)$ – with $u, v \in V$. In principle, extensions of our approach to directed weighted graphs should be trivial. With A we indicate the adjacency matrix of G, with $A_{uv} = 1$ if $(u, v) \in E$, zero otherwise. Since the graph does not contain self loops, the diagonal of A is equal to zero.

Our problem is dynamic: we are observing the status of the system at different moments in time. We use t_i to refer to the time step i. In our problem definition, the topology of the graph is static. For any $i \neq j$, $G = G_{t_i} = G_{t_j}$, meaning that the sets of nodes and edges are the same.

Let us assume that there exists a function f which takes as input a time step t_i and a graph G. The function returns a vector of length $|V|$, which represents the activation state of the nodes in the graph. f can represent any real world phenomenon affecting the nodes of a network. For instance, in a social network, f could have non-zero elements to indicate the nodes currently affected by a disease. For simplicity, in many tests we will add a constraint: f returns relative activation states, i.e. $\sum f(t_i, G) = 1$. However this is not a strict requirement for our framework.

f returns different activation states at different times, or $f(t_i, G) \neq f(t_j, G)$. In our example, it means that the people affected by the disease at time j might be different from the infected set at time i. Specifically, some people might have contracted the disease from their neighbors, while others might have recovered. Informally, in this paper we want to define a distance measure that can estimate how much $f(t_i, G)$ differs from $f(t_j, G)$. How quickly does the disease spread and do individuals recover in the network?

Formally, our problem definition is:

Definition 1 *Given a graph $G = (V, E)$ and a function f determining the activation state of the nodes of G at time t, define a metric $\delta(f(t_i, G), f(t_j, G))$, which takes as input two node vectors and returns their distance calculated using G's topology.*

Note that we want δ to be a metric, thus it has to satisfy the defining characteristics of a proper metric:

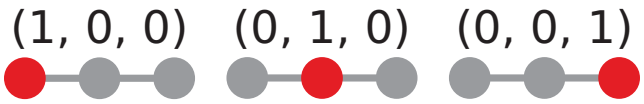


Figure 2: Three points in a three dimensional space, in their vector form and in their representation on a chain graph.

1. Non-negativity, meaning that if $f(t_i, G) \neq f(t_j, G)$ then $\delta(f(t_i, G), f(t_j, G)) > 0$. Comparing two distinct node vectors will always result in a non-zero distance.
2. Identity of indiscernibles: $\delta(f(t_i, G), f(t_i, G)) = 0$. If we are comparing a vector to itself – or to another vector identical to itself –, we expect it to have a distance of zero.
3. Symmetry, which implies that: $\delta(f(t_i, G), f(t_j, G)) = \delta(f(t_j, G), f(t_i, G))$. The distance between two vectors is the same regardless which vector we consider as origin and which we consider as destination.
4. Triangle inequality, meaning: $\delta(f(t_i, G), f(t_k, G)) \leq \delta(f(t_i, G), f(t_j, G)) + \delta(f(t_j, G), f(t_k, G))$, assuming $i \neq j \neq k$. We want δ to be a true metric, where the space is defined by the topology of the network.

Methods

Generalized Euclidean

The core issue in this paper is estimating the distance between two vectors, a and b. The most obvious choice is assuming that the vectors live in an n-dimensional Euclidean space. The number of dimensions is the length of the vector which, in our case, is the number of nodes in the network: $|V|$. Then, one can simply calculate the Euclidean distance between the two points identified by the vectors: $\delta(a, b) = \sqrt{(a-b)^T(a-b)}$. In this formula, $(a - b)$ is the element-wise difference of the vectors a and b, while $(a - b)^T$ is its transpose.

The problem with the Euclidean distance is that each dimension contributes equally to the spatial distance between the points. In a network, this is not the case. Since each dimension is a node in the network, some dimensions contribute less to the distance than others. If two vectors only differ along two dimensions, it makes a difference whether the two corresponding nodes are connected or not. In Figure 2, the middle and the right vectors are equidistant from the left vector if we use the Euclidean formula – their distance is $\sqrt{2}$. However, on the graph topology, the rightmost vector should be farther from the left vector than the middle vector, as the two nodes are farther from each other.

The Mahalanobis distance solves the problem of differential contribution to the total distance by different dimensions (Mahalanobis 1936). In the Mahalanobis distance, we multiply the squared vector difference by the inverse of the vectors' covariance matrix S: $\delta(a, b) = \sqrt{(a-b)^T S^{-1}(a-b)}$. The interpretation is that some dimensions are correlated with each other and thus contain less unique information than others. Therefore, each of them should contribute less to the overall distance. S only depends

on the vectors we are comparing, thus it also ignores G's topology, as does the Euclidean distance.

In this paper we propose to replace the covariance matrix S with a matrix Q which contains the graph's topological information. One constraint we have to respect is that Q needs to be positive (semi)definite, otherwise the x^TQx product could be negative for some vector x, which would result in a nonsensical distance estimation. For this reason, we cannot use the adjacency matrix of the graph, which is not positive semidefinite – unless the graph is empty.

