Supplementary Material for "Node Attribute Analysis for Cultural Data Analytics: a Case Study on Italian XX-XXI Century Music"

Michele Coscia[1*]

[1*]CS Department, IT University of Copenhagen, Denmark.

Corresponding author(s). E-mail(s): mcos@itu.dk;

1 Data

1.1 Data Sources

Our first data source is Wikipedia, by default the Italian version[1] given our focus on Italian music, but we can also use the English version if it contains information that is not in the Italian version. We save all Wikipedia pages of records that contain credit information. This is the first source we check. If a record is on Wikipedia and has credit information, that is the information we include in our database. If a record is not on Wikipedia – or it is on Wikipedia, but has no credit information, then we use Discogs[2]. Discogs is a crowdsourced record database which aims at including also bootlegs and off-label records. Discogs was originally created to cover only electronic music, but today it strives to include all genres. Discogs periodically releases an XML dump with their entire database, which we use for recovering for each record both the credits and the genre of the record.

However, the combination of Wikipedia plus Discogs is still incomplete and has an English-speaking bias, which makes it more likely that there are Italian records without credits, or that are simply missing. For those, we attempt to find credit information online on the official webpages of the bands or the labels, but most of those records remain outside of our database. Finally, there are also some cases of bands that are completely unknown to both Wikipedia and Discogs. At the moment of writing this paragraph, one example is the Calabrian band *I Musicanti del Vento*. Since we need

[1] https://it.wikipedia.org
[2] https://www.discogs.com/

a Discogs band ID to infer the genres played by the band, we cannot do anything else but excluding these bands from the network.

1.2 Data Model

As mentioned in the main paper, we model the data as a bipartite network $G = (V_1, V_2, E)$. The nodes in the first class V_1 are artists. An artist is a physical real person. If an artist works under different pseudonyms, we do our best to disambiguate them (or we use the pseudonym if it is much more recognizable than their real name, as it is the case e.g. for the singer Alice, born Carla Bissi). If an artist is an abstract collective, then each of the people in the collective is reported separately – if the information about their identities is available. Since we use the name to identify an artist, we sometimes have multiple people with the same name; for instance we have found four different Paolo Rossis. We do our best to disambiguate names also in this case, distinguishing different people with different names with a given ID – Section 1.4.1 has more details about this process. We cannot use the Discogs artist IDs, as they are not assigned with the same standards we require – e.g. Discogs assigns different IDs to different aliases of the same person.

The nodes in the second class V_2 are bands, which are identified by their name. Note that we consider solo artists as bands, and they are logically different from the artist with the same name. One example is Mina, who is both an artist and a band. The band refers to a different entity than the person: it is the collective of people who played a role in at least a record published under the name Mina – which include the actual person named Mina and many other people. Since it is a different entity, there are two Mina nodes: one of class artist and one of class band. The two should not be confused with one another.

As mentioned in the main paper, each edge is weighted by the number of years in which the band-artist connection existed. This can be different from the time interval in which the edge was active. For instance, Ginevra Di Marco participated in her solo records from 1999 until 2009, a period of ten years, which is longer than the 1993-2001 period in which she participated in CSI's records. However, CSI released nine records in that period, against Ginevra Di Marco's (the band) two, and that is why the edge connecting Di Marco to CSI is thicker than with her own solo albums.

Regarding the node attributes, we need additional details about the region and the year attributes. The region attribute records, for bands the region where they were formed, regardless of the geographical provenance of its members. For solo artists it is their region of birth. If they are foreign born, we use the place where they grew up. This attribute is then one-hot encoded, which results in twenty binary attributes – since there are twenty Italian regions. The value of the attribute is one if the band originated in that specific region, and zero otherwise. Occasionally, the information about the region of origin is unavailable on Wikipedia and Discogs. In this case we look at the official website of the bands, Bandcamp, Soundcloud, or similar other sources. If the region of origin cannot be found, it will be set to zero for all regions. In our dataset, 1.47% of the bands have no region information.

The temporal attributes provide a distinct binary attribute per year, set to one if the band released at least a record in that year, and zero otherwise. This results in

122 additional attributes, one per year, starting from 1902 until 2024 (there are no records in year 1903, so that year does not generate a node attribute).

1.3 Data Collection Design

In order to collect the data, we need to define the criteria of what needs to be included and what should be excluded. These criteria are necessary to obtain a coherent network and for practical purposes. Here we detail these criteria. There are three main classes: what counts as a published record, what counts as an Italian, and what counts as a record credit that can generate a link in the bipartite network.

