Chapter 14

Summarizing and comparing networks

Realistic networks are rich in information. Often too rich for all that information to be easily conveyed. Summarizing the network then becomes useful, often necessary, for communication and understanding but, being wary, of course, that a summary necessarily loses information about the network. Further, networks often do not exist in isolation. Multiple networks may arise from a given dataset or multiple datasets may each give rise to different views of the same network (Ch. 7). In such cases and more, researchers need tools and techniques to compare and contrast those networks.

Summaries can fulfill this purpose: by reducing networks to summaries, we make it easier to compare one network to another. Comparing networks becomes critical when contrasting the results of a study (networks before and after treatment), when networks change over time (network in week two versus network in week one), and when the upstream task is explored (network from data processed by method one vs. by method two). It's very common to have multiple networks, or multiple versions of a network, making comparison a common task.

14.1 Summarizing networks

Given a network, how to generate a suitable summary? Answering this nuanced, complex question is our goal here. For any network of sufficient complexity, a summary must necessarily involve a tradeoff. Not all the features of the network can be contained in the summary; we must pick and choose what to leave in and what to leave out.[1] Choosing a summary first means determining the proper medium of the summary. Should it be a table of summary statistics? Should it be another (smaller) network? Perhaps a visualization of the network (Ch. 13) is a valuable summary?

Let's tackle these each in turn.

[1] Apologies to R. Seger.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

14.1.1 Summarizing networks with statistics

The focus of Ch. 12 was on understanding the structure of a network using a variety of measures and methods. Many network statistics exist to describe the network structure. For instance, the average degree, the transitivity, the average path length, and more; even simple statistics such as the number of nodes and links. The numeric values of these statistics allow us to quantify the size, connectivity, and density of the network. A network with a high average degree (relative to the number of nodes) will be quite dense. Likewise, a network that exhibits many triangles will show a high transitivity. These quantities alone, the number of nodes and edges, the average degree, and the transitivity, while just scratching the surface, already provide a basic summary of a network:

	Nodes	Edges	Average degree	Transitivity
Zachary's Karate Club	34	78	4.588	0.2557
Plant–pollinator (bipartite)	104	81	1.558	0
Developer collaboration	679	3628	10.69	0.4475
Flavor network	376	917	4.878	0.4719
HuRI	8,272	52,548	12.71	0.05585
Malawi sociometer (weighted[2])	86	347	8.07	0.3699

By choosing a suitable set of statistics, and using the values of those statistics as the basis of your summary, you are in effect projecting networks down into a space of reduced dimension—a "summary space." Is doing so effective? It depends in part on the goals of your summary and the choices of your summary statistics.

Some diagnostics can guide you if you wish to interrogate your summary statistics. If you make a small change to the network, such as rewiring an edge (Ch. 11), how does the position of the network in your summary space change? Does it have no effect, a small effect, or does it drastically change the location of the network? Depending on the type and magnitude of the change to the network, you may expect only a small change. If so, and you observe a large change, then you may want to revisit your choice of summary. Perturbing the network in this way is a small-scale example of some of the null models used in Ch. 11, and another way to interrogate your summary is to check the same statistics on those null models.

Lastly, another question to ask is whether your set of statistics is appropriate. Do the values capture the information you need? Is something missing? Often there are connections between certain statistics. For example, the average degree is a function of the number of nodes and links in the network, and relying on those three alone captures some, but not all, aspects of the network: your summary may be information poor. Perhaps all you need to compare is the size of the network and its density, but you will often want (or need) to incorporate further information into the summary. Consider adding information on the clustering coefficient, or average shortest path length, or degree correlations. These statistics all capture further information about

[2] We discuss the transformation of this temporal network to a static, weighted network in Ch. 15.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

the network's structure and organization. The most effective summary is not always maximally succinct, but instead maximally readable and informative.

Consider using *network cards*, a tabular network summary and reporting tool we have proposed[3] [22]. Network cards are concise, readable, and broadly applicable to all types of networks.

