Harmadik házi feladat

Hálózatelemzés kurzus

Készítette: Pipis Bence

Heller Farkas Szakkollégium

2019/20 ősz

1. Feladat

Az első feladatban felhasznált temporális hálózatom forrása a Stanford University adatbázisa. A hálózat a University of California egyetem felhasználói által egymásnak küldött üzeneteket tartalmazza, irányított gráfként. A teljes időintervallum 193 nap. A gráf összesen 1899 csúcsot tartalmaz, és 59835 élet.

Időrend szerinti sorba rendezés után létrehoztam az első Ne/2 élből álló statikus hálózatot, majd meghatároztam a kért mutatószámokat. A link prediction módszer lényege, hogy különböző hasonlósági mutatók alapján megpróbáljuk megbecsülni, hogy a két csúcs között mekkora eséllyel lesz él a későbbiekben.

A common neighbors mutatót egyszerűen, az adjecency mátrix négyzetre emelésével megkaptam minden csúcspárra, majd ezt a mátrixot súlyozott adjecency mátrixként értelmezve létrehoztam belőle egy segédgráfot. Ebből kinyerve az edgelistet, majd csökkenő sorrendbe rendezve megkaptam az első Ne/2 legnagyobb valószínűségű prediktált élet.

Az Adamic-Adar hasonlósági mutató meghatározásához segítségül vettem egy igraph beépített függvényt, mellyel lehetőség van meghatározni a bemenő és kimenő fokszámokra is a mutatót. Meghatároztam mind a kimenő, mind a bemenő Adamic-Adar hasonlósági mutatót, majd ezeket összeadtam minden csúcspárra. Az így kapott mátrixból a fent már leírt módon kaptam meg a sorba rendezett edgelistet. Egyetlen különbség az volt, hogy az AA mutató számításakor a főátlóba is kerültek értékek, így a segédmátrix létrehozásakor ezt nem vettem figyelembe, mivel az önmagába mutató éleknek, elküldött üzeneteket tekintve nem sok értelme van.

A local path index meghatározásához is elég volt az adjecency mátrixot különböző hatványokra emelni. A következő egyenletben a bétát 0.1-re állítottam, hogy nagyobb súlyt kapjanak a rövidebb utak. (Aziz et al, 2020.)

$$S^{LP} = A^2 + \beta * A^3$$

A mutatót tartalmazó mátrix kiszámítása után a fentiekkel analóg módon kaptam meg a sorbarendezett edgelistet.

Összehasonlítva a három módszer predikciós képességét megvizsgáltam, hogy a prediktált élek közül az első Ne/2 közül mennyi szerepel ténylegesen a második Ne/2 élből létrehozott eredeti hálózatban, majd ezt az értéket elosztottam Ne/2-vel, hogy megkapjam, mekkora hányada jelent

meg az első Ne/2 prediktált élnek a valós gráfban. Az így kapott mutatókat az alábbi táblázat tartalmazza.

method	fraction
Common neighbors	0.02172678
Adamic-Adar	0.02222816
Local path index	0.04442290

Látható, hogy a local path index magasan a legjobban teljesít, melyet az indokol, hogy ez nem egy lokális, hanem egy kvázi-lokális mutató. Az Adamic-Adar nagyon kevéssel, de felülmúlja ugyan a common neighbors mutatót, de közel azonos szinten teljesítenek.

