Exploratory analysis of global banking exposure data

David & Will

February 27, 2022

Overview

1. Network properties
   - We analyse how several network properties change over time, such as most important nodes, and link these observations to real-world events
2. Measuring global systemic risk
   - We introduce a novel measure of global risk using a Markov chain model, as an extension to SinkRank

Approach

- We split the data into quarters and create a network for quarter
- We then analyse each of the quarters and create a time series to analyse changes in the network

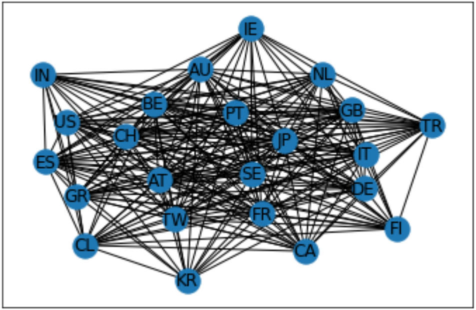
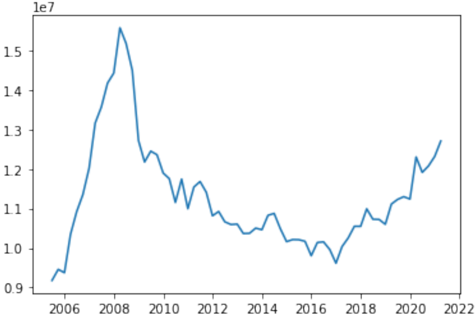


Figure: Example network

Total Value of Foreign Holdings of Assets

- We calculate the total value of assets held by foreign banks by summing up all the weights of a network in each period
- Note that this rises very rapidly up to a peak in 2008 and falls down after the financial crisis spreads worldwide and banks are less willing to lend to each other across borders
- This also links to the communities that are observed later



Most Important Countries

- We calculate most important countries by eigenvector centrality, which is a measure of how connected a node is to other connected nodes
- Dominant countries include the United States, the United Kingdom, and Japan

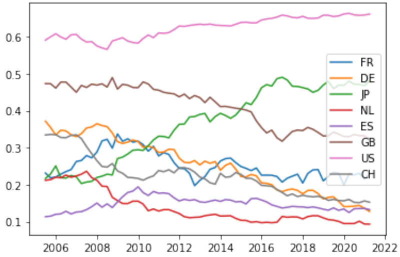


Figure: Most Important Countries

Most Important Countries

- Note the dramatic rise in the importance of Australia, with a noticeable jump during the 2008 financial crisis
- This may be because Australia was less impacted by the financial crisis compared to Europe and the Americas, so more banks are willing to deal with Australian banks in the wake of the crisis

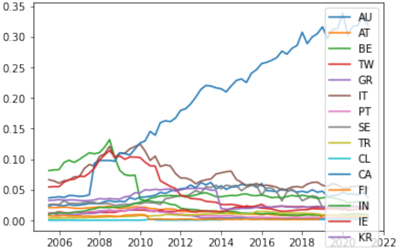


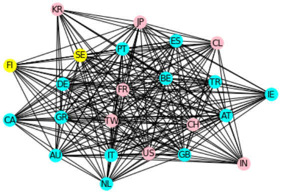
Figure: Least Important Countries

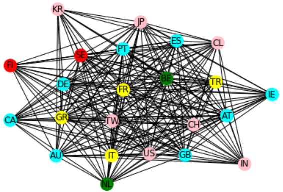
Community Structure

- We split countries in the network according to how much they are linked to each other
- We can then view how these groupings, and the number of groupings, change over time

Community Structure

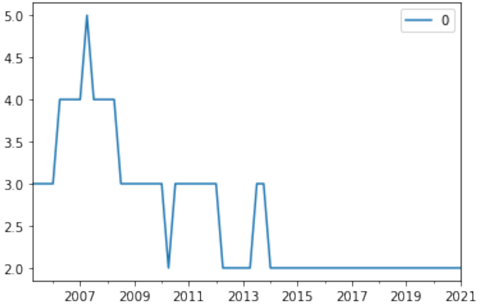
- Groupings occur mostly within region
- e.g. Finland and Sweden
- However, there are exceptions, such as France being included with the non-EU group in 2005
- Although not included, communities are relatively stable after 2014
- Figures for 2005 Q2 and 2007 Q2





Community Structure

- The number of communities in the network represents howfragmented the financial system is
- The number spikes around 2008, representing the increasing isolation of each countries' financial system
- The financial has become much more integrated again after2014 as the global economy recovers and bounces back



SinkRank+: model specification

- We introduce an extension of SinkRank to capture the level of global systemic risk at a year, using a Markov chain model
- In the context of this dataset containing 23 reporting countries, there are four parameters of this model:

$$I = \{1, 2, \cdots, 23\},$$

$$A = \{A_1, A_2, \cdots, A_{23}\},$$
$$P = \{p_1, p_2, \cdots, p_{23}\},$$
$$Y = \{y_1, y_2, \cdots, y_{23}\}.$$

Each i represents a node containing a country.
Each A_i is a 22×1 vector containing the volume of cross-border claims of country i against other countries.
Each p_i is a scalar of the chance of country i to default.
Each y_i is a scalar containing the GDP of country i.