14.1.2 Summarizing networks with visualizations

A visualization can be a powerful means to convey information about a given network. Many of the statistics discussed in the preceding section (and Ch. 12) are amenable to graphical plots. In addition, network visualizations are useful summaries. We dedicated Ch. 13 to the details of network visualization, including techniques and advice on how to generate useful, readable visualizations.

Chapter 13 also discusses the limitations of network visualizations and those limitations hold for summaries as well. Fundamentally, basic visualizations fail when networks become very large and very dense, forcing us to try to tease out meaning from "hairballs" or "ridiculograms."

With that said, if your network appears to be suitable for a network visualization, it is likely that the visualization will also be a useful summary of the network. At a glance, a reader familiar with network drawings will be able to glean useful information about the size and density of the network.

14.1.3 Summarizing networks with other networks

Deriving a new network from an existing network can also be an effective form of summarization. Very large networks are difficult to visualize. Their size makes it slow to draw the layout and the resulting visual may be too dense to read clearly. A smaller summary network can alleviate these problems.

One approach is to create a "super network" by exploiting the structural properties of the original network. Community structure is a natural starting point (Ch. 12). First, infer the communities using an algorithm.[4] Finding the communities means you have a mapping from each node to the community to which it belongs.[5] Use this mapping to define the super network by creating a new network where each node represents a community in the original network and these "super nodes" are linked when the nodes in the original network are linked.[6,7]

Beyond using community structure to derive a smaller summary by coarse-graining the network, other approaches are possible. An "effective network" can be defined

[3] https://github.com/network-cards

[4] This makes the result depend on the algorithm being used. Hopefully, the algorithm is appropriate, but this is something the researcher should determine.

[5] Or communities, if nodes can belong to multiple communities.

[6] This can also define a weighted network where the weights are the number of links between the two communities or another summary of their connections.

[7] A similar definition can be used for *overlapping* communities: two super nodes are connected when the original communities overlap by sharing nodes.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

by identifying connections between nodes that capture some notion of similarity and building a network topology from that. Often dynamical information is used to find this network. For instance, a network with time series associated with nodes can lead to an effective network extracted from correlations between nodes' time series. If many time series exhibit weak correlations, the resulting network will be dense but not necessarily meaningful, so filtering the network by removing weak links can also be considered a summary using another network. However, filtered networks sit on a subset of the original nodes and therefore we don't feel this cleanly fits the idea of using a different network as a summary. See Ch. 10 for more on thinning overly dense networks.

14.1.4 Other summaries

Combinations Don't underestimate the power of combining summaries. A small table of network statistics included within or alongside a network visualization can lead to a reinforcement effect. Bring the readers attention to the most salient details by highlighting a key detail in both the network visualization and the summary table, such as by using node size to capture node degree in the visualization and reporting the mean, median, and variance of the degree distribution in the summary table. Many such options are possible.

Model parameters When models are fit to the network, parameters are usually used to govern the fit. Usually these are statistical parameters, such as the matrix of edge probabilities for a stochastic block model (Ch. 23) or a vector of regression coefficients for an exponential random graph model (Ch. 23), inferred by a fitting procedure. Among other uses, these parameters can serve as summaries of the network being fitted, in effect, acting as additional statistics (Sec. 14.1.1) to describe the network. An interpretable model can be especially informative as a summary, but regardless, it is important to quantify the goodness-of-fit of the model, as appropriate; a poorly fitting model will generally not be suitable as a summary.

Metadata For networks with associated attributes, metadata describing nodes and links (Ch. 9), summaries of those metadata and using those metadata are both worth exploring. In other words, we can treat the metadata as data in its own right, and summarize it using any manner of descriptive statistics. We can also relate those metadata to the network itself, for example, reporting an assortativity using a particular node attribute (Sec. 9.3).

