Report: Comtrade - Big Data

statswiki.unece.org/display/bigdata/Report:+Comtrade

Abstract

International Merchandise Trade Statistics (IMTS) is considered as one of the oldest statistical domains. The UN Statistical Division has compiled official trade statistics in their Comtrade database since 1962. This database now contains billions of records. Due to interest in measuring economic globalization through trade, trade data has been used to analyse interlinkages between economies. Thanks to big data technologies, it is now feasible to process complex calculations derived from trade data. This project aimed to analyse and visualize regional global value chains (focusing on trade in intermediate goods) using tools in the sandbox.

Description

In terms of data source, IMTS compilers have been relying on Customs declarations (part of administrative data) supplemented by border or enterprise surveys. Depending on the size of national economies, there may be thousands or more declarations each day translating into millions of data points in a year. In this regard, trade statisticians and analysts are not new to dealing with the compilation and analysis of huge volumes of data. However, due to limitations of computing power and tools, this takes a long time to complete. Data often need to be split into smaller chunks for easier handling. UN Comtrade (comtrade.un.org) compiles all official trade data produced by national institutions, standardizes them and makes them available publicly. This project is intended to explore the use emerging tools in sandbox (that can process huge amount of data) to identify regional global value chain networks and analyse their properties.

Data characteristics

The input data for the project consists of detailed trade data (by partner and by product), and total trade. Data sources are UN Comtrade (comtrade.un.org) for detailed trade data and total trade data sets from UNSD (http://unstats.un.org/unsd/trade/data/tables.asp#monthlytotal). UN Comtrade consists of billions of trade data records in multiple trade classifications since 1962. The database is regularly updated. All of the data are publicly available and can be accessed from UN Comtrade Public API (http://comtrade.un.org/data/Doc/API).

The data schema is as follows:

Variable Name	Data Type	Variable Description

classification	string	Commodity Classification
year	smallint	Reference Year
period	int	Indicate months/years
period_desc	string	Description of the period
aggregate_level	smallint	Level of commodity codes
is_leaf_code	boolean	Indicate if the commodity code is the most detailed level or not
trade_flow_code	smallint	Code of trade flow
trade_flow	string	Description of trade flow
reporter_code	smallint	Code of Reporter
reporter	string	Country/territory that reports the trade data
reporter_iso	string	ISO code of Reporter
partner_code	smallint	Code of trading partner of the Reporter
partner	string	Trading partner description
partner_iso	string	ISO code of the Reporter
commodity_code	string	Code of commodity
commodity	string	Description of commodity
qty_unit_code	smallint	Code of quantity unit
qty_unit	string	Description of quantity unit
qty	bigint	Value of quantity
netweight_kg	bigint	Netweight
trade_value_us	bigint	Trade value in US$
flag	smallint	Quantity estimation flag

The data stored in the sandbox are from the year 2000 to 2010, according to the Harmonized System (HS) classification for all available reporters . Due to data gaps (not all countries report data in a specific year), the project team estimated the missing trade data in order to achieve global coverage. The total number of data points is around 325 millions records.

Activities

Tools

HADOOP: Trade data are characterized by their huge volume. They have many variables, but are highly structured. This combination of data characteristics fit with the Hadoop technology: easy to process and analyse data in raw text format, without the need to import such data files into a specialized database. *Used as data storage.*

PIG: This scripting-based language is suitable to clean incoming data files (updating the text qualifiers). *Used in data cleaning phase.*

HIVE: This tool make it easy for SQL-specialists that have been using traditional relational database management systems (RDBMS) to query and analyse data. Complex SQL queries with sub-queries and nested statements are supported, and it enables calculation of complex methods (such as data estimation). Regarding the performance, the SQL queries that run under sandbox are quite good, and probably better compared to traditional RDBMS (with good design of data structure supplemented by indexes). *Used to perform data preparation.*

RHADOOP-MAP/REDUCE: Open source R package that allows to write and trigger MapReduce jobs in R. It was conveniently used to write the MapReduce script for symmetry analysis directly in RStudio, and to retrieve datasets from HDFS in R for producing visualizations. Negative aspects are a general instability and the lack of a coherent documentation. However, the practical experience from this project could be part of material for training. *Used to carry out analysis of bilateral asymmetry.*

SPARK: An open source cluster computing framework. In contrast to Hadoop's two-stage disk-based MapReduce paradigm, Spark's multi-stage in-memory primitives provides performance up to 100 times faster for certain applications. *Used to calculate trade network properties, and visualize them.*

GEPHI: This software provides an interactive visualization and exploration platform for the easy creation of social data connectors to map community organizations and networks. It runs on Windows, Linux and Mac OS X and is based on NetBeans UI. *Used to visualize network structures.*

D3: Javascript library for developing interactive visualizations. It can handle a large number of data points and it is highly customizable and for these features it is widely used in the visualization of big data. There is quite a big community that continuously maintains existing and adds new visualization libraries. *Used to visualize network structures.*

1 - Data Acquisition

The first activity was to define the periods for analysis and preparation of compressed CSV files from UN Comtrade. It was decided that analysis of 15 years of data is sufficient for the first attempt (translated to 325 million records). Data were extracted from the UN Comtrade database and organised into 15 data files (one year one file). The data files were then FTP-ed to the sandbox environment.

//**Update Nov 2015, UN Comtrade now supports bulk data API http://comtrade.un.org/data/Doc/api/bulk . Therefore data preparation and transfer can be simplified**//

All of those data files were moved into Hadoop file systems, and were made accessible through Hive. Even though, the data format is CSV, it was necessary to develop a specific script to "clean up" text qualifiers in CSV (it seems that Hive does not support double quote as a text qualifier in CSV files, and no similar problem in RStudio).

Pre-processing script in pig

```
a = load '/datasets/comtrade/annual/hs' using PigStorage() as (text:chararray);
b = filter a by (STARTSWITH(text,'Classification')==FALSE);
b1 = FOREACH b GENERATE REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(
   REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE( REPLACE(text, '""',''),
'"smart card"', 'smart card'),
   '"Smart card"', 'Smart card'), '"dental wax"', 'dental wax'), '"dental waxes"',
'dental waxes'),
   '"dental impression compounds"', 'dental impression compounds'), '"Agarbatti"',
'Agarbatti'),
   '"electronic"', 'electronic'), '"herring-bone"', 'herring-bone'),
   '"homogenised"', 'homogenised'), '"reconstituted"', 'reconstituted'), '"scale"',
'scale') as text;
b2 = FOREACH b1 GENERATE REPLACE( REPLACE( REPLACE(text, 'China, Hong Kong SAR',
'China- Hong Kong SAR'),
   'China, Macao SAR', 'China- Macao SAR'), 'Other Asia, nes', 'Other Asia- nes') as
text;

c1 = FOREACH b2 GENERATE FLATTEN(STRSPLIT(text, '"',3)) as (l, c, r);
c1s = FOREACH c1 GENERATE l, REPLACE(c, ',', 'PLACEHOLDER') as c, r;
c1c = FOREACH c1s GENERATE (c is not null ? CONCAT(CONCAT(l,c),r) : l) as text;
c2 = FOREACH c1c GENERATE FLATTEN(STRSPLIT(text, '"',3)) as (l, c, r);
c2s = FOREACH c2 GENERATE l, REPLACE(c, ',', 'PLACEHOLDER') as c, r;
c2c = FOREACH c2s GENERATE (c is not null ? CONCAT(CONCAT(l,c),r) : l) as text;
c3 = FOREACH c2c GENERATE FLATTEN(STRSPLIT(text, '"',3)) as (l, c, r);
c3s = FOREACH c3 GENERATE l, REPLACE(c, ',', 'PLACEHOLDER') as c, r;
c3c = FOREACH c3s GENERATE (c is not null ? CONCAT(CONCAT(l,c),r) : l) as text;
d = FOREACH c3c GENERATE REPLACE( REPLACE(text,',','|'), 'PLACEHOLDER', ',');
store d into '/datasets/comtrade2/annual/hs';
```

Pig was used because it could process data in raw text format and as such could handle substitution of text bits containing commas and quotes, that could not be correctly interpreted when imported in Hive.

