Journal of Financial Stability 70 (2024) 101195

Contents lists available at ScienceDirect

Journal of Financial Stability

journal homepage: www.elsevier.com/locate/jfstabil

From liquidity risk to systemic risk: A use of knowledge graph[☆]

Ren-Raw Chen [a,*], Xiaohu Zhang [b]

[a] Gabelli School of Business, Fordham University, 45 Columbus Avenue, New York, NY 10019, USA
[b] Scotiabank 44 King St W, Toronto, ON M5H 1H1

ARTICLE INFO

JEL Classification:
Code: G01
G12
G21

KeyWords:
Knowledge graph
Liquidity index
Systemic risk
Global crisis
Machine learning

ABSTRACT

In this paper, we use knowledge graph (KG) to study systemic risk in the banking industry. KG provides a graphic representation of the connections of entities of interest (known as vertices or nodes) with the strengths of connections being reflected by the lines connecting them (known as edges) or distances between them. As a result, KG is a natural tool for visualizing the relationships among financial institutions. Furthermore, various data and graph choices can present how differently entities of interest can be connected. In this paper, we draw KGs on two datasets: liquidity index and volatility and three different embedding methods: locally linear embedding, spectral embedding and principal component analysis. Our empirical results show, not surprisingly, that volatility and liquidity index are not similar in explaining how banks are connected. Embedding methods also matter.

1. Introduction

In this paper, we use knowledge graph (KG) to study systemic risk in the banking industry. KG provides a graphic representation of the connections of entities of interest (known as vertices or nodes) with the strengths of connections being reflected by the lines connecting them (known as edges) or distances between them. As a result, KG is a natural tool for visualizing the relationships among financial institutions. Furthermore, various data and graph choices can present how differently entities of interest can be connected.

The origin of KG is the graph theory by Leonhard Euler in 1735 when he solved Konigsberg's (now Kaliningrad) bridge problem.[1] Even since, the development that follows Euler has been exploding. Largely there are three areas of development: probability graphic models (PGM), KG, and KG database.

In the first case, PGMs are similar to other network models (e.g. Bayesian networks). The general idea is the expression a multi-dimensional probability distribution of entities of interest (which describes how entities are connected) in a series of conditional probability distributions. The focus of this line of research is generally to estimate such a network and establish statistical inferences. Hence, there is no need to present graphs. Also worth noting is that predominantly in this line of research normality (i.e. Gaussian distribution) is assumed. This differs from the second approach (see next paragraph) where non-parametric algorithms (i.e. machine learning) are adopted.

In the second case (knowledge graphs), graphs are essential in presenting the main result. A model (i.e. knowledge) is applied to create connections among entities of interest (e.g. stocks or banks). There is a huge selection of graphs, depending upon the purpose of the study. In general, various machine learning models are used (e.g. embedding and clustering methods) to create "knowledge". Textual analyses (e.g. NLP models such as LDA[2]) are also often used to create knowledge. In the next section, a Harry Potter characters graph is given as an example in which all figures in the first three novels of the Harry Potter series are plotted in a graph. The purpose of the graph is to see how different characters are connected and the relative importance of their roles in the story. It can be seen then that multiple models are used in conjunction to create the graph. The models are not parametric (unlike the first approach above) which provide the flexibility of combining models.

Lastly is to use KG to build databases. Such databases are essential for search engines. In a traditional database where tables/columns are used (a.k.a. relational database), "keys" are how various variables (i.e.

☆ We thank the editor Iftekhar Hasan for his encouragement and the anonymous referee for very insightful suggestions that help position the paper in the literature greatly.

* Corresponding author.
E-mail addresses: rchen@fordham.edu (R.-R. Chen), xzhang25@fordham.edu (X. Zhang).

[1] See Alexanderson (2006) for a nice review.
[2] NLP stands for natural language processing and LDA stands for latent Dirichlet allocation.

https://doi.org/10.1016/j.jfs.2023.101195
Received 28 January 2023; Received in revised form 1 November 2023; Accepted 7 November 2023
Available online 14 November 2023

columns in tables) are connected. However, in a KG database, "adjacent matrices" (i.e. connections) are used in connecting variables. This revolutionizes how search engines such as Google can provide speedy and relevant search results.

While PGMs (including network models) have been used in finance for quite some time,[3,4] KG is relative new. This is because machine learning tools are only introduced to finance recently. As mentioned earlier, the main difference between PGMs and KG is that the former assumes normality (parametric) and the latter use various machine learning methods to create knowledge (non-parametric).

While PGMs are more suitable for portfolio analysis (see footnote 4), KG is more suitable for studying systemic risk, especially for the financial industry. This is because financial institutions are connected in a rather convoluted manner. They hold each other's assets and share common vulnerability against macro economic conditions. Furthermore, many large financial institutions are global, which make they expose to various country risks. These convoluted connections in a large and wide variety of risk factors (many of which are not even numerals (e.g. political risks)) make the study of their connections impossible to be parametrical.

Financial institutions, given their specific inter-connectivity, present a substantial systemic risk that is vulnerable to crises as in 2008. KG which provides a network map is a natural tool to represent relationships among banks and as a result ideal to studying the systemic risk. Following the literature (to be reviewed in the next section) that uses KG to study systemic risk of the financial sector, in this paper, we draw a set of KGs of all financial firms in our sample using data between 1996 ~ 2013 (216 months). While using KG to measure the systemic risk is a consensus, what variables (features) to use remains a challenge. In KG, the measurement for the relationships is "edges" of which a higher value represents a closer relationship, and vice versa.

As mentioned earlier, there are a wide variety of choices for the variables to connect financial firms, such as firm fundamentals (e.g. profitability, credit risk, etc.), technicals (e.g. seasonality, momentum, etc.), management (e.g. governance, size, etc.), among numerous others; and descriptive variables such as news, lawsuits, analyst reports, and others. Ideally, all of these variables should be included in building a network model using KG. Unfortunately, in the literature, there has been no such work. Furthermore, even in the very limited literature, stock prices/returns are often chosen as the only variable in building a KG. It is understandable to use returns in that correlations are measured by returns. Yet return is an aggregated measure and too broad to be informative in many situations such as systemic risk. There are a few studies that do not use returns including a dissimilarity index (Boss et. al. (2000), volatility (Ahelegbey, 2016), stress indices (Nicola et al., 2020), non-performing loans (Dolfin et al., 2019), and interest rates (Caccioli et al., 2018).

We propose to use an alternative variable – the liquidity discount measure by Chen (2012) which focuses on the co-movement of the liquidity of financial firms during the 2008 crisis (see Chen et al., 2013 and Chen et al., 2016). As a robustness test, we also use volatility, as suggested by Ahelegbey (2016). In addition, we draw KGs with three different embedding methods: locally linear embedding, spectral embedding and principal component analysis. Our empirical results show, not surprisingly, that volatility and liquidity index are not similar in explaining how banks are connected. Embedding methods also matter.

2. Motivation and literature

As mentioned in the Introduction, the use of PGMs has had a long history, and is reviewed by Caccioli et al. (2018).[5] Giudici and Parisi (2016) summarize various parametric models.[6] Yet KG is relatively new and the literature is rather limited. The focus of this paper is to build a KG using machine learning methods.[7]

To our knowledge, Boss et al. (2004) is earliest KG used on banks. They, using only September 2002 data, present a KG of Austrian banks using clustering method (for the connections) and Zhou's dissimilarity index for edges.[8] Birch et al. (2015) also and planar maximally filtered graphs to analyze DAX 30 stocks for the time period 2001–2012. In addition, they use minimum spanning tree and asset graph to compare results. Soramäki et al. (2016) also use planar maximally filtered graph and minimum spanning tree to analyze major U.S. market indices.[9]

Tumminello et al. (2010) use hierarchical clustering and Kullback-Leibler distance to analyze 10 U.S. stocks from January 2001 to December 2003.[10] They also discuss the differences between their methods and the methods used earlier (i.e. minimum spanning tree and planar maximally filtered graph).

Nicola et al. (2020) who use mutual information and cross entropy to study the stock prices of top 74 U.S. large listed banks in a period January 2003 to May 2017. They calculate a different graphical model for each market day based on the data of the 90 previous days and leverage the graphical models information comparing the measures extracted from its structure with well known financial stress indexes and performed a causality analysis. Their finding confirms the common wisdom that during the crisis period (2007–2010) banks are more interconnected.

Zhan et al. (2021) utilizes a wide variety of machine learning (e.g. spectral embedding, graphic LASSO, and clustering) methods to study stock portfolios. They study tech, financial, and energy stocks in the S&P 500 index from January 1, 2012 till January 1, 2020.

Our paper is similar to the literature reviewed in this section. Our contributions are the following. First, we focus on how financial firms are connected, especially during the crisis period (i.e. 2008 Lehman crisis). We differ from Nicola et al. (2020) in that we use different graphic methods. Secondly, we use a liquidity index by Chen (2012). As documented by Chen et al. (2013), 2016), Chen's liquidity index has a strong explanatory power of the crisis. Thirdly, we also compare the KG from the liquidity index with the volatility (Ahelegbey (2016)).

Finally, omitted here due to limitation of space and yet highly important is the broad literature of how machine learning in general has impacted the financial research in the past 20 years. These studies may or may not directly adopt graphs or networks and yet do relate in various degrees to our paper here. Given the extremely large volume of studies,

[3] See Caccioli et al. (2018) for an extensive review of how graphic models are used in systemic risk.

[4] A Gaussian PGM is a natural extension to the classical Markowitz-Sharpe portfolio theory in that they share the basic mean-variance assumption and PGM can "better identify" (or "clean up") the relationships between stocks. This is because the inverse of the variance-covariance matrix, known as the precision matrix, can detect conditional independence between any two stocks. A Gaussian PGM (with regularization such as LASSO) can identify conditional independence (a.k.a. partial correlation) and hence can more clearly and cleanly describe the relationships (in some cases, dependencies) among stocks. This helps isolate out those stocks that are actually not connected even though they have non-zero correlations. As a result, a better portfolio can be constructed.