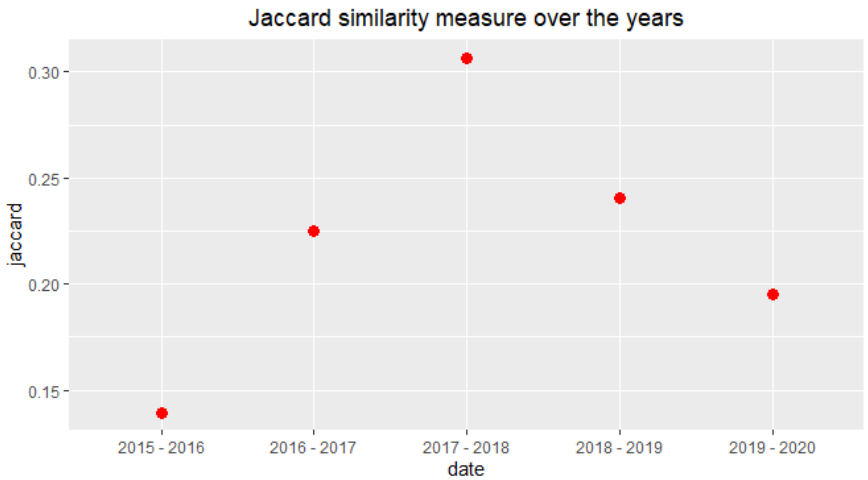
2. Feladat

A feladatban az S&P 500 első 50 részvényét töltöttem le 2015 és 2020 között. A sorrendet a 2020 novemberi piaci kapitalizáció alapján állítottam fel. A napi loghozamok kiszámítása után kiszámoltam az éves korrelációt a hozamok között, majd ebből megalkottam a d=1-corr mátrixot. Ezt a távolságmátrixot súlyozott adjecency mátrixként értelmezve létrehoztam minden évre a teljes, irányítatlan gráfot, hiszen itt minden csúcspár között számoltunk korrelációt. Ebből ezt követően kiszámoltam a minimális feszítőfát a beépített *mst()* paranccsal, mely képes különböző algoritmusok szerint meghatározni a fát. Itt a *prim* algoritmust választottam, mely a távolságokat minimalizálja. Ez nekünk éppen megfelelő, hiszen a távolságmátrixban ezek a távolságok szerepelnek, melyeket súlyként hozzáadtunk az élekhez.

Miután elkészültek a minimális feszítőfák, az egymást követő évekre összehasonlítottam őket a Jaccard hasonlósági mutató szerint, melyet a következő képlettel számoltam ki.

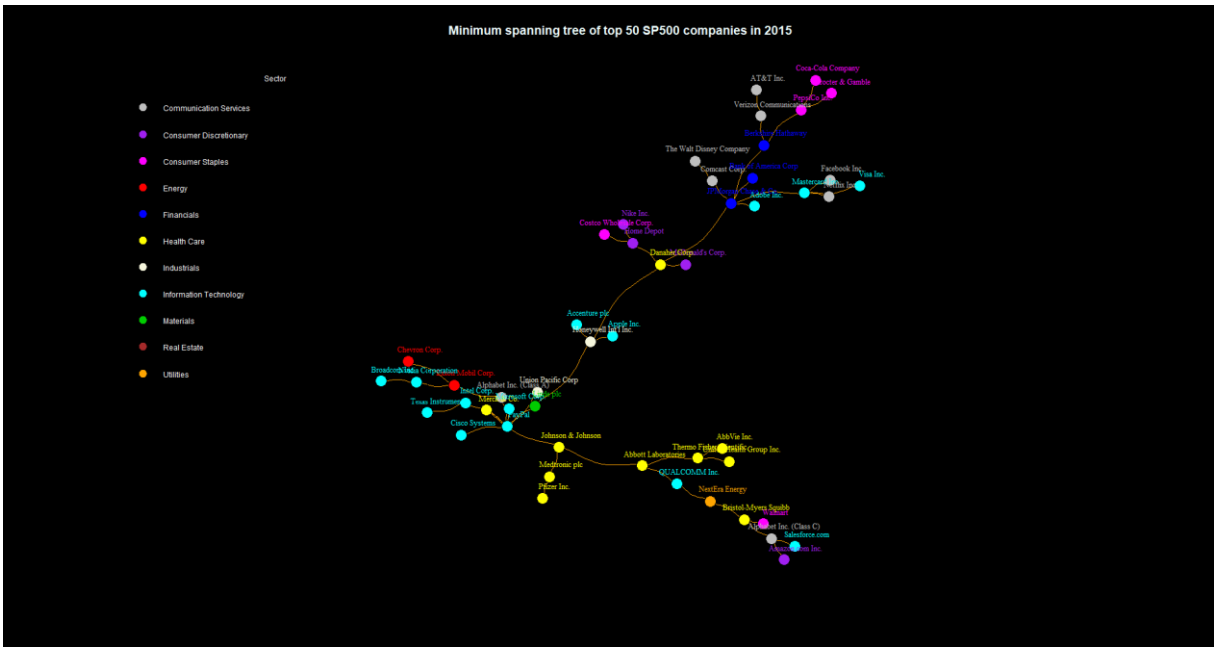
$$S^J = \frac{E_i \cup E_j}{E_i \cap E_j},$$