We focus on the graph Laplacian L, due to its relationship with diffusion processes (Coifman and Lafon 2006). The intuition behind the use of the graph Laplacian is that it can be interpreted as a matrix representation of a particular case of the discrete Laplace operator. This means that we can use the graph Laplacian to describe the heat exchange between nodes until we reach an equilibrium. If f is the function assigning the heat to each node, then:

$$\frac{df}{dt} = -kLf,$$

where k is the heat capacity. In other words, the change df at each discrete interval of time dt is regulated by L. Thus, we can see the node vector distance as the process of transferring the heat from the origin to the destination nodes, and the graph Laplacian is what regulates such exchange.

The graph Laplacian L is the degree matrix D (a matrix with the node's degree on the main diagonal and zeros everywhere) minus the adjacency matrix A: $L = D - A$. The smallest eigenvalue of L is zero, making L positive semidefinite. However, L represents relations between nodes: it tells us how much heat flows from a node to another. We need the opposite: a measure of the distance between them. Thus we would need to have $Q = L^{-1}$. This is not possible, because L is singular and singular matrices cannot be inverted. We can approximate L^{-1} by calculating L's Moore-Penrose pseudoinverse L^+.

If $Q_1\Sigma Q_2^T = L$ is the single value decomposition of the Laplacian matrix L, then $Q_2\Sigma^+Q_1^T = L^+$ is its Moore-Penrose pseudoinverse. Here, Σ is a diagonal matrix containing L's singular values, the solution of L's singular value decomposition problem (SVD). Σ^+ is the diagonal matrix containing the reciprocals of A's singular values. It holds that $LL^+L = L$ and that $L^+LL^+ = L^+$.

Thus, if $f(t_i, G)$ and $f(t_j, G)$ represent the two node vectors of which we want to estimate the distance, our proposal for the δ function is:

$$\delta(f(t_i,G), f(t_j,G), G) = \sqrt{(f(t_i,G) - f(t_j,G))^T L^+ (f(t_i,G) - f(t_j,G))}, \quad (1)$$

with L^+ being the pseudoinverse of the Laplacian of G.

This δ is a proper metric. It is non-negative, because L^+ is positive semidefinite and thus $x^TL^+x \geq 0$ no matter x. It respects the identity of indiscernibles, as $0^TL^+0 = 0$ no matter what L^+ is. It is symmetric, as $(a-b)^TL^+(a-b) = (b-a)^TL^+(b-a)$. It also inherits the triangle inequality property from the Mahalanobis form by means of the Cauchy-Schwarz inequality.

When it comes to computational complexity, the core of the method is the computation of the pseudoinverse of the Laplacian, which is in turn dominated by the complexity of solving the SVD problem. Since L is a $|V| \times |V|$ square matrix, the time complexity of solving SVD – and, therefore, estimating the generalized Euclidean distance – is $\mathcal{O}(|V|^3)$. While this is a hefty price to pay, as we show in the last part of the Experiments section this cost has to be paid only once per network: the pseudoinverse can be cached to solve an arbitrary number of distance estimations between any node vectors on that network.

Alternative Approaches

In this paper, we compare our generalized Euclidean (GE) approach with two alternatives: Earth Mover Distance (EMD) and the Graph Fourier Transform approach (GFT).

Earth Mover Distance EMD is a way to solve the optimal transportation problem. In practice, EMD is trying to minimize the number of edge crossings to transport all the weights from $f(t_i, G)$ to $f(t_j, G)$, and returning the number of such edge crossings. More formally, in EMD we want to find a set of movements M such that:

$$M = \arg\min_{m_{u,v}} \sum_u \sum_v m_{u,v} d_{u,v},$$

where u and v are the weighted entries of $f(t_i, G)$ and $f(t_j, G)$, respectively; $m_{u,v}$ is the amount of weights from u that we transport into v; and $d_{u,v}$ is the distance between them (more on this below). Then:

$$EMD(f(t_i,G), f(t_j,G)) = \frac{\sum_u \sum_v m_{u,v} d_{u,v}}{\sum_u \sum_v m_{u,v}},$$

where the $m_{u,v}$ movements come from the M we found at the previous step. Finding the optimal M is hard and approximations exist in the literature. Here we use the one[2] formulated by Pele and Werman (Pele and Werman 2008; 2009). The thing we are left to determine in the EMD formula is the distance function $d_{u,v}$ between pairs of nodes. In this paper, we choose this to be the length of the shortest path between u and v. This is zero if $u = v$.

Graph Fourier Transform Suppose that we have a signal s on a graph, which is an activation pattern of its nodes. In this scenario, each node is a sensor and edges express dependencies between sensors – i.e. their results are correlated. Thus, we should expect the true signal $\hat{s}$ to be distorted by such correlations. The aim of the Graph Fourier Transform is to reconstruct the original signal. This is achieved by the following operation: $\hat{s} = \Phi s$.

Here, Φ is the matrix of generalized eigenvectors of L, the graph Laplacian of G. If $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_n$ are the sorted eigenvalues of L and $l_0, l_1, \ldots, l_n$ are the corresponding eigenvectors, then $\Phi = (l_0, l_1, \ldots, l_n)$. Once we reconstruct the true signal $\hat{s}$, we can filter it so that we take into

[2] https://github.com/wmayner/pyemd

account the topology of the graph. This is usually achieved by filtering the signal in the spectral domain, multiplying it with the diagonal matrix of the Laplacian's eigenvectors (Λ). This is the Laplace operator.

