Understanding Supply Chain Networks

Investigate supply network structures & their performance consequences

13th March 2017

Name: Animesh Biyani

PGID: 61710155

Contents

Acknowledgements

I would like to thank Prof. Sridhar Seshadri for his invaluable guidance and insights. I would also like to thank Prof. Vaidyanathan for his help of the project.

A special thanks to Chandra Prakash Shukla and Prashanth Hariharan for helping me out at every stage and clearing all my doubts. I would also like to thank my friends A Lasya Priya and Sriharsha Chillara for their help in writing the code.

Introduction

The objective of this project is two-fold:

1. to develop a visualization tool for the supplier/customer network of various public firms
2. to understand the impact of network parameters on inventory performance

The visualization tool has been developed using the library "shiny" available with the programming language R. The degree of the directed network is close to 30,000 and it contains about 80,000 edges. The tool allows the user to see the networks (both suppliers and customers) of various firms and provides additional functionality described later in the report. In addition to visualization, the tool also allows the user to view network centrality measures such as Eigenvector centrality and Betweenness centrality, clustering coefficients and global properties such as characteristic path length.

To understand the impact of network on inventory performance we used "inventory turnover" as the proxy for inventory performance. We then ran a regression with inventory turnover as the dependent variable and the network parameters (betweenness centrality, number of suppliers, number of customers and eigenvector centrality) as the independent variables. We find that the firms with high influence (high eigenvector centrality) tend to have a higher inventory turnover.

Literature Review

In 1998, Watts and Strogatz formalized the concept of "small-world" networks and since then network science has gained immense popularity and has been adopted in several disciplines such as Finance, Economics and Operations Management. In this project, we focus on the implication of network structure on inventory performance.

In network analysis, the "centrality" measure is useful is identifying the most important nodes. We will focus on two of these centrality measures:

i) Betweenness centrality: denotes how many times a node occurs on the shortest path between all other pairs of nodes, e.g. betweenness centrality of node "v" can be calculated as:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} denotes the number of shortest paths between nodes s and t; $\sigma_{st}(v)$ denotes the number of shortest paths between s and t that pass through v

ii) Eigenvector centrality: denotes the influence of a node in the network (see e.g. Google's PageRank). This is a relative measure and assigns scores based on the concept that connections to high-influence nodes contribute more than connections to low-influence nodes.

To measure inventory performance we will use the inventory turnover ratio. Inventory turnover ratio tells us how fast a company is able to convert its inventory into sales. This value is usually

compared to the industry average – a higher value could indicate strong sales or heavy discounts, while a lower value indicates weak sales.

$$Inventory\ turnover = \frac{Sales}{Average\ Inventory}$$

Apart from firm-specific factors, this ratio also depends on the industry in which the firm operates. Therefore, the sample we took adjusts for the fact that different industries require different levels of inventory.

Data Collection & Cleaning

The data on network of suppliers and customers has been collected from the S&P Capital IQ database. Capital IQ provides information on suppliers/customers/vendors/creditor/licensor, their primary industry, subsidy involved, and business description. For the purposes of this project, we will use only the suppliers and customers along with their primary industry.

Capital IQ segregates its entire database into 11 primary industries (which are further divided into sub industries), viz. Consumer Discretionary, Consumer Staples, Energy, Financial, Healthcare, Industrials, IT, Materials, Real Estate, Telecommunication and Utilities. This allows us to scrape the data industry-wise from the database. The scraping program is divided into 2 parts: i) first we need the unique ID assigned by Capital IQ to each firm; ii) get the suppliers and customers data using the unique ID

To obtain the unique ID we used a library called "BeautifulSoup" available for Python (see Appendix 1). We parse the HTML source code for each firm in the list using beautifulsoup and extract the ID from the appropriate tag. To simulate a browser environment we used a library called "Selenium", also available for Python. It is necessary to have a valid capital iq username and password for the script to work. Once we have the company IDs and a browser environment, we can download the supplier/customer information (see Appendix 2 & 3).

The supplier/customer files downloaded from capital iq contain a lot of redundant columns such as ticker, entity, customer name, business description. Furthermore, each file follows a different format which makes it difficult to extract information. Therefore, we first wrote a script to remove all the redundant information and convert all the files into the same format (see Appendix 4 for the Python code).

The second set of information that we needed was the inventory turnover ratio for various firms. For this we used the Factset database provided by WRDS (Wharton Research Data Services). From this database we were able to obtain the inventory turnover for 52 firms over a period of 2 years – from 2009 to 2011 (see Appendix 5). The Factset database requires ticker names as inputs to its query. We can obtain a complete list of tickers from the capital iq data and provide it as an input to the Factset database. Once we obtained the inventory turnover data from the database, the final step was to group the data based on ticker name and take an average of the inventory turnover across the two years.

