Initial thoughts - Finding communities using the yelp dataset.

Overall goal: Using temporal information in a dynamic network to derive information about static communities with varying activation profiles. We assume that over the timescale sampled, community membership is fixed relative to the activity of the communities themselves, which can vary in activity or be on/off.

1. Motivation: Why is this an important project? What are the practical applications?

A number of techniques exist to find communities in static graphs, however less work has been done on dynamic communities. There are a number of settings where fixed communities with varying activation may be a good model: connections within the brain, which themselves are relatively static but are active dependent on stimuli; a workplace email network, where the employees/management is fixed but communication is dependent on important events; cellphone call networks, where the number of people newly acquiring a phone is very small over a short period, but call volumes (especially between individuals) vary over the day/week; and many other potential settings. Incorporating time directly gives us access to the increased data available in multiple timesteps, but aggregating data means the algorithm is sensitive to noise, or may miss communities that are strongly connected but rarely active. The activation profiles themselves can be useful, for instance in classification tasks. Consider fMRI data on individuals in different categories (say healthy and (pre-)Alzheimer's); activation profiles of different neural structures could differ enough between groups to be able to classify them, which could have important medical uses.

2. Project definition:

Formal (using mathematical symbols) definition of the following:

2.1) Input data

A sequence of networks or graphs G_t consisting of nodes which are connected if a communication has been observed between them at time step t. We can represent this as a sequence of adjacency matrices X^t, where $X_{ij} > 0$ (which can be either binary or a count of communication events) if we observe a communication between nodes i and j at time t (this can be aggregated internally, in that “at time t” may mean “between times t-1 and t”)

2.2) Output solution

The result of a community detection should give both the community structure and the activation profile of each community. The community structure can be represented as an $n \times k$ matrix C, where C_{ik} gives the weight of node i’s membership in community k. An activation function $A(t)$ yields a $1 \times k$ vector at time t, where $A(t)_k$ gives the activation level of community k (which can be either binary or weighted as the community memberships).

2.3) Measure of quality of the solution (if applicable)

The method here produces what is essentially a compression of the original dynamic network into communities and their activations. The quality of this compression can be measured by decompressing and comparing to the original sequence of graphs - we use C and $A(t)$ to generate a sequence of adjacency matrices $\hat{X}^t$, and can derive a measure of fit as $\sum_t \left| X^r - \hat{X}^t \right|$

3. Evaluation plan

3.1) Baseline techniques (E.g. solutions from the book, random solutions, other naive solutions)

At the very least we should perform better than random (i.e. generate random $\hat{X}^t$ and compare to the original)

We also want to compare to methods that do not consider time in the same way:

Cluster the network after aggregating all time steps into a single one using an existing community detection method, generate a sequence of $\hat{X}^t$ ignoring the concept of activation

Cluster as above, then fit the best activation profile for each community based on available temporal data. This will likely be more sensitive to noise

3.2) Key metrics (this could be simply what you have in 2.3, but also time of execution, quality of approximation if approximate solution, etc.)

The quality measure of 2.3 is key

Since the number of communities k is not known, we may want to consider how the quality changes with respect to k (and how that curve compares to the baseline solutions) - can we achieve good quality with relatively small k, since otherwise the compression doesn't compress much at all?

We can measure execution time, however our method may lose out by virtue of complexity; we likely want to focus on accuracy gains for now

3.3) Data description, where is your dataset, how big is it (number of entities, etc), basic statistics

Source:

Dataset Yelp Academic dataset.

Can be downloaded at : https://www.yelp.com/dataset_challenge/dataset

Data Set Information:

The dataset Is divided into 6 files, all in JSON format. The relevant files are :

yelp_academic_dataset_user.json

Data Set Characteristics:	Multivariate	**Number of Instances:**	686556	**Area:**	Social
Attribute Characteristics:	Categorical, Integer, String, Float	**Number of Attributes:**	23	**Missing values**	Yes

Attributes Information :

"user_id":"encrypted user id", "name":"first name", "review_count":number of reviews, "yelping_since": date formatted like "2009-12-19", "friends":["an array of encrypted ids of friends"], "useful":"number of useful votes sent by the user", "funny":"number of funny votes sent by the user", "cool":"number of cool votes sent by the user", "fans":"number of fans the user has", "elite":["an array of years the user was elite"], "average_stars":floating point average like 4.31, "compliment_hot":number of hot compliments received by the user, "type":"user"

yelp_academic_dataset_review.json

Data Set Characteristics:	Multivariate	**Number of Instances:**	2685066	**Area:**	Social

Attribute Characteristics:	Categorical, Integer, String	**Number of Attributes:**	10	**Missing values**	Yes

Attributes Information :

"review_id":"encrypted review id", "user_id":"encrypted user id", "business_id":"encrypted business id", "stars":star rating, rounded to half-stars, "date":"date formatted like 2009-12-19", "text":"review text", "useful":number of useful votes received, "funny":number of funny votes received, "cool": number of cool review votes received, "type": "review"

yelp_academic_dataset_business.json

Data Set Characteristics:	Multivariate	**Number of Instances:**	85901	**Area:**	Social
Attribute Characteristics:	Categorical, Integer, String	**Number of Attributes:**	23	**Missing values**	Yes

Attributes Information :

"business_id":"encrypted business id","name":"business name", "neighborhood":"hood name", "address":"full address", "city":"city", "state":"state -- if applicable --", "postal code":"postal code", "latitude":latitude, "longitude":longitude, "stars":star rating, rounded to half-stars, "review_count":number of reviews, "is_open":0/1 (closed/open), "attributes":["an array of strings: each array element is an attribute"], "categories":["an array of strings of business categories"], "hours":["an array of strings of business hours"], "type": "business"