Putting this together, we have that:

$$\hat{f}(t_i, G) = \Lambda \Phi^T f(t_i, G).$$

Once we apply this transformation to both $f(t_i, G)$ and $f(t_j, G)$, we have encoded G's topology in the vectors. The Euclidean distance between $\hat{f}(t_i, G)$ and $\hat{f}(t_j, G)$ is the node vector distance that we are looking for.

Experiments

In this section we test our approach on a number of dimensions. First, we ask whether these measures make intuitive sense. Second, we use them to recover salient characteristics of different network processes we can observe (epidemics, viral marketing campaigns, etc). Then, we perform a case study on real world data. Finally, we test their scalability.

A word of warning when interpreting the results. We expect the GE approach we propose to perform better than the spectrum GFT method, because it is better tailored to the actual problem definition. However, we do not expect GE to outperform the EMD approach. In fact, we expect the opposite: EMD is an optimization approach and thus allows for more precise solutions. The reason why we propose and prefer generalized Euclidean over EMD is in its promising scalability, which is an improvement over EMD.

Intuitiveness

We start with synthetic network tests. The objective is to create simple networks. In these networks, a human would be able to tell which vector pairs are more distant from each other than another pair of vectors. We then compare the results from each measure with our expectation.

Chain Test The first test we run is the simplest possible: the chain test. In the chain test we create chain graphs of progressive lengths. The first vector occupies one end of the chain and the second vector occupies the other end. Clearly, the longer the chain the more the two vectors are distant from each other. Figure 3 shows a depiction of this setup.

We calculate the distance between the vectors for growing chain sizes: we start with a chain from two to one hundred nodes. Figure 4(a) shows the result for all the measures we consider. The Euclidean distance (in gray), as we know, ignores the graph's topology and considers all vectors to be at the same distance from each other: $\sqrt{2}$.

Our generalized Euclidean measure (in red), instead, grows sub-linearly with the chain's length. In fact it scales as $\sqrt{|E|}$, giving it a rather intuitive meaning: in this test, the distance between the vectors is the square root of the number of edges you need to cross to go from one to the other. GFT (in blue) has a different profile: it quickly grows as small chains become slightly less small, but the relative difference between a chain of length 90 with one of length 100 is negligible. The EMD approach has instead a perfectly linear relationship with the number of edges.

Random Chain The chain test is intuitive for a human, but it might be too simple to benchmark a distance measure. With the random chain test we aim at maintaining the intuitive aspect of the test, introducing random fluctuations.

In this test, we generate a random network by specifying a number n of columns. Each column contains 10 nodes. Nodes in column i can only connect to nodes in column $i+1$ and $i-1$, with the exception of the first and last columns, which only connect to one column. For each column pair, we extract 20 random edges. We vary n from 2 to 21 columns.

For each network of n columns, we set the source vector as occupying the first column and the target vector as occupying the nth (last) column. The expectation would be that, in networks with fewer columns, the source and target vectors are closer to each other than in networks with more columns. We thus expect a positive correlation between number of columns and covered distances.

Figure 4(b) reports the results. Since the networks are random, we repeat the experiment ten times and we report the average and standard deviation. The figure shows that the generalized Euclidean approach behaves as it should, with increasing distances for increasing column counts. The GFT approach, on the other hand, has only a mild correlation with the number of columns. This proves that it is not robust enough when the topology of the network becomes more complex. EMD, like generalized Euclidean, has a tight relationship with what a human would expect.

Small World A harder test uses small world random networks following the Watts-Strogatz model (Watts and Strogatz 1998). In this model, nodes are placed in a low-dimensional space at regular distances. Each node is connected to its k nearest neighbors, creating a regular lattice. Then, with a random probability p, an edge can be rewired so that it will connect two random nodes. One can see that, if $p = 0$, the network has long shortest path lengths, because the paths need to traverse the regular lattice. As p grows, more and more random shortcuts are added, decreasing the average shortest path length.

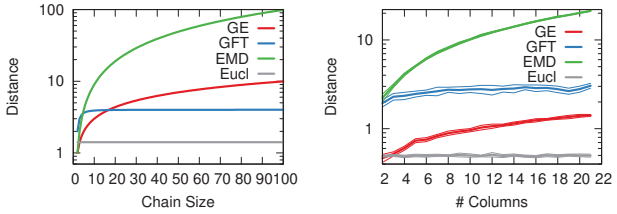
Since nodes are placed and connect to each other based on their position in a space, we can set our source and target vectors to be at the antipodes of this space. Then, intuitively, the lower the average path length in the network the closer the two vectors are. Thus we can plot the distance between the two vectors against the average path length of the network and expect to find a positive correlation.

Figure 5 shows this relationship. We can see that our generalized Euclidean has a tight relationship with the average path length in the network, while the GFT spectrum approach can distinguish between a high and low path lengths, but with (i) a non-linear relationship, and (ii) failing for very long path lengths. The EMD approach has also a tight relation with the average path length, but not as tight as the generalized Euclidean.