Visualization Tool

The network visualization has been developed using the programming language R. The following libraries were used to develop the various functionalities:

i) igraph: for generating the networks, plotting, and calculating the centrality parameters

ii) shiny: to develop the UI for the tool

The first step was to generate the network and store it in a format readable by the tool. The network that we are interested in is a directed network with the following property: an incoming edge is from a supplier and an outgoing edge is to a customer. To capture this directionality, we store the edges of the network in a csv file as a vector with the following property: customers are placed at even indices and suppliers are placed at odd indices.

Alongwith the edges file, we also create a mapping for each firm and its industry and store in a separate file. The industry information is used to color the nodes in the network.

Using igraph we can also calculate the betweenness score for each node in the network. However, the algorithm takes a long time to run. Therefore, to prevent the tool from crashing, we will store these values in another file. The eigenvector centrality can be calculate in real time as it is not as computationally intensive as betweenness centrality.

The next step is to develop the UI using shiny. The tool contains two panels: a side panel where the user can enter the inputs, and a main panel to display the network and network parameters (Appendix 6)

The tool requires 4 inputs:

i) name of the company

ii) number of layers (denotes how many layers deep do you want to plot the network, 1 means immediate suppliers/customers)

iii) layout of the plot (options include layout.fruchterman.reingold, layout.random, etc)

iv) type of graph (only suppliers, only customers, both)

To display the network parameters, the user can click on any node on the plot and the relevant information will be displayed in a box below the plot (Appendix 6).

Regression Model

The next step was to check if the network structure has an impact on the inventory performance of a firm. For this, we developed a simple linear regression model with Inventory turnover as the dependent variable. We chose the following independent variables:

i) number of customers

ii) number of suppliers

iii) eigenvector centrality

iv) betweenness centrality

The summary of the model is as shown below:

Coefficients:					
	Estimate	Std. Error	t value	Pr(> |t|)	Significance
(Intercept)	1.46E+01	4.99E+00	2.923	0.00582	**
BETWEENNESS	-6.59E-08	1.88E-07	-0.351	0.72753	
NUM_SUPPLIERS	-4.96E-02	3.82E-02	-1.299	0.20166	
NUM_CUSTOMERS	-9.16E-02	7.51E-02	-1.219	0.2304	
EIGEN	9.78E+01	3.23E+01	3.03	0.00438	**

From the summary it is clear that the coefficient of eigenvector centrality is highly significant and positive. Therefore, we can say that as the influence of a firm increases in the network, there is an increase in its inventory turnover cycle.

Appendix

Appendix 1 – Python code to obtain IDs

```
browser = webdriver.Chrome(executable_path='/Users/abiyani/Downloads/chromedriver')
url = 'https://www.capitaliq.com/ciqdotnet/login-sso.aspx'
browser.get(url)
username=browser.find_element_by_id('username')
username.send_keys(enter username here)
password=browser.find_element_by_id('password')
password.send_keys(enter password here)
loginButton=browser.find_element_by_id('myLoginButton')
loginButton.click()
time.sleep(5)
link = browser.find_element_by_xpath('//a[@href]')
link.click()
time.sleep(2)
username=browser.find_element_by_id('username')
username.send_keys(enter username here)
password=browser.find_element_by_id('password')
password.send_keys(enter password here)
loginButton=browser.find_element_by_id('myLoginButton')
loginButton.click()
time.sleep(10)

f = open('/Users/abiyani/Downloads/Nasdaq_NorthAmerica.txt', 'a')

i = 0
while i < len(tickr_list):
    print i
    try:
        searchbar = browser.find_element_by_id('SearchTopBar')
    except NoSuchElementException:
        continue
    except TimeoutException:
        continue

    searchbar.clear()
    searchbar.send_keys(name_list[i])
    time.sleep(2)
    soup = BeautifulSoup(browser.page_source)
    search_result = soup.find('a', {'class':'acResultLink'})
    try:
        href = search_result['href']
    except:
        i += 1
        continue

    str = re.search(r'company(.+)=(\d+)', href)
    if str:
        f.write(tickr_list[i] + ':' + str.group(2) + '\n')
    else:
        f.write(tickr_list[i] + ':\n')
    i += 1
```