SinkRank+: model specification

- The model is a complete directed weighted graph containing 24 nodes, including each of the 23 countries and an absorbing node representing the state of insolvency
- At node i, liquidity has a probability p_i of becoming insolvent and $1 - p_i$ of moving to another country
- Given liquidity has not become insolvent, it has probabilities $(q_1, q_2, \cdots, q_{23}) = A_i / \sum A_i$ of moving to each other country
- The probabilities of default p_i can be inferred using interest rate spreads, following from the no arbitrage principle
- Given p_i's are nonzero, liquidity eventually gets absorbed in the insolvent state after a finite number of steps

SinkRank+: model specification

- The SinkRank+ can be seen as the expected number of steps liquidity goes through before becoming insolvent
- We assume that the probability liquidity starts at a given node i is proportional to its GDP. Let this be denoted l_i
- It can then be calculated the expected number of steps it takes for liquidity to become insolvent, starting from each node i. Let this be denoted e_i
- The final SinkRank+ score is calculated by taking the expectation of overall steps, where

$$SinkRank+ = \sum_i l_i e_i.$$

- Clearly the lower the *SinkRank+* score the more sysmetic risk.

SinkRank+: analysis

- Since we were only able to gather GDP data but not complete interest rate spread data, this remains a proof of concept
- However, by using arbitrary probabilities of insolvency, we can explore the type of predictions made by this model

SinkRank+: an hypothetical scenario

Scenario: The risk of default of one country is 0.5, 0.01 for others

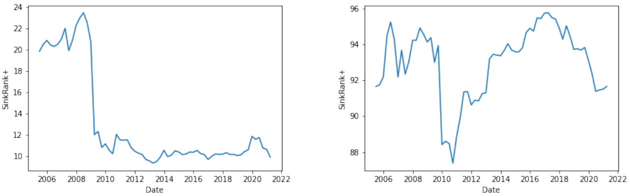


Figure: Left: USA high default risk, Right: Greece high default risk

- Observation 1: stability fell at around 2008 and 2010 respectively, corresponding to GFC and Sovereign debt crisis
- Observation 2: USA (because of its larger GDP) has a larger impact overall, bringing the score to the range from 10 to 24 as opposed to 88 to 96

SinkRank+: summary

- As seen in the previous slide, scores produced by the model are reasonable
- Overall, the model is an appropriate measure of global risk because it takes into account of
    - Interdependence (A_i's): the more interconnected a country is the larger the global impact of its default
    - Wealth (y_i's): the wealthier a country is the larger the global impact of its default
    - Country-specific risk (p_i's): the riskier investments are in a country the less stable the global system is
- The code we used to create this metric can be found in Appendix 1.

Conclusion

- We explored centrality and measures of community in the global banking exposure network
- Changes over time in these measures were found, particularly around the 2008 financial crisis
- Further, we introduced a novel metric SinkRank+ to capture global systemic risk level
- This metric considers the interdependence, wealth, and country-specific risk of each country, but remains as a proof of concept due to lack of interest rate data

Appendix 1: SinkRank+

- Due to lack of time, we will attach screenshots of codes used to construct SinkRank+ only

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx

edges = pd.read_csv('bis_arcs-1.csv')
vertices = pd.read_csv('bis_vertices-1.csv')
gdp = pd.read_csv('gdp.csv')

edges = edges[edges['to_id'] != '5A']
```

Figure: Step 1: importing libraries and datasets

Appendix 1: SinkRank+

```
def get_long_run_state(pi, k, P):
    return pi @ np.linalg.matrix_power(P, k)

def extract_Q(P):
    indices_without_1_in_diagonal = np.where(P.diagonal() != 1)[0]
    Q = P[indices_without_1_in_diagonal.reshape(-1,1), indices_without_1_in_diagonal]
    return Q

def compute_N(Q):
    number_of_rows, _ = Q.shape
    N = np.linalg.inv(np.eye(number_of_rows) - Q)
    return N

def compute_t(P):
    Q = extract_Q(P)
    N = compute_N(Q)
    number_of_rows, _ = Q.shape
    return N @ np.ones(number_of_rows)
```

Figure: Step 2: Function to calculate absorption time (Adapted from online)

Appendix 1: SinkRank+

```python
country_list = list(edges['from_id'].unique())

def transform_to_transitional_matrix(df, p_default):
    df = df.drop('net_id',axis=1).pivot(index ='from_id', columns ='to_id')
    df.columns = df.columns.droplevel()

    # The adjacency matrix
    adjacency = pd.DataFrame(columns=country_list, index=country_list)
    for col in adjacency:
        if col in df.columns:
            adjacency.loc[:,col] = df[col]
    adjacency = adjacency.fillna(0)

    adjacency = adjacency.to_numpy()

    # Normalising the adjacency matrix for Markov processes, considering risk of default
    transition = (adjacency/adjacency.sum(axis=1, keepdims=True))*(1-p_default)
    transition = np.column_stack([transition, p_default])

    # The [0,0,...,1] row vector which ensures the absorption of the default state
    absorb = [0]*(transition.shape[0]+1)
    absorb[-1] = 1

    # Final transitional matrix with possibility of default
    transition = np.vstack([transition, absorb])

    return transition
```

Figure: Step 3: Function to transform parameters into a transitional matrix representing the model

```python
np.dot(compute_t(transform_to_transitional_matrix(quarter_data, p2[quarter].to_numpy())),gdp_norm[quarter])
```

Figure: Step 4: Calculate the score given data from a quarter