1.3.1 Published Record

We need to define the observations we use to fill the data model. In our case, we use only records published in the music market. This means that we do not record any artist-band collaboration for movie soundtracks. The reason is that movie soundtracks follow different dynamics than a record published as a stand-alone cultural product. Including them would be capturing two different phenomena in the same structure, likely confusing the discussion of the results of our analysis.

We also need to use flexible criteria when it comes to define what counts as a record. Different industries and different times used different media to publish their records. For instance, mainstream music usually focuses on album releases. But more contemporary electronic music releases single tracks directly, either as downloadable files or via streaming platforms. Similarly, in older times a variety of different formats were used and the release of singles was more common than full albums. Therefore, for each combination of genre and time period we focus on the main medium used to publish music.

Finally, we exclude records with no or only one credited participants. This is because they would not generate any edge weight in the projection of the bipartite network. We still count those records when it comes to analyzing the genres, but they play no structural role defining collaboration, by definition.

1.3.2 Italian Record

Now that we know what a record is, we need to define what counts as an *Italian record*. For this, we rely on a combination of multiple factors: the record has been published in Italy, it is in Italian, it is by an Italian band, and any combination of these conditions. A band is Italian if they primarily target the Italian music market. Here, it is worthwhile make a couple of clarifying examples:

- The artist Dalida was at all effects French. Most of her records were published in France and are sung in French. For this reason, they are not included. However, her eponymous 1966 record was published in Italy, it was sung in Italian, and has mostly Italian artists working on it. For this reason, we include it in the data.
- On the other hand, Rocco Granata is Italian, born in Italy from Italian parents. He published some songs in Italian, most notably Marina, but he published all his records in Belgium, targeting the Belgian market, singing mostly in Dutch. For this

reason, he is not part of the network as a band. Having no collaborations with other Italian bands, he is also absent as an artist.

We decide to drop bands that have published only one record or have only one year of recorded activity. This is because we want to capture enduring musical projects, not ephemeral one-time-only events that do not indicate a real collaboration. For instance, we ignore the humanitarian single published by the superband LigaJovaPelú in 1999, as the three famous artists Ligabue, Jovanotti, and Piero Pelú never continued the project.

1.3.3 Record Credits

We want to include only record credits that show a real collaboration relationship for the final recorded songs in the record. For this reason, not all credits attached to a record are included in the final network.

We include people playing instruments – even if they are programmed, e.g. drum machines – and working on the production of the song – arranging, producing, mastering, mixing, etc. We exclude people who do not have a strong connection with the songs: artists designing the record cover, photographers, people working with costumes, etc. Songwriters sometimes have a strong connection to the song, but in many cases they do not – covers, inspirations, other unclear rules of who counts as a songwriter and who does not. Since it is not possible from the credits alone to distinguish these two cases, we opt to be conservative and to ignore songwriters altogether.

1.3.4 Exploring the Network

Notwithstanding the fact that we attempt to give as complete of a picture as possible, the network is only a sample of the full structure. This is due to the fact that data cleaning – as we will see – can only be done semi-automatically, to ensure quality. Since we focus on quality – trying to allow in as little noise as possible – we have to somewhat sacrifice quantity. This implies that the way we decide to explore the network of collaboration matters. We decided to explore bands in descending order of popularity in number of records owned on Discogs. Since Discogs has a genre bias – being more focused on electronic – we explore several genres in parallel.

1.4 Cleaning

The data comes with a certain amount of errors, both on nodes and on edges.

1.4.1 Errors on Nodes

There are two possible errors on nodes: homonyms and pseudonyms/misspellings – both already mentioned in Section 1.2.

To find homonyms, we manually inspect edges with suspiciously high betweenness centrality. These edges connect bands that would otherwise have nothing in common and would be placed in completely different areas of the network. Since the network – as we will see – has a strong temporal and genre organization, these nodes indicate either unreasonably eclectic or long-lived artists. Of course, with this procedure we

miss all homonyms that lived nearby in time and worked on similar genres. That is one reason why we operate not directly on the bipartite network but on its projection, as we can turn these node errors into edge weight errors – for which we have a good solution via backboning (see Section 1.4.2).