Appendix 2 – Python code to download suppliers/customers data

```
chrome_options = Options()
chrome_options.add_argument('download.default_directory=/Users/abiyani/Downloads/Suppl
iers')

# Global variables
global browser
company_id_start = 24000
company_id_end = 30000
current_company_id = company_id_start
max_login_attempts = 3

def capital_iq_login():
    global browser
    url = 'https://www.capitaliq.com/ciqdotnet/login-sso.aspx'
    browser =
webdriver.Chrome(executable_path='/Users/abiyani/Downloads/chromedriver',
chrome_options=chrome_options)
    browser.get(url)
    time.sleep(10)
    username = browser.find_element_by_id('username')
    username.send_keys('chandra_shukla@isb.edu')
    time.sleep(2)
    password = browser.find_element_by_id('password')
    password.send_keys('Isb@1234')
    time.sleep(2)
    loginButton = browser.find_element_by_id('myLoginButton')
    loginButton.click()
    time.sleep(5)
    if 'Welcome' in browser.title:
        return 1
    if 'error' in browser.title.lower():
        return 0
    link = browser.find_element_by_xpath('//a[@href]')
    if 'Login' in link.text:
        link.click()
        time.sleep(5)
        username = browser.find_element_by_id('username')
        username.send_keys('chandra_shukla@isb.edu')
        time.sleep(2)
        password = browser.find_element_by_id('password')
        password.send_keys('Isb@1234')
        time.sleep(2)
        loginButton = browser.find_element_by_id('myLoginButton')
        loginButton.click()
        time.sleep(5)
        if 'Welcome' in browser.title:
            return 1
        else:
            return 0

def get_suppliers():
    global current_company_id, browser
    url =
'https://www.capitaliq.com/CIQDotNet/BusinessRel/Suppliers.aspx?CompanyId='+str(curren
t_company_id)
    try:
        browser.get(url)
    except TimeoutException:
        return 0
```

```
    if 'error' in browser.title.lower():
        current_company_id += 1
        return 1

    time.sleep(2)

    try:
        soup = BeautifulSoup(browser.page_source)
        suppliers_list = soup.find_all('table', {'class':'cTblListBody'})
        if len(suppliers_list[0].findChildren()) <= 3 and
len(suppliers_list[1].findChildren()) <= 3:
            current_company_id += 1
            return 1
    except IndexError:
        return 0

    try:
        browser.find_element_by_xpath('//*[@title="Suppliers Report in
Excel"]').click()
        time.sleep(2)
        current_company_id += 1
    except NoSuchElementException:
        current_company_id += 1
        return 1

login_attempt = 1
isLoggedIn = capital_iq_login()
while isLoggedIn == 0 and login_attempt <= max_login_attempts:
    login_attempt += 1
    isLoggedIn = capital_iq_login()

if login_attempt > max_login_attempts:
    sys.exit('Unable to login')

dict_of_id_parsed = {}
while current_company_id <= company_id_end:
    isSuccess = get_suppliers()
    try:
        WebDriverWait(browser, 2).until(EC.alert_is_present(), 'Popup detected.
Restarting.')
        alert = browser.switch_to_alert()
        alert.accept()
        browser.close()
        capital_iq_login()
    except TimeoutException:
        pass
    if isSuccess == 0:
        browser.close()
        capital_iq_login()
    else:
        dict_of_id_parsed[current_company_id-1]=1

browser.close()
```

Appendix 3 – Sample data

Supplier Name	Exchange:Ticker Symbol	Relationship Type	Primary Industry
Cuckoo Electronics Co., Ltd. (KOSE:A192400)	KOSE:A192400	Supplier	Household Appliances
Elematec Corporation (TSE:2715)	TSE:2715	Supplier	Technology Distributors
Tongda Group Holdings Limited (SEHK:698)	SEHK:698	Supplier	Electronic Components
Tsann Kuen (China) Enterprise Co. Ltd. (SZSE:200512)	SZSE:200512	Supplier	Household Appliances