Scaling Vectors So far, for the sake of intuition, we assumed that all tested vectors sum to the same value: one. In other words, the $f(t, G)$ vectors are normalized: they represent relative activation states. This was done not to conflate changes in *position* on the network with changes in *intensity*

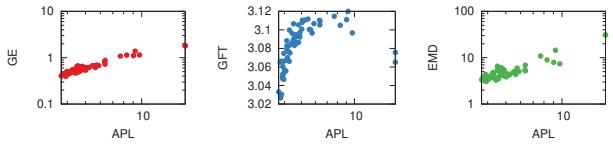
(a) Short distance. (b) Medium distance. (c) Long distance.

Figure 3: The expectation of the chain test. Origin node in red, destination node in green. As the chain gets longer, we expect the distance measure to return higher values.



(a) (b)

Figure 4: The distance measure behavior (y axis) for increasing: (a) chain sizes (in number of nodes); and (b) column counts.



(a) (b) (c)

Figure 5: The relationship between the distance between vectors at the antipodes of a Watts-Strogatz model (y axis) and its average shortest path length (x axis) for: (a) GE, (b) the GFT spectrum approach, and (c) EMD.

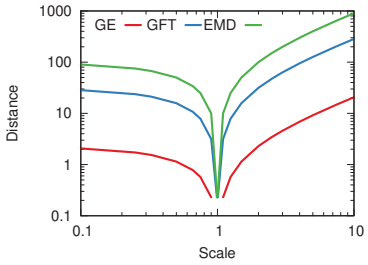


Figure 6: The distance measure behavior (y axis) for different c scaling factors of the original vector (x axis).

Method	Chain	Rnd Chain	SW	Scale
GE	1.00	0.98	0.93	1.00
Spectrum	1.00	0.55	0.53	1.00
EMD	1.00	0.99	0.77	1.00
Euclidean	N/A	-0.05	N/A	N/A

Table 1: The Spearman correlation between the distance values obtained with alternative measures and the intuitive human expectation, explained in the text, for various tests.

in the signal. However, whether the distance measure is able to properly capture changes in scale is an important property for many applications. For instance, in viral marketing, the value on a specific node could be the absolute level of engagement of the user in the campaign.

In this test we create a vector on a network and we scale it by a factor of c which can be smaller or larger than one: $f(t_2, G) = c \times f(t_1, G)$. Here, for simplicity, G is a Erdos-Renyi random graph. We then track the distance values as c changes (Figure 6). We can see that all measures are well behaved. The more we shrink or expand the original vector, the further the origin and destination move apart. The generalized Euclidean approach is less sensitive to shrinking/expanding. Whether this is a desirable property or not is left as a consideration on a case by case basis.

To sum up the results for all the tests conducted so far, we calculate the Spearman rank correlation between expectation and measurements, and report the results in Table 1. The table tells us the relationship between the intuitive distance expectation between two vectors and the distance as measured with different techniques. Recall that the expectations are as follows: (i) chain – the longer the chain, the most distant its origin-destination nodes are; (ii) random chain – the longer the chain, the most distant its origin-destination node columns are; (iii) small world – the higher p, the more shortcuts we add, thus the closer origin-destination nodes are; (iv) scaling – the farther from 1 the scaling factor is, the farther origin-destination nodes are.

GE manages to consistently match our expectation, while GFT is more erratic. It works well in simple cases, but breaks down as the test networks become more and more complex. EMD is more consistent than GFT, but it has its own failure scenarios.

Potential Applications

Estimating Infectivity In the first application scenario, we look at simple contagion. This is an SI model of an epidemic outburst scenario. In an SI model, nodes can be in two states: Susceptible (S) and Infected (I). If a susceptible node has at least an infected neighbor, it will transition to the infected state with probability p_{si}.

This is simple contagion because nodes need no reinforcement to be infected: a single contact is sufficient. The idea is that we can recover the infectiousness p_{si} of the disease by monitoring how quickly it spreads through the network. The farther the disease spreads, the more infectious it is.

We create a series of networks with 100 nodes and roughly 500 edges, with different topologies: Erdos-Renyi (ER), Barabasi-Albert Scale Free (SF), Power-law Clustered

Topology	GE	GFT	EMD
ER	0.4581***	0.3107***	0.6678***
SF	0.4568***	0.2465***	0.6145***
PC	0.4670***	0.2264***	0.6288***
CM	0.5712***	0.3974***	0.5992***

Table 2: The Spearman correlation coefficients (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$) with the infection parameter in the SI model, for all types of random networks. All differences between coefficients are significant – estimated via bootstrapping.

(PC), and CaveMan (CM). We pick 10 random nodes in the networks. We set p_{si} to a uniform random value between 0 and 1. We perform five steps in the SI model. We then compare the vector distance between the initial state with the final state. We repeat the process approximately $1,000$ times per network topology. We then correlate p_{si} with the start-end distance as a predictor.

Table 2 reports the results, specifically the Spearman correlation coefficient. All methods return distance estimations significantly related to p_{si} ($p < 0.001$), i.e. they are good predictors. The EMD approach has a tighter relationship with p_{si}, given by its higher correlation coefficient, but GE is not far behind. We can conclude that the generalized Euclidean is a proper predictor of infectivity in the SI model, as GFT and EMD are.