For pseudonyms, we rely on Discogs. While Discogs has different IDs for the different pseudonyms of the same person, it still reports the various aliases an artist has used, and we choose one as the canonical name – if possible, their given name. For misspellings, we calculate the string edit distance between all pairs of nodes, using the Ratcliff-Obershelp algorithm [1] and we manually inspect the strongest matches. We need the manual step, because occasionally extremely similar names indeed refer to different people – e.g. Alessandra Simoncini and Alessandro Simoncini are two different people, the former a female vocalist for a heavy metal band, the latter a male violinist who worked for multiple pop artist.

1.4.2 Errors on Edges

We know we have missing collaborations because there are plenty of records which do not report credits. On the other hand, there could be artist-band connections that are not really present – as we mention in Section 1.4.1. Both errors can be handled by projecting the bipartite network and then using network backboning, specifically the noise-corrected approach [2].

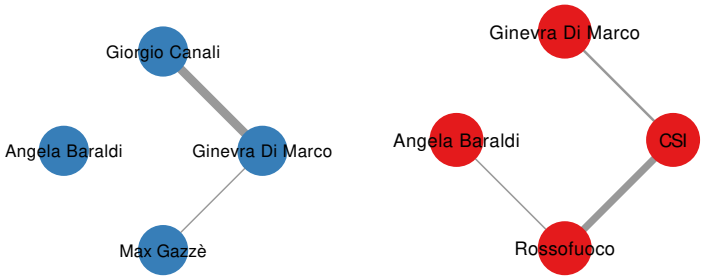
Let us assume that we projected the bipartite artist-band bipartite network into a band-band unipartite one. Two bands are now connected with a weighted edge, counting how many artists the two bands shared. Spurious edges in the bipartite network will inflate the edge weight, while missing credits would deflate it – possibly to zero. The noise-corrected backboning technique allows us to create a statistical test, telling us the likelihood of the edge weight being zero.

If an edge was not supposed to be in the projection, it means we are dealing with a spurious artist-band connection. Since the assumption is for these errors to be more rare than the signal, the edge weights of these spurious edges will be compatible with the null hypothesis and therefore they will be eliminated from the network.

At the same time, missing credits would weaken an edge weight, possibly below the threshold of significance. By imposing a high significant bar for an edge to be included in the projection, we reduce the incidence of these errors as well: we require collaborations to be so strong that a few missing credits would not affect their significance level.

For this reason, we allow in the band projection only edges with $p < 2 \times 10^{-7}$ probability of being compatible with the null hypothesis of no connection between the bands. The threshold is instead $p < 10^{-6}$ for the artist-artist projection, due to the fact that there are more nodes in this projection and therefore the resulting structure is more sparse.

Not to lose nodes, we add to the network backbone the single most significant connection of all the nodes that did not have a single edge surviving the significance threshold. The methods we want to use require the graph to be connected, so we need to connect the various disconnected components of the graph. We do so by looking at



(a) (b)

Fig. S1: The two projections of the bipartite example from Figure 1 in the main paper. The edge's thickness is proportional to its weight. (a) Artist projection. (b) Band projection.

all node pairs and connect the one with the most significant edge that would join two disconnected components. We repeat until we have a single connected component.

This operation does not distort the network too much. The total number of edges we need to add this way are $1,001$ for the artist projection and 181 for the band projection, representing only 0.71% and 2.77% of the final edge sets, respectively. The average significance of the edges added this way is 9.09×10^{-6} for the artist projection and 6.31×10^{-6} for the band projection, which makes these added edges still highly significant.

1.5 Bipartite Projections

We project our bipartite network into an artist projection and a band projection. There are many ways to perform this projection [3–5], which result in different edge weights. Since, as we explain in Section 1.4.2, we plan to correct errors using the noise corrected backboning method, using simple counts as edge weights is the most appropriate way, because it allows us to match the null expectation of the backboning method with the edge generating process of the projection.

For the artist projection, this choice results in a trivial operation. The edge weight in the projection is the number of records in which the two artists both appeared. Note that here we require a common record, not just the band. Figure S1a shows the artist projection of the toy bipartite network from Figure 1 in the main paper. Even if Angela Baraldi has played for Rossofuoco, she is not connected with Giorgio Canali due to the fact that they do not share a record – in this toy example Baraldi only played for Rossofuoco after Canali's period[3].