Qualitative summaries Lastly, don't discount the usefulness of *qualitative summaries*. Our focus is strictly on quantitative measures, but a simple human-readable written description of the network is often helpful. We can think of this as a captioning process narrated by the researcher and guiding the reader through the key facets of the network. One of the main criticisms of qualitative methods, of course, is the potential for bias and subjectivity, so good practices should be followed whenever possible [462, 86]. In particular, it should always be clear to the reader that the summary is qualitative in nature.

14.2 Comparing networks

Comparing two or more networks is a problem closely associated with summaries. By reducing the complexity of the network down into a summary, we can define a comparison or similarity measure between two networks by comparing their summaries, often an easier task. Here we discuss why and how to compare networks.


Defining and measuring *graph similarity* is an integral part of network comparison.

14.2.1 Why compare networks?

On occasion, a researcher working with network data may have a single network to deal with, but this is the exception and not the rule. Of course, multiple datasets, say from different experiments, will lead to multiple networks. But even if you are working with a single dataset, in general, you can expect to work with multiple networks.

The most obvious reason for encountering multiple networks in a single dataset comes from the upstream task (Ch. 7). Working from your original data, you defined processing steps that brought you from that data to the network now before you. Were those steps correct? What does the network look like if you change part of your upstream processing? Studying these questions will lead you to alternative networks, created by varying your upstream processing. And you will now be faced with multiple networks. Comparing them can give you insight into the effects of your network processing steps.

Another reason for encountering multiple networks comes when you want to understand different parts of a single network. In a multiplex or multilayer network, you may want to compare one layer against another. For example, how does the neural network derived from an organism's electrical junctions differ from the network formed by its chemical synapses [101]? Or how does a social network extracted from mobile phone data compare to one gathered from email records? You may also wish to compare different components of a disconnected network: how do the largest and second largest components differ?

Dynamic networks are a further situation where comparison can come into play. Suppose you have data describing how a network changes over time. How does the network look at the beginning of your data window compared to the end? For instance, what's happening in the social network of students over their summer break compared to during the school year? By taking "snapshots" of the network over different time periods, we reduce the complexity of the evolving network to a collection of static networks, and comparing those networks can help us understand how a network changes over time.

(Dynamic networks are of such importance we dedicate Ch. 15 specifically to them.)

14.2.2 Comparing pairs of networks

Given two networks G_1 and G_2, a comparison can be made by several means. Comparisons may be qualitative but usually we consider quantitative comparisons, meaning we seek a numeric measure of similarity (or, equivalently, distance) between the two networks.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

When nodes are consistent between G_1 and G_2, such as when $V_1 = V_2$, we can compare their edge sets directly. This can be as simple as counting the number or fraction of edges present in either network that are present in both networks (e.g., *Jaccard index*). A more challenging problem is when we do not know that nodes are the same between the two networks, but we still have options. This problem has been studied extensively in the field of *graph kernels*, which refer to a class of functions that can compute the similarity of two graphs. Here, we do not discuss graph kernels in detail but instead introduce a few simpler approaches.

Using statistics or features Earlier, we discussed summarizing networks by computing a set of statistical quantities or features. This summary can also power a comparison measure. For network G_i, let $\mathbf{v}_i$ be a vector of statistics. Then, introduce a vector *similarity function* $s(\mathbf{v}_i, \mathbf{v}_j)$ that compares pairs of vectors. Vectors are given a higher value of similarity by this function the more similar, or closer, that they are. If every statistic is numeric, this could be a cosine similarity or other common measure of similarity (or distance), but you may also find it helpful to define a bespoke similarity function. For instance, if one of your statistics is a binary quantity, such as a true or false attribute, you may want to build that into your similarity function. In general, a lot of flexibility is introduced with all the possible similarity functions; we recommend taking care in your choices and try to pick a function that is meaningful for your application as well as simple and interpretable, to the extent possible.