[5] More recent such work can be found in Ahelegbey (2015, 2016), Cerchiello and Giudici (2016), Engel (2019), Denev et al. (2020), Zhou (2020), Smitshoek (2021),

[6] They also provide empirical evidence on partial correlation in the two crisis periods of the EU nations, using various government and corporate interest rates between 2003 and 2015.

[7] We also exclude the theoretical network models such as Eboli (2007), Tabak et al. (2011), Benazzoli and Di Persio (2016), Detering et al. (2019), Dolfin et al. (2019), and Yu and Zhao (2020), among others.

[8] Zhou, H (2003), "Distance, Dissimilarity Index, and Network, Community Structure," e-print: arXiv:physics/0302032.

[9] Real Estate, High Yield, S&P, Gold, VIX, and Euro-USD, Fixed Income.

[10] AIG, IBM, BAC, AXP, MER, TXN, SLB, MOT, RD, and OXY.

we can only cite an extremely few as examples. Broadly speaking, machine learning has been used in the following four areas:

- Portfolio Optimization: Machine learning methods, such as clustering algorithms and neural networks, have been used to optimize portfolios by identifying asset relationships and predicting returns. For instance, Ravi and Ravi (2015) surveyed machine learning techniques to predict stock prices and optimize portfolios based on the predicted prices.
- Algorithmic Trading: Reinforcement learning algorithms and deep learning models have been applied to develop algorithmic trading strategies. For example, see Kissell (2020) of how machine learning is used in algorithmic trading.
- Credit Risk Assessment: Machine learning techniques, such as decision trees, support vector machines, and neural networks, have been employed to assess credit risk by analyzing credit data and predicting default probabilities. A notable example is the work by Yeh and Lien (2009), which compared the performance of various machine learning methods for credit scoring.
- Fraud Detection: Machine learning algorithms, including unsupervised and supervised techniques, have been used to detect financial fraud by analyzing transaction data and identifying anomalous patterns. For instance, Phua et al. (2010) surveyed the application of machine learning methods for fraud detection in credit card transactions, insurance claims, and securities trading.

3. Knowledge graph

A knowledge graph (KG) is a graph where it displays (usually in a two dimensional diagram) how each node (vertex) is connected with other nodes. The line connecting two nodes is known as an edge whose value represents how close the two nodes are related (higher value represents closer relationship). The main advantage of KG is its visualization. It allows the users to see visually how each node is connected to other nodes. Fig. 1 is an example taken from a blog of neo4j which is a popular KG database.[11] In Fig. 1, all the characters in J.K. Rowling's first four Harry Potter books are displayed in a graph. Each character is a node (vertex) and each line is an edge. The size of the bubble of each node represents how important a character is and the distance between any two nodes measures how close the relationship is. It is obvious that Harry Potter is the most important character in the 4-book series and therefore is placed in the center of the graph (and hence it is to be noted that the coordinates (i.e. x-y axis) in the graph carries no meaning in Fig. 1, which is not necessarily so in other KGs).

By its name, a KG must contain "knowledge". Knowledge is just a model that creates the connections. In Fig. 1, the knowledge used is all the texts in the first four Harry Potter books. By going through the entire four books, the connection of any two characters (e.g. Hermione Granger and Ron Weasley) is determined by how many times they are mentioned together. This requires use of models in NLP (natural language processing).,[12,13]

There are various ways to generate a KG. If the location (i.e. coordinates) on the graph matters (i.e. the axes have meaning – this is usually the case where dimension reduction techniques are applied and hence the axes represent the most important features (or linear combinations of all features, e.g. PCA)),[14] then the distance between any two vertices represents the "closeness" of the two vertices. If the location has no meaning (i.e. vertices are randomly placed by the graphic software for the sake of a nice visual), then there are two ways to present closeness. First, edges are drawn with different degrees of thickness – a thicker edge representing a closer relationship. In such a case, various machine learning methods (such as LASSO, clustering, K-means, among numerous others) can be used to gauge the "closeness" of the vertices. Secondly, distance can be used in measuring closeness as in Fig. 1. In such graphs, vertices are arranged optimally[15] for visualization (i.e. closer vertices are more connected than farther vertices and yet where they are located in the x-y plane is not important).

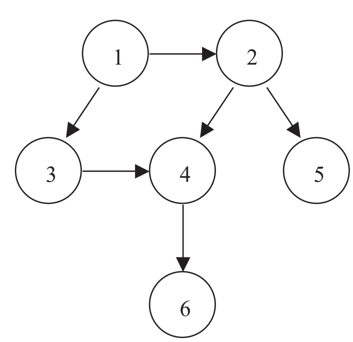
As mentioned earlier, one can use various machine techniques to describe vertices and edges. These machine learning techniques may not be related, which provides one a large amount of flexibility in build a KG. This is drastically different from PGMs which must obey a parametric structure.

In sum, there is no standard way to present a KG. Authors can choose any visualization as numerous packages are available for selection. Apparently this raises issues such as robustness, stability (stationarity), and other statistical concerns. We shall discuss this matter in Conclusion. In the remainder of this section, we briefly describe directed and undirected graphs which are mostly concerned by PGMs (a standard KG is usually an undirected graph) and a Gaussian PGM which is the most popular PGM. These are all parametric (or semi-parametric) models and not directly related to this paper.

3.1. Directed and undirected graphs

As the names suggest, a directed or undirected graph is to present a relationship in a symmetrical (latter) or asymmetrical (former) way. Typical examples in finance are joint defaults (former) and return correlations (latter). In the studies of joint defaults, conditional probabilities are used to describe dependences between two companies. For example, company A defaults may lead to the default of company B but not vice versa. In such situations a directed graph can be more suitable. On the other hand, return correlations are symmetrical and hence an undirected graph is more suitable.

Directed graphs can be modeled via a series of conditional probabilities. The following graph depicts the basic idea of a directed graph:



where arrows demonstrate dependencies. For example, node #1 depends upon nodes #3 and #0 but it is depended upon by nodes #2 and #4.

[11] By Tomaz Bratanic in https://neo4j.com/developer-blog/turn-a-harry-potter-book-into-a-knowledge-graph/

[12] For example, the author uses co-reference resolution which replaces the pronouns with the referenced entities.

[13] Here, knowledge graph embedding is used. It is a technique used in natural language processing and machine learning to represent entities and relations in a knowledge graph as low-dimensional vectors in a continuous vector space. Details of such embedding is provided in the Appendix.

[14] We use "feature" and "explanatory variable" interchangeably. PCA is shorthand for principal component analysis.

[15] That is, the chosen graphic software will place the vertices optimally for a nice visual. Fig. 1 is a "spring graph" in which least connected vertices will be put in the far sides of the graph and the Euclidian distance is not a good measure for closeness. In another example, " graph" will place all vertices on the circumference of a circle (given that location has no meaning) and the closeness is entirely represented by the thicknesses of the edges.

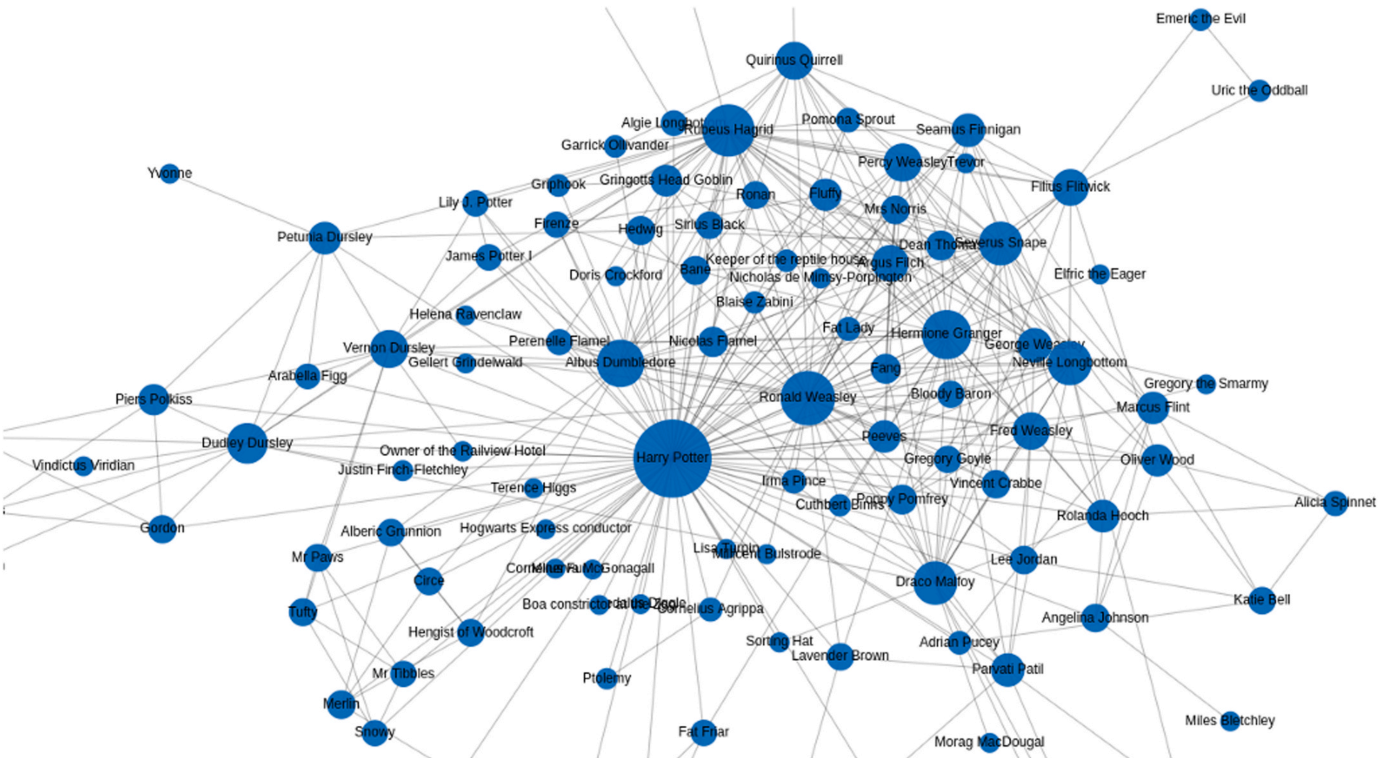


Fig. 1. Harry Potter Knowledge Graph.
Source: https://neo4j.com/developer-blog/turn-a-harry-potter-book-into-a-knowledge-graph/.