Viral Campaign Post-Mortem In this test we repeat the setup of the previous section. We have four network topologies, a contagion process, and we use the distance measures to predict the infection parameter regulating the process. The difference is that, in the previous section, we used a simple SI model. In simple contagion a node is activated with probability p_{si} if it has at least one infected neighbor and cannot revert from infected to susceptible.

Here, instead, we perform complex contagion: a cascade model. In the cascade model, a node needs at least a fraction β of active neighbors to be active. If $\beta = 0.2$ a node needs at least two out of five neighbors infected to transition to the infected state. A node with a single neighbor out of five infected will not transition. Moreover, the node will transition back into the susceptible state every time its fraction of infected neighbors falls lower than β.

This prediction task is harder because β controls two aspects of the process: the infection of susceptible nodes and the recovery of infected nodes. Thus this test shows a different facet of the same problem. It models a viral marketing campaign where sustained usage from the target audience is required for the campaign to be successful. Sustained usage comes from peer pressure.

Table 3 reports the results, specifically the Spearman correlation coefficient. We see that this task was more challenging, as the coefficient are lower. Moreover EMD fails in all but one case. Our generalized Euclidean approach was able to be the best predictor in all cases but the random graph. Thus one could use GE to examine the results of a viral marketing campaign and infer what is the level of peer pressure that sustained the cascade.

Topology	GE	GFT	EMD
ER	0.1662**	0.2629***	-0.0046
SF	0.3426***	0.2854***	0.0467
PC	0.3373***	0.2406***	0.0150
CM	0.5844***	0.3557***	0.5291***

Table 3: The Spearman correlation coefficients (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$) with the infection parameter in the cascade model, for all types of random networks. All differences between coefficients are significant – estimated via bootstrapping.

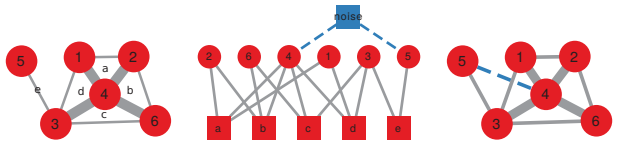


Figure 7: The procedure to generate a noisy projection. From left to right: original network with marked maximal cliques, bipartite version connecting nodes to their cliques (with a noisy element added in blue), noise projection adding the unusually weak connection to node 4 that the filtering should catch and filter even if there are non-noisy connections in the network with the same weight.

Evaluating Bipartite Projections In many common scenarios, we do not observe a network directly. We instead have entities with a collection of common attributes and we perform a network projection: we connect two entities if they are similar to each other, i.e. they have common attributes. Then, since attributes can be noisy or too common, we usually filter the network by keeping only statistically significant similarity values. This projection and filtering strategy is tricky, due to different edge generating processes and possible sources of noise. Thus there are many different ways to solve the problem (Coscia and Rossi 2019). A natural question is: how do we know whether we performed a good projection and filtering?

GE can provide an answer. Suppose we have data about memes spreading on a social network. Unfortunately we cannot observe the social relationship directly, rather people connect in a bipartite network to specific attribute values (their age, gender, etc). A network projection of people reflects some true relationship if the meme does not jump long distances over the network's topology, under the assumption that the meme "infects" people with similar characteristics. In this section we test this application scenario.

We do so by creating a ground-truth unipartite network. We detect all maximal cliques in the network, with edges being cliques of size two. We create a bipartite version of this network, connecting each node to the maximal cliques it belongs to. At this point, the bipartite network has no noise: if we were to simply project it into a unipartite one, and perform no filtering, we would obtain the original network. Thus, we add noise to the bipartite structure.

For each node in the bipartite structure we note down its average number of common cliques shared with its neigh-

Projection	Filter	EMD	GE	GFT
hyper	Disparity Filter	0.0862	0.0430	0.3690
hyper	Naive Threshold	0.1053	0.0969	0.3968
hyper	Noise-Corrected	0.0447	0.0233	0.0983
probs	Disparity Filter	0.0543	0.0257	0.2711
probs	Naive Threshold	0.0543	0.0277	0.2853
probs	Noise-Corrected	**0.0351**	0.0531	0.1095
simple	Disparity Filter	0.1085	0.0849	0.3238
simple	*Naive Threshold*	*0.4840*	*1.0057*	0.2067
simple	**Noise-Corrected**	0.0415	**0.0218**	**0.0974**
ycn	Disparity Filter	0.1798	0.1002	*0.5356*
ycn	Naive Threshold	0.1468	0.0607	0.4818
ycn	Noise-Corrected	0.1170	0.0479	0.1838

Table 4: The absolute relative error between the distance as computed on the ground truth and the one on the noisy projection, for each projection and filtering technique. Best performing combinations in bold, worst performing in italics.

bors in the unipartite structure. Then, we pick randomly a pair of nodes and we attempt to create a set of noisy bipartite edges. We connect the two nodes with half of their combined expected common neighbors count. This would generate an edge in the projection with a weaker weight than expected. Thus the noisy projection would have extra noisy edges and the filtering phase should be able to identify them, because their weight is lower than expected – although it can be higher than non-noisy edges. Figure 7 shows an example.

We then apply a collection of projecting and filtering strategies, obtaining a collection of different unipartite projections, all with the same number of edges – which is the same number of edges in the ground truth – with one exception, discussed later. We perform a simple one-edge spread process on the ground truth and we calculate its distance using our three measures (GE, GFT, EMD). Then, for each noisy projection, we record its absolute relative error with the distance calculated from the ground truth. The closer the projection was to the ground truth, the better it performed.