2 - Data Processing/Analysis

As indicated earlier, not all countries regularly report data to UN Comtrade. Therefore, there are some data gaps for the combination of reporter-period. The step of data preparation is intended to estimate those data gaps. Furthermore, it is also necessary to map the HS codes into intermediate goods (through BEC classification). See below:

Macro-Phase	Step	Details
Data preparation	1 - Calculate average of trade data for specific period of years	In order to smoothen possible one-time jump/drop in trade (e.g., due to global crisis) and to increase data availability, the trade data is grouped into ranges of years (2000-2005, 2005-2008, 2009-2011, 2011-2014). The trade values are expressed as averages for those year ranges.

	2 - Estimate missing countries by looking at partners and reversing import/export (mirror statistics). Weight by total trade amount for country.	Not every country reports data to UN Comtrade. Therefore, it is necessary to estimate such data gaps. One method is using mirror statistics (estimate trade using data reported by trading partners). The mirror data are then inflated or deflated proportionally by the known total trade of the country. Dataset needed: Total Trade
	3 - Define category of intermediate goods by aggregating harmonized system codes into Broad Economic Category (BEC) codes.	This can be done by mapping the existing HS to BEC correlation table made available by UNSD. Each HS code is assigned a specific BEC code. Finally to obtain BECs for intermediate goods, some BEC codes are aggregated. Datasets needed: HS to BEC correlation table, and BECs Intermediate goods definition. Provided by UNSD in CSV format.
Analysis	4 - Establish the network of imports of intermediate goods. Use a visualization tool, define the data structure for design-representation	For the purpose of this study, only a subset of imports of intermediate goods is needed. Re-imports (imports from itself) is excluded so that the graph does not contain multiple loops. The final graph would be a simple directed graph with no loops and no multiple arcs. At this time, it is possible already to visualise the networks [TARGET 1]. It is expected that many countries would report imports of intermediate goods, therefore it is important to take into account the trade values or share of intermediate goods to total trade (thus creating weighted networks). Needed: Visualization tools, Data structure/design/representation of the network
	5 - Analysis of existinggroups of countries	The networks can be split into several regional networks based on arbitrary approach of geography (e.g., Europe, Asia, America, Africa), development status (e.g., Developed, Developing, LDC), or trade bloc (e.g., EU, ASEAN, COMESA, CARICOM). The results can be visualized along with their networks properties (e.g., density, degree, no of arcs, closeness centralisation) [TARGET 2]. Needed: List of country groups and Formula
	6 - Identification of networks using automated programs implementing data mining techniques.	Using a loop, identify clusters of nodes using a clustering coefficient. The coefficient should take into account the absolute and/or relative values of intermediate goods [TARGET 3].

3 - Data Imputation

One of the first tasks to undertake within the Data Preparation macro-phase was to evaluate the quality of the original information provided in the *annual_hs* table. This table includes in each record the commodity, reporter country, partner country, type of flow (import, export, re-import or re-export), year, and value of each transaction flow, jointly with a number of codes and labels for these variables. Starting with a simple observation of the table, some errors (code errors, invalid blank fields...) were discovered.

The first issue, also the most simple, is to check the coherence between the data provided by the exporter and the importer country, because each commodity import (respectively export) flow reported by a country, should have an export (respectively import) counterpart in the corresponding partner country. The value of this counterpart might not be exactly equal because it is recommended that values for exports are recorded as FOB-type (including the transaction value of the goods plus the value of services performed to deliver goods to the border of the exporting country) while values for imports should be CIF-type (including the FOB-value plus the value of services performed to deliver the goods from the border of the exporting country to the border of the importing country). UNSD commissioned a report on the analysis of bilateral asymmetries in international merchandise trade statistics (IMTS) for the International Conference on the Measurement of International Trade and Economic Globalization at the end of September 2014 in Aguascalientes, Mexico. The report described three main causes of asymmetries, namely partner country attribution, valuation and differences in trade system. It also pointed out that the recommendations of IMTS adopted in 2010 addressed the issues of partner attribution and valuation. It is known that some countries do not follow these recommendations, but, in any case, it could be interesting to see the magnitude of the differences.

The *annual_hs* table is kept in the Sandbox in the *Comtrade* database. The first steps to access the table have been carried out through Apache Hive SQL while more advanced steps, with smaller files that can be more easily processed, have been carried out using Python.

To check the information of a specific commodity, all the flows referring to it are summarised by reporter country, partner country, type of flow and year, using Hive SQL. The resulting dataset is later treated through a Python script to obtain, for each year, a table showing the discrepancies.

Table 1 presents an example of a commodity for several countries in 2014. It shows the percentage discrepancy between the sum of exports of the commodity to the second country (column) declared by the first country (row), and the sum of imports from the first country (row) declared by the second country (column). The magnitude of differences ranges from an acceptable 29% on the exports of the commodity from Philippines (PHL) to Slovenia (SVN), to 381963% from Poland (POL) to Slovakia (SVK), and -100% from Slovenia to Slovakia (there are missing data whenever there is no information from at least one of the countries).

Table 1

Discrepancy of information by importer and exporter country (%) 2014

Commodity 392490: Plastic household, toilet articles not table, kitchen

	PAN	PHL	POL	PRT	PRY	ROU	SLV	SRB	SVK	SVN
PAN	-	-	-	-	-	-	-	-	-	-

PHL	-	-	-	-	-	-	-	-	-	29
POL	-93	-	-	-	-	-	-	-91	381,963	-90
PRT	-	-100	-	-	-	-	-	163,039	28,786	-84
PRY	-	-	-	-	-	-	-	-	-	-
ROU	-	-	-	-	-	-	-	-83	-28	142
SLV	-	-	-	-	-	-	-	-	-	-
SRB	-	-	1,259	-	-	-	-	-	-93	-
SVK	-	-	170	-	-	-	-	-	949	-99
SVN	-	-	649	-	-	-	-100	136	-	-98

The information from these tables can be used to complete the missing data of a (commodity, reporter country, partner country, year, flow) when the corresponding mirror data from the opposite flow exist. Given the magnitude of the differences -which is a common feature through all the commodities- the use of the tables for data editing purposes was shelved until there was information about which data are more reliable. To get an idea of this magnitude, the differences percentage for the aggregation of all the 6-digits level commodities in the same countries is shown in Table 2.

