Community Detection in Networks

Hesam Ipakchi

Imperial College London

hesam.ipakchi10@imperial.ac.uk

24th June, 2014

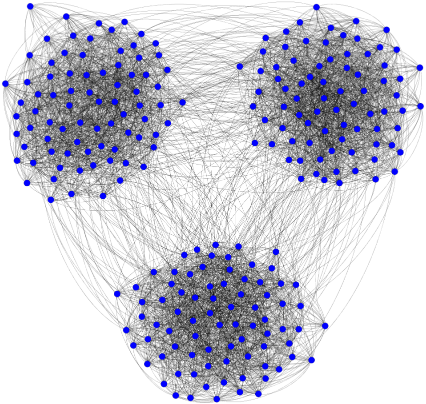
Outline

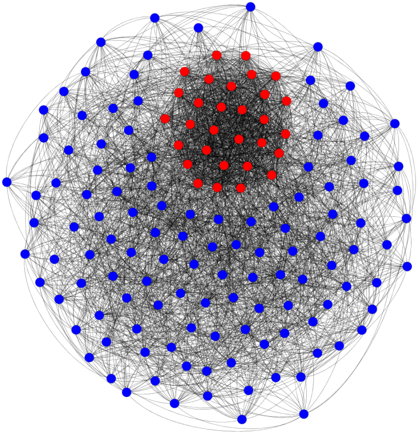
- Community structure in networks
  - Intuitive Definition
  - Example illustrations
- Community detection algorithms
  - Algorithms studied
  - Synthetic data testing set-up
  - Comparison using results from testing
- Application on financial networks
  - Motivation
  - Application on a static network
  - Extension to dynamic (time-evolving) networks

Community structure in networks

- Networks are represented by graphs, consisting of nodes and edges
- **Communities** are groups of nodes with denser connections within groups and sparser connections between groups
- **Community detection** algorithms involve partitioning the network into communities
- Many real-world applications including social networks, biological networks, financial networks...

Example illustrations





Community detection algorithms

Study a range of algorithms based on different techniques:

- Spectral clustering
- Modularity Optimisation
  - Greedy algorithms
  - Spectral algorithms
  - Simulated annealing
- Belief propagation
- Non-linear power iteration

NOT considering massive data sets or overlapping communities!

Statistical block models

Compare algorithms in a synthetic data testing framework

- Use statistical block models to generate random graphs which exhibit community structure
- Tweaking model parameters captures varying network properties (e.g. size, sparsity, number of communities, edge-occurrence probabilities)
- Provides theoretical setting to test and compare algorithms
- Two popular models: planted partition model, 'hidden clique model'

Synthetic data testing set-up

For each community detection algorithm:

- Decide upon an appropriate generative model for this specific algorithm
- Construct synthetic data set by generating various networks with different underlying parameters of the model
- Apply the algorithm to each network in the data set and measure accuracy

Comparison of algorithms

Algorithm	**Advantages** ✓	**Disadvantages** ✗
Spectral Clustering	Simple	Accuracy $\downarrow$ as sparsity $\uparrow$ Quite slow Need no. of communities as an input
Greedy method	Simple Fast Works on larger graphs	Accuracy $\downarrow$ as sparsity $\uparrow$
Belief propagation	Very good accuracy Very fast (for sparse) Works on larger graphs	

Problem: generative models are not well representative of real-world networks!

Financial Networks: Motivation

Consider portfolio selection problem for investor: how to select assets to form the 'best' portfolio that aligns with risk and return preferences?

- Famous technique: **mean-variance portfolio theory**
- **Idea:** construct portfolio which generates the mean return desired but with lowest variance of all possible selections.
- Ideal case: make portfolio with lowest possible inter-asset correlations
- Use sample estimates of mean, variance and cross-correlation of asset returns from historical data
- Also beneficial for dynamic rebalancing of portfolio for risk management

Financial Networks: Construction

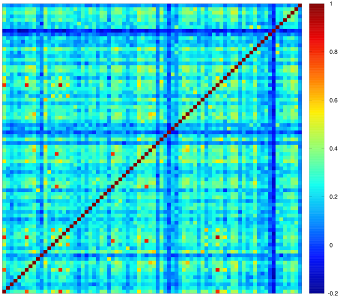
One possible approach: construct a undirected, fully connected and weighted graph!