One particular feature that can power comparison is the distribution of motifs (Sec. 12.6). First, select a set of motifs that are relevant to your use case. Then compute the number of occurrences of each motif in each network. This gives you two vectors of counts, one for each network, which you can then compare against using a similarity function. (Often you want to normalize the vectors so they sum to one or have unit "length.") Comparison with motif counts can be quite effective [381, 427].

Using network portraits The distance or shortest path length distribution is an important summary of a network's topology, capturing a wealth of information on the organization of the network at all scales. This distribution powers network portraits (Ch. 13), a useful tool for visualization. But portraits can also be used for network comparison: a similarity between two networks can be defined using a similarity between their corresponding portraits. Given that a portrait is a graph invariant (isomorphic graphs have the same portrait) [27], comparison using portraits is quite natural—no matter how we "label" or "order" the nodes in the network, we get the same portrait similarity. Bagrow and Bollt [27] define and explore a distance measure for comparing portraits.

Using embedding Similar to comparing with features, any network embedding[8] can power a comparison (in fact, summarizing a network with a set of statistics is *essentially* a kind of embedding). More generally, an embedding maps each

[8] Here we mean an embedding of the entire network, as opposed to an embedding of all the nodes, say, within a network.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

network to a position in the *embedding space*. Then comparison follows by defining a similarity measure or distance between the positions of two points[9] in the embedding space. We discuss embedding methods further in Ch. 16 and Ch. 26.

This is not an exhaustive list of comparison measures, but covers most practical approaches. One missing measure is *graph edit distance*. This quantity counts how many edits (edge insertions, deletions, etc.) must be made to one network to turn it into the other network.[10] Edit distance was quite popular, but has somewhat fallen out of favor, at least for the larger networks we typically study, as computing it is far too expensive for such networks.

From any of these approaches, one can define a similarity measure $s(G_1, G_2)$ to quantify just how similar are the two networks. Usually s is non-negative, $s \geq 0$, with larger values for more similar networks, but some similarities are normalized such that $s = 1$ for identical (or at least maximally similar) networks and $s \to 0$ as network pairs become maximally different.

14.3 Clustering networks

One particular application powered by comparison measures is network clustering. Specifically, given a large collection of networks, clustering means to group related networks together. By determining the relationships between the different networks, we can learn more about the properties of those networks—what similarities or differences they have, what do "typical" networks in the collection look like, whether some networks are highly unusual, and so forth. And a network comparison measure determines whether or by how much two networks are related.

Here network clustering should not be confused with clustering within a network, measured with transitivity or the clustering coefficient. An unfortunate collision of terminology.

Suppose you have n networks $G_1, G_2, \ldots, G_n$. For each pair of networks G_i, G_j, $i \neq j$, choose a comparison measure s and compute $s_{ij} := s(G_i, G_j)$. We assume that more similar network pairs have a higher value of s_{ij}. Otherwise, if the comparison measure is interpreted as a distance or dissimilarity measure, we can rescale or transform it in some way, for instance by $s_{ij} = 1/d_{ij}$ or $s_{ij} = d_{\max} - d_{ij}$, so that larger values indicate more similar networks. We also assume that s is a symmetric function ($s_{ij} = s_{ji}$) and that $s_{ij} \geq 0$.

Given our comparison measure which we now interpret as a similarity measure, we can construct from the n networks an $n \times n$ similarity matrix $\mathbf{S}$—this matrix is then sent to clustering methods. The similarity matrix is the key here. While any of a variety of

[9] Here we're talking about pairwise comparisons, but other statistics could be useful. Suppose you have a n networks, looking at the average of the positions (the *centroid*) of their n embedding points can tell us how atypical a network is.

[10] Specifically, the minimum number of edits to make the two graphs *isomorphic*.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

clustering methods can be used,[11] as we discuss below, it is the choices that go into the similarity matrix that play the strongest role. As you work, consider both the effects of the clustering algorithm you are using and the features that go into the similarities.