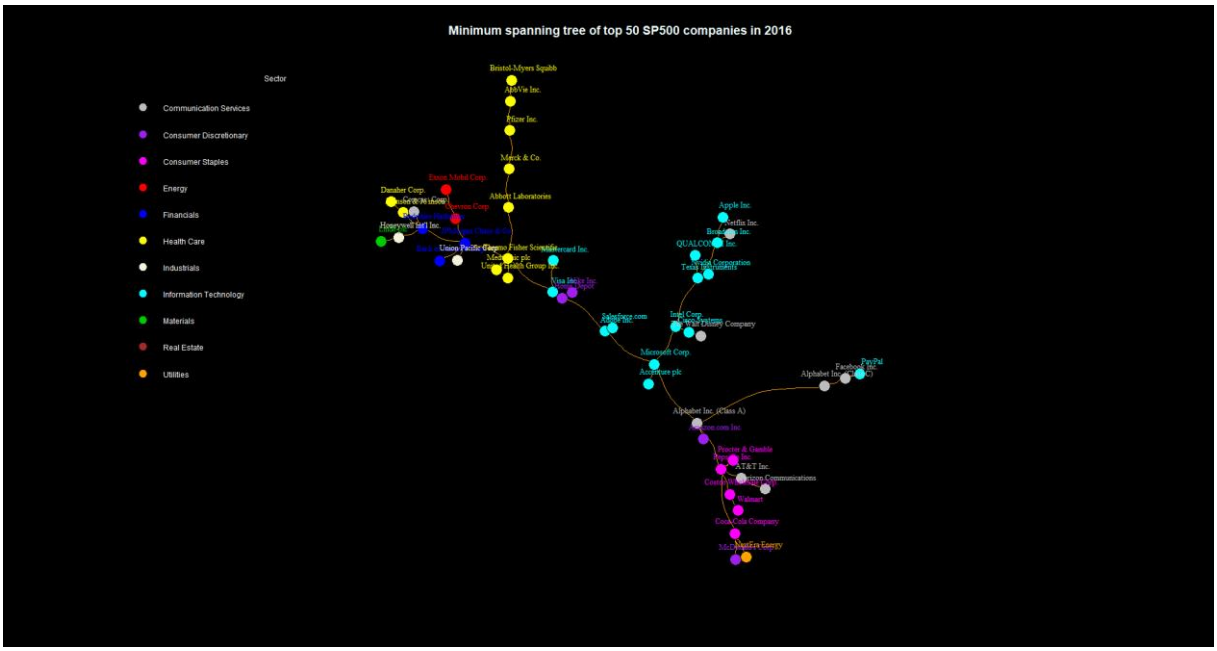
ahol Ei és Ej az i és j hálózatban (itt egymást követő évek feszítőfáiban) szereplő éleket jelölik. A Jaccard hasonlósági mutatókat ábrázolva a következő diagramot kapjuk.