For the band projection, the operation is a bit more complex. Two bands should be connected by the number of artists they share. However, we should count shared artists differently, because it is different sharing two artists which both only appeared in a record of the band and two artists who appeared in ten. For instance, in Figure 1 in the main paper, the link between Rossofuoco and CSI should be stronger than the one

[3] Note that this is not true in the actual data, but we elected to alter the data here for illustrative purposes.

Variable	Value
# Bands	2,447
# Artists	29,014
# Edges	101,441
# Years	122
# Band AVG Year	6.9

Table S1: Summary statistics for the bipartite network.

between Rossofuoco and Angela Baraldi, even if they both share a single artist. In the first case, that artist is a key member participating in all records of both bands, while in the second case the shared artist only participated in one record of Rossofuoco.

Our solution is to use the minimum importance overlap criterion. Artist a was active in w_{a,b_1} years for band b_1 and for w_{a,b_2} years in band b_2. The strength with which a brings b_1 and b_2 together is the minimum number of years in which a was part of either, $\min(w_{a,b_1}, w_{a,b_2})$. If we say that the set of shared artists between b_1 and b_2 is $b_1 \cap b_2$, then the weight of the edge between the two bands in the band projection becomes:

$$w_{b_1,b_2} = \sum_{a \in b_1 \cap b_2} \min(w_{a,b_1}, w_{a,b_2}),$$

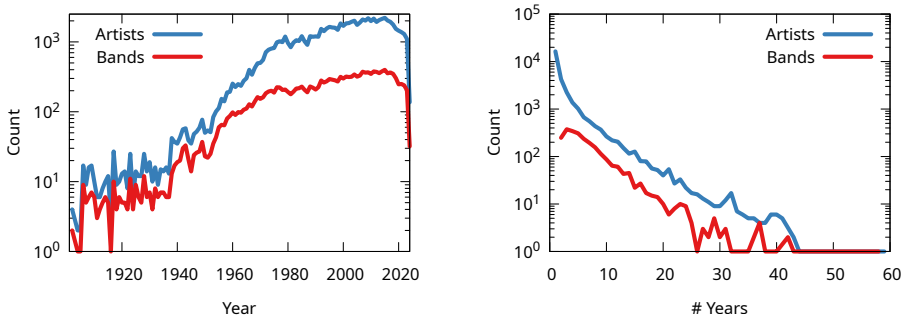
which matches the intuition that bands have stronger edges only if they share many artists and those artists have contributed a lot to the bands' histories.

Figure S1b shows the result of the band projection operation of the bipartite network in Figure 1 in the main paper. Note how Angela Baraldi still connects her solo band to Rossofuoco even if she did not directly collaborate with Giorgio Canali, because sharing Angela Baraldi (the artist) is enough to connect Angela Baraldi (the band) to Rossofuoco. Similarly, the edge weight between Rossofuoco and CSI is 7 (the number of records Canali contributed to for Rossofuoco) and not 12 (the number of CSI records in which Canali appears), because we take the minimum overlap.

2 Summary Description

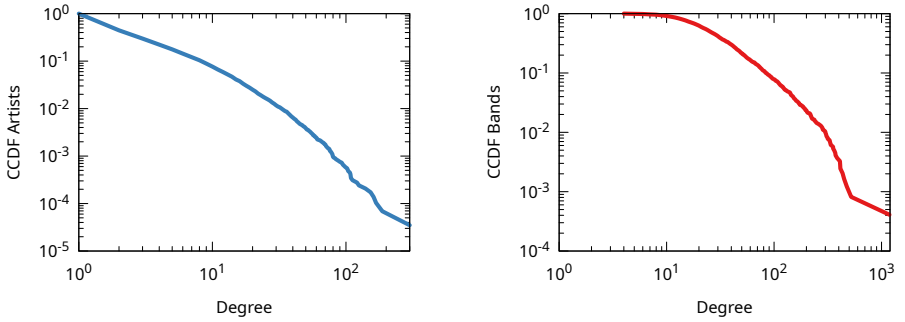
Table S1 summarizes the basic statistics of the bipartite network. From the table we can see that there are around 11.85 distinct unique artists per band and a band will put records out for an average of 6.9 years.

Figure S2a shows the temporal evolution in our data. We can see that there is little activity before the end of the Second World War, due to the infancy of the Italian record industry. The activity steadily increases over time. There is an artificial peak in our data in the late 2010s. This is an artifact caused by two factors. First, there is a lag between the publishing of a record and its entry on Discogs or Wikipedia. Second, since we only focus on artists with at least two records in the data, we tend to undersample new artists. For this reason, we will not discuss the data as representative for the last five years of the Italian record industry. The overall picture – given the other 117 years – should not be significantly affected.