Tabla2

Discrepancy of information by importer and exporter country (%) 2014

	PAN	**PHL**	**POL**	**PRT**	**PRY**	**ROU**	**SLV**	**SRB**	**SVK**	**SVN**
PAN	782,959	1,303	1,282	-	-	-100	-	-	1,160	4
PHL	-	-	2,133	3,165	-31	393	1,408	323,714	1,340	-90
POL	-96	-	-	4884	-100	26,570	-100	130,761,624	420	-88
PRT	45	-99	-	-	-92	2,526	103	-73	292	-89
PRY	-3	-100	423	-	-	486	-79	-	-	-93
ROU	-98	-98	561	21,768	-	-	-100	362,458	220	-36
SLV	56,269	125	-21	132	200	-	-	-	3,029	-99
SRB	-96	-100	811	16,180	-100	1,079,091	-	-	-38	753
SVK	-76	-100	1,471	4,063	-100	891,262	-100	-	-29	-70
SVN	-43	-99	985	1,968	-99	1,911,978	-100	-	-	-72

4 - Bilateral Asymmetry Analysis

An alternative approach for analyzing the discrepancies between flow records has been implemented using RHadoop. RHadoop is an open source project based on two R packages, rmr2 and rhdfs, that allow to integrate R with an Hadoop platform and to write MapReduce jobs directly in R. The scripts are streamed to the Hadoop infrastructure and executed in the cluster. Although this is a low level approach that should be justified only by certain requirements, its use can help with solving problems that do not necessarily fall into the standard categories tackled (mainly data manipulation jobs such as aggregations, joins and filtering). MapReduce jobs written in R are more compact with respect to their Java counterparts and can benefit to the compact syntax and the wide range of specialized functions and packages that R offers.

The code written for analyzing symmetry of flows is very compact and allowed the whole dataset to be handled in a very straightforward way, without resorting to intermediate steps of processing or producing additional datasets. The code is shown in the following figure and explained below.

Code block

```
library(rmr2)
library(rhdfs)
hdfs.init()

bp =
list(
hadoop =
list(
D = "mapred.map.child.ulimit=2097152",
D = "mapred.reduce.child.ulimit=2097152",
D = "mapreduce.map.memory.mb=2048",
D = "mapreduce.reduce.memory.mb=3096"))
rmr.options(backend.parameters = bp);
rmr.options("backend.parameters")

input.file = '<HDFS path to input file>'
output.file='<HDFS path to output file>'

ct.map=function(k,v) {
flow = v[,3]
reporter = v[,5]
partner = v[,7]
commcode = v[,9]
tradevalue= v[,10]
tradevalue.int = as.integer(tradevalue)
if(is.na(tradevalue.int)) return
key = NULL
if(flow==1)
key = paste(reporter,'-',partner,'-',commcode, sep="")
else if(flow==2)
key = paste(partner,'-',reporter,'-',commcode, sep="")
if(!is.null(key))
keyval(key, tradevalue.int)

}

ct.reduce=function(kk,vv) {
if(length(vv)==1)
keyval(gsub("-",",",kk),-1)
else if(length(vv)>0){
diff = abs(vv[1]-vv[2])
perc = diff/vv[1]
keyval(gsub("-",",",kk),paste(diff,perc,sep=","))
}
}
mr.result=mapreduce(
input=input.file,
input.format=make.input.format("csv",sep=","),
output=output.file,
output.format=make.output.format("csv",sep=",", quote=FALSE),
map=ct.map,
reduce=ct.reduce
)
```

The idea is very simple: In the map phase couples of records that correspond to the same flow are mapped to a same intermediate key, i.e. import of a reporter-partner pair for a specific commodity code are matched to the partner-reporter export. In the reduce phase, the list of values associated to a flow are analysed. If only one value is present it means that one record is missing. If two records are in the list, the difference is computed in both absolute and percentage terms, and written to disk.

The downside of this approach is that the execution appears to be slower with respect to jobs implemented in Hive-Pig (e.g. it took more than 1 hour to process only 1 year of data - 1Gb approx). Moreover, documentation is very basic and covers mainly simple tutorials. As such, it is not sufficient to prepare to write production-level scripts for complex processing of real-world datasets. In general, everything that deviated from the default behavior described in the tutorial required a lot of trial and error and so this experience was important to document a real use case of RHadoop, that can be shared in the community.

5 - Data Visualization with Gephi

The Gephi network visualization software has been chosen to visualize the data at an aggregated level. For this purpose, the table *yearrange_bec* uploaded in the *Comtrade* database has been used as starting point. This table summarises the **imports** appearing in the table *annual_hs* by groups of years (2000-2005, 2006-2008, 2009-2011, 2012-2014), reporter country, partner country, and BEC code. It has been completed when there was missing information using mirror statistics.

The table has been summarised on its own, selecting exclusively the BEC codes corresponding to **intermediate goods**(111, 121, 21, 22, 31, 322, 42 and 53) and computing totals through Hive SQL.

The obtained table is of medium size and does not need any big data specific software in order to be analysed. All the rest of the pre-processing has been carried out using *Python* for data analysis on Linux and Windows.

Other files including the information on countries, continents, geographical regions in UN terms, and classification of countries by *developed, developing* and *least-developed* have been downloaded from the United Nations web page (http://unstats.un.org/unsd/methods/m49/m49regin.htm). These datasets have been combined with the prepared dataset on the **imports of intermediate goods** (by group of years, reporter country and partner country) to prepare the files for executing the social networks analysis.

A customized social network analysis for the world trade exists in http://wits.worldbank.org/globalnetwork.aspx (built using customized D3 libraries). The graphs below depict the trade network for a selected HS product (or aggregate) from a global perspective. There are two possible viewpoints: buyer and seller. The buyer side shows that the role of each country as source of demand, and node size is proportional to import market share. The seller side shows the role of each country as supplier, and node size is proportional to export market share. For a given period (year) the graphs report the global trade network in the selected commodity identified using the preferred classification/ nomenclature. Underlying information may come from two sources: export declarations or mirror imports declarations (trade flows).

All these graphs analyze the trade network at the country level. Hence, our aim was to prepare the analysis at a more aggregated level, considering geographical regions and classification of countries by economic development.

The files to introduce the data into the *Gephi* social networks software have been prepared through *Python* scripts. Figure 1 shows the network of trade of intermediate goods (2012-2014 average) by geographical regions. The node size is proportional to the export market share in the period, while the edge thickness is proportional to the total world trade share. Network graphs having a similar structure have been obtained for the other periods considered.