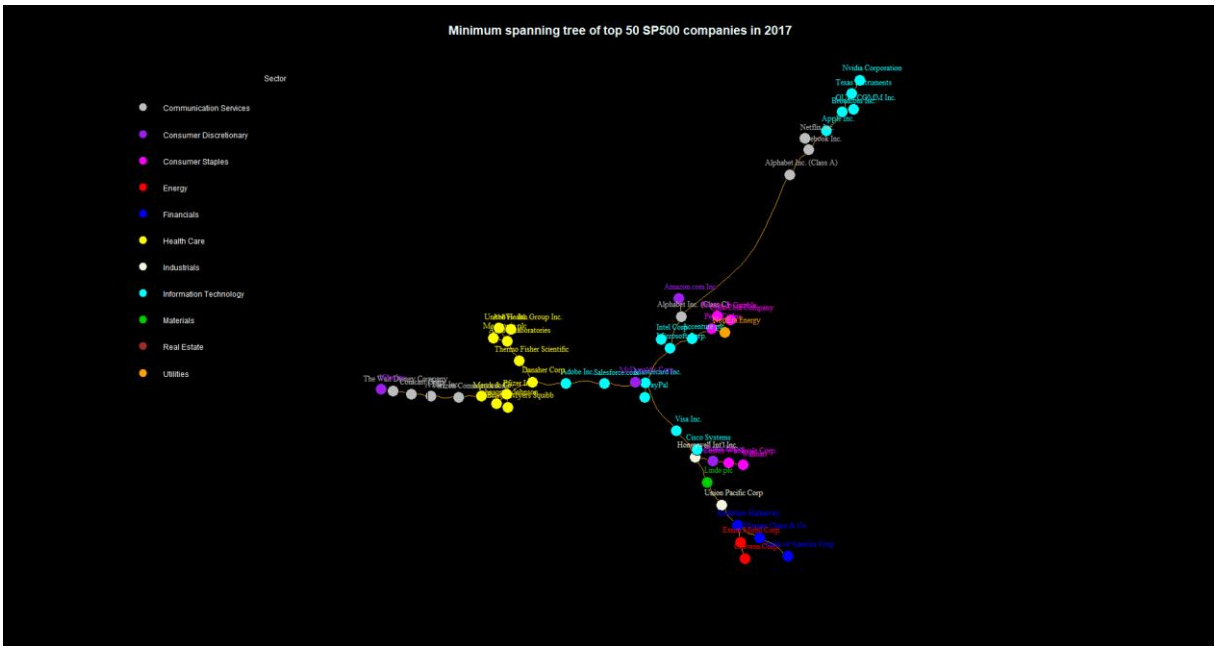


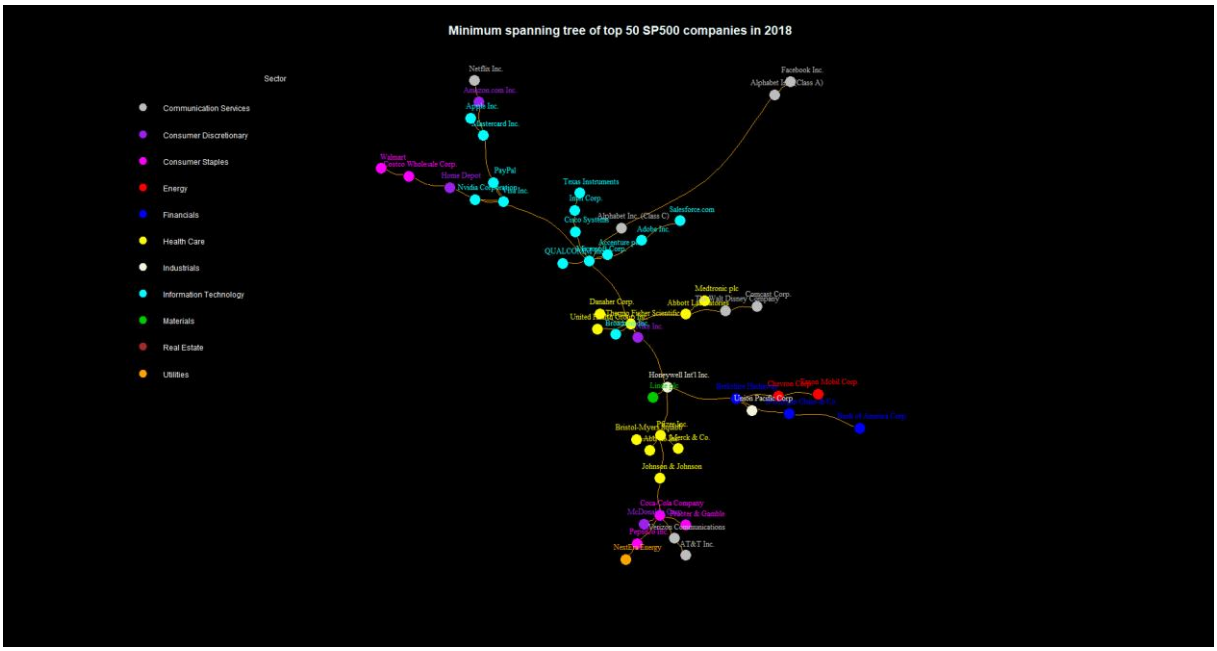
Az ábrát értelmezve azt láthatjuk, hogy az S&P 500 top 50 részvényének hozamaiból képzett minimális feszítőfa 2017 és 2018 között mutatott a legnagyobb hasonlóságot, míg a legnagyobb változás 2015 és 2016 között ment végbe. Közgazdasági értelemben azt mondhatjuk, hogy a részvények hozamai közötti páronkénti korreláció is akkor maradt hasonló az évek során, ha a Jaccard hasonlósági érték magas. Akkor alakult át jobban a piac, amikor a Jaccard mutató értéke alacsony volt (például 2015-2016 között), hiszen a feszítőfa nagymértékű megváltozása azt jelenti, hogy mások lettek a legjobban korrelált részvények. Tehát ezt a mutatót felfoghatjuk egyfajta piaci változást mutató értéknek.