The joint probability of all six nodes can be shown, as a demonstration, as:

$$p(x_1, \ldots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)p(x_5|x_2)p(x_6|x_4) \tag{1}$$

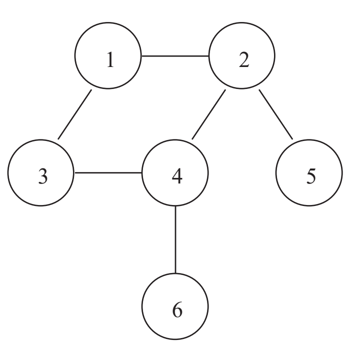
To reflect the dependencies,[16] in this example, vertex 2 and vertex 3 are independent conditional on vertex 1, usually labeled as $x_2 \perp x_3 | x_1$.

In general, a set of vertices have the joint probability as:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_1 | x_{\mathrm{Pa}(i)}) \tag{2}$$

where $\mathrm{Pa}(i)$ represents the parents of i as shown in graph (and i represents a vertex). For example, $\mathrm{Pa}(4) = \{2, 3\}$ and $\mathrm{Pa}(2) = \{1\}$ and $\mathrm{Pa}(1) = \phi$. The random variable $x_A = (x_1 : i \in A)$. Hence if $A = \{2,3\}$, then $x_A = (x_2, x_3)$. Let $A = \mathrm{Pa}(4) = \{2, 3\}$ and then $x_{Pa(4)} = (x_2, x_3)$.

Undirected graphs, a.k.a. random fields, are depicted as an example as follows:



.

Clearly, in an undirected graph, relationships/dependencies are symmetrical. As mentioned earlier, the edges can be presented with different degrees of thickness (for a better visual). The edges can be estimated parametrically (e.g. using a Gaussian graphic model) or via machine learning methods (e.g. graphic LASSO).,[17,18]

3.2. Embedding

In machine learning studies, it is customary to use a large amount of data with a large number of features. Given the complex nature of (non-parametric) these data and features, in many cases, features are not linearly related and need to be transformed in order to obtain accurate results. This is known as graph embedding. As a result, graph embedding is generally understood as a dimension reduction tool to map a complex graph into a usually 3- or 2-dimensional drawing for easy visualization. It is well known that any finite graph can be embedded in 3-dimensional Euclidean space and a planar graph is one that can be embedded in 2-dimensional Euclidean space.[19]

Formally, an embedding of a graph G on a surface Σ is a representation of G on Σ in which points of $\Sigma$are associated vertices and arcs are associated with edges in such a way that

[16] This example is taken from https://www.youtube.com/watch?v=A7Ypw5d9580: (ML 13.2) Directed graphical models - introductory examples (part 2)

[17] Please see the appendix for a short discussion of two types of the undirected graph.

[18] Often edges are modeled as partial correlation (inverse of the covariance matrix, or precision matrix) or any definition of "distance".

[19] This is taken from Wikipedia. see Cohen, Robert F.; Eades, Peter; Lin, Tao; Ruskey, Frank (1995), "Three-dimensional graph drawing", in Tamassia, Roberto; Tollis, Ioannis G. (eds.), Graph Drawing: DIMACS International Workshop, GD '94 Princeton, New Jersey, USA, October 10–12, 1994, Proceedings, Lecture Notes in Computer Science, vol. 894, Springer, pp. 1–11, doi:10.1007/3–540–58950–3_351, ISBN 978–3–540–58950–1.

- the endpoints of the arc associated with an edge *e* are the points associated with the end vertices of *e*,
- no arcs include points associated with other vertices, and
- two arcs never intersect at a point which is interior to either of the arcs.

Knowledge graph embedding is a type of representation learning between entities and relations in a knowledge base. The entities and relations are mapped into a low- dimensional space representing the semantic information between entities and relationships. We classify knowledge embedding into two broad areas. The first is unfolding. The most famous case is the Swiss roll example where a roll is unfolded into a plane.[20] This includes isomap, locally linear embedding, spectral embedding, Hessian eigenmapping, local tangent space alignment, multi-dimensional scaling (MDS), t-distributed stochastic neighbor embedding (t-SNE), among others.

The second is to investigate the relation of any two nodes using textual data, known as translation distance models. Both entities and relations can be represented as vectors in the same space. This includes DistMult, TransE, TransH, TransR, ComplEx, ConvE, KG2E, among others. Note that these methods use textual data. These knowledge graph embedding methods aim to capture the semantic and structural information of entities and relations in knowledge graphs. These embeddings can then be used as features for various downstream tasks, such as knowledge graph completion, entity recommendation, and question answering.

In the Appendix, we provide further details of these various embedding methods.

Finally we should note that graphs are often used, like other machine learning tools, to perform classification, clustering, regression, anomaly detection, feature learning, among others. All of these tasks have their counterparts in network analysis. Researchers in network science have traditionally relied on user-defined heuristics to extract features from complex networks (e.g., degree statistics or kernel functions). However, recent years have seen a surge in approaches that automatically learn to encode network structure into low-dimensional embeddings, using techniques based on deep learning and nonlinear dimensionality reduction. These network representation learning (NRL) approaches remove the need for painstaking feature engineering and have led to state-of-the-art results in network-based tasks, such as node classification, node clustering, and link prediction.

4. Liquidity discount model

In this paper, we use the liquidity discount model developed by Chen (2012). In an empirical study, Chen et al. (2016) summarized how various banking sectors react to the crisis.

In Chen (2012), the liquidity discount is modeled as a put option as Fig. 2 demonstrates. The illiquid price (horizontal axis) is lower than the liquid price (vertical axis) in a put option style.[21] When the liquidity squeeze is severe, the curve is higher and when the liquidity squeeze is milder, the curve is lower.

To explicitly carry out the liquid and illiquid evaluations, he adopts a functional form of the asset under evaluation with respect to a fundamental (or underlying) economic variable. The way to do it, according to Chen (2012), is to create two different expectations, one of which is under illiquid trading and the other under liquid trading. Let the liquid price be $\widehat{X}(t)$ and the illiquid price be $X(t)$. The terminal payoff is the same for both prices which is a function of a chosen economic variable: $X(T) = \widehat{X}(T) = f(V(T))$. The liquid price is computed as:

$$\widehat{X}(t) = e^{-r(T-t)}\widehat{\mathbf{E}}_t\left[X(T)\right] \tag{3}$$

under the risk-neutral expectation $\widehat{\mathbf{E}}_t[\cdot]$ (conditional at time *t*) where r is the risk-free rate. The illiquid price is computed as:

$$X(t) = e^{-\xi(T-t)}\mathbb{E}_t\left[X(T)\right] \tag{4}$$

under the physical expectation $\mathbb{E}_t[\cdot]$ (conditional at time *t*) where ξ is the risk-adjusted rate.

We note that if both expectations are evaluated under continuous trading, then they must be equivalent as in continuous trading, no-arbitrage condition holds. Equation (3) is known as "certainty equivalent" method and equation (4) is known as the "risk-adjusted" method.[22] Unfortunately, under illiquid trading, equation (4) cannot evaluated in continuous time and hence differences occur.

Chen (2012) then uses the CAPM as follows.[23].

$$\begin{aligned} e^{\xi(T-t)} &= \mathbb{E}_t\left[\frac{X(T)}{X(t)}\right] \\ &= e^{r(T-t)} + \beta\left[\mathbb{E}_t\left[\frac{V(T)}{V(t)}\right] - e^{r(T-t)}\right] \end{aligned} \tag{5}$$

where r is the (continuously compounded) risk-free return and β (beta) is the systematic risk:

$$\begin{aligned} \beta &= \frac{\operatorname{cov}\left[\frac{X(T)}{X(t)}, \frac{V(T)}{V(t)}\right]}{\operatorname{var}\left[\frac{V(T)}{V(t)}\right]} \\ &= \frac{V(t)}{X(t)}\frac{\operatorname{cov}[X(T), V(T)]}{\operatorname{var}[V(T)]} \end{aligned} \tag{6}$$

Substituting (5) back into (4), we obtain:

$$X(t) = e^{-r(T-t)}\mathbb{E}_t\left[X(T)\right] - \beta^{\$}\left\{e^{-r(T-t)}\mathbb{E}_t\left[V(T)\right] - V(t)\right\} \tag{7}$$

where

$$\beta^{\$} = \frac{\operatorname{cov}[X(T), V(T)]}{\operatorname{var}[V(T)]} \tag{8}$$

Chen (2012) shows that if $X(T)$ is linear in $V(T)$, then under illiquid trading, there is no liquidity discount: $X(t) = \widehat{X}(t)$. Chen and Li (201x) assume two explicit non-linear functional forms of $X(T)$ in terms of $V(T)$:

$$\begin{aligned} X(T) &= \max\left\{K - V(T), 0\right\} \\ X(T) &= aV(T)^2 + bV(T) + c \end{aligned} \tag{9}$$

In Chen et al. (2016), the former function is used to calculate liquidity discounts for financial firms. In Chen, Lin and Wei, a firm's equity is modeled as a call option on its firm (market) value which is under liquidity squeeze and the liquidity discount is evaluated via the above equations. Details of the implementation can be found in Chen, Lin and Wei. We are grateful for their liquidity discount data.

5. Empirical work

In this research, we present a series of KGs for all financial firms (551 firms, according to Chen et al., 2016) as well as the top 25 financial firms in the U.S. We generate KGs using both liquidity index and volatility. We analyze KGs for the crisis period as well as the non-crisis period as comparisons.

[20] See, for example, https://scikit-learn.org/stable/auto_examples/manifold/plot_swissroll.html.