Figure 1

Trade of intermediate goods by geographical regions

2012-2014

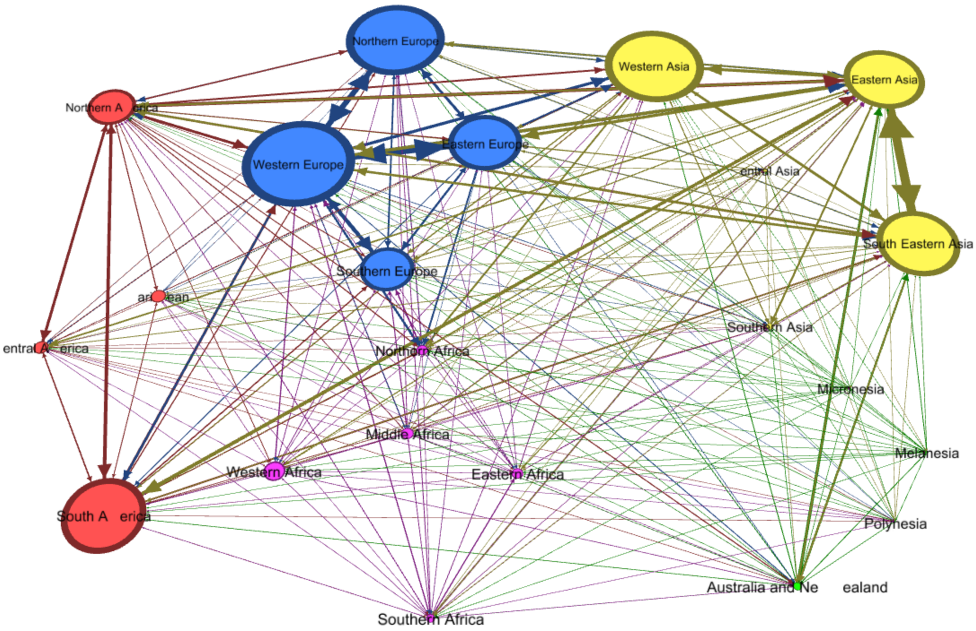
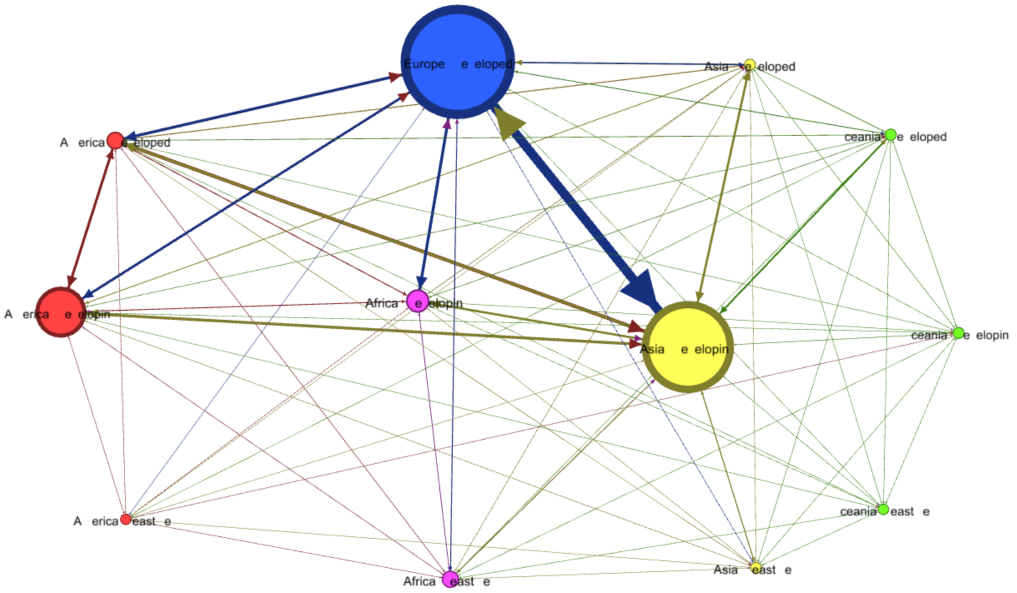


Figure 2 shows the network of trade of intermediate goods in the 2012-2014 average by economic development regions. The graphs for previous periods shows also that there is almost no evolution over time.

Figure 2

Trade of imports of intermediate goods by economic development regions

2012-2014 average



6 - Network Analysis and Visualization with Spark

Generally speaking, graphs can be visualised but also described and analysed from a statistical point of view, using some specific measures and indicators. For large graphs, the Apache project Spark includes a component designed for the analysis of graphs : **GraphX**. Apache project Spark API can been requested with scala, but also mostly with python. Some of the functionalities are now connected with R. But GraphX is, at the time, only available in scala. Among all the measures existing for the description of a graph, some are directly available as properties of the object 'graph' implemented in the library graphx.

We will focus on these measures, **no other measures have been directly implemented.** But it is worth noting that not all the classic measures are, at the time, implemented in the library. For example the betweenness centrality of the graph - which assigns a high score to nodes that are strategically connected on the shortest paths between each pair of nodes - is not implemented, and some classic algorithms for communities detection are not included either. Regarding the latter, some implementation may be found on the web, its use requires some extra understanding of scala and graphx API, so it hasn't be tested in this study.

International trade by category of intermediate goods can be represented as graphs, using countries as vertices and trade flows as directed edges. **Temporality of the data allows us to analyse networks evolution on a yearly basis**. For each category and each year, we have computed the graph of the imports between countries using table annual_hs_bec.

The scala code is organised as follows : first, libraries are loaded, then some variables are defined (list of BEC categories, matrix for storing results..), then we implemented a double loop on years and categories. Inside the loops we did the following operations : filtering the table via sql function, defining the 'couples' variable representing the edges, converting to the 'toLong' format necessary to the use of the 'edges' property of the `graph` object, then defining the 'graph' variable itself. With the graphx library, there are different ways of defining a graph, with the function 'fromEdgeTuples', we built the graph from the list of edges only. The identification of the vertices is automatic.

The most basic indicators we first computed are: **vertices and edges counts**. The 'count()' property is directly applied to edges and vertices which are direct properties of our graph. From this, it is possible to compute a **density** measure which corresponds to the number of actual edges divided by the total of possible edges. This measure has to be computed but the calculus is elementary so there is no difficulty here.

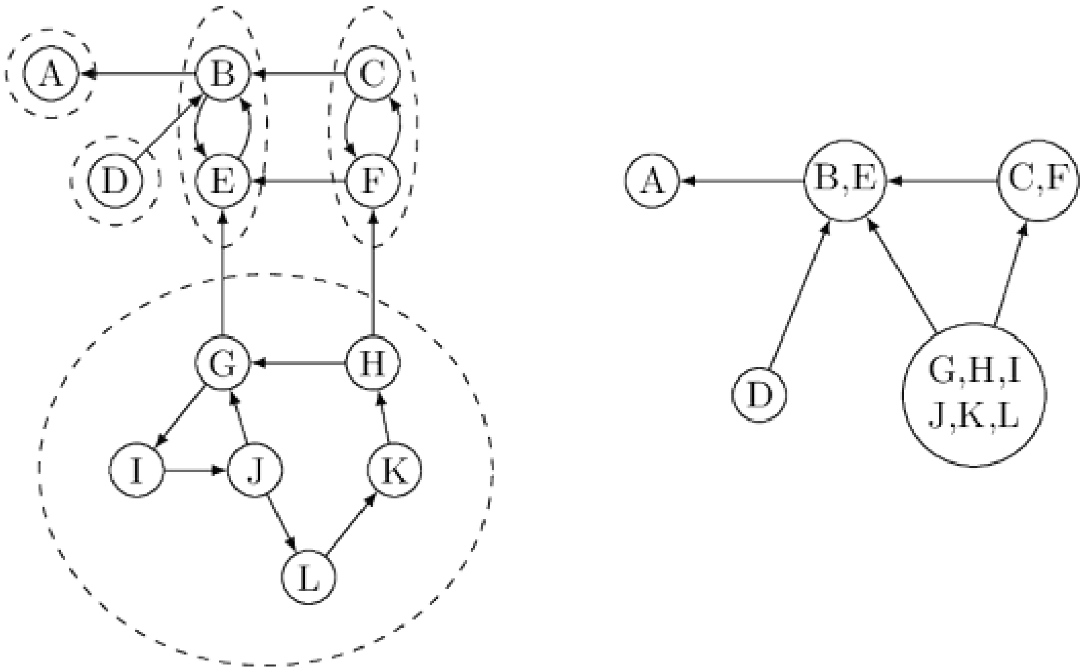
The second natural indicator is the **degree**. For each vertex, the degree is the number of edges between this one and other connected vertices. The distribution of the degrees will give information about the heterogeneity of the position of the different countries in one market. The 'degree' is an available property for the graph object, with the function 'stats()' we can compute some classic indicators like mean, standard deviation, max and min. We computed these four indicators for the degree distribution of each network, but also for the distributions of in-degrees (the number of incoming edges) and out-degrees (the number of outgoing edges) since our graphs are directed.