The bipartite projection methods we use are: simple projection, i.e. multiplying the bipartite adjacency matrix with its transpose; hyperbolic (Newman 2001); ProbS (Zhou et al. 2007); and YCN (Yildirim and Coscia 2014). The filtering strategies we use are: naive, simply filtering out edges with lowest weights; disparity filter (Serrano, Boguná, and Vespignani 2009); and noise-corrected (Coscia and Neffke 2017).

The procedure we used to generate the bipartite structure and its noise closely reflects the simple projection and the noise-corrected backboning assumptions, thus we should expect our measure to pick the simple+NC as the best performing one. On the other hand, in the simple projection with naive threshold it was impossible to find a threshold to obtain the same number of edges as the other cases. This projection contains less than half the number of edges as the other projections. Therefore, we expect the simple+naive strategy to return the highest errors.

Table 4 shows the absolute relative errors of each projection, averaged over ten repetitions of the experiment. GE and GFT pick up the simple+NC strategy as the best. EMD ranks it as second best, which is a failure of this measure. On the other hand, GE and EMD correctly spotted the worst projection (simple+naive), while GFT could not. To sum up, the only measure that identified both the best and the worst projections was our proposed metric, GE.

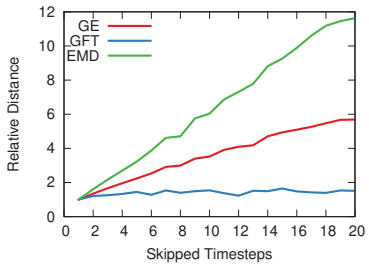


Figure 8: The distance measure behavior (y axis) for different observation gaps sizes (x axis).

Temporal Granularity When we observe a spreading phenomenon on a network we need to choose the observation frequency. If it is too often we do not observe enough change in the system. If it is too rare, the spreading event appears to jump longer distances than it actually does. We could use the methods presented in this paper to make an informed choice on the proper observation frequency.

To test this scenario, use use the random chain network model we presented earlier. We create a network with 20 columns of 10 nodes each. We start the process by occupying the first column. Then, at each natural time step, we pick 5 random nodes and we move them to the next column. We repeat the process until all nodes reach the final column.

We then introduce a skip parameter s. With $s = 1$ we observe the natural evolution of the system. With $s = 2$ we make an observation only at every other time step. With $s = 3$ we skip two steps, and so on. The real distances are the ones computed for $s = 1$. For $s > 1$, we overestimate the speed of the phenomenon. We expect a proper measure to show higher distances between snapshots as s gets larger.

Figure 8 shows the behavior of each measure as s grows. We show the ratio between the distance for $s = x$ and the one we would get for $s = 1$. The figure shows that both EMD and GE grow as we would expect: higher s values imply a higher overestimation of the average distance between observations. GFT is much flatter and thus would fail to notice that our observation interval is longer than appropriate.

Social Media Case Study

In this case study we use data coming from the Anobii website. Anobii is a social media where users keep online bookshelves of all the books they read. Anobii also has a social network feature through which users can add each other as friends. The data was collected to study the effects of anonimity and bots in social media (Aiello et al. 2010). The data contains snapshots taken at 15 days intervals, starting from Sept 11 2009 until Dec 24 2009. Anobii is particularly popular in Italy and that is why most of the examples of popular books we see in this section are Italian books.

Rank	Title	Author	GE
1	Narcissus and Goldmund	H Hesse	0.003284
2	The Old Man and the Sea	E Hemingway	0.002955
3	The Late Mattia Pascal	L Pirandello	0.002852
4	Eva Luna	I Allende	0.002794
5	I Malavoglia	G Verga	0.002790
6	Zeno's Conscience	I Svevo	0.002789
7	Two Out of Two	A De Carlo	0.002776
8	Angels & Demons	D Brown	0.002768
9	I Kill	G Faletti	0.002767
10	Reads like a novel	D Pennac	0.002736

Table 5: The longest biweekly jumps in the Anobii dataset.

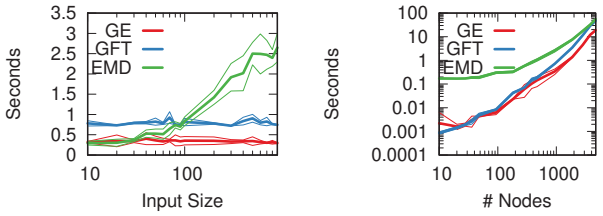
Rank	Title	Author	GE
1	Millennium	S Larsson	0.001887
2	Norwegian wood	H Murakami	0.001726
3	Breaking dawn	S Meyer	0.001691
4	Slaughterhouse-Five	K Vonnegut	0.001552
5	Narcissus and Goldmund	H Hesse	0.001538
6	Fight Club	C Palahniuk	0.001514
7	The Girl Who Played w Fire	S Larsson	0.001513
8	Eclipse	S Meyer	0.001496
9	Reads like a novel	D Pennac	0.001467
10	The Rotters' Club	J Coe	0.001425

Table 6: The top 10 books with the highest average biweekly speed in the Anobii dataset.