[21] That is, the difference between the liquid and illiquid prices is similar to a put option. See the original discussion in Chen (2012).

[22] Discussions on risk-adjusted versus certainty-equivalent methods can be found in any standard corporate finance textbook under the "capital budgeting" topic.

[23] This is Merton (1973).

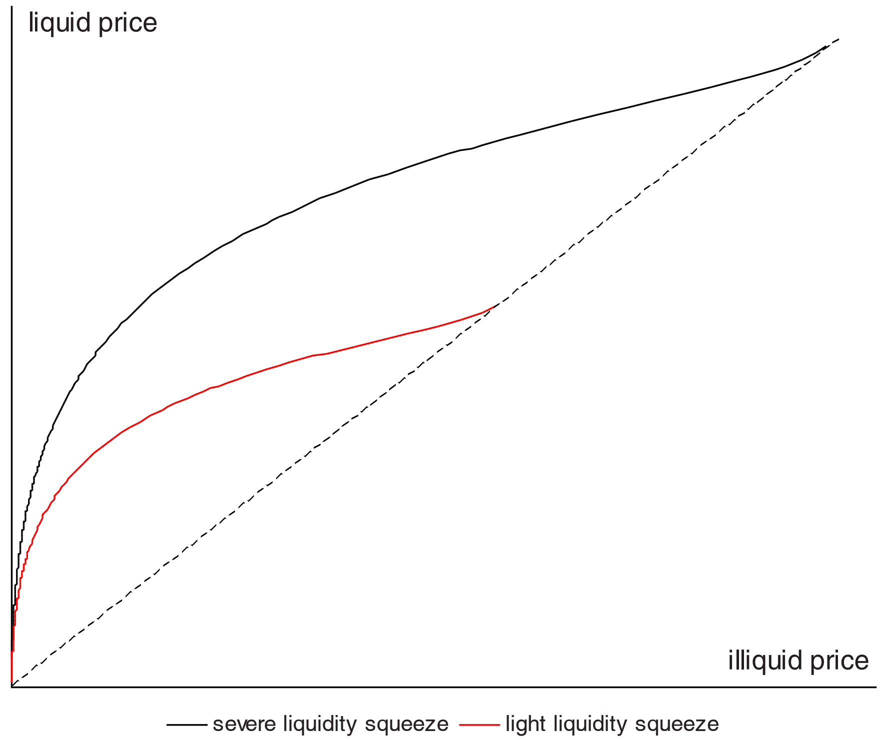


Fig. 2. Liquidity Discount Model of Chen (2012). This is Exhibit 8 in Chen (2012).

5.1. Data and Main Methodology

The main data used in this study is liquidity discount index compiled by Chen et al. (2016).[24] In their liquidity discount file, there are a total of 75,331 observations containing 551 financial firms over the period monthly 1996 ~ 2013 (216 months). Out of 75,331 observations, 250 observations have 0 debt and hence cannot compute liquidity discount and are removed from the data (resulting 75,081 observations). 4750.

For the purpose of our study (which is their interconnectedness during crisis), we only select there period of 2006–2012 (84 months) which is 2 years before and 3 years after the 2008 Lehman crisis. We study the whole sample as well as two sub-samples: crisis sub-sample (2007–2009) and non-crisis sub-sample (all except for 2007–2009). The summary statistics of the liquidity discount index are provided in Table 1.

In Panel (a) of Table 1, it is clear that liquidity risk reached its peak in 2009 (mean discount is at 74.6% and standard deviation is at 34.0%), followed by 2008 (mean discount 95.3% and standard deviation 16.3%) which is when crisis began. The monthly time series plots in Chen et al. (2016)[25] depict how liquidity discounts start to appear in late 2007 and last till end of 2009. This is why we define the crisis sub-period to be 2007–2009.

Volatility (Panel (b) of Table 1) tells a similar story. The peak of the volatility is in 2009 (and also the highest variation). The difference is in the relative magnitudes. This can be seen in the measure of coefficient of variation (or CoV which is standard deviation divided by the mean). The CoV for liquidity discounts is more than 3-fold (0.2668 for the crisis period versus 0.0888 for the non-crisis period) and yet it is only 30% for the volatility (0.6906 versus 0.5307 for the crisis and non-crisis respectively). In essence, both volatility and liquidity discounts tell the same story about the crisis. They both reflect the market turmoil during the crisis period. The difference is merely a non-linear transformation from one measure to another. Yet this is what a model is meant for. The basic phenomenon can always be observed from the raw data (in this case volatility) and yet a model can better gauge the severeness of the problem. In this case, the raw data (volatility) apparently does not reflect the severeness of a global crisis (from 30% to 300%).[26]

We finally compare the two indices in Panel C.[27] For the top 25 banks, the correlation between their volatility and liquidity discount is quite substantial. They are highly negatively correlated. The smallest (in absolute value) correlation is –0.4533 (Franklin Resources, Inc. (BEN)) and the highest is –0.9882 (Citi Bank (C)). The average is –0.8427 with a standard deviation of 0.1327. The median is –0.8652 suggesting that the distribution is quite symmetric.

To draw KGs, we follow the suggestions by Zhan et. al. (2021) that we first implement an embedding method to remove any non-linearity in the data. Then we use the precision matrix (combined with LASSO – known as graphic LASSO in Python) to detect any conditional independence. Finally, a clustering method is used to highlight the relationships among the firms (represented by various colors). We should note that the clustering method is not necessarily consistent with the precision matrix and yet it is a useful auxiliary method to validate the result from the precision matrix. This is common in generating KGs in that various machine learning methods are used in combinations.

In the following, we briefly describe our methodologies and then report our empirical findings.

5.2. Embedding

Embedding in graph theory means that a complex (non-linear) data structure in a high dimension might “embed” a simpler (linear) structure in a lower dimension. Hence, embedding refers to techniques that

[24] We thank Chen et al. (2016) for providing us their data.
[25] Exhibit 2 in Chen et al. (2016).
[26] We should note that comparing means or standard deviations (but not combined) separately is meaningless in that the two measures have different units and magnitudes.
[27] We thank the referee for suggesting this, as the high correlation between variables could suggest bias in either variable.

Table 1
Summary Statistics of Liquidity Discounts and Volatility.

a) Liquidity Discount	mean	median	std.dev	# of obs.	coef.var.
2006	0.9982	1.0000	0.0050	4313	0.0050
2007	0.9973	0.9935	0.0099	4162	0.0099
2008	0.9530	0.9972	0.1626	4041	0.1706
2009	0.7462	0.9830	0.3404	3882	0.4562
2010	0.9645	0.9845	0.1347	3786	0.1396
2011	0.9880	1.0000	0.0749	3739	0.0758
2012	0.9855	0.9955	0.0850	3694	0.0862
all	0.9483	0.9988	0.1770	27617	0.1866
2007–2009	0.9018	0.9970	0.2406	12085	0.2668
all but 07–09	0.9845	0.9995	0.0875	15532	0.0888
b) Volatility					
	mean	median	std.dev	# of obs.	coef.var.
2006	0.2303	0.2138	0.0727	4313	0.3155
2007	0.2581	0.3236	0.0917	4162	0.3553
2008	0.5468	0.3690	0.2851	4041	0.5215
2009	0.9470	0.7460	0.3898	3882	0.4116
2010	0.4667	0.4072	0.2511	3786	0.5381
2011	0.3519	0.2874	0.1474	3739	0.4189
2012	0.3627	0.3727	0.1427	3694	0.3934
all	0.4481	0.3104	0.3181	27617	0.7099
2007–2009	0.5759	0.4064	0.3977	12085	0.6906
all but 07–09	0.3487	0.2872	0.1850	15532	0.5307

c) Correlation between Liquidity Discounts and Volatility	
mean	-0.8427
median	-0.8652
std.dev.	0.1327
maximum	-0.9882
minimum	-0.4533

The minimum correlation is Franklin Resources, Inc. (BEN) and the maximum is Citi Bank (C)).

"unfold" the original data from a non-linear structure to an approximation of linear structure in a lower dimension. One advantage is that it is then able to present the structure in low (2 or 3) dimensions via a graph. This allows human eyes to easily recognize the relationships among the vertices.

We try various embedding methods. Various methods yield different results. As a consequence, we experiment with two popular methods – locally linear embedding and spectral embedding. We also compare these two embedding methods with a popular dimension reduction method – principle component analysis (PCA). Note that not only do different embedding methods apply different unfolding techniques, they also apply different graphic methods. Graphs generated by locally linear embedding and spectral embedding are (Laplacian) eigenmaps and isomaps respectively, in which the coordinates are generated in such a way that is best matched with the unfolding techniques. PCA can be regarded as a linear embedding method in that principle components (PC) are linear transformations of original coordiates. The first two or three PCs carry the most relevant information among vertices. Each method is briefly explained in the Appendix.

5.3. Precision matrix

Usually it is via the inverse of the covariance matrix (known as the precision matrix). However, edges can be estimated via other methods. Edges measure the closeness of two vertices. Normally an edge is regarded as "distance" between any two adjacent vertices. This "distance" can be viewed as Euclidian distance or any other measure (such as "precision" in portfolio analysis).

5.4. Clustering

A KG usually gets help from clustering analysis. Clustering puts vertices in groups (clusters). As a result, vertices are color-coded to represent their closeness. Note that a standalone clustering analysis frequently outputs a tree diagram. Yet, combined with KG, it puts colors on vertices where each color represents a cluster.

There could be contradictions between the result of clustering analysis and the precision matrix. Hence, clustering used KG is often serves as an auxiliary tool whose quantitative results are not used by the model. There are various methods in clustering and we use k-nearest neighbors (KNN).

6. Results

We have two sets of results. The first set of graphs is plots of the results generated by various embedding methods. We draw KGs for all firms in our sample. The second set of graphs is a spring graph (or force-directed graph). The reason to choose different methods for graphs when different data are involved is that we would like to reach maximum visual effect so how these financial firms are connected can be easily seen. In a large dataset, the number of edges is large and hence the best presentation of the connectedness is to use firms' geographic locations to present their closeness. When the sample size is small, then edges can be added. With edges measuring connectedness, locations do not matter and hence spring graph is a better choice. Details are given below.