It is also pretty straightforward to analyse the **connexity** of the graph, which means counting the components of the graph. One graph is said to be connected if every pair of vertices is connected even indirectly. All of our graphs appear to be connected so we will not comment any further on this result.

More interesting, it is possible to study the strongly connected components. The search for the **strongly connected components** is the algorithm searching for subsets of graph vertices. There is a path between any two vertices of a specific subset (this time, direction is taken into account). At the same time, there is no path between the vertices of different subsets.

Figure 1

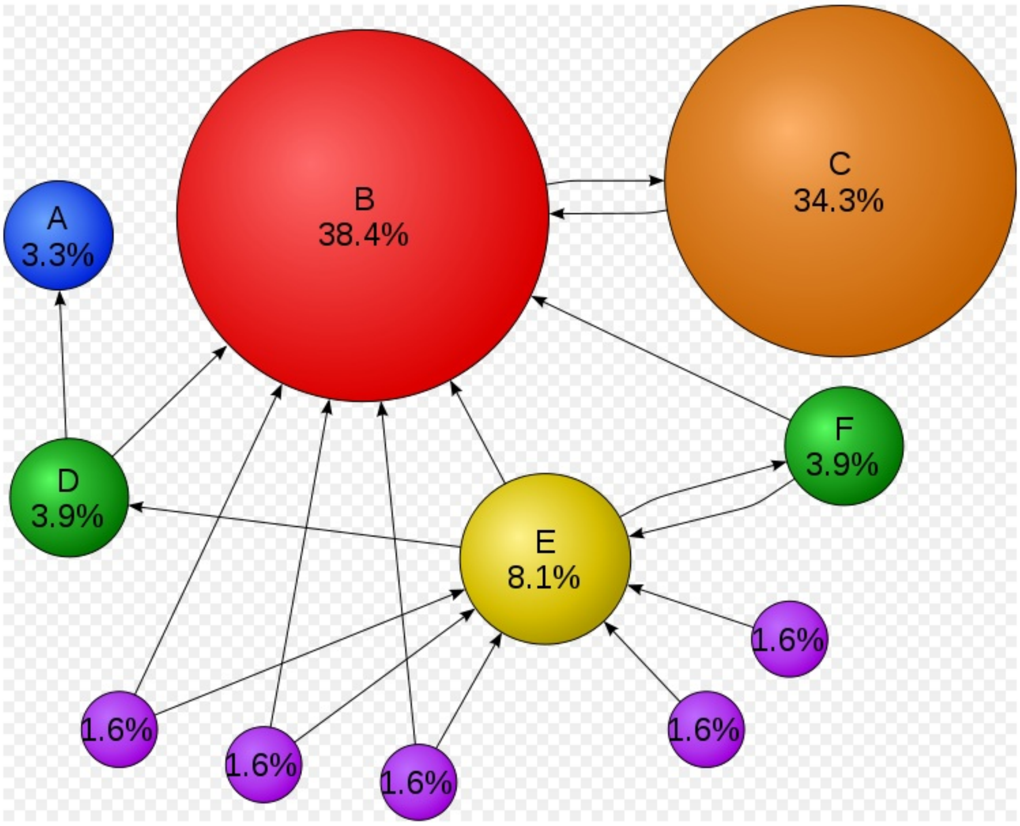
Example of the strongly connected components of a graph (source)



The last algorithm we applied is the **PageRank** algorithm. PageRank is a well-known algorithm for computing the “authority” of a vertex in a graph. It was offered by Google in 1998. For a long time, it has been used to rank the search results. The principle is to score every page proportionately to the number of times a user would cross the page clicking randomly through the web. A page has a high pagerank when the sum of the pageranks of the vertices pointing to it are big. This is a centrality measure.

Figure 2

Illustration of the PageRank algorithm (source: wikipedia)



The **transitivity measure** (the probability that two nodes connected to another one are connected together) is apparently available as a property from the graph object but we couldn’t make it work.

Results have been exported to HDFS and Hive have been used to build the tables from which the following figures have been plotted.

As a result of this analysis, we can attest a clear heterogeneity between the markets corresponding to the different BEC categories. Some include less agents than others and also less connections even with respect to the number of vertices (the density appears heterogeneous too). The impact of the financial crisis can also be noted.

Figure 2

Description of the networks of imports of intermediate goods between 2000 and 2013

