On Aggregation of Network Multiplexity

Haochen Wu
hcwu@ucdavis.edu
Department of Computer Science

Abstract

In this project, we studied the consequences of aggregating two layers of network together. By investigating the simulation of the networks of chaotic maps, we compared the difference between aggregating networks that sharing the same edge property and networks that having the same topology. Simulations show that the latter might be more appropriate to aggregate.

1 Introduction

Many dynamic systems that involve interacting entities can be modeled as networks. By this method one can relate a lot of different systems and it has been proved to be beneficial to use this representation. However, recent study of network science shows that the simplest single layer network may not capture enough features and a multiplex extension is necessary in some cases [1] [2]. This kind of extension can distinguish the different kind of relationship between entities and group them into different layers of network thus provide more information about the original system.

The above arguments raised an important question: what is a proper multiplex representation. In some sense, each edge of a network is unique. For example, in a social network, the relationship between any of two individuals differs in some aspects. It's not possible to find two pairs of people that have exactly the same relationships. We could, for sure, let each edge be in a different layer, but this kind of formulation won't give us any insight about the network. Alternatively, we could also just ignore the difference between all the relationships and put all of them in a single layer network, but this is shown to be insufficient for some analysis. Therefore the right thing to do is to group these relationships into different categories and use those categories as different layers of network. Then we go back to the question we asked at the beginning of this paragraph.

The answer to this question depends on the circumstances and the purpose of using the network. Currently this is done by applying domain specific knowledge or grouping by well recognized standard. However, there is no guarantee that this will give the correct behavior. It is beneficial to have a principled way to achieve this. Recently there are only some exploratory work[3] there.

There are two situations that things can go wrong. The first one is that we have more layers than we actually need. More layers typically means more complex, it will require more computational resources and prone to over-fitting. More common case is that we have less layers than we need. There is really no good ways to deal with it other than to add some information and create more layers. We will need to know what is the information we should use to create them.

In this project we are trying to understand when two layers of network can be treat as just one. We will compare the difference between structure and edge property heterogeneous and try to answer the following question: Is it more appropriate to aggregate the network with similar structure or the network with similar edge property?

2 Background

Previously we've done some work trying to aggregate US airline network based on the similarity of network structure using the domestic flight data from the Bureau of Transportation Statistics. The network is quite intuitive. Each node is an airport, different carriers form different layers. If there is a flight run by some carrier flying between two airports, we add an edge in that carrier's layer and the weight of the edge will be the number of passengers in that flight.

We investigated how the passengers distributed among all those edges in different layers and trying to aggregate those layers with similar distributions together. The Jensen-Shannon entropy between two layers is defined as:

$$D_{JS}(\mathcal{N}_{L_1} \parallel \mathcal{N}_{L_2}) = \frac{1}{2}D_{KL}(\mathcal{N}_{L_1} \parallel \mathcal{N}_{L_{12}}) + \frac{1}{2}D_{KL}(\mathcal{N}_{L_2} \parallel \mathcal{N}_{L_{12}}) \qquad (1)$$

where $\mathcal{N}_L$ is normalized weight distribution over edge set. and L_{12} refers the network generated by aggregating L_1 and L_2. This metric can be understood as the information loss in aggregation of the two individual layers.

Combining this metric and the hierarchical clustering algorithm, we created a dendrogram (See Figure 1) of how the carriers are grouped together.

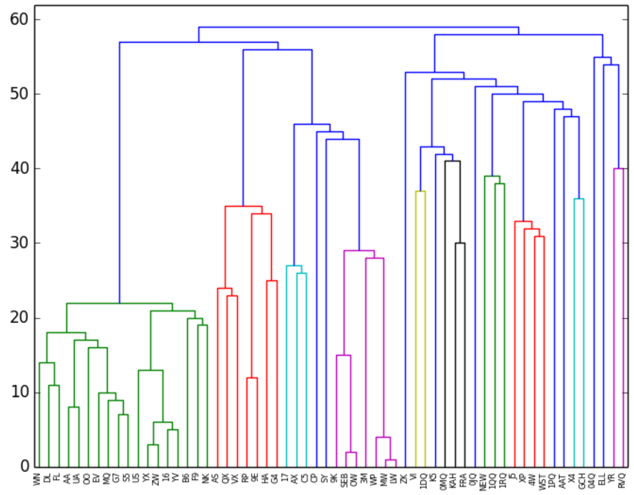

Figure 1: Dendrogram of hierarchical aggregation.