The social network represents the space through which books spread. A book "infects" a user when the user puts the book on their bookshelf. With GE we calculate the distance covered by the book over time. To perform this analysis we extract the k-core of the social network, setting $k = 4$: we recursively remove nodes with degree lower than 4. This is done to avoid overestimating distances covered by books mostly present in the fringes of the network. We also focus exclusively on the most popular books: books that are in at least $1,000$ bookshelves in every observed snapshot. The final network view contains $14,805$ nodes, $79,646$ edges, and we track a total of 113 books. Due to the size of the network, we are not able to report results for the EMD and GFT approaches.

We perform two analyses. First, we calculate the speed of all books from one snapshot to another and take the maximum distance covered. Table 5 shows the ten longest jumps in the network. These are the books experiencing the longest biweekly jumps. One common characteristic in this list is that it predominantly contains classics of Italian literature, or other books very popular in Italy, such as Hemingway's. Interestingly, *all* of these jumps happen in the last week of observation, which is the beginning of the school winter break in Italy, also the period in which students are more likely to have to read a classic book as holiday homework.

In the second analysis we instead calculate the average speed of the books across the entire observation period. To do so, we average all the biweekly speeds of each book. Table 6 shows the ten books with the highest average speed. One can see that there is some sort of overlap between the two lists, with notable and meaningful differences. The books with higher average speed in the final quarter of 2009



(a) (b)

Figure 9: Average and standard deviation of ten runs of the runtime of each distance measure in seconds (y axis) against: (a) the number of non-zero entries in the input vectors; and (b) the number of nodes in the input network.

are books that have a reasonable external push which might cause such drift. For instance, the movie adaptation of "The Girl who Played with Fire" came out on Sept 25 2009 in Italy, which is right at the beginning of our observation period. Interest in the movie is likely to have spurred interest in the Millenium trilogy of books written by Stieg Larsson, which in fact experienced the highest average speed and the seventh highest speed, respectively.

This case study shows that the estimation of propagation distance and speed in the network can be used to identify events. In this particular case, it could be used to estimate whether a marketing campaign is having an effect.

Time Efficiency

Here we analyze the runtime of the various distance measures on two dimensions. First, we test how much the size of the input vector affects the running time, meaning the number of non-zero entries in $f(t, G)$. Then we test the consequences of calculating the distances on larger networks, with increasing number of nodes but keeping the average degree constant. In both cases, we test on Erdos-Renyi random graphs with an average degree of three.

A note on the implementations. Both GE and the GFT are implemented using `scipy` Python standard functions. These make use of optimized linear algebra routines, which use all available CPUs. To calculate EMD, we need to calculate the lengths of all pairs of shortest path. To make the comparison fair, we naively parallelize the problem, calculating the single origin shortest path for each node separately on each available CPU. Moreover, for the Earth Mover Distance we only need to know the distances between nodes with a non-zero entry in $f(t, G)$, in some cases drastically reducing the number of shortest path calculations.

Figure 9(a) shows the runtimes for all measures, varying the number of non-zero entries in the input vectors. We set the number of nodes to $1,000$ and the number of edges to $1,500$. Neither GE nor the GFT are affected by the input size, because the computationally expensive parts of estimating these distances reside in operations that must take the whole of the graph as input. In case of GE, this is the pseudoinversion of the Laplacian. For GFT, it is calculating the Laplacian's eigenvectors, which is more expensive than pseudoinverting, hence GE is more time efficient than GFT.

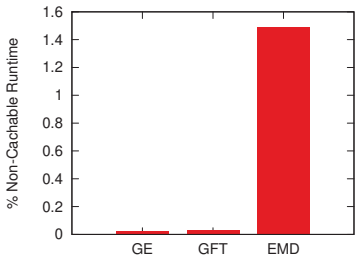


Figure 10: The portion of the runtime that cannot be cached for each distance measure.

On the other hand, EMD is affected, as the input size determines the number of shortest path calculations necessary to estimate it. GE and GFT are suited for vectors with many non-zero entries, where EMD might become unusable. EMD is, instead, more applicable to the case of large networks in which node activity is rare (Zhou, Zha, and Song 2013).

Figure 9(b) shows the runtimes for all measures, varying the number of nodes in the input network. We set the number of non-zero entries in the input vector to be 80% of the number of nodes of the network. In this case all distance measures are affected. While EMD has an unfavorable multiplicative factor, the trend for the measures is asymptotically the same. Thus, keeping fixed the number of non-zero entries in the input vector, generalized Euclidean can guarantee at best a multiplicative speedup.

The time efficiency comparison above, however, may not be the best reflection of real-world conditions. The analytic scenario we consider in this paper is the one where the network's topology stays constant, but one has potentially many events changing the nodes' statuses: many different campaigns happening on the same social network, or one campaign observed at regular intervals for a long period of time. Thus, while there is one single network, there can be potentially hundreds or thousands vector pairs to compare.

The most expensive part of all three frameworks has to do with the network's topology: generalized Euclidean needs to calculate the pseudoinverse of the Laplacian, GFT needs to calculate its eigenvectors, EMD needs to compute all shortest paths. The result of all these operations can be cached and reused if the topology of the network does not change. Thus, they only need to be run once.