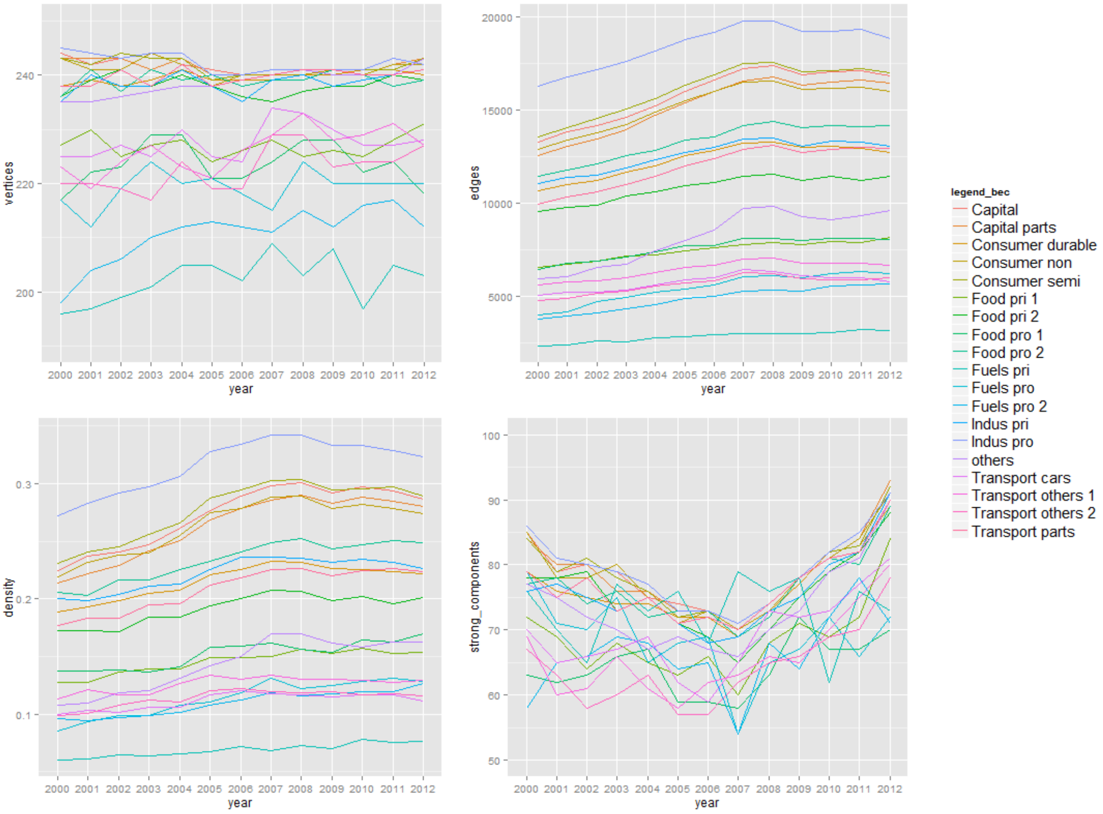
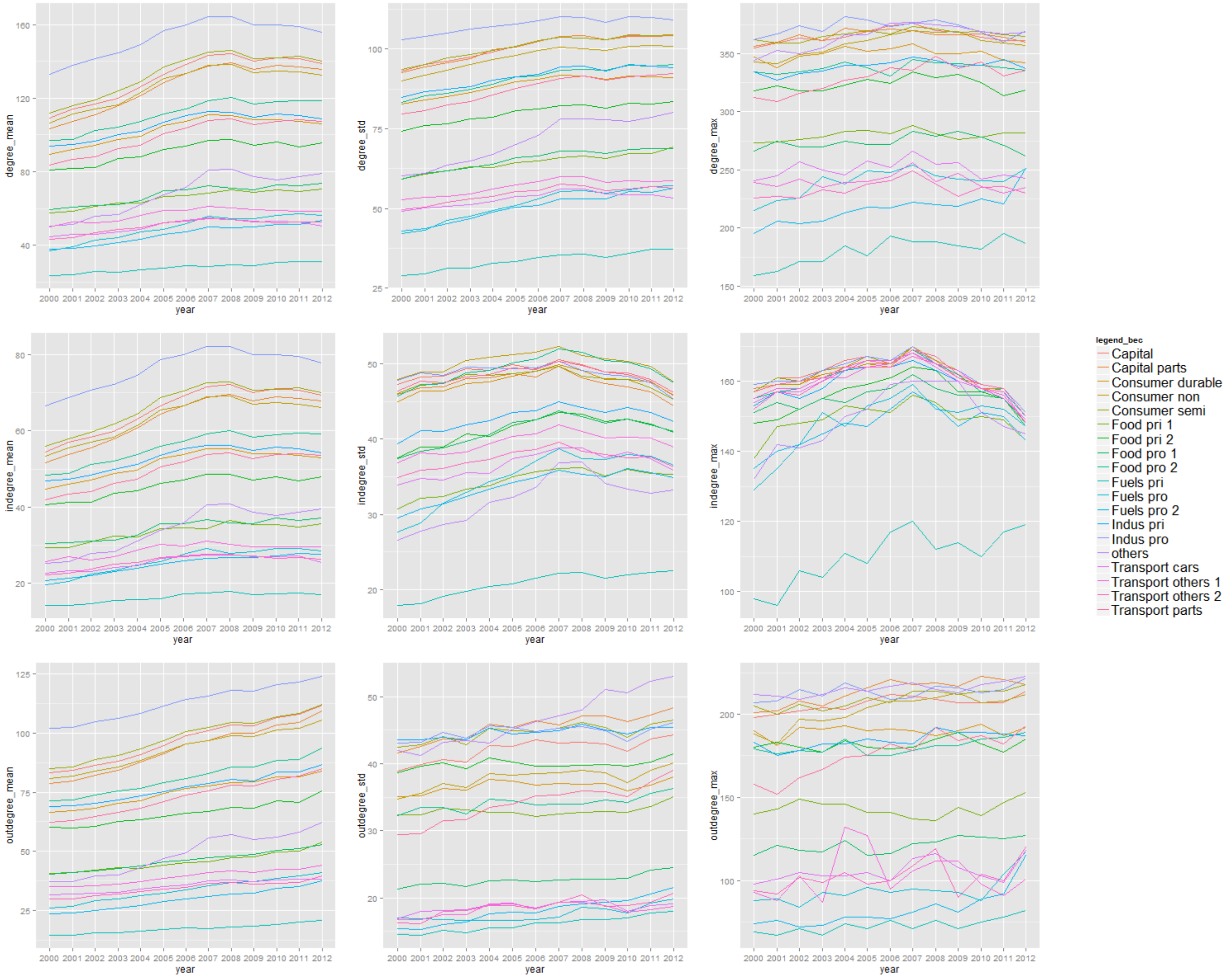


Figure 3

Distributions of the degrees of the networks between 2000 and 2013: mean, st dev, max for degrees (top panel), indegrees (medium panel), outdegrees (bottom panel)



As to the degree distribution, we can see for example that the average number of import partners decreased after the financial crisis whereas the average number of export partners kept increasing.

The pagerank algorithm supplies ranks for the vertices of one network, we kept the ten vertices with highest ranks for each networks for a given year and drew the histogram of appearances of these countries in the different rankings. This aims at spotting the countries that have an important position in several markets.

Figure 4

Histogram of appearances of the countries that were counted at least one time in the ten best ranks after application on the PageRank algorithm on each network for 2005.