Briefly, we discuss two possible classes of clustering methods to use on the similarity matrix.

14.3.1 Hierarchical clustering

Hierarchical clustering goes beyond finding groups to finding subgroups and supergroups, and builds a hierarchical tree (Sec. 12.8) relating, in our application, the n networks. Hierarchical clustering methods fall into two categories, agglomerative or divisive, based on whether you begin with n different clusters each containing a single point (network) that you iteratively merge together to build a tree, or you begin with one cluster containing n points that you iteratively divide apart to build a tree.

Here we describe an agglomerative method. Initially, each network is in its own cluster and our similarity matrix describes the similarities between any pair of clusters. Take the two most similar clusters (the maximum off-diagonal entry in our similarity matrix) and merge those clusters together. Now we have $n - 1$ clusters, with one cluster containing two networks. We can then repeat this step, merging the next two most similar networks, and the next two after that. But this describes merging two individual networks into a cluster of two. What happens when we want to merge two clusters that themselves contain multiple networks? In principle this may happen while we still have unmerged individual networks. In fact, it might be quite common. Imagine the two most similar networks are actually tied, or nearly tied with a third network, forming a triple of very similar networks. We may want to merge those three networks together into a cluster before we begin building clusters with the other networks. This is where the hierarchical nature of the clustering becomes so useful, giving us new information on the relatedness of the set of networks.

To use the similarity matrix to merge together clusters, and not just individual networks, we define a *linkage criterion*. First, represent each cluster as a set (an unclustered network can be considered a set of size 1). Then, for any two clusters U and V, the linkage criterion we use to determine whether to merge them is a function of the similarities of the networks $u \in U$ and $v \in V$. For example, *single-linkage clustering* uses

$$\max_{u \in U, v \in V} s_{uv}. \tag{14.1}$$

Here we look for the largest similarity among each pair of networks with one network in either cluster. At each step of the algorithm, we determine which pair of clusters has the largest criterion and we merge those clusters. Repeating these mergers then builds the hierarchical tree of networks. Thus the linkage criterion (single linkage is one of many) acts as a function mapping us from pointwise similarity to clusterwise similarity. Single-linkage clustering is the prototypical agglomerative hierarchical clustering method.

[11] The prototypical clustering algorithm k-means (Sec. 25.7) is not directly applicable, however. Here we have the matrix of similarities between n points, but k-means requires the points themselves, their coordinates in a space. That said, one method we discuss will create such points which we then cluster using, for example, k-means.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

Note that many common clustering methods, in particular k-means clustering, require more information than the similarity matrix. The k-means method, for instance, works by placing k new points, known as "centroids," down alongside the n points to be clustered, and then iteratively moving the centroids until they are located at the centers of k clusters. Here the actual coordinates of the n points being clustered are needed, because you are computing distances not between other points (which is effectively given by transforming the similarity matrix) but between points and the centroids, which are not present in the similarity matrix.

Such a limitation can be overcome by creating a meaningful space in which to position your networks. This is the approach used by spectral clustering, which "embeds" the networks based on eigenvectors and then applies a clustering algorithm such as k-means. Keep in mind that such an embedding may be unneeded, if you can go back from your similarity matrix to the underlying similarity measure. For example, if you defined a similarity based on a set of network summary statistics, you can keep those statistics as the "space" in which the centroids are placed and moved about. All this is to say, as you compare networks, don't throw the baby out with the bathwater.

14.3.2 Spectral clustering

Working from the $n \times n$ similarity matrix $\mathbf{S}$, spectral clustering uses its matrix properties to extract information about the clustering of points. Generally, another $n \times n$ matrix is computed from $\mathbf{S}$, and the k most "important" eigenvectors of this matrix are assembled into an $n \times k$ matrix. This new matrix represents the n networks in a k-dimensional space, and finally we can now apply a clustering method to the points in this space. Even more, the similarity matrix is often modified such that it become what is in effect another network! Nodes in this network, however, represent entire networks from the original n networks. Such an interpretation is often fruitful.