Figure 10 shows the percentage of running time for each distance measure spent on parts that are dependent on the node vectors and thus cannot be cached. A measure for which this percentage is lower is better, because it would result in higher time savings when reusing the cached content. From the figure, we can see that for both GE and GFT the running time involved in non-cachable operations is negligible: lower than 0.1%. Thus one could calculate more 1,000 distances on the same network and only doubling the running time. On the other hand, EMD has a larger portion of non-cachable operations, 1.5% of the total, due to its nature as an optimization strategy. Thus running times would double after just a little more than 66 distance calculations.

Conclusion

In this paper we consider the problem of calculating the network distance between activation states of nodes. Nodes that are directly connected are closer than nodes separated by many edges, thus a process that travels across many connections is covering longer distances. Estimating the distance between node activation states is a crucial problem in network science. Experiments show that it can inform us about the infectiousness of a disease or a viral marketing campaign, and whether we are observing a process with the correct time granularity. In the paper, we observe that there are methods to solve this problem in the literature, but they are not explicitly designed for this particular scenario. The Earth Mover Distance is an optimization approach that is accurate but computationally too expensive, while the Graph Fourier Transform is as efficient, but largely imprecise. Thus, we propose a new generalized Euclidean distance based on the Mahalanobis distance, which we prove being intuitive and bringing together the best of EMD and GFT without the downsides. Additional experiments show the usefulness of the measure in an online social media scenario.

Our work opens the way for possible future extensions. While viral marketing and network epidemiology are two important problems, we can envision the application of such distance metrics on networks in many other scenarios. Now that we defined the problem as separate from the optimal transportation problem on a graph, we could investigate possible distance measures based on shortest paths without worrying about the optimization constraint. Moreover, because it is computationally efficient and modular, our generalized Euclidean measure could be further optimized to scale up to networks of millions, rather than thousands, of nodes.

Acknowledgements. The author wishes to thank Andres Gomez, James McNerney, and Frank Neffke for extended discussion on the topic. We thank Clara Vandeweerdt for insightful comments on the manuscript.

References

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

Banavar, J. R.; Colaiori, F.; Flammini, A.; Maritan, A.; and Rinaldo, A. 2000. Topology of the fittest transportation network. *Physical Review Letters* 84(20):4745.

Barabási, A.-L., and Bonabeau, E. 2003. Scale-free networks. *Scientific american* 288(5):60–69.

Barrat, A.; Barthelemy, M.; and Vespignani, A. 2008. *Dynamical processes on complex networks*. Cambr Univ Press.

Coifman, R. R., and Lafon, S. 2006. Diffusion maps. *Applied and computational harmonic analysis* 21(1):5–30.

Colizza, V.; Barrat, A.; Barthélemy, M.; and 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 of the United States of America* 103(7):2015–2020.

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

Coscia, M., and Rossi, L. 2019. The impact of projection and backboning on network topologies. *arXiv preprint arXiv:1906.09081*.

Erbar, M.; Rumpf, M.; Schmitzer, B.; and Simon, S. 2017. Computation of optimal transport on discrete metric measure spaces. *arXiv preprint arXiv:1707.06859*.

Fortunato, S. 2010. Community detection in graphs. *Physics reports* 486(3-5):75–174.

Galas, D. J.; Dewey, G.; Kunert-Graf, J.; and Sakhanenko, N. A. 2017. Expansion of the kullback-leibler divergence, and a new class of information metrics. *Axioms* 6(2):8.

Hammond, D. K.; Vandergheynst, P.; and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30(2):129–150.

Hitchcock, F. L. 1941. The distribution of a product from several sources to numerous localities. *Studies in Applied Mathematics* 20(1-4):224–230.

Kempe, D.; Kleinberg, J.; and 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*, 137–146. ACM.

Leskovec, J.; Adamic, L. A.; and Huberman, B. A. 2007. The dynamics of viral marketing. *ACM TWEB* 1(1):5.

Lü, L., and Zhou, T. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390(6):1150–1170.

Mahalanobis, P. C. 1936. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India, 1936* 49–55.

Monge, G. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*.

Newman, M. E. 2001. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical review E* 64(1):016132.

Pele, O., and Werman, M. 2008. A linear time histogram metric for improved sift matching. In *European conference on computer vision*, 495–508. Springer.

Pele, O., and Werman, M. 2009. Fast and robust earth mover's distances. In *Computer vision, 2009 IEEE 12th international conference on*, 460–467. IEEE.

Pennacchioli, D.; Rossetti, G.; Pappalardo, L.; Pedreschi, D.; Giannotti, F.; and Coscia, M. 2013. The three dimensions of social prominence. In *International Conference on Social Informatics*, 319–332. Springer.

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40(2):99–121.

Serrano, M. Á.; Boguná, M.; and Vespignani, A. 2009. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the national academy of sciences* 106(16):6483–6488.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30(3):83–98.

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

Yildirim, M. A., and Coscia, M. 2014. Using random walks to generate associations between objects. *PloS one* 9(8):e104813.

Zhou, T.; Ren, J.; Medo, M.; and Zhang, Y.-C. 2007. Bipartite network projection and personal recommendation. *Physical Review E* 76(4):046115.

Zhou, K.; Zha, H.; and Song, L. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, 641–649.