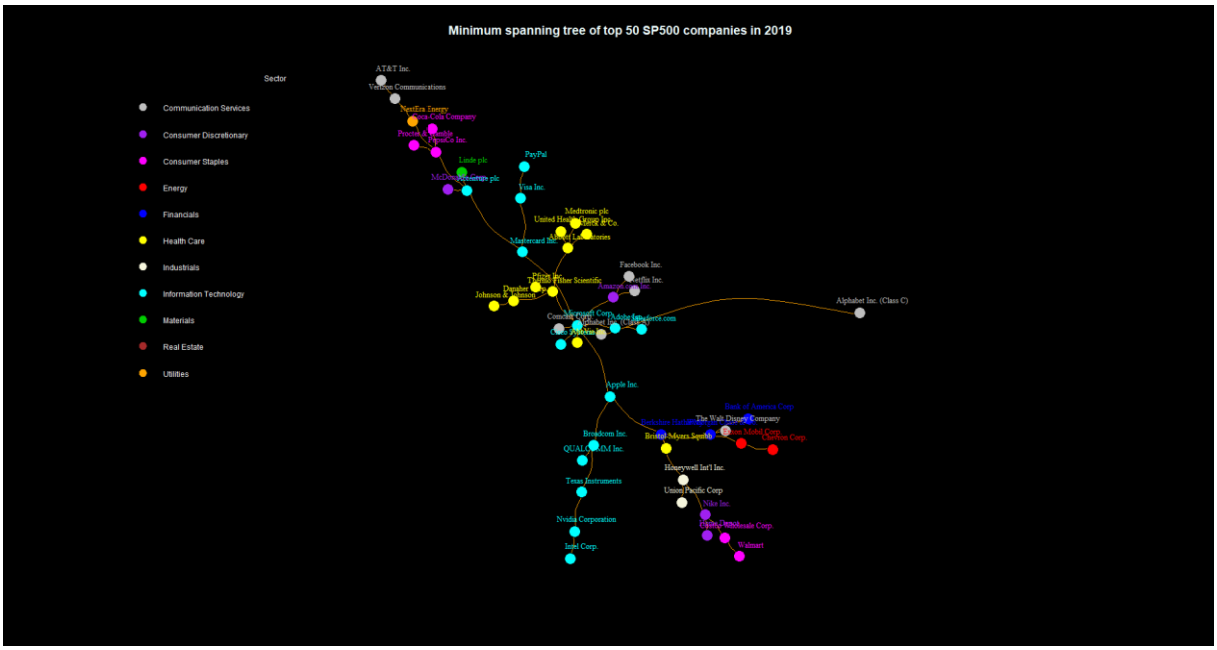
Feszítőfák

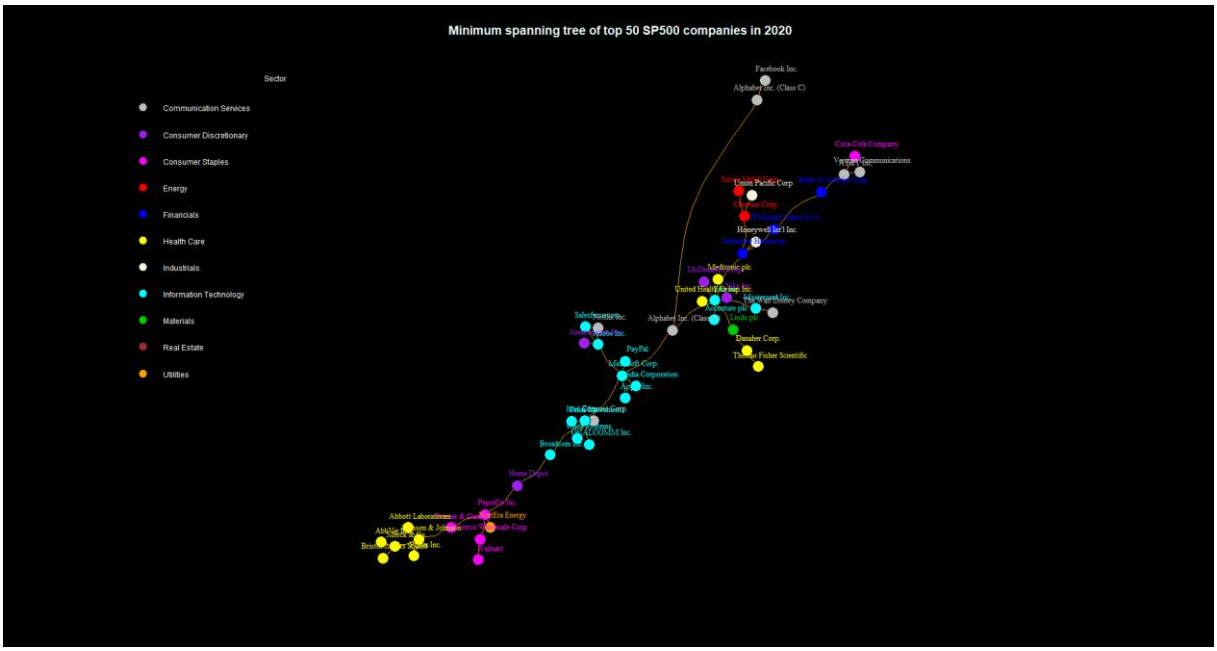





Minimum spanning tree of top 50 SP500 companies in 2017
Sector
Communication Services
Consumer Discretionary
Consumer Staples
Energy
Financials
Health Care
Industrials
Information Technology
Materials
Real Estate
Utilities
Nvidia Corporation
Texas Instruments
Apple Inc.
Alphabet Inc. (Class A)
Amazon.com Inc.
Alphabet Inc. (Class C)
Thermo Fisher Scientific
Danaher Corp.
PayPal
Visa Inc.
Union Pacific Corp


Minimum spanning tree of top 50 SP500 companies in 2018
Sector
Communication Services
Consumer Discretionary
Consumer Staples
Energy
Financials
Health Care
Industrials
Information Technology
Materials
Real Estate
Utilities
Netflix Inc.
Facebook Inc.
Amazon.com Inc.
Apple Inc.
Mastercard Inc.
Walmart
PayPal
Texas Instruments
Salesforce.com
Alphabet Inc. (Class C)
Medtronic plc
Comcast Corp.
Danaher Corp.
Honeywell Int'l Inc.
Union Pacific Corp
Johnson & Johnson
Coca-Cola Company
Verizon Communications
AT&T Inc.