This approach showed that we could use some information measure to reduce the network model. However, the result is far from satisfying. We don't know what kind of process will have similar result in the reduced network. Therefore we can't apply this method to a more realistic application. More properties about these kind of aggregation must be studied before we can actually use it.

3 Dynamical System

In this project we use networks of 1D chaotic map as a proxy of real networks. In those networks, each node has a chaotic map and they are connected through edges by coupling. For node i, the value x_i at time $t+1$ is:

$$x_i(t+1) = (1-\epsilon)f(x_i(t)) + \frac{\epsilon}{k_i}\sum_{(i,j)\in G} f(x_j(t)) \tag{2}$$

f is a chaotic map, edge (i,j) is linear coupling and k_i is the degree of node i.

In order to have a multiplex setting, we adopt a time-division multiplexing approach. For each time step, the nodes' values are only updated by the edges in one single layer.

$$x_i(t+1) = (1-\epsilon_{L(t)})f(x_i(t)) + \frac{\epsilon_{L(t)}}{k_i} \sum_{(i,j)\in G^{L(t)}} f(x_j(t)) \tag{3}$$

All the notations are the same as equation 2. $L(t)$ is a function to pick a specific layer for time t.

In the simulations, we used the logistic map with r = 4.0.

$$f(x) = rx(1-x) \tag{4}$$

4 Methods

As the first step of understanding the aggregation, we focused on the aggregation of a two-layer network, trying to figure out the consequences of aggregating them together. Specifically, two different settings are studied and compared. One setting is two layers with different structures but the same coupling parameter. Another is two layers with identical structures but different coupling parameter.

We used the parity of time step t as the function to pick layer, therefore the edges in two layers will take effects alternately.

$$L(t) = \begin{cases} Layer\ 1 & t = 2N+1 \\ Layer\ 2 & t = 2N \end{cases} \tag{5}$$

In the first scenario, the two layers of network are independently generated and share a same coupling parameter ϵ. We choosed $\epsilon = 0.05$ for the simulation. In the second one, the edges in two layers are identical but using different coupling parameters. $\epsilon_1 = 0.05$ and $\epsilon_2 = 0.1$. The aggregated network has $\epsilon = 0.075$.

The smallest two-layer network that has different structure yet the same number of edges for each layer is a 3-node network. Therefore we started from this simplest case with two edges in each layer. Then we come to the Erdos-Renyi random network with 100 nodes. The topology of network can be found in next section.

The quantity we used to measure the consequences are calculated from the sequence generated by the chaotic maps. In order to find out excess entropy, we first convert the real value to a finite alphabet using the generating partition of our logistic map given by:

$$\sigma_i(t) := \begin{cases} 0 & 0 \le x_i(t) < \frac{1}{2} \\ 1 & \frac{1}{2} \le x_i(t) < 1 \end{cases} \tag{6}$$

Each node starts from a random value and then iterates for 50,000 times. This gave us a 100,000-symbol sequence for each nodes. We used the excess entropy calculated from length 10 as a proxy for the process's property. According to the simulation, it seems the excess entropy hasn't converged yet, but generating more data require more computation resources, and since we have similar dynamics in all the nodes, the excess entropy at length 10 can give some information.

5 Results

Figure 2 3 4 and 5 show the results of the simulation. In each figure, the upper one shows the original two-layer network and the lower one shows the aggregated network. The color of edges indicates the layers they belong to. Blue and red edges are those only exist in one layer and purple ones are those in both layer. The color of nodes gave the information of estimated excess entropy. Red ones have higher excess entropy than blue ones.

For three-node network, when the two layers have different topology, the node coupled with different node in different layer has a higher excess entropy. Once we aggregated two layers of network together, all three nodes are symmetric therefore they will all have similar excess entropy.

When the two layers of network have same topology, the node in the middle always has a lower excess entropy no matter whether they are simulated in one layer and two layers.