Our first empirical result is to show how all financial firms in our dataset are connected. We compare three different embedding methods – locally linear embedding in Panel (a); spectral embedding in Panel (b); and principle component analysis (PCA) in Panel (c). As mentioned in the previous section, embedding is a method that "unfolds" the data (in other words, to transform non-linear data to linear data so linear models can be applied). Furthermore, the processed data can be plotted in a lower dimension (e.g. 2d or 3d) for a nice visual. In this step, edges are not computed. This is given in Fig. 3.

Given a large number of vertices and for the sake of a nice visual, edges are not computed for the graphs in Fig. 3 (while edges are computed for top 25 financial firms later). Also these graphs are 2d graphs (top 2 features). Finally, clustering is performed and 10 clusters are chosen. Vertices with the same color belong to the same cluster. The clustering analysis is an auxiliary analysis to cross-check the embedding result. We select only 10 clusters using KNN (we also select 30 clusters due to high similarity in 10 clusters and the result is available upon request).

At the first glance of the graphs, it is quite surprising that (1) the graphs are not like a typical spring graph (such as Fig. 1) where a spider-web is observed and (2) different embedding methods yield very different graphs. Panel (a) of Fig. 3 presents two approximately linear lines. Given that Euclidian distances here are meaningful, this graph indicates that the first two most important explanatory factors are orthogonal and all the firms can be categorized into two groups. Panel (b) of Fig. 3 presents a very different graph. Instead of two lines, it is now triangular. The clustering analysis (similar colors belong to one cluster) indicates that closely connected firms are also physically located closer together. The graph in Panel (c) of Fig. 3 presents a more scattered plot. This could be because the two axes under PCA are orthogonal. Different from the two previous graphs, the majority of firms under PCA belong to cluster #3 (green).

For a comparison, we also use volatility to generate KGs. The results are presented in Fig. 4.

The results are just as surprising as those of the liquidity index. Panel (a) of Fig. 4 presents the graph of a hyperbola-like shape. Firms in different parts of hyperbola do show different colors indicating that the clustering analysis is consistent with the embedding method. Panel (b) of Fig. 4 is a circle with different clusters mixed together, although within a cluster, firms do have similar locations. Panel (c) of Fig. 4 presents several rings. It is like an expanded Panel (a). Now we can see more clearly how similar (more connected) firms cluster together with the clustering analysis.

In summary, three conclusions are drawn. First, the graphs are different with different embedding methods This is true with both liquidity index data and volatility data. Secondly, graphs using liquidity index data are also quite different from those using the volatility data. Thirdly, the independent clustering analysis verifies the results of the graphs – similar firms do stay together, confirming that geographical locations of the firms represent the closeness of these financial firms.

The next set of results focus on top 25 firms by their market capitalization, since there is a substantial interest in only large firms during the 2008 Lehman crisis. We select the top 25 firms which are given in Table 2.

Fig. 5 plots the KGs of top 25 firms. We look at the whole sample (2006–2012) as well as two sub-samples (crisis period of 2007–2009 and the non-crisis period of 2006, 2010–2020) in order to examine any change in interconnectedness in these large firms during the crisis. As noted earlier, spring graphs are used for these firms. It is a force directed layout that uses a spring and electrical forces. "The spring force tries to enforce a certain distance between connected vertices. The electrical force repels the vertices which are close to each other."[28] ". [The] purpose is to position the nodes of a graph in two-dimensional or three-dimensional space so that all the edges are of more or less equal length and there are as few crossing edges as possible, by assigning forces among the set of edges and the set of nodes, based on their relative positions, and then using these forces either to simulate the motion of the edges and nodes or to minimize their energy."[29]

We also examine two cut-offs (low and high). The reason for doing so is that spring graphs will reinforce both strong (pulling) and weak (pushing) edges, which is then presents a much clearer picture for the graph. Note that the cutoffs are different in different graphs. This is because same cutoffs produce inconsistent graphs. In different samples, data (i.e. liquidity index and volatility) are different and hence same cutoff will generate very different graphs. As a result, we vary the cutoffs so the visuals of the graphs are relatively similar to each other. Panels in

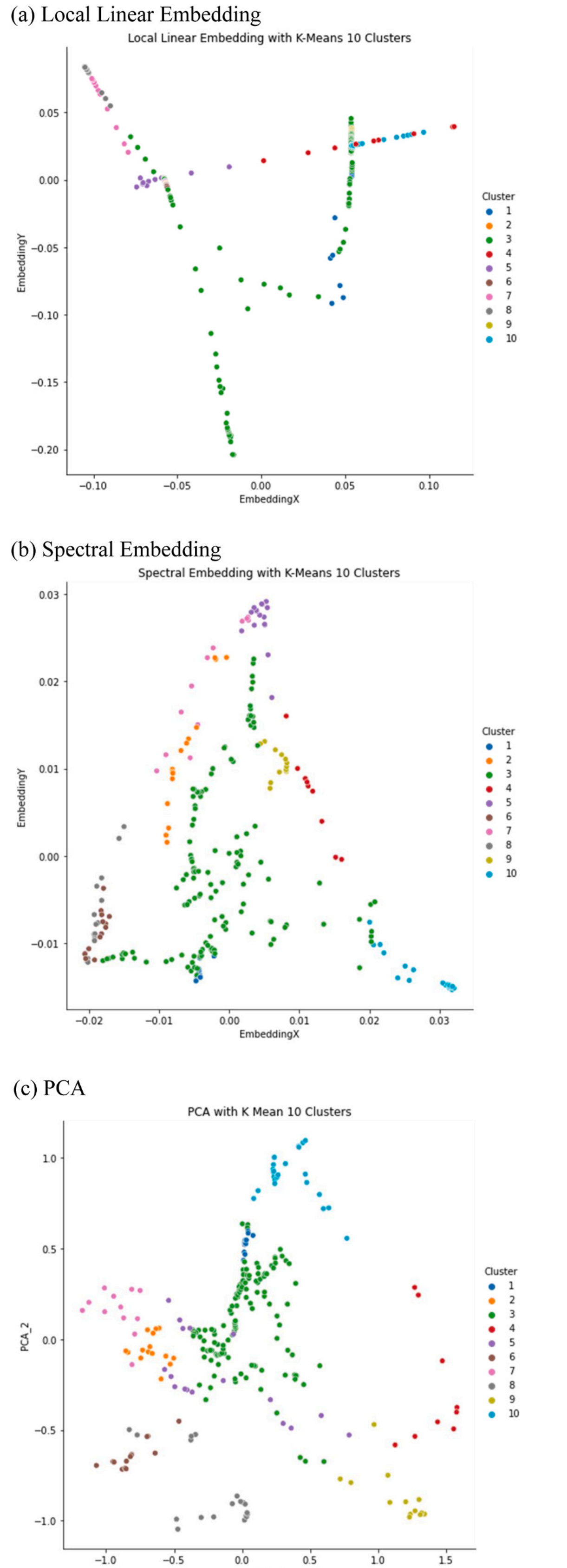


Fig. 3. All Banks Knowledge Graph (Liquidity Discount) – Whole Period (2006–2012), All Banks. Note: These graphs are generated using sklearn. manifold.[embedding method]. For example, in spectral embedding, a Laplacian eigenmap is plotted.

[28] https://www.nevron.com/products-dot-net-diagram-gallery-automatic-layouts-spring-graph-layout.aspx

[29] https://en.wikipedia.org/wiki/Force-directed_graph_drawing

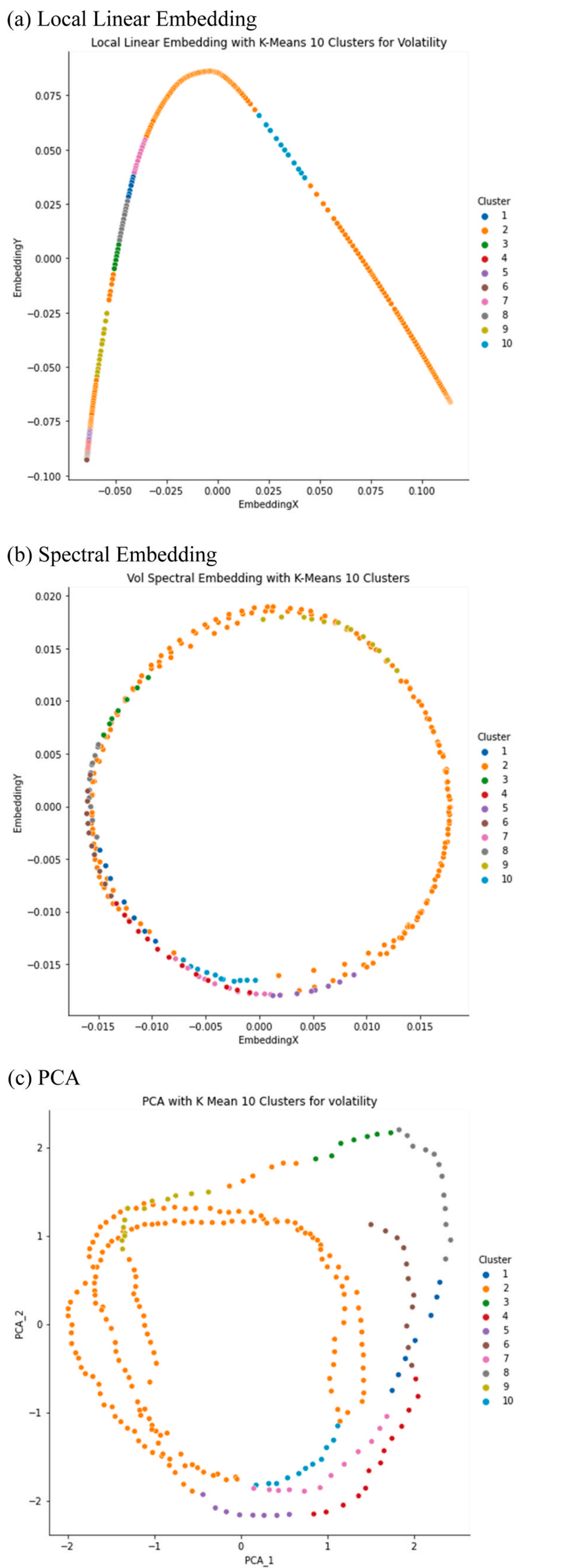


Fig. 4. All Banks Knowledge Graph (Volatility) – Whole Period (2006–2012), All Banks.

