**Project1 - venture capital(vc) industry network analysis**

**1.dataset description:**

**firm:** string, identifier of specific firm being invested

**vc:** string, identifier of specific VC making investment

**invtime:** date of investment event, year-month-day, including year 1991-2016

**ctg1:** string, lower level sub-industry category the firm belong to

**ctg2:** string, higher level sub-industry category the firm belong to

**round:** string, investment round; including A,B,C,D,E,F,G,PIPE, angel, Pre- A, strategy, board; 707 out of 44887 are missing value.

**2.project description:**

**2.1 VC network (mono-partite)**

**2.1.1 Code to draw the network**

The node is each VC (visualize the node size in total investments number)

There are 2\*3\*2\*25 whole networks (not divided by category) and 2\*3\*2\*25\*i specific networks(divided by category):

(no need to actually produce the jpg version as long as there is the python code)

**WHOLE NETWORK\_(e,s,c,t) and SPECIFIC NETWORK \_****(e,s,c,t,ctg) for sub-divided specific category:**

* **e---two way of Edge type**

Edgetype1: an edge exists if two VCs invested together in the same firm.

Edgetype2: an edge exists if two VCs invested together in the same firm for the same round.

*(Note here edge has weight because multiple edges can exist between same pair of VCs if they co-invest more than once. please visualize the edge weight )*

* **s---three way of time span:**

timespan1: network\_t includes all the edges formed before time t

timespan2: network\_t includes all the edges formed 10 years before time t

timespan3: network\_t includes all the edges formed 5 years before time t

* **c---two ways to identify the category either in in classification criteria 1 or classification criteria 2**

ctg1 means network is divided by classification criteria ctg1

ctg2 means network is divided by classification criteria ctg2

* **t---**(t=every half year from 01/01/2005 to 01/01/2017 (25 snapshots in total), e.g.01/01/2005; 06/01/2005….)
* **ctg---specific categories**

ctgi means network is limited to all investments that are from the specific “ctgi” category (there are new ctgi coming in in different time)

**2.1.2 Code to generate network attributes and the outputting txt file:**

* **network level: density and clustering coefficient in WHOLE NETWORK\_(e,s,c,t) and SPECIFIC NETWORK \_(e,s,c,t,ctg)**
* **node level: degree centrality, closeness centrality, betweenness centrality,** **eigenvector centrality ( Bonacich’s (1987) eigenvalue measure; Katz centrality, pagerank, HITS algrithm),** **structural hole ,clustering coefficient**
* in WHOLE NETWORK\_(e,s,c,t)
* in SPECIFIC NETWORK \_(e,s,c,t,ctg) : node attributes for each vc are generated from the specific networks whose categories that the vc belongs to.

e.g. if vc 1 belongs to category 010100 and 020200 in time t, it’s degree centrality is generated from the WHOLE NETWORK\_(e,s,c,t) and SPECIFIC NETWORK \_(e,s,c,t,010100) and SPECIFIC NETWORK \_(e,s,c,t,020200)

* can you think of any other node attributes that is often used? skip the red attributes if it is not applicable.
* **node level: change of eigenvector centrality in above measure**

change of eigenvector centrality in t= (eigenvector centrality in t) – (eigenvector centrality in t-1)

* should I use the standardization of eigenvector centrality in the calculation?
* This is a simple measure, I am looking for better algorithm to capture the centrality change in dynamic network which can be discussed later
* **possible title of the table (or can you think of better format to store the data?):**
* for WHOLE NETWORK\_(e,s,c,t)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vc | Edge type | Time span | T | classification | density | degree centrality | closeness centrality | Between  ness centrality | eigenvector centrality | Change of eigenvector centrality | structural hole |
| vc | Edge type1 | Time span1 | 01/01/2017 | ctg1 |  |  |  |  |  |  |  |
|  |  |  |  | Ctg2 |  |  |  |  |  |  |  |

* for SPECIFIC NETWORK\_(e,s,c,t,ctg)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vc | Edge type | Time span | T | classification | category | density | degree centrality | closeness centrality | Between  ness centrality | eigenvector centrality | Change of eigenvector centrality | structural hole |
| vc | Edge type1 | Time span1 | 01/01/2017 | ctg1 | 010100 |  |  |  |  |  |  |  |
|  |  |  |  |  | 020200 |  |  |  |  |  |  |  |
|  |  |  |  |  | … |  |  |  |  |  |  |  |
|  |  |  |  | Ctg2 |  |  |  |  |  |  |  |  |