(a) (b)

Fig. S2: (a) The number (y axis) of artists (blue) and bands (red) active in a given year (x axis). (b) The number (y axis) of artists (blue) and bands (red) with a given number of years of recorded activity (x axis).



(a) (b)

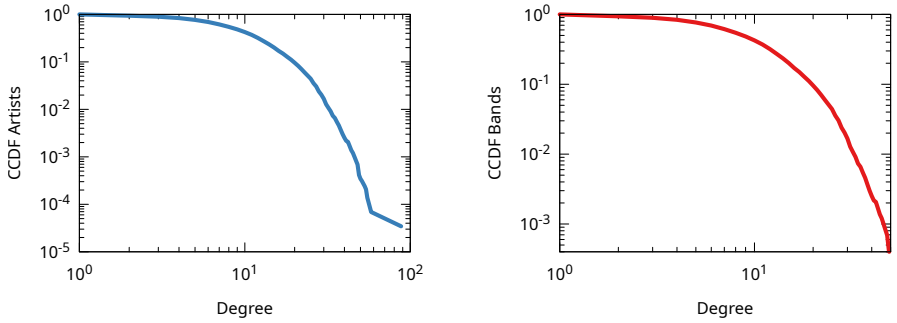
Fig. S3: The probability (y axis) of a node having a given degree value (x axis) or higher in the bipartite network for the two node types: (a) artists, (b) bands.

There is heterogeneity in the number of years we see an artist or a band active. Figure S2b shows that the mode for artists is to be active only for a single year, while the artist with most years of activity (Mina, the person) was present in 59 years – note that this is the count of distinct years in which she appears and gap years are not counted. The band distribution is censored by our choice of dropping bands with less than two years of activity. The mode number of years is three, and the band with most years of recorded activity (Mina, the band) was present in 58 years.

The heterogeneity is present not only in number of years of activity, but also in number of collaboration links. The degree distributions of both node types are skewed – see Figure S3. The average band degree (Figure S3b) is 41.45, with the band having most connections is Mina with degree 1,185. Note that this is not the number of artists

Variable	Artist	Band
# Nodes	29,014	2,447
# Edges	141,712	6,512
Avg Deg	9.8	5.3
Density	0.0003	0.0022
Clustering	0.5567	0.4160
Modularity	0.8812	0.8437

Table S2: Summary statistics for the projected networks.



(a) (b)

Fig. S4: The probability (y axis) of a node having a given degree value (x axis) or higher in the projected networks for the two node types: (a) artists, (b) bands.

playing in the band Mina, but rather the number of artist-year pairs: the same artist is counted twice if it played for the band in two years.

Again, the band distribution is censored by our filter choice. The artist distribution (Figure S3a) is not censored. As a result, the average degree is much lower (3.5). The artist participating in most band-year combinations is Lele Melotti – a popular drummer –, with 303 connections.

The summary statistics for the artist projection is in the first column of Table S2, whose second column reproduces the values for the band projection we discuss in the main paper. With our significance threshold, we end up with a sparse but highly clustered network. Performing a community discovery with the Louvain algorithm, we obtain a quite high modularity value. These factors can be trivially explained by the fact that the network tends to be composed by cliques interconnected with each other, as we expect the artists collaborating in a record to be all connected to each other.

Figure S4a shows the degree distribution of the artist projection – compared with the band projection. While we do see a broad distribution as we did for the degree distribution in the bipartite network, here the distribution shape shows a higher prevalence of nodes with medium degree and a lower prevalence of nodes with degree one – again due to the tendency of nodes to form cliques.

#	Degree	Closeness	Betweenness
1	Agostino Balestrieri	Alfredo Golino	**Paolo Fresu**
2	Simone Mularoni	Lele Melotti	Dino Olivieri
3	Noyz Narcos	Alberto Radius	Gianluca Petrella
4	Luché	Claudio Pascoli	Carlo Ubaldo Rossi
5	Mirella Freni	Paolo Donnarumma	Pino Calvi
6	Ennio Morricone	**Paolo Fresu**	Claudio Simonetti
7	**Paolo Fresu**	Alessandro Colombini	Lorenzo Urciullo
8	Fabri Fibra	Angelo Carrara	Maria Di Donna
9	Gianfranco Bortolotti	Paolo Costa	Danilo Vigorito
10	Lino Capra Vaccina	Giorgio Cocilovo	Franco Battiato

Table S3: The top 10 artists in the artist projection according to different centrality measures. In bold we have nodes central in multiple measures.