Table 2
Top 25 Banks.

Name	Ticker
ACE	ACE LIMITED
AFL	A F L A C INC
AIG	AMERICAN INTERNATIONAL GROUP INC
AMT	AMERICAN TOWER CORPORATION
AXP	AMERICAN EXPRESS CO
BAC	BANK OF AMERICA CORP
BBT	TRUIST FINANCIAL CORP
BEN	FRANKLIN RESOURCES INC
BK	BANK NEW YORK INC
BLK	BLACKROCK INC
BRK.B1	BERKSHIRE HATHAWAY INC DEL
C	CITIGROUP INC
CB	CHUBB LIMITED
COF	CAPITAL ONE FINANCIAL CORP
HCP	HEALTH CARE PROPERTY INVESTORS INC
JPM	JPMORGAN CHASE & CO
MET	METLIFE INC
PNC	P N C FINANCIAL SERVICES GRP INC
PRU	PRUDENTIAL FINANCIAL INC
PSA	PUBLIC STORAGE (PSA)
SPG	SIMON PROPERTY GROUP INC NEW
STT	STATE STREET CORPORATION (STT)
TRV	ST PAUL TRAVELERS COS INC
USB	U S BANCORP DEL
WFC	WELLS FARGO & CO NEW

Note that ACE and CB merged on January 14, 2016. Hence they are two different insurance companies in our sample. BBT was formerly known as Bankers Trust and merged with SunTrust Bank in December 2019.

Fig. 5 present the following KGs:

Panel (a) of Fig. 5 (whole sample period, 2006–2012) presents a fairly complex web. All firms are well-connected except for Chubb Limited (CB) and Bankers Trust (BBT). Within the web, we can find two sub-webs (located left and right) and they are connected via American Tower (AMT), Bank of New York (BK), Healthcare Property Investors (HCP), and PNC bank (PNC).

Panel (b) of Fig. 5 presents the KG of firms during the crisis period (2007–2009). It is clear that all firms now are more closely connected. There are no longer two sub-webs. The outliers in this period are Metlife (MET), State Street (STT), and Berkshire Hathaway (BRK.B).

Lastly is Panel (c) of Fig. 5 which presents the KG for the non-crisis sub-period (2006 and 2010–2012). Immediately we can see that firms are not as connected. Only a few (such BlackRock (BLK), Bank of America (BAC), and Bankers Trust (BBT)) are closely connected while the remaining firms are not.

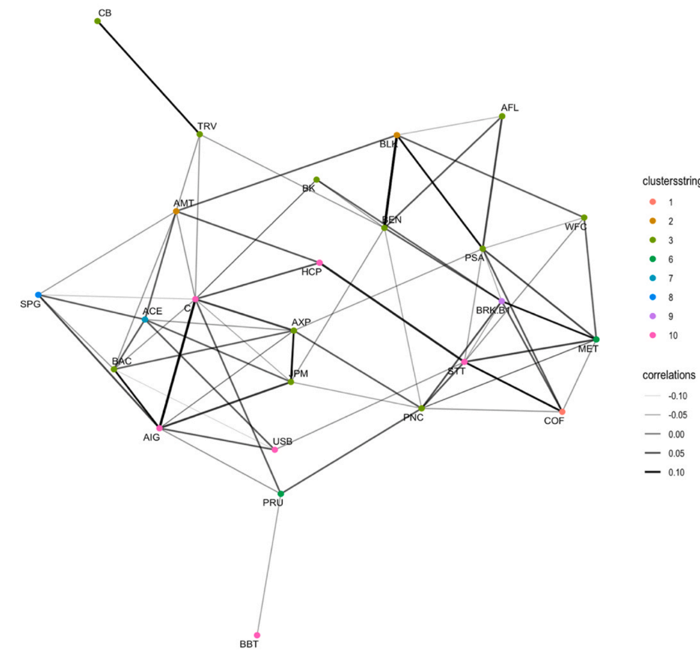
Fig. 6 presents the same set of graphs but with volatility. Ahelegbey (2015, 2016) provides a similar graph for the top European banks. We try to be conservative and use either same or lower cutoffs than the cutoffs used in the liquidity KGs.

Two observations are made. First, the non-crisis period (2006 and 2010–2012) has a more complex web than both the whole period (2006–2012) and the crisis period (2007–2009). This is counter intuitive. One would expect financial firms are more connected during the crisis and during normal times. Secondly, for the crisis period, firms not connected. More than half of the firms are not connected to the web.

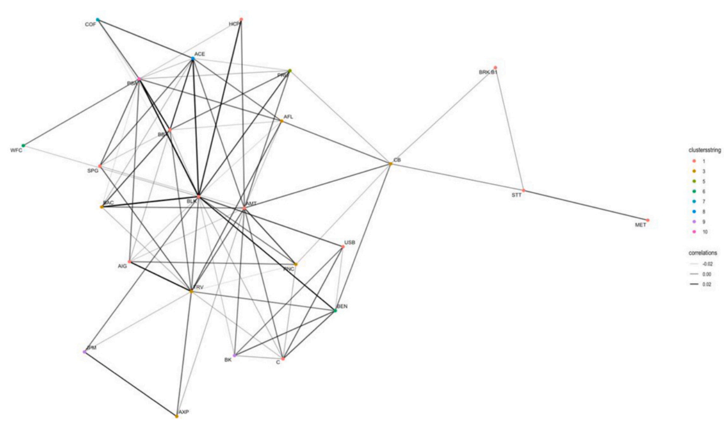
This result is understandable in that volatility does not reflect properly how firms are connected. During a crisis, firms have very high volatility and the magnitudes of their volatility may not move together as much as liquidity indices do. In other words, volatility may be more idiosyncratic than liquidity index does, hence resulting a counter intuitive graph for the crisis period.

We also lower the cutoffs as a robust check. Lower cutoffs lead to more complex webs. We find the results are similar. These graphs are presented in an appendix and attached at the end of the paper.

(a) Full Sample (2006-2012), Top 25 with 0.03 Cutoff



(b) Crisis Subsample (2007-2009), Top 25 with 0.0085 Cutoff



(c) Non-crisis Subsample (2006, 2010~2012), Top 25 with 0.0125 cut-off

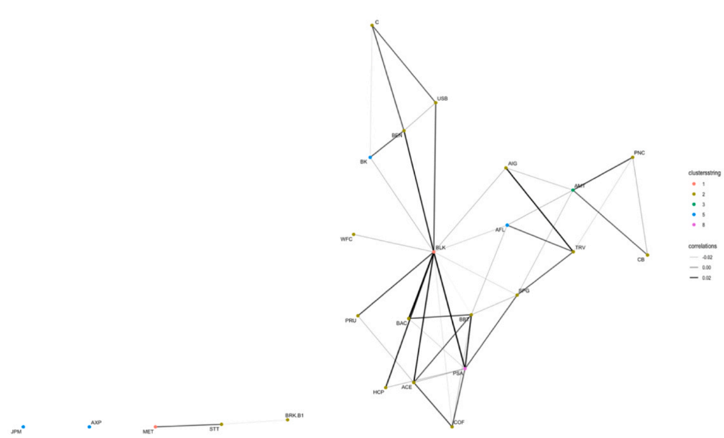


Fig. 5. Selected Banks Knowledge Graph (Spring Graph Network).