CTG1

Firm

VC

**2.2 Category similarity**

**2.2.1 Code to calculate the similarity matrix (asymmetric)**

1. **Similarity from category i to job category j:**

* where i and j represent occurrences of categories in categories i and j before time t (timespan1), respectively. Therefore, the similarity from categories i to j is equal to the number of times both category i and j occur in past history of a VC, all VCs summed, and divided by the total number of occurrences of category i before time t through history.
* Similarity is asymmetric, that is, category i can be more similar to j than j is to I; the diagonal are all 1; 0 < Similarityi,j,t< 1.
* I calculated this measure for categories with more than 100 total investments by time t. Categories with fewer than 100 investments were set at a minimum similarity of 0 from all other categories.
* There are three ways to calculate similarity in terms of what we define “co-occurrence”:
* **Measure 1 (Similarity1):** occurrence is number of investment that is belong to a specific type; time of co-occurrence of i and j on object k (VC) in Simijt equals to times of i occurences if i and j both occurred in k’s history.

e.g. if the sample in time t include vc 1,2,3 whose investment history is:

1: A A B A A B B A D C (successively before time t, different investments can be belong to same category)

2: A B A

3: B B D

4: A C E

Sim1A,B,t = 7/8 Sim1B,A,t = 4/6

* **Measure 2(Similarity2):** occurrence is number of investment that is belong to a specific type; time of co-occurence of i and j on object k (VC) in Simijt equals to times of the smaller number of occurences (of i and j) if i and j both occurred in k’s history。

in the above example, Sim1A,B,t = 4/8 Sim1B,A,t = 4/6

* **Measure 3(Similarity3):** occurrence is number of objects (VC) that has invested the specific type; time of co-occurence of i and j on object k (VC) in Simijt equals to 1 if i and j both occurred in k’s history.

in the above example, Sim1A,B,t = 2/3 Sim1B,A,t = 2/3

Here is a more specific calculation example:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Measure 1** | | | | **Measure 2** | | | | **Measure 3** | | | |
| VC investment history | A∩B | A | B∩A | B | A∩B | A | B∩A | B | A∩B | A | B∩A | B |
| 1: A A B A A B B A D C | 5 | 5 | 3 | 3 | 3 | 5 | 3 | 3 | 1 | 1 | 1 | 1 |
| 2: A B A | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3: B B D |  |  |  | 2 |  |  |  | 2 |  |  |  | 1 |
| 4: A C E |  | 1 |  |  |  | 1 |  |  |  | 1 |  |  |
| SUM | 7 | 8 | 4 | 6 | 4 | 8 | 4 | 6 | 2 | 3 | 2 | 3 |
|  | Sim1A,B,t = 7/8 | | Sim1B,A,t = 4/6 | | Sim1A,B,t = 4/8 | | Sim1B,A,t = 4/6 | | Sim1A,B,t = 2/3 | | Sim1B,A,t = 2/3 | |

1. **Distance** **from category i to job category j:**

1. **Visualize the distance map:**

I am not sure whether we can visualize it in Python, you may refer to following prescriptions:

“I used NetDraw (Borgatti 2002) and the MDS (multi-dimensional scaling) algorithm set to node repulsion to depict distance between job categories. The node repulsion option optimized distances between nodes to minimize overcrowding, thereby allowing visibility of the labels by de-emphasizing the close clustering of highly similar job categories. Because the two-dimensional space can represent only one distance, I display the average of the two category pair of asymmetric distances (i.e., Dij + Dji / 2). The closer two categories are, the more similar; the farther apart, the less so.”

1. **Distance history of VC:**

* For VC k’s measure of distance history at time t is the average distance it moved between categories, calculated by dividing the total distance it moved by the total number of categories (minus one) it completed. The total consecutive distance of the path is calculated by summing all distances of each step, from 1 to N (N being the total number of chronologically ordered categories completed by VC k at time t).

* There are two type of measures in terms of how we define the distance of each step:
* **Step Distance Measure 1: distance of step in is the average distance between category i and all previous categories.**

For example: VC 1: A A B A A B B A D C

VC 1’s distance history = (DisAB+ DisBA+ DisAB+ DisBA+ (DisAD+ DisBD)/2+(DisDC+ DisAC+ DisBC)/3)/6

* **Step Distance Measure 2: distance of step in is the average distance between category i and the last previous category. If category in equals jn+1, the distance is zero.**

for example: VC 1: A A B A A B B A D C

VC 1’s distance history = (DisAB+ DisBA+ DisAB+ DisBA+ DisAD+ DisDC)/6

* **possible title of the table (or can you think of better format to store the data?):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| vc | T | Ctg history | Similarity Measure | Step distance measure |  |
| vc | 01/01/2017 | A A B A A B B A D C | Similarity 1 | Step distance measure1 |  |
|  |  |  |  | Step distance measure2 |  |