Table S3 summarizes the top 10 artists as ranked by three different centrality measures – degree, closeness centrality, and betweenness centrality. These reflect different ways in which an artist can be important: either because they are part of many records with lots of other artists (degree), or they participate in those records that are in the core of the network (closeness), or they participate in those records that play a critical role in keeping the structure connected (betweenness).

The only artist appearing in all three top tens is Paolo Fresu, which we could consider the most important Italian artists in a structural sense.

3 Methods

Most of the node attribute analysis methods we use were already developed in the literature and we briefly outline how they work and how we use them in Section 3.1. On the other hand, in our final analysis we want to discover temporal eras. This is a special case of the node attribute clustering task which has not been tackled yet in the literature. So we detail our methodology to discover node attributes areas in Section 3.2.

For this section, it is convenient to establish some notation. If $G = (V, E)$ is our graph, we use A to indicate its adjacency matrix, which has an entry of 1 for connected node pairs and 0 otherwise. D is the degree matrix, a matrix with the degrees of the nodes on the main diagonal, and zero elsewhere. The graph Laplacian is $L = D - A$, a matrix with -1 for the entries corresponding to an edge, and the nodes degrees on the main diagonal. The effective resistance matrix Ω tells us the resistance distance between two nodes in a graph – i.e. their resistance assuming the network is a finite electric circuit and each edge is a 1 Ohm resistor. For what concerns this paper, $\Omega_{u,v} = \Gamma_{u,u} + \Gamma_{v,v} - 2\Gamma_{u,v}$, with $\Gamma = \left(L + \frac{1}{|V|}\mathbb{1}\right)^{\dagger} - \mathbb{1}$ being a matrix of ones. The $\dagger$ operator is the Moore-Penrose pseudoinverse – a technique to calculate the inverse of singular matrices.

3.1 Node Attribute Analysis

3.1.1 Node Attribute Variance

The variance is a measure of how much a variable's values disperse. One can calculate the variance of a variable x as:

$$var_x = \frac{1}{2}\sum_{u,v} x_u x_v d_{uv}^2,$$

where d is the Euclidean distance between the observations u and v on the line of real numbers. One can estimate a network variance by replacing the Euclidean distance with a proper metric on a graph [6]. The best choice is the effective resistance matrix Ω, given that it is a proper metric and more stable than the shortest path distance to small fluctuations in the graph structure – such as the addition or removal of an edge [7]. Hereafter, we intend node attribute variance as

$$var_{x,G} = \frac{1}{2}\sum (x \otimes x)\Omega^2,$$

with $\otimes$ being the outer product operation.

3.1.2 Node Attribute Distance

The methodology to calculate the effective resistance matrix – specifically the pseudoinverse of the Laplacian $L^\dagger$ – can be used to estimate Euclidean distances between node attributes on a graph. Suppose that we have two node attributes x and y. Their Euclidean distance in a flat Euclidean space is:

$$\delta_{x,y} = \sqrt{(x-y)^T I (x-y)},$$

with I here being the identity matrix and representing the fact that the dimensions are all equally important and are not related to each other in any way. We can modify this by allowing the complex space in which x and y live to be represented by the graph G. This results in the formula:

$$\delta_{x,y,G} = \sqrt{(x-y)^T L^\dagger (x-y)},$$

which is a proper analogue to the Euclidean distance on G – and a proper metric satisfying the triangle inequality [8].

3.1.3 Node Attribute Correlation

The Euclidean distance is useful in many contexts, but sometimes its units might not be the easiest to interpret. For instance, in the case of the formula just presented, a unit can be interpreted as the (square root) of the number of edges required to transport the values of one attribute value to the values of the other in a chain graph. However, as soon as the graph is more complex than a chain, this interpretation becomes harder.

To understand the relationship between two variables in a more intuitive way, one might want to calculate their correlation coefficient. One example of such a correlation coefficient is Pearson's linear correlation, which is +1 for perfectly correlated variables,

0 for unrelated variables, and -1 for perfectly uncorrelated variables. The classical Pearson correlation coefficient still assumes that the variables live in a flat Euclidean space, just like we have shown for the Euclidean distance:

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

here $\hat{x}$ is the centered version of x ($\hat{x} = x - \bar{x}$, or the subtraction of x's mean from x); and σ_x represents the standard deviation of x ($\sigma_x = \sqrt{var_x}$, the square root of the variance).