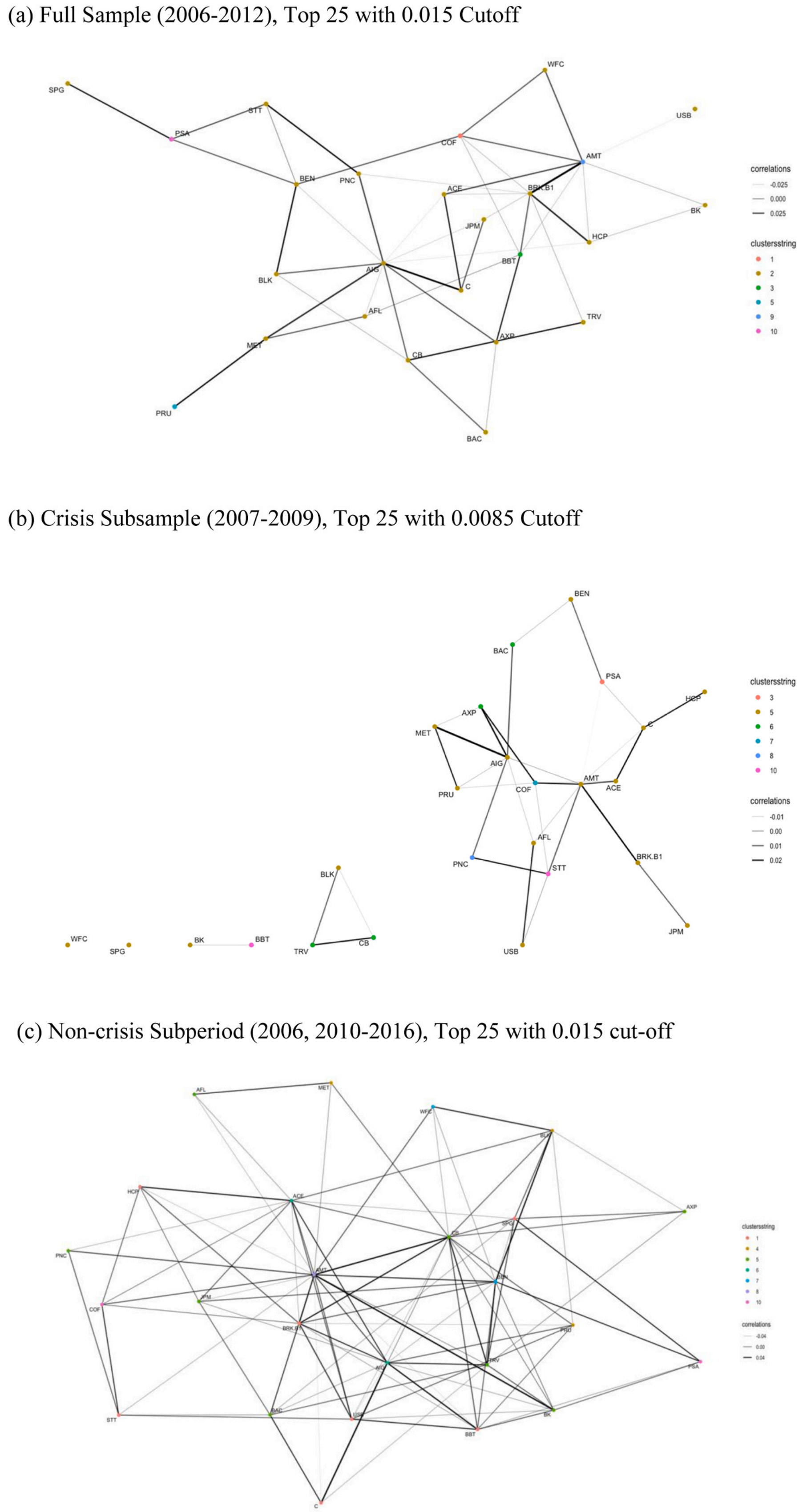


Fig. 6. Knowledge Graphs (Volatility).

7. Conclusion, deficiencies and future research

In this paper, we present a set of knowledge graphs (KG) for the financial firms during the 2008 Lehman crisis. KG is particularly useful in visualizing how firms are connected and hence perfectly suitable for studying the systemic risk during a crisis time. We use two different types of KG – embedding and spring graph.

Our results (spring graphs of top 25 firms) verify the literature and the common wisdom that during crises, firms are more connected than non-crisis times. In addition, we also present results under various embedding methods (all 551 firms) and examine the sensitivity of how KG reacts to embedding. We discover mixed results: (1) closely connected firms are located close to each other but (2) Euclidian distances from one another are not necessarily measure properly their connectedness.

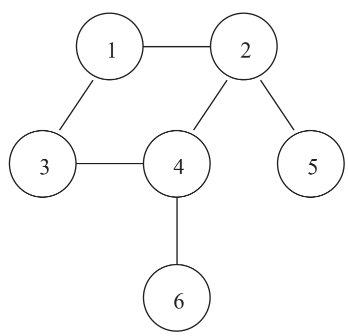
KG, like any other machine learning methods, is static. It is very difficult to analyze the dynamic behavior of a KG and also hard to establish any meaningful statistical inferences. That said, a certain number of hypotheses can still be established and tested. For example, can a feature (features can be ranked using, e.g., random forest) be statistically significant in connecting financial firms? In other words, this paper can be extended to include other features (Cerchiello and Giudici, 2016 use balancesheet data) but this is left to future research.

Due to space limitations, we left off interesting investigations for future research. First, data limitation prevents us from examining how graphs look like in recent episodes after 2008 (such as European crisis in 2015 and the pandemic crisis in 2020). Second, we are not able to compare various distress indices such as, for example, dissimilarity index, stress index, and non-performing loans. Both of these deficiencies are due to lack of easy access to data. Note that such indices are easily accessible (they are not raw data but rather require complex models to compile). Third, we are not able to experiment other embedding methods in that we need relevant textual data. All of these interesting further work is left for future research.

Appendix

A short discussion of undirected graph

Even though the relationships in an undirected are two-way, they are not necessarily symmetrical. In parametric estimations, one can distinguish between the dependencies of two vertices. For example, vertex 4 is connected to both vertices 3 and 6 and hence the probabilities from vertex 4 to vertices 3 and 6 must be both non-negative and sum to 1. On the other hand, the probability from vertex 6 to vertex 4 is 100% since other than vertex 4, vertex 6 has nowhere else to go.



.

In non-parametric estimations, the situations are entirely different. Edges now only measure how tightly any two vertices are related. They do not represent probabilities. Hence, the edges are symmetrical. In many cases, edges are calculated as correlation or any definition of "distance".

In the parametric case, edges are often formulated as a Markov process with the following transition matrix (numbers in the table are probabilities and are arbitrarily given):

	1	2	3	4	5	6
1		0.6	0.4			
2				0.5	0.5	
3	0.4			0.6		
4			0.6			0.4
5		1				
6				1		

If the underlying distribution assumption is Gaussian, then many desirable statistical inferences can be drawn in a dynamic setup.

Embedding

In this appendix, we briefly discuss two (rough) groups of embedding techniques: one that is based upon only numeral data and is used to unfold a complex network (into a lower dimension) and the other that is based upon textual data where connections (i.e. edges) of features are retrieved. This appendix is only to provide a high level description and interested readers can trace the references given here for more details.

The first group of methods includes local linear embedding and spectral embedding. Embedding is meant to find a lower dimension representation of a highly dimensional data structure. This often involves "unfolding" a highly non-linear data structure (residing in a high dimensional space) to its linear (hence lower dimensions) equivalent. The second group[30] of methods includes various translation tools which take advantage of textual data (via natural language processing algorithms) to explain the relations of features. We shall describe each method as follows.

Locally Linear Embedding (LLE) is a method of non-linear dimensionality reduction proposed by Roweis and Saul (2000). LLE seeks a lower-dimensional projection of the data which preserves distances within local neighborhoods. It can be thought of as a series of local Principal Component Analyses which are globally compared to find the best non-linear embedding.[31] It performs the nearest neighbors search using Isomap. There are three different nearest neighbor searches – `BallTree`, `KDTree`, and brute-force (available in scikit-learn). It constructs the weight matrix involving the solution to a $k \times k$ linear equation for each of the N local neighbors.

Spectral Embedding is Laplacian Eigenmaps in action. Laplacian Eigenmaps is considerably similar to Isometric Feature Mapping (also referred to as Isomap). The primary difference between Isomap and Laplacian Eigenmaps is that the goal of Isomap is to directly preserve the global (non-linear) geometry, but the goal of Laplacian Eigenmaps is to preserve the local geometry (i.e., nearby points in the original space remain nearby in the reduced space). There are crucial three steps in achieving spectral embedding:

- Constructing the adjacency graph
- Choosing the weights
- Obtaining the eigenmaps

Finally an Isomap (Isometric Mapping) uses graph distance to the approximate geodesic distance between all pairs of points. Through eigenvalue decomposition of the geodesic distance matrix, it finds the low dimensional embedding of the dataset. In non-linear manifolds, the Euclidean metric for distance holds good if and only if neighborhood structure can be approximated as linear. If neighborhood contains holes, then Euclidean distances can be highly misleading. In contrast to this, if one measures the distance between two points by following the manifold, one will have a better approximation of how far or near two points are.

Translation embedding is often described via a triplet (h, r, t) which represent head entity (i.e. a node), relation (i.e. edge), and tail (a node) respectively so that:

$$t \approx h + r \tag{A1}$$

The objective function to be minimized is the margin-based ranking loss:

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r',t')\in S'} \max \left\{ \gamma + d\left(h + r, t\right) - d\left(h' + r', t', 0\right) \right\} \tag{A2}$$

where S is the set of observed true triples, $S' = \phi(h, r, t)$ is the set of corrupted triples for a given true triple (h, r, t), γ is the margin, and $d(a, b)$ is a distance measure between a and b (e.g., Euclidean or Manhattan distance).

Because translation embedding models includes a one-to-one relation between two entities, in order to build a large-scale knowledge graph embedding, the relationship between many entities has to be constantly added. Recently, several methods have been applied in knowledge graph completion. Elhammadi (2020) demonstrates how word embedding is extracted from Wikipedia and a knowledge base was trained using the TransE model.[32] This model makes the word representation corresponding to the entity in the text as close as possible to the entity representation in the knowledge base. In addition to TransE, there are others various embedding methods:

- TransE: Translating Embeddings for Modeling Multi-relational data:
  - o TransE represents each entity and relation as vectors.
  - o It assumes that relations can be seen as translations in the embedding space.
  - o The model learns to minimize the energy of true triples (entities and relations that exist in the knowledge graph) while maximizing the energy of false triples (entities and relations that do not exist).
- TransH: Knowledge Graph Embedding by Modeling Hierarchical Structures of Relations:
  - o TransH extends TransE by modeling relations in a hyperplane instead of directly translating them.
  - o It introduces a hyperplane for each relation to capture the structural information of the relations.
- TransR: A Unified Model for Knowledge Graph Embedding and Reasoning:
  - o TransR takes a different approach by learning separate projection matrices for entities and relations.
  - o It projects entities and relations into different subspaces, allowing for more expressive modeling of complex relationships.
- ComplEx: Complex Embeddings for Simple Link Prediction:
  - o ComplEx extends the idea of complex numbers to knowledge graph embeddings.
  - o It represents entities and relations as complex-valued vectors and uses the conjugate operation to model various relationships.
- ConvE: Convolutional 2D Knowledge Graph Embeddings:
  - o ConvE utilizes 2D convolutional neural networks to capture local patterns and dependencies in knowledge graphs.
  - o It operates on the 2D matrix representation of entities and relations.
- KG2E: Knowledge Graph Embedding with Entity Types:
  - o KG2E incorporates entity types or categories into the embedding process to enhance the representation of entities.
  - o It leverages both entity and type embeddings.

[30] We thank the referee for suggesting this group of embedding methods.
[31] Taken from scikit-learn 1.2.0 documentation.
[32] For a good reference, also see Bordes et. al. (2013).