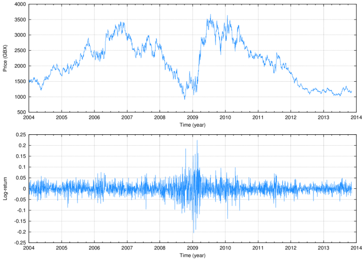
- Each node represents an asset, with the (standardised) time series of returns for the asset associated with the node
- Weight of an edge connecting two nodes is the cross-correlation between the two time series' associated with those nodes (based on time averages)
- Weighted adjacency matrix of the graph is the empirical correlation matrix of the asset returns!

Idea: communities in financial networks represent groups of assets whose aggregate average correlation is higher within groups and lower between groups
$\therefore$ provide investors with (small number of) 'baskets' of assets

Our data set

- Collected daily price data for 80 stocks traded on FTSE 100 exchange between 01/01/2004 and 11/11/2013





Community detection approach

Underlying technique: **modularity optimisation**

- First consider two 'naive' modularity maximisation methods: greedy algorithm and spectral relaxation
- Then consider two 'modified' modularity methods, tailored for this specific application: spectral clustering and Louvain method
- **Result:** able to detect finer-tuned and more communities using tailored approach $\implies$ higher quality partitions and notable improvement for adapted modularity techniques

Problem: only considered one static network so far! Require **dynamic** community detection to capture time-evolving correlation structure

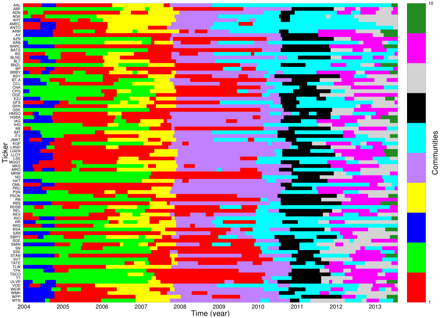
Extension to dynamic networks

Need to alter our approach:

1. Construct time-evolving networks from data set
   - Use **time-windowing** procedure to generate set of 'network slices' and create one correlation matrix per time window
   - For our data set: window size of 100, overlap length of 10 $\implies$ 240 correlation matrices
2. Apply dynamic community detection algorithm
   - Consider the **generalised Louvain method** previously applied to empirical neuroscience data in the literature
   - Similar procedure to Louvain method but designed to optimise a generalised notion of modularity for dynamic networks
   - Use modified modularity matrices as input for each network slice to tailor the method for our application

Evaluation

- Able to understand **temporal evolution of communities** with smooth transitions in community memberships obtained
- Similar procedure is applicable for **different asset classes** if data sets are collected



But... the generalised Louvain method is slow and performance depends on appropriate parameter choice

Future work

Some directions for research in this area...

- Lack of a consensus on precise definition of a ‘community’ $\implies$ **no single benchmark** to compare algorithms exists
- Little focus on **overlapping communities** which could lead to better models for real-world networks
- Increase in availability of time-stamped network data sets enables the study and application of **dynamic community detection** algorithms
- Significant improvements in **computational complexity** enables partitioning of networks with up to millions of nodes, but algorithms are approximate methods and not very reliable

Summary

- Introduced concept of community structure in networks and motivated use for community detection algorithms
- Mentioned several community detection algorithms and described the synthetic-data testing framework used to compare them
- Considered a real-world application of financial networks
- Identified time-evolving communities consisting of FTSE 100 stocks over the last decade
- Several interesting and important research directions exist

Thanks for listening!

Any Questions?