Similar observations can be made for the Erdos-Renyi network with 100 nodes. When aggregated two layers of network with same edge property together, the excess entropy for different nodes changed without obvious pattern. However, for the network with the same topology in different layers, most nodes stay at a similar excess entropy after aggregation.

From those simulation results, it seems that the dynamics changed a lot when we aggregate two layers of network with the same edge property but different structures. Meanwhile, aggregation of two layers of network with the same topology but different edge property gives a good approximation.

6 Conclusion

It is hard to draw a final conclusion from such a simple simulation with a lot of arbitrary assumptions. However, the results still shed some light on how we could approach the bigger problem of aggregating multiplex networks. Grouping the network layers with similar topology is hypothesized to be a

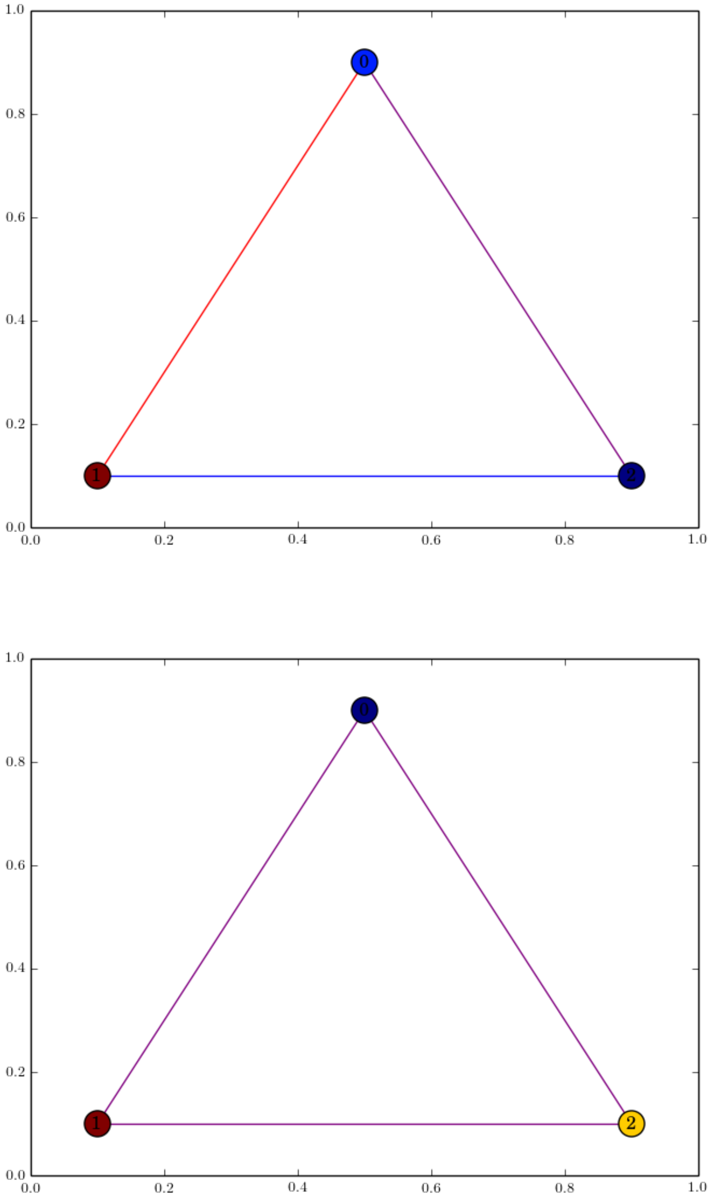

Figure 2: Small network with same edge property.