**2.3. Regression eigenvector centrality change of VC on its distance historyk,t**

can use different type of regression model

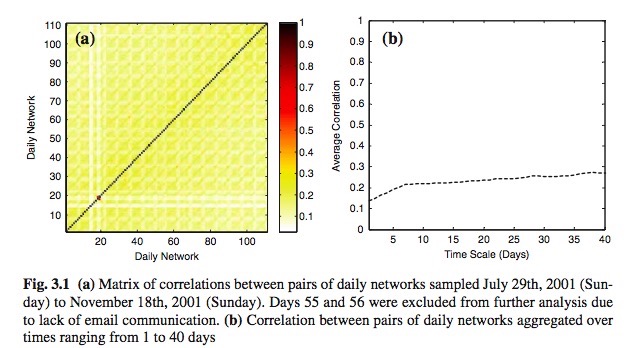
**2.4 Solution 1 to analyze dynamic network**

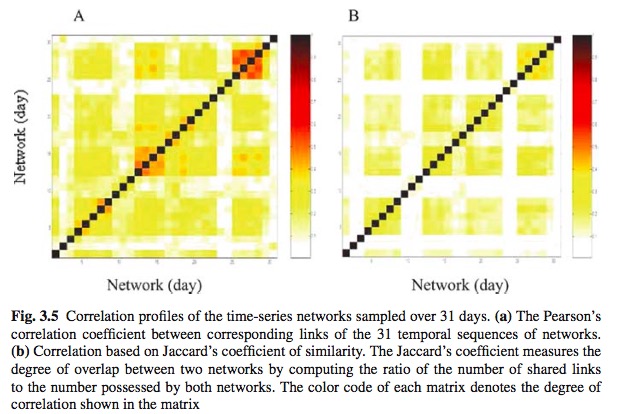
Previous we treat graph as static, this time we try to analyze the dynamics by simple solution in which we treat time as discrete and compare the evolution of network:

* all the following network would use edgetype1 and no category specification.
* discrete network: We set the time scale to one month, thus creating 216 discrete consecutive monthly networks. 01/1999 to 01/2017 (18\*12=216 in total, each graph includes nodes and edges that appear in the investment events during that specific period).

**2.4.1 network correlation**

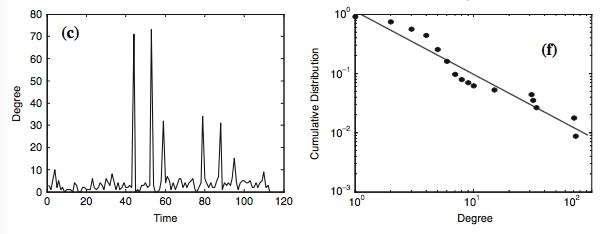
get the correlation **matric and figure1** between corresponding edges of the 216 monthly networks. And then get the average correlation when setting the time scale from 15days/1 month/2months/3 months/6 months/1year (we can define the correlation of different graph based on overlap of nodes or edges like Jaccard’s coefficient of similarity in the following second example, similar figure would be the following two examples.)





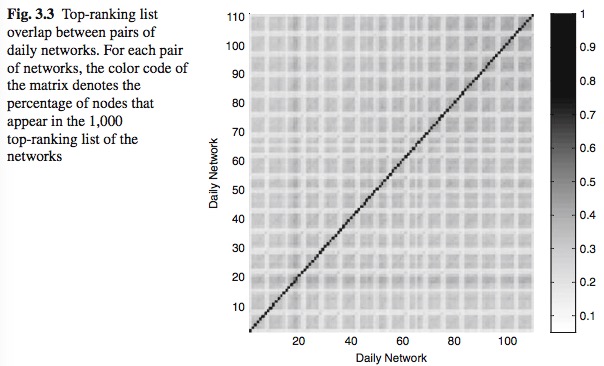
**2.4.2 degree distribution of nodes with high centrality**

First, we generate top 100 nodes (from aggregated network in that nodes with highest eigenvector centrality before 01/01/2017 with edgetype1 and timespan1, no category specification), Second, get the **table and figure2** of their eigenvector centrality(EC) from month to month(based on the discrete 216 networks) over the duration of 216 observations. Third, we get the corresponding distribution EC of each node and visualize it in figure3 (similar figure 2 and figure3 with only one node would be the following example).



**2.4.3 eigenvector centrality overlap**

First we identify the top n(n=10;50;100;500;1000) nodes for each of the monthly networks (216). We then get the percentage of nodes that appear in both top-ranking lists (“eigenvector centrality overlap”) for each pair of monthly networks (overlap=0 if some monthly network don’t have enough nodes, e.g. some discrete network’ nodes number < n ), then generate the five(n=10;50;100;500;1000) **matrix and following graph**(example).



**2.4.4 comparing discrete network and aggregate network**

We compare 216 monthly networks with the aggregated network (network with all the nodes and edges appeared before 01/01/2017 with edgetype1 and timespan1, no category specification). We can find which nodes in the monthly 1,000 top-ranking list also appear in the top-ranking list of the aggregate net- work, obtaining the binary image as the following example figure a(ignore b).