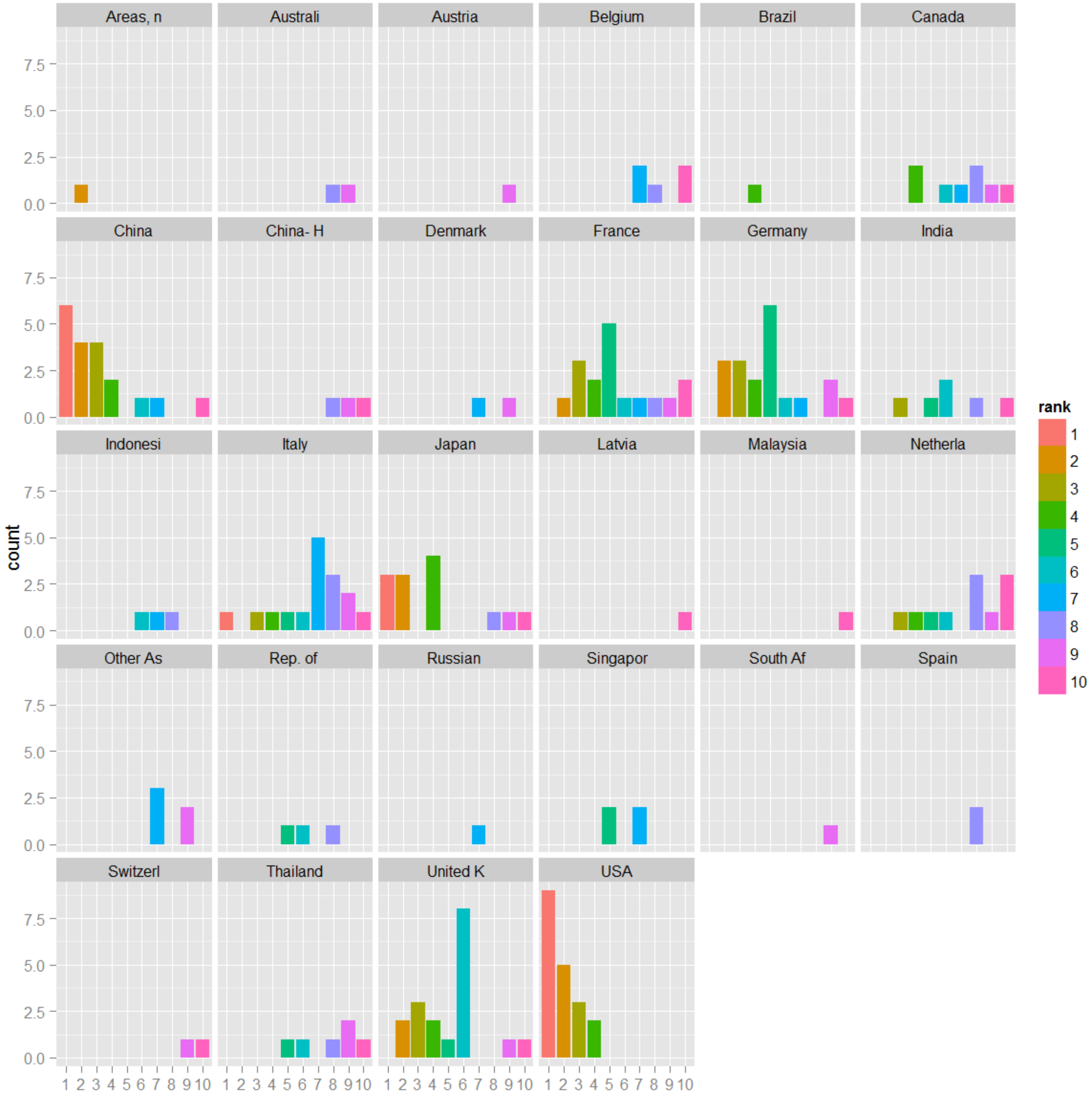
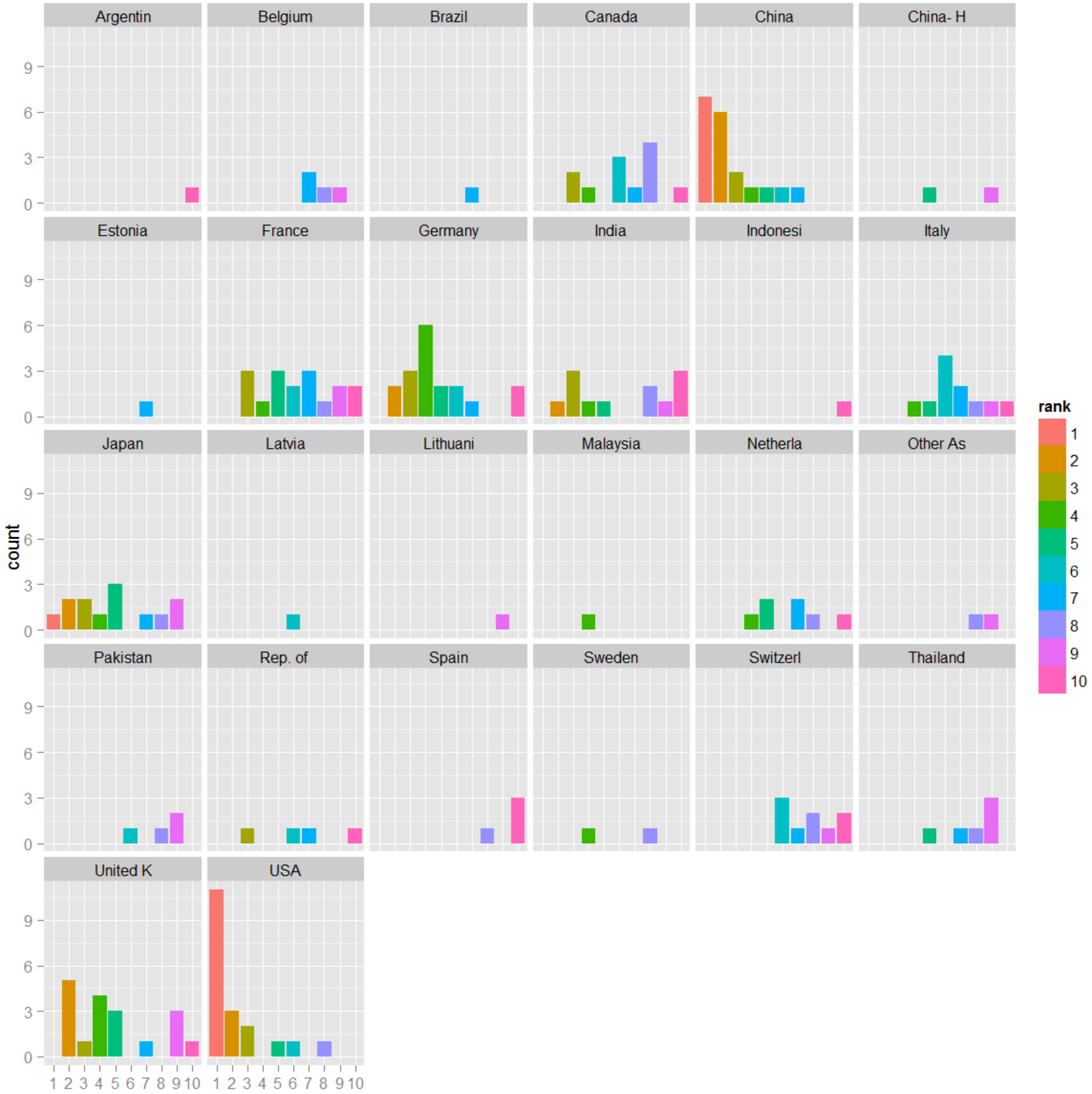


Figure 4

Histogram of appearances of the countries that were counted at least one time in the ten best ranks after application on the PageRank algorithm on each network for 2012.



We limited our study to the basic description of the graphs but some **future work** could be carried out on the following subjects :

- **Link prediction:** it may be interesting to be able to predict what edges will appear in our graph in the future (for example which new market would appear). For each pair of randomly selected vertices, we could try to predict the probability that there will be an edge connecting them, using the features and the description of vertices. For instance, one of the features can be an intersection of subsets of partners, or the Jaccard index (suggestion from http://kukuruku.co/hub/algorithms/social-network-analysis-spark-graphx).
- **Communities determination or ‘clusters’.** There are plenty of methods to solve this problem, from the simple selection of connected components (which we did), but we may want to gather vertices with similar properties or the closest to one another with respect to some measure (the Louvain algorithm mentioned before is a classic tool for this task).
- **Shortest Paths** in graphs: This problem is also classic and is implemented in various services that help to find the shortest path in a graph, between two countries here.

To conclude, the current implementation of SparkX contains **few implemented**

algorithms. Therefore, it is still relevant to use other tools. However, we can hope that GraphX will be improved in the future. As it provides the capability to cache data in memory, it could become very popular in solving graph problems.

Outputs

List of tables in Hive

annual_hs: base table with all fields for all years in HS classification, derived from the original Comtrade DB - pre-processed for acquisition in Hive

annual_hs_bec: only codes for countries and commodities (no descriptions), mapping of hs codes (filtered to level 6) to bec codes

yearrange_hs: imports and exports, aggregated on year ranges

yearrange_hs_withestimates: derived from yearrange_hs, added imputed values obtained from mirror statistics (imports only)

yearrange_bec: derived from yearrange_hs_withestimates, aggregating hs to bec (imports only)

yearrange_bec_exp: derived from yearrange_hs, aggregating hs to bec (exports only)

The codes below are used to impute the non-reported pair of country-period through the method of mirror statistics (data as reported by its trading partner).