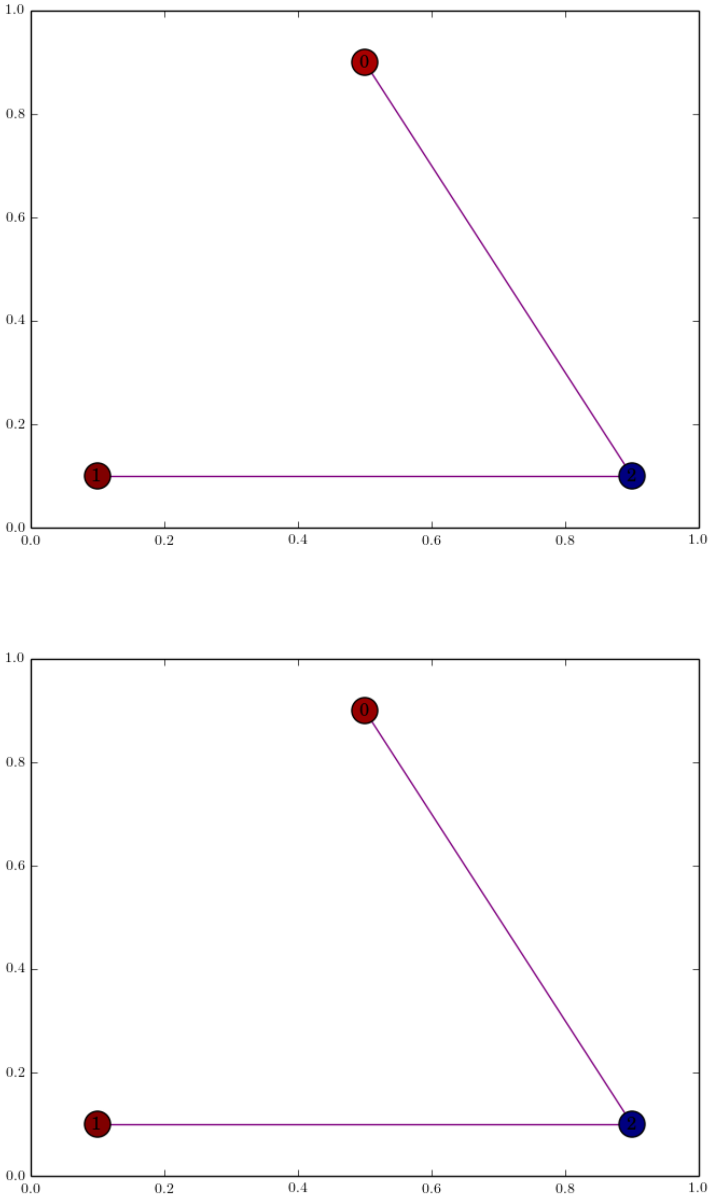

Figure 3: Small network with same topology.