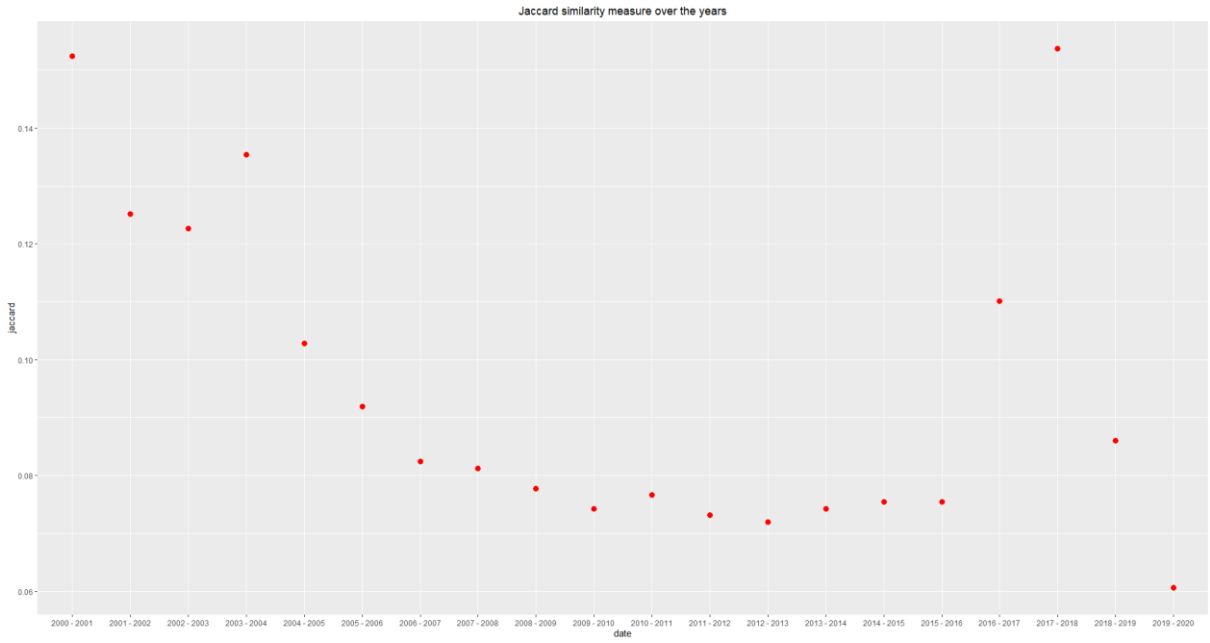


A feszítőfákról megfigyelhető, hogy bár volt változás az évek során, de az azonos színek nagyrészt egy helyen csoportosulnak, és van köztük él. Ebből azt a következtetést lehet levonni, hogy az azonos szektorban szereplő vállalatok hozamai között az elmúlt 6 évben nagy volt a korreláció.

3. Kettes feladat továbbgondolása

Ha a kettes feladatban megfogalmazott vonalon továbbindulunk, érdemes lehet megvizsgálni a teljes S&P 500 részvényeinek hozamkorrelációját, hiszen egyes statisztikák szerint ez az amerikai piac 70%-át leírja. Nézzük meg ezt egy kiterjesztett időszakra, 2000 és 2020 között, majd ábrázoljuk a Jaccard hasonlósági mutatót az évek során. Itt természetesen torzítást okoz, hogy a 2020-as részvényeket használjuk fel, pedig az S&P 500 összetétele időben változik. Ez további kutatás lehet, hogy minden évben az aktuális komponenseket vesszük figyelembe.

Elvégezve ezt az elemzést a következő ábrát kapjuk.



Az ábráról az látható, hogy a 2000-es évek elején viszonylag hasonló maradt a piac évről évre, majd ez elkezdett lecsökkenni. Egyre többet változott évről évre a minimális feszítőfa az amerikai részvénypiacon az S&P 500 részvényeket tekintve, azaz átalakult, hogy mely részvények mivel korrelálnak. Ez valamilyen szinten a piac instabilitásának tekinthető. Ha megnézzük az ábrát, 2008-2009 környékén volt a legalacsonyabb a Jaccard mutató értéke, azaz itt változott legjobban a piac. Ekkor volt a nagy gazdasági válság, mely megmagyarázhatja a piac instabilitását. Érdemes megfigyelni az utolsó értéket az ábrán, mely az idei év változását mutatja a tavalyi évhez képest. Itt is nagyban megváltozott a feszítőfa, más részvények lettek erősen korreláltak, mely változás mögött a koronavírus okozta sokk sejthető. Természetesen ezek nagyon felületes következtetések, és ezek részletesebb vizsgálata lenne szükséges az állítások teljes mértékű alátámasztásához.

4. Források

Aziz, F., Gul, H., Uddin, I. et al. Path-based extensions of local link prediction methods for complex networks. Sci Rep 10, 19848 (2020). https://doi.org/10.1038/s41598-020-76860-2 letöltve: 2020.12.01.

Laishram, Ricky, "Link Prediction in Dynamic Weighted and Directed Social Network using Supervised Learning" (2015). Dissertations - ALL. 355. https://surface.syr.edu/etd/355 letöltve: 2020.11.28.