Many questions come to mind. What matrix is computed? Which eigenvectors to use? Which clustering method to use? All are worth asking. The standard approach is to threshold the similarity matrix to make an adjacency matrix $\mathbf{A}$ for a new network:

$$A_{ij} = \begin{cases} 1 & \text{if } S_{ij} > \sigma, \\ 0 & \text{otherwise,} \end{cases} \tag{14.2}$$

where σ represents some similarity threshold. The graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the degree matrix, a diagonal matrix with diagonal entries $D_{ii} = \sum_j A_{ij}$ is commonly used, as the eigenvalues and eigenvectors of $\mathbf{L}$ capture lots of information about the new network we've made (Ch. 25). Taking the first k eigenvectors of $\mathbf{L}$ and making a $n \times k$ matrix $\mathbf{U}$ whose columns are eigenvectors of $\mathbf{L}$, we then apply k-means clustering to points that are the rows of $\mathbf{U}$. Several alternatives and options exist, but this is the core idea behind spectral clustering.

We discuss spectral clustering (and k-means) in greater detail in Ch. 25 (Sec. 25.7). For more information on the broader problem of clustering, please see our bibliographic remarks at the end of this chapter.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

14.4 Summary

Networks are complex and we rely on tools to understand them. Summarizing such complexity, with rare exceptions, must lose information about the network. Our goal when searching for the best summary is, in effect, to lose only inessential information, where our problem or purpose controls what is and what is not essential. It is worth always keeping in mind what information is essential and what is not, to guide us towards worthwhile summaries.

To complicate matters even more, networks rarely exist in isolation and many research problems have multiple networks to study. Comparing different networks is often a key step to address a research problem. Network summaries connect directly with network comparison, as many comparison methods work by first summarizing the networks then comparing the summaries.

Bibliographic remarks

The roots of network summaries and network comparison extend back well into questions from graph theory, including graph edit distance and graph matching. Gao et al. [172] provide a useful, recent survey on graph edit distance while Conte et al. [116] review the landscape of graph matching.

Liu et al. [278] give a wide-ranging survey of network summarization. While a wealth of algorithms are available, more work is needed, including standards and best practices, when using network summaries to report on data [22].

Network comparison and graph similarity are likewise broadly studied areas. Readers may wish to consult Hartle et al. [203] for a recent survey and Soundarajan et al. [440] for useful guidance on comparing networks. Ghosh et al. [181] is a recent survey of graph kernels.

For those interested in learning more on clustering, a general and foundational problem in statistics and machine learning, we recommend Xu and Wunsch [499], von Luxburg [478] on spectral clustering in particular, and Kaufman and Rousseeuw [238].

Exercises

14.1 (**Focal network**) Take a focal network of interest and summarize it using (i) a brief, written description and (ii) 3–5 network statistics (including some beyond those used in this chapter). For the statistical summary, choose statistics that capture distinct aspects of the network's structure. Provide a brief justification for each statistic.

14.2 Define (with justification) an appropriate similarity measure using the statistics from Ex. 14.1 According to this measure, which two focal networks are most similar? Least similar? (Given the focal networks, is this question even meaningful?)

14.3 (**Focal network**) The plant–pollinator network is bipartite. A basic summary using statistics may need to be adapted for such a bipartite network. Describe

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press

a few ways to specialize a summary for a network that is bipartite. What is the most important difference you should capture in the set of statistics compared to summarizing a unipartite network?

14.4 Comparing networks using summary statistics of network features may require normalizing those features: if one feature covers a much larger range than another, it may dominate in the final distance or similarity calculation. Consider the summary table shown in Sec. 14.1.1.

(a) Which of those statistics may end up overwhelming the others and why?

(b) Provide a normalization scheme to correct for this.

https://doi.org/10.1017/9781009212601.017 Published online by Cambridge University Press