Appendix 4 – Python code to clean the data

```
category = ['Consumer Discretionary Sector Customers', 'Consumer Staples Sector
Customers', 'Energy Sector Customers', 'Financial Sector Customers', 'Healthcare
Sector Customers', 'Utilities Sector Customers']
dir = '/Users/abiyani/Documents/firp data/'
new_dir = '/Users/abiyani/Documents/firp data updated'
for cat in category:
    os.makedirs(os.path.join(new_dir, cat))
    for f in os.listdir(os.path.join(dir,cat)):
        try:
            wb = openpyxl.load_workbook(os.path.join(dir,cat,f))
        except InvalidFileException:
            print f.title()
            continue
        new_wb = openpyxl.Workbook()
        new_ws = new_wb.active
        ws = wb.get_sheet_by_name("Sheet")
        new_ws_column = 1
        isFound = 0

        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Customer Name":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=1).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Customer Name":
                for j in range(1, 14):
                    new_ws.cell(row=j, column=1).value = ws.cell(row=j,
column=i).value
                for j in range(16, ws.max_row+1):
                    new_ws.cell(row=j-2, column=1).value = ws.cell(row=j,
column=i).value
```

```
        new_ws_column += 1

        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Exchange:Ticker Symbol":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=2).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Exchange:Ticker Symbol":
                for j in range(1, 14):
                    new_ws.cell(row=j, column=2).value = ws.cell(row=j,
column=i).value
                for j in range(16, ws.max_row+1):
                    new_ws.cell(row=j-2, column=2).value = ws.cell(row=j,
column=i).value

        new_ws_column += 1

        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Relationship Type":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=5).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Relationship Type":
                for j in range(1,14):
                    new_ws.cell(row=j, column=5).value = ws.cell(row=j,
column=i).value
                for j in range(16,ws.max_row+1):
                    new_ws.cell(row=j-2, column=5).value = ws.cell(row=j,
column=i).value

        new_ws_column += 1

        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Primary Industry":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=6).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Primary Industry":
                for j in range(1,14):
                    new_ws.cell(row=j, column=6).value = ws.cell(row=j,
column=i).value
                for j in range(16,ws.max_row+1):
                    new_ws.cell(row=j-2, column=6).value = ws.cell(row=j,
column=i).value

        new_ws_column += 1
```

```
        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Source":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=7).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Source":
                for j in range(1,14):
                    new_ws.cell(row=j, column=7).value = ws.cell(row=j,
column=i).value
                for j in range(16,ws.max_row+1):
                    new_ws.cell(row=j-2, column=7).value = ws.cell(row=j,
column=i).value

        new_ws_column += 1

        for i in range(1, ws.max_column+1):
            if ws.cell(row=14, column=i).value == "Business Description":
                for j in range(1, ws.max_row+1):
                    new_ws.cell(row=j, column=8).value = ws.cell(row=j,
column=i).value

        for i in range(1, ws.max_column+1):
            if ws.cell(row=16, column=i).value == "Business Description":
                for j in range(1,14):
                    new_ws.cell(row=j, column=8).value = ws.cell(row=j,
column=i).value
                for j in range(16,ws.max_row+1):
                    new_ws.cell(row=j-2, column=8).value = ws.cell(row=j,
column=i).value

        new_ws_column += 1

        new_wb.save(os.path.join(new_dir,cat,f))
```

Appendix 5 – Inventory Turnover (source: WRDS)

Company	inv_turn
AMERCO	29.93766667
Agilysys Inc	63.69683333
Anadarko Petroleum corporation	30.36033333
Ascena Retail Group Inc	3.901083333
Ashland Inc	5.224333333
Avery Dennison corporation	8.651666667
Avon Products Inc	3.357083333
BB&T corporation	0.573916667
Baxter International Inc	1.810916667
BorgWarner Inc	12.01108333
Boston Scientific corporation	2.0575
Briggs & Stratton corporation	3.319583333
Cabot corporation	5.07025
Carmike Cinemas Inc	159.789
Chart Industries Inc	3.55725
Cincinnati Bell Inc	29.79008333
Citigroup Inc	0.250833333
Cliffs Natural Resources Inc	8.819416667
Columbus McKinnon corporation	3.693
Costco Wholesale corporation	11.36975
Crown Holdings Inc	5.385
Dover corporation	4.98425
Duke Energy corporation	4.496166667
EQT corporation	119.3024167
General Cable corporation	4.342916667
General Dynamics corporation	3.203416667
Harsco corporation	7.842333333
Hasbro Inc	4.993583333
Hawaiian Holdings Inc	84.48366667
Hecla Mining Co	6.409916667
Hologic Inc	2.84025
Jabil Circuit Inc	7.283
Kaiser Aluminum corporation	5.314083333

Lincoln Electric Holdings Inc	5.21875
Masco corporation	6.19975
Microsoft corporation	10.18591667
Murphy Oil corporation	6.423583333
Murphy USA Inc	75.013
Newpark Resources Inc	4.00975
Nucor corporation	6.729416667
Omnicom Group Inc	9.625833333
Owens & Minor Inc	9.86875
PPL corporation	6.946
Pilgrim's Pride corporation	8.579416667
Pitney Bowes Inc	14.53158333
Quest Diagnostics Inc	41.73883333
Range Resources corporation	58.62641667
Sanmina corporation	6.612583333
Scholastic corporation	2.482
Sotheby's	2.14275
Standard Motor Products Inc	2.423916667
Starbucks corporation	11.90783333
Steel Dynamics Inc	5.109916667
Stryker corporation	1.98075
Union Pacific corporation	18.02383333
Unit corporation	173.39
Veeco Instruments Inc	4.43575
Western Digital corporation	7.338833333
Weyerhaeuser Co	8.90475

Appendix 6 – Visualization tool