Code model

```
--INPUT: yearrange_hs, total_imports (Nov 15 version)--OUTPUT:
yearrange_hs_withestimates (imports only)-- (STEP 1) to make it a table (STEP 1)
THIS IS TO GET MIRROR STATISTICS FOR MISSING COUNTRY-PERIOD PAIR, ONLY GET 6 digits
CREATE table mm_yearrange_hs_mirror as
select yearrange_hs.yearrange, yearrange_hs.partnercode as reportercode,
yearrange_hs.partner as reporter,
yearrange_hs.reportercode as partnercode, yearrange_hs.reporter as partner,
yearrange_hs.commoditycode, yearrange_hs.aggregatelevel, yearrange_hs.tradevalue,
b1.total_import
FROM yearrange_hs
JOIN
(
 SELECT total_imports_nov15.country_id, total_imports_nov15.yearrange,
total_imports_nov15.total_import from total_imports_nov15
 LEFT JOIN (
 Select reportercode, yearrange from yearrange_hs where tradeflowcode=1 group by
reportercode, yearrange
 ) b
 ON total_imports_nov15.country_id=b.reportercode and
total_imports_nov15.yearrange=b.yearrange
 WHERE b.reportercode is NULL
) b1
ON yearrange_hs.partnercode = b1.country_id and yearrange_hs.yearrange=b1.yearrange
WHERE tradeflowcode=2 and aggregatelevel=6
-- (STEP 2) THIS IS TO ESTIMATE DATA GAPS of individual country 6 digits (STEP 2)
CREATE table mm_yearrange_hs_mirror_adjusted as
SELECT yearrange, reportercode, reporter, partnercode, partner,commoditycode,
sum(adjtradevalue) as tradevalue, 'estimate' as flag FROM
(
SELECT
mm_yearrange_hs_mirror.yearrange, mm_yearrange_hs_mirror.reportercode,
mm_yearrange_hs_mirror.reporter,
mm_yearrange_hs_mirror.partnercode, mm_yearrange_hs_mirror.partner,
mm_yearrange_hs_mirror.commoditycode, mm_yearrange_hs_mirror.aggregatelevel
, mm_yearrange_hs_mirror.tradevalue*a.total_ratio as adjtradevalue FROM
mm_yearrange_hs_mirror
LEFT JOIN
(
select reportercode, yearrange, SUM(tradeValue) as sumTradeValue, MAX(total_import)
as total_import, MAX(total_import)/SUM(tradeValue) as total_ratio
FROM mm_yearrange_hs_mirror
GROUP BY reportercode, yearrange
) a
ON mm_yearrange_hs_mirror.reportercode=a.reportercode and
mm_yearrange_hs_mirror.yearrange=a.yearrange
) x
GROUP BY yearrange, reportercode, reporter, partnercode, partner, commoditycode
-- (STEP 3) ADD ESTIMATED DATA TO yearrange_hs
CREATE table yearrange_hs_withestimates as
SELECT yearrange, reportercode, reporter, partnercode, partner, commoditycode,
tradevalue, flag from mm_yearrange_hs_mirror_adjusted
UNION ALL
SELECT yearrange, reportercode, reporter, partnercode, partner, commoditycode,
tradevalue, 'reported' as flag from yearrange_hs WHERE tradeflowcode=1 and
aggregatelevel=6 and partnercode <> 0
```

Findings

The findings of the project are categorised in terms of Relevance (how useful is this project related to official statistics), Technology (how effective are the tools), Methodology (methods used to derive the results), Source (evaluation of data sources, including easiness to access and richness of the information) and Quality (overall quality of the output of this project).

Relevance

The measurement of Economic Globalization and International Trade is high on the agenda of UN Statistical Commission (see Decision 46/107 International trade and economic globalization statistics at the 46th session of the UN Statistical Commission). The comprehensive analysis of global value chains through trade networks in all economic sectors is crucial to better understand international trade. Therefore, the project is relevant to the global programme on International Trade Statistics.

Technology

Eight different tools/languages were used to work with the data. Starting from data in basic text format made it easy to switch from one tool to another. Processing data in the Sandbox provided evident advantages in terms of processing time and manageable size compared to the current tool used at UNSD (Relational DB) or other research institutes (splitting data into manageable chunks, and analysing them using traditional statistical software, such as Mathlab).

Methodology

The methodology used to prepare and analyse trade data has been developed over the years, and has been used by various organizations and research institutes. The sandbox environment enables comprehensive analysis of trade data (instead of chopping in to data chunks). However, due to time constraints, we have not yet implemented "automatic" detection of network clusters. This would certainly need new methodology, such as machine learning (this would be a continuity of this project).

Source

UN Comtrade data is available publicly, regularly updated, and now supports the bulk API. With this feature, data can be regularly synced/updated through script in PIG/PYTHON.

Quality

The analysis of quality is broken down by quality dimensions:

- **Timeliness**: Even though UN Comtrade is regularly updated, the frequency of the data sets is annual or monthly (with some time lag), and not comparable to "real-time" big data feeds such as from social media

- **Coverage**: UN Comtrade covers most of the countries with some delays (recent periods have fewer reporters than past periods). Therefore, there is need to estimate data gaps (using extrapolation/regression and/or mirror statistics)
- **Accessibility**: All data are available publicly, and a bulk data API is offered
- **Accuracy**: Data received from countries have undergone data cleansing, standardization, aggregation, and verification before being published in UN Comtrade. Nevertheless, there are possible methodological quality issues in underlying data (such as territorial and commodity coverage, suppression of data due to confidentiality, deviation to international standards, etc.). All of these contribute to the comparability issues across countries
- **Accuracy in conversion of commodity classification**: The purpose of the study is to analyse imports of intermediate goods. The original data reported by the country is normally according to the Harmonized System (such as potato, or car), and there is no indication of the end-use. We used generic HS to intermediate goods conversion tables that may not be accurate in specific countries. For an example, a potato can be seen as final goods (if it is consumed by people) or an intermediate goods (if it is used to make potato chips)
- **Relevance**: The analysis of global value chain is very much in demand, especially in the measurement of economic globalization and international trade
- **Methodological soundness**: We do not attempt to invent new methodology to analyse trade networks (we use an established network theory). However, we would like to explore the use of machine learning technology to detect and analyse clusters of trade networks in near future.

Conclusions

In brief, we established three substantive goals:

1) Establish the network of imports of intermediate goods

2) Analyse and visualize trade networks of existing group of countries

3) Automatic detection of trade clusters

and one technical goals:

1) Exploration on the use of big data tools and technologies in analyzing of trade data.

Substantive goals 1 and 2 were achieved, but due to time constraints we were not able to implement goal 3. In addition, we are pleased to report that it is possible to use big data tools and technologies (Hadoop, Rhadoop, Pig, Hive, Spark and Gethpi) to process and analyse huge volume of trade data. The easiness of setting up the data environment, powerful computing power and availability of built-in libraries to analyse networks may change the way trade analysts work, and how UN Comtrade will offer services to users in the future (in addition to purely data services, UN Comtrade might also offer an analytical platform through the use of Hadoop, Pig, Hive, Spark, and other big data technologies). We would like to continue working with Comtrade data in the sandbox to achieve goal no.3, the use of big data tools to automatically detect of trade clusters, and refine the procedures to acquire raw data (now through API) and to to streamline data cleansing activities. In addition, other groups may be interested to use Comtrade@Sandbox for other purposes, not only analyzing trade networks (i.e., analysis of unit values).

How far are we from something that can be published?The objective was not to produce statistics. However analysis such as those produced in this experiment are normally published by international research centers and institutions.

What case has been made for the future of the Sandbox?We showed that it is possible to use big data tools and technologies in processing and analyzing large volumes of trade data. The easiness of setting up the data environment, powerful computing power and availability of built-in libraries to analyse networks may change the way trade analysts work.

Task Team members

- Michael Behrman (un)
- Stéphanie Combes (fr)
- Markie Muryawan (un)

- Pilar Rey-del-Castillo (es)

- Toni Virgillito (it)

No labels

Report inappropriate content

{"serverDuration": 559, "requestCorrelationId": "e8b13f97f2e5a0"}