There exists a version of this correlation that can allow for the observations to live in a space defined by the network, just like the variance and distance measures discussed above. The formula is:

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

with $\sigma_{x,W} = \sqrt{\sum W \times (\hat{x} \otimes \hat{x})}$ being a similarly defined version of the network's standard deviation. The portion of this formula embedding G's structure in this formula is W, the matrix assigning a proper notion of network similarity between pairs of nodes. Recent work showed that only a proper metric can be used for W, resulting in a natural choice being $W = e^{-\Omega}$ – using shortest path distances instead of the effective resistance Ω is invalid and would result in imaginary correlation values [9].

3.1.4 Node Attribute Clustering

Having a notion of distance between numerical vectors on a graph opens the possibility of performing network-based unsupervised learning. In the problem of data clustering, one wants to find groups of observations – numeric vectors – that organize in segregated groups – clusters. Points in a cluster are on average closer to other cluster members than they are on average to non-cluster members. Using the node attribute distance defined in Section 3.1.2 allows us to find clusters of node attributes depending on how they distribute in a network.

This, however, presents a problem. In this framework, the nodes of the network represent the dimensions of analysis. Networks tend to have many nodes. Even if our band projection is of modest size – with $|V| = 2,447$ – this still represent a large number of dimensions. The curse of dimensionality is the observation that each additional dimension makes using distances harder, because the more dimensions there are the more observations tend to get farther apart from each other. In this scenario detecting clusters becomes harder.

A proposed solution [10] uses dimensionality reduction techniques. Each node attribute starts as a $|V|$ dimensional vector. One could use a dimensionality reduction technique of choice to reduce it to a two or three dimensional embedding – here we use t-SNE [11] but Isomap [12] or UMAP [13] are valid choices as well. Once reduced, the embeddings can be clustered with a clustering algorithm of choice – here we use hierarchical agglomerative clustering with a Ward linkage criterion.

By default, dimensionality reduction techniques operate in an Euclidean space. To achieve proper node attribute clusters in the non-Euclidean space defined by the

network it is critical that we use the previously defined node attribute distance. This is possible, as the dimensionality techniques can use a custom distance function to estimate the distance between the observations. Once the node attributes have been reduced to a lower dimensional space, their embeddings take into account the network structure, and can then be clustered.

3.2 Discovering Node Attribute Eras

Finding eras in node attributes means to cluster temporal information into periods that are characterized by low distances internally and separated from the preceding and following period by a jump in distances. Essentially this means to represent each temporal snapshot as a node attribute – setting the value for a node to 1 if the node was active in that snapshot and 0 otherwise. Then one can perform hierarchical clustering on the node attributes.

3.2.1 Building the Dendrogram

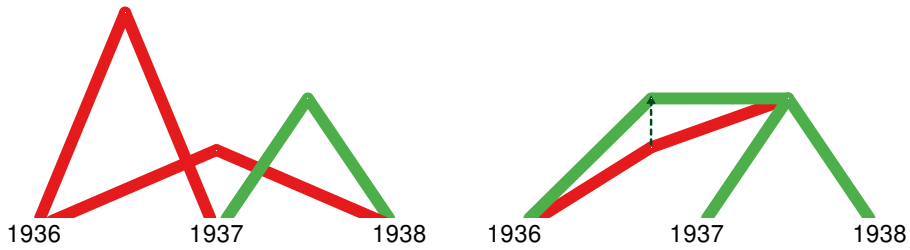
This clustering has additional requirements to the ones presented in Section 3.1.4, and it has not been discussed before in the literature/ For this reason, we report here additional details about these requirements. These are:

1. While general node attribute clustering can use any technique, here we restrict it to be an agglomerative hierarchical clustering.
2. The only candidates for merging at any point in the agglomerative clustering procedure are only snapshots – or clusters of snapshots – adjacent in time.
3. There is no conceptual difference between a snapshot and a cluster of snapshots, since they can be both represented as node attributes. As a result, we do not need special linkage rules, but we can simply calculate the node attribute distance between two (clusters of) snapshot(s) using our node attribute distance.