DistMult is another popular model for knowledge graph embedding, not specifically for matrix embedding. The DistMult model is based on the idea of bilinear diagonal interactions and is designed to capture the interactions between entities and relations in a knowledge graph. It's important to note that DistMult is designed specifically for knowledge graph embedding and is not a general-purpose method for matrix embedding. If you are looking for matrix factorization techniques or methods for embedding matrices, there are other approaches and models available, such as Singular Value Decomposition (SVD), matrix factorization, or various neural network-based matrix factorization methods. These methods are used for tasks like recommendation systems and matrix completion, which are different from knowledge graph embedding.[33]

The scoring function for the triplet (h, r, t) is given by the element-wise product of the head, relation, and tail embeddings, followed by summing the results:

$$\phi(h, r, t) = \sum\nolimits_{i=1}^{k} h_i \cdot r_i \cdot t_i \tag{A3}$$

where h_i, r_i, and t_i are the ith components of the embeddings of the head entity, relation, and tail entity, respectively, and k is the dimension of the embedding space.

The objective function to be minimized is the negative log-likelihood of the true triples:

$$L = -\sum_{(h,r,t)\in S} \ln\left[p(h, r, t)\right] \tag{A4}$$

where S is the set of observed true triples and $p(h, r, t)$ is the probability of the triplet being true, which can be computed using the softmax function:

$$p(h, r, t) = \frac{\exp\{\phi(h, r, t)\}}{\sum_{h',r',t'} \exp\{\phi(h', r', t')\}} \tag{A5}$$

Gaussian (Probabilistic) graphical models

Graphical models are part of the KG family and use probabilistic modeling for the knowledge. Gaussian graphical models are such models that further employ Gaussian distributions for the underlying variables.

The major advantage of the graphical models is convenience to create "partial correlation" or conditional independency, a key feature in KG. Formally speaking, a multi-variate Gaussian distribution can be written as:

$$\begin{aligned} p(\underline{x}) &= \left((2\pi)^n |\Sigma|\right)^{-1/2} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu})'\Sigma^{-1}(\underline{x}-\underline{\mu})\right) \\ &\propto |\Phi|^{1/2} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu})'\Phi(\underline{x}-\underline{\mu})\right) \end{aligned} \tag{10}$$

where

$$\Sigma = \begin{bmatrix} V_1 & R \\ R' & V_2 \end{bmatrix}$$

and

$$\Phi = \Sigma^{-1} = \begin{bmatrix} K_1 & H \\ H' & K_2 \end{bmatrix}$$

with the following

$$\begin{aligned} K_1^{-1} &= V_1 - RV_2^{-1}R' \\ H &= -K_1RV_2^{-1} \\ K_2^{-1} &= V_2 - R'K_1^{-1}R \end{aligned}$$

and

$$\begin{aligned} \mu_{A|B} &= \mu_A + \Sigma_{A,B}\Sigma_{B,B}^{-1}\left(x_B - \mu_B\right) \\ \Sigma_{A|B} &= \Sigma_{A,A} - \Sigma_{A,B}\Sigma_{B,B}^{-1}\Sigma_{B,A} \end{aligned}$$

[33] See Yang et. al. (2015). https://arxiv.org/pdf/1412.6575.pdf

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jfs.2023.101195.

References

Ahelegbey, Daniel Felix, 2015 (July), Bayesian Graphical Models with Economic and Financial Applications, Dissertation, Department of Economics, Ca' Foscari University of Venice. #833894–1170527.pdf.

Ahelegbey, Daniel Felix, 2016. The econometrics of bayesian graphical models: a review with financial application. J. Netw. Theory Financ. 2 (2), 1–33 (#The_Econometrics_of_Bayesian_Graphical_M.pdf).

Alexanderson, Gerald L., 2006. About the cover: euler and konigsberg's bridges: a historical view (October). Bull. (N. Ser.) Am. Math. Soc. 43 (4), 567–573.

Benazzoli, Chiara, Persio, Luca Di, 2016. Default contagion in financial networks. Int. J. Math. Comput. Simul. Volume 10, 217552812.pdf.

Birch, Jenna, Pantelous, Athanasios A., Soramäki, Kimmo, 2015. Analysis of correlation-based networks representing DAX 30 stock price returns. Comput. Econ. 1–25, 80778428.pdf.

Bordes, Antoine, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko, 2013, Translating Embeddings for Modeling Multi-relational Data, Universite de Technologie de Compiegne – CNRS and Google.

Boss, Michael, Elsinger, Helmut, Summer, Martin, Thurner, Stefan, 2004. Network topology of the interbank market. Quant. Financ. 4, 677–684.

Caccioli, Fabio, Barucca, Paolo, Kobayashi, Teruyoshi, 2018. Network models of financial systemic risk: a review. J. Comput. Soc. Sci. volume 1 pages 81–114. #s42001-017-0008-3.pdf.

Cerchiello, Paola, Giudici, Paolo, 2016. Conditional graphical models for systemic risk estimation. Expert Syst. Appl. Volume 43. Pages 165-174. #1-s2.0-S0957417415006041-main.pdf.

Chen, Ren-Raw, 2012. Valuing a liquidity discount (Winter). J. Fixed Income Vol. 21 (No. 3), 59–73.

Chen, Ren-Raw, Filonuk, William, Patro, Dilip, Yan, An, 2013. Valuing financial assets with liquidity discount: an implication to basel III (December). J. Fixed Income Vol. 22 (No. 3), 45–63.

Chen, Ren-Raw, Lin, Wenlin, Wei, He, 2016. A liquidity index (Spring). J. Fixed Income 25 (4), 5–19.

Denev, Alexander, Papaioannou, Adrien, Angelini, Orazio, 2020. A probabilistic graphical models approach to model interconnectedness (July). Int. J. Risk Assess. Manag. 23 (2), 119–133. SSRN-id3078021.pdf.

Detering, Nils, Thilo Meyer-Brandisy, Konstantinos Panagiotouy, and Daniel Rittery, 2019 (April), Systemic Risk in Networks, University of California, Santa Barbara. systemic-risk-in-networks.pdf.

Dolfin, Marina, Damian Knopoff, Michele Limosani, and Maria Gabriella Xibilia, 2019 (August), Credit Risk Contagion and Systemic Risk on Networks, Mathematics. mathematics-07–00713.pdf.

Eboli, Mario, 2007 (September), Systemic Risk in Financial Networks: A Graph-theoretic Approach, Facoltà di Economia, Università G. d Annunzio. Systemic_risk_in_financial_networks_a_graph-theore.pdf.

Elhammadi, Sarah, 2020, Financial Knowledge Graph Construction, Dissertation, University of British Columbia. ⟨https://open.library.ubc.ca/soa/cIRcle/collections/ubctheses/24/items/1.0392614⟩.

Engel, Janina, 2019, Financial Network Models: Assessing Systemic Risk, Dissertation, Technical University At Munich. 1524120.pdf.

Giudici, Paolo, and Laura Parisi, 2016, CoRisk: Measuring Systemic Risk through Default Probability Contagion, University of Pavia and NYU Stern School of Business. GIUDICI.pdf.

Kissell, Robert, 2020, Algorithmic Trading Methods: Applications Using Advanced Statistics, Optimization, and Machine Learning Techniques, Academic Press.

Nicola, Giancarlo, Cerchiello, Paola, Aste, Tomaso, 2020. Information network modeling for U.S. banking systemic risk. Entropy 22 (11) entropy_22_01331.pdf.

Phua, Clifton, Vicent Lee, Kate Smith, and Ross Gayler, 2010, A comprehensive survey of data mining based fraud detection research," Monash University. ⟨https://arxiv.org/ftp/arxiv/papers/1009/1009.6119.pdf⟩.

Ravi, Kumar, Ravi, Vadlamani, 2015. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowl. -Based Syst. 89, 14–46. ⟨https://www.sciencedirect.com/science/article/abs/pii/S0950705115002336⟩.

roboZhou, Zhipu, 2020, Graphical Models in Financial Market and Portfolio Allocation: Applications and Considerations, Dissertation, UC Santa Barbara. qt5156c57j.pdf.

Roweis, Sam T., Saul, Lawrence K., 2000. Nonlinear dimensionality reduction by locally linear embedding. Publ. der Univ. Bielef.

Smitshoek, R. 2021 (March), Modeling Systemic Risk in Banking Using a Graphical Extremal Model, Master Thesis, Erasmus University Rotterdam. Thesis_511299_adjusted.pdf.

Soramäki, Kimmo, Samantha Cook, and Alan Laubsch, 2016, A Network-based Method for Visual Identification of Systemic Risks, Journal of Network Theory in Finance 2 (1), 67–101. A_network_based_method_for_visual_identification_of_systemic_risks.pdf.

Tabak, B.M., M. Takami, J.M.C. Rocha and D.O. Cajueiro, 2011, Directed Clustering Coefficient as a Measure of Systemic Risk in Complex Banking Networks, Banco Central Do Brasil. wps249.pdf.

Tumminello, Michele, Lillo, Fabrizio, Mantegna, Rosario N., 2010. Correlation, hierarchies, and networks in financial markets. J. Econ. Behav. Organ. 75, 40–58, 0809.4615.pdf.

Yang, Bishan, Yih, Wen-tau, He, Xiaodong, Gao, Jianfeng, Deng, Li, 2015. Embedding entities and relations for learning and inference in knowledge bases. Cornell Univ. Microsoft Res. ⟨https://arxiv.org/pdf/1412.6575.pdf⟩.

Yeh, I.-Cheng, Lien, Che-hui, 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst. Appl. 36 (2), 2473–2480.

Yu, Jiannan, and Jinlou Zhao, 2020, Prediction of Systemic Risk Contagion Based on a Dynamic Complex Network Model Using Machine Learning Algorithm, Complexity (online). 6035372.pdf.

Zhan, Ni, Yijia Sun, Aman Jakhar, and He Liu, 2021, Graphical Models for Financial Time Series and Portfolio, in Toronto Machine Learning Series (TMLS), Carnegie Mellon University. 2101.09214.pdf.