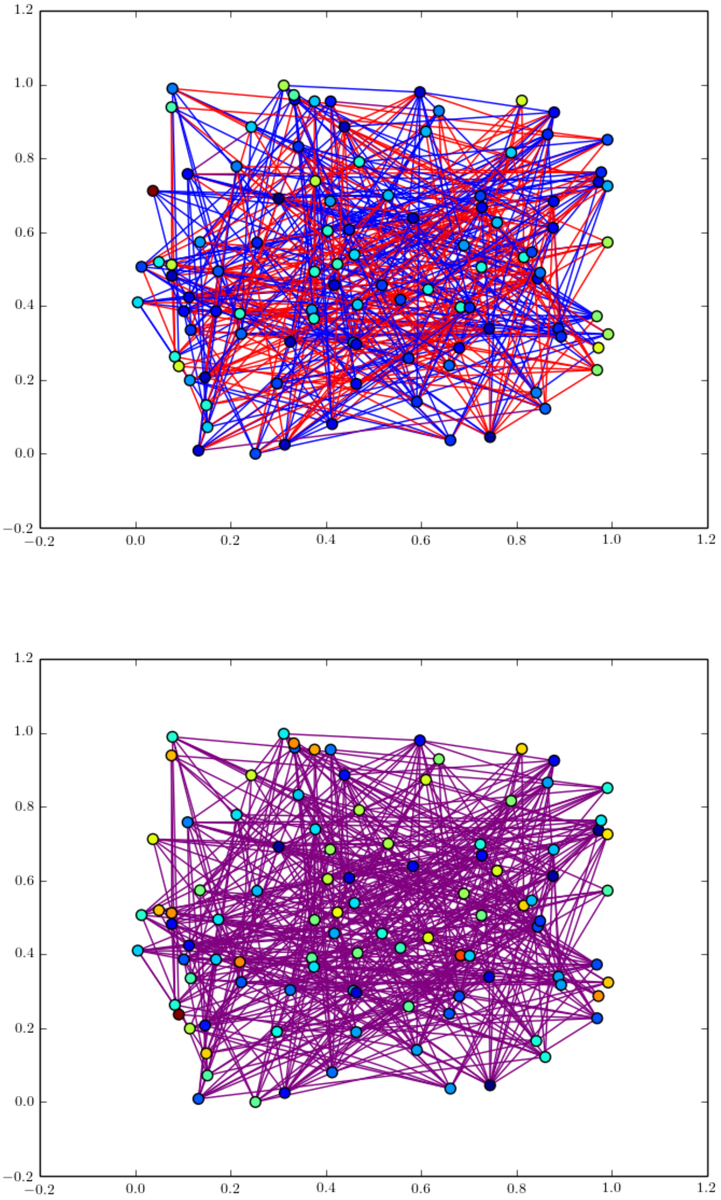

Figure 4: Random network with same edge property.

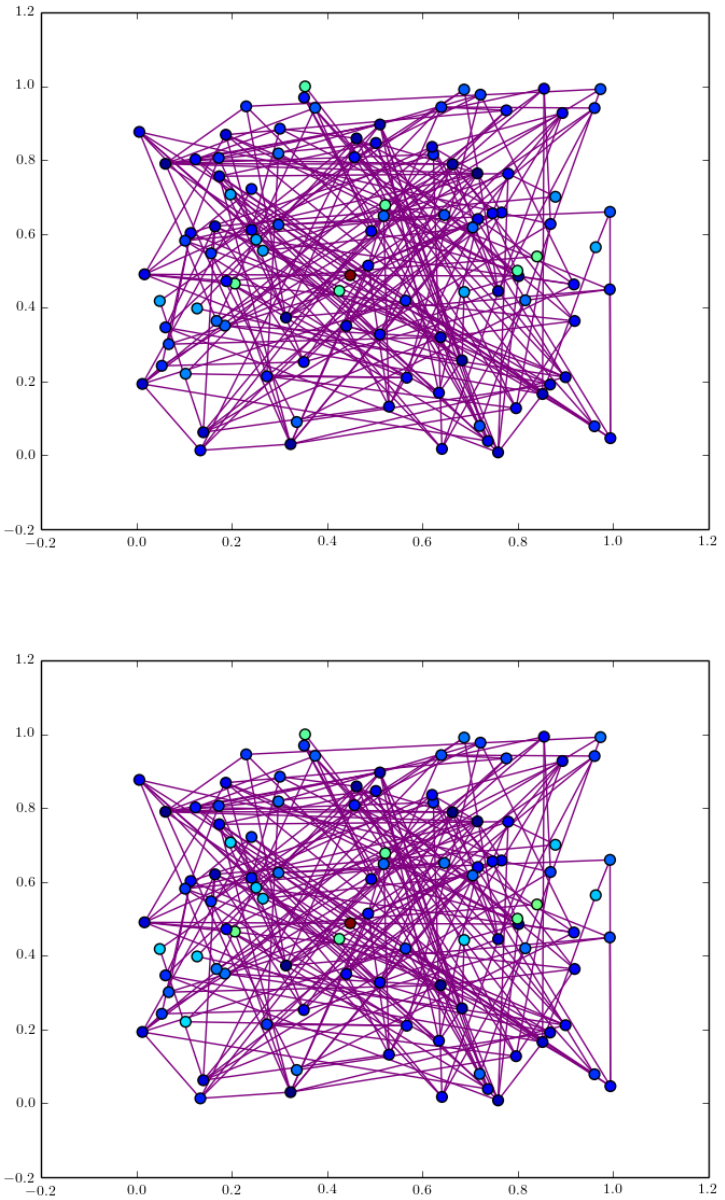

Figure 5: Random network with same topology.

promising way of reducing the network model. The future work could follow this line and try to validate this assumption with real data.

References

[1] Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco Del Pozo, and Stefano Boccaletti. Emergence of network features from multiplexity. *Scientific reports*, 3, 2013.

[2] Massimiliano Zanin. Can we neglect the multi-layer structure of functional networks? *Physica A: Statistical Mechanics and its Applications*, 430:184–192, 2015.

[3] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature Communications*, 6, 2015.