Constraint #2 simply means that we will not attempt to merge two non-adjacent years, e.g. 1936 with 1938, or non-adjacent clusters, e.g. 1934-1936 with 1938-1942. This ensures that the eras will not have gaps in them. Figure S5a shows an example, where we merge 1938 with 1937 and not with 1936, even if 1937 is father to 1938 than 1936 is, just because 1936 is not adjacent to 1938. This is a fairly standard constraint in clustering, but it has never been explored in network-based clustering.

Constraint #3 means that, once we merge two years, we can create a new node attribute representing both years. This can be the average of the values in the merged years. For instance, suppose we merge years 1937 with 1938. The 1937-1938 cluster is a new node attribute. If a band published a record in 1938 but not in 1937 (or vice versa), then their entry will be 0.5.

This constraint has a peculiar consequence. The node attribute 1937-1938 might become closer to 1936 than 1937 originally was. This implies that the merging distance δ is not monotone: the distance between 1936 and 1937-1938 could be smaller than the one between 1937 and 1938. This breaks one of the fundamental properties of regular agglomerative clustering. We fix the issue by deriving a δ' distance function from our network distance function δ.



(a) Step #1 (b) Step #2

Fig. S5: Two steps of the era clustering procedures. Edge color indicates the status of the proposed merge (red = rejected, green = accepted). The height at which the merge happens represents the distance between the observations.

Suppose we are trying to merge clusters X and Y. Suppose also that Y was built merging Y_1 with Y_2, at distance $\delta'_{Y_1,Y_2,G}$. Then, the merging distance of X and Y is:

$$\delta'_{X,Y,G} = \max\left(\delta_{X,Y,G}, \delta'_{Y_1,Y_2,G}\right).$$

This ensures that δ' is a monotone function. If X and Y both only contain one element, then $\delta' = \delta$. Figure S5b shows how δ' behaves in a graphical way: we are pushing up the merging distance to get a well-defined dendrogram.

3.2.2 Cutting the Dendrogram

Once the dendrogram is built, we need to have a criterion to cut it and obtain the final eras. Normally this is done by establishing a threshold that maximizes some quality function. Merges above the threshold are rejected and we only keep the groups we obtain if their merge distance is below the threshold. However, due to data sparsity and noise, the dendrogram is unbalanced. This means that having the same cut in all areas of the dendrogram is suboptimal, as in some parts of the dendrogram we might want to have a higher threshold than in others. The risk otherwise is to have either too many singletons if we choose a small threshold, or one giant cluster if we choose a high one.

For this reason, we decide to perform an adaptive cut, which means that we use different cut heights for different parts of the dendrogram [14]. We found that a simple algorithm, for our case, produces balanced results:

1. Always accept merging a singleton year to any cluster;
2. Reject merging two clusters if they both contain more than $x = 10$ years;
3. Once a merge is rejected, the two clusters are final and will not accept any other merge, even if the merge would pass check #2.

References

[1] Ratcliff, J.W., Metzener, D., *et al.*: Pattern matching: The gestalt approach. Dr. Dobb's Journal **13**(7), 46 (1988)

[2] Coscia, M., Neffke, F.M.: Network backboning with noisy data. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 425–436 (2017). IEEE

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

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

[5] Yildirim, M.A., Coscia, M.: Using random walks to generate associations between objects. PloS one **9**(8), 104813 (2014)

[6] Devriendt, K., Martin-Gutierrez, S., Lambiotte, R.: Variance and covariance of distributions on graphs. SIAM Review **64**(2), 343–359 (2022)

[7] Devriendt, K.: Effective resistance is more than distance: Laplacians, simplices and the schur complement. Linear Algebra and its Applications **639**, 24–49 (2022)

[8] Coscia, M.: Generalized euclidean measure to estimate network distances. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 14, pp. 119–129 (2020)

[9] Coscia, M., Devriendt, K.: Pearson correlations on networks: Corrigendum. arXiv preprint arXiv:2402.09489 (2024)

[10] Damstrup, A.S.R., Madsen, S.T., Coscia, M.: Unsupervised learning via network-aware embeddings. arXiv preprint arXiv:2309.10408 (2023)

[11] Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

[12] Tenenbaum, J.B., Silva, V.d., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)

[13] Leland, M., John, H., James, M.: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)

[14] Rohlf, F.J.: Adaptive hierarchical clustering schemes. Systematic Biology **19**(1), 58–82 (1970)