https://snap.stanford.edu/data/CollegeMsg.html

5. Kódok

```
library(quantmod)
library(dplyr)
library(igraph)
library(ggplot2)
library(matrixcalc)

#https://rstudio-pubs-
static.s3.amazonaws.com/362044_903076131972463e8fdfcc00885fc9a6.html

setwd("C:/Users/csirk/OneDrive/Documents/Kurzusok,k?dok/H?l?zatelemz?s")

#############
##1. feladat##
#############
msg<-read.table(gzfile("CollegeMsg.txt.gz"))
colnames(msg)<-c("source","target","sec_since_epoch")
msg$day<-(msg$sec_since_epoch-min(msg$sec_since_epoch))%/%(60*60*24)+1
#making sure data is in increasing order in time
msg<-msg[order(msg$day,decreasing=F),]
#creating graph from all edges and counting all edges

total<-graph.data.frame(msg,directed=T)
Ne<-gsize(total)
Ve<-gorder(total)
#creating graph from first Ne/2 edges in edgelist
static<-graph.data.frame(msg[1:(Ne%/%2+1),],directed=T)
#performing link prediction methods

#common neighbors
adj<-as.matrix(as_adjacency_matrix(static))
colnames(adj)[1:5]
adj2<-matrix.power(adj,2)

CN<-get.data.frame(graph.adjacency(adj2,weighted = T))
```

```
CN<-CN[order(CN$weight,decreasing = T),]
head(CN)
nrow(CN)

#Adamic-Adar
AAinmatrix<-similarity(static,method = "invlogweighted",mode="in",loops=F)
colnames(AAinmatrix)<-colnames(adj)
rownames(AAinmatrix)<-rownames(adj)

AAoutmatrix<-similarity(static,method = "invlogweighted",mode="out",loops=F)
colnames(AAoutmatrix)<-colnames(adj)
rownames(AAoutmatrix)<-rownames(adj)

AAmatrix<-AAinmatrix+AAoutmatrix

AA<-get.data.frame(graph.adjacency(AAmatrix,weighted = T,diag=F))
AA<-AA[order(AA$weight,decreasing = T),]

#Local path index with parameter ß=0.1 ir order to give higher weight for shorter paths. source:
https://www.nature.com/articles/s41598-020-76860-2
beta<-0.1
LPmatrix<-adj2+beta*matrix.power(adj,3)
LP<-get.data.frame(graph.adjacency(LPmatrix,weighted = T))
LP<-LP[order(LP$weight,decreasing = T),]
head(LP)

#evaluating link prediction methods
#since i created the static network from Ne%/%2+1 edges to make sure Ne/2 edges appear, now
I am going to check for the residuum, Ne-(Ne%/%2+1)
Nehalf<-Ne-(Ne%/%2+1)
#creating vector for all measures
fraction<-c()
#creating benchmark graph from original graph
BM<-graph.data.frame(msg[(Nehalf+2):Ne,],directed=T)

#common neighbors
fraction[1]<-gsize(intersection(graph.data.frame(CN[1:Nehalf,1:2],directed=T),BM))/Nehalf

#Adamic-Adar
fraction[2]<-gsize(intersection(graph.data.frame(AA[1:Nehalf,1:2],directed=T),BM))/Nehalf

#Local path index
fraction[3]<-gsize(intersection(graph.data.frame(LP[1:Nehalf,1:2],directed=T),BM))/Nehalf

result<-data.frame(method=c("Common neighbors","Adamic-Adar", "Local path
index"),fraction=fraction)

############
##2. feladat##
############
```

```
date_start<-"2015-01-01"
date_end<-"2020-11-01"

components<-as.data.frame(read.csv("constituents.csv",sep=";"))
samp_size<-50
samp<-sample_n(components[,1:3],samp_size)
top<-components[1:samp_size,1:3]
data<-c()

#downloading necessary data
data<-as.data.frame(do.call(cbind,sapply(seq(1,samp_size), function(x)
getSymbols(as.character(top$Symbol[x]), src="yahoo", from=date_start, to=date_end,
auto.assign = FALSE )[,6])))

#500 stock calculation block
#data_full<-data #for total sp500 calculation. once downloaded and saved
#top_full<-top #for total sp500 calculation. once downloaded and saved
#data <-data_full
#top<-top_full
#firstdates<-c("2000-01-01","2001-01-01","2002-01-01", "2003-01-01", "2004-01-01","2005-
01-01","2006-01-01","2007-01-01","2008-01-01","2009-01-01","2010-01-01","2011-01-
01","2012-01-01", "2013-01-01", "2014-01-01","2015-01-01","2016-01-01","2017-01-
01","2018-01-01","2019-01-01", "2020-01-01", "2021-01-01")
####

ncol(data)
#return calculation
returns<-as.data.frame(sapply(data, function(x) diff(log(x))))

colnames(returns)<-top$Name
rownames(returns)<-as.Date(rownames(data[2:nrow(data),]))

firstdates<-c("2015-01-01","2016-01-01","2017-01-01","2018-01-01","2019-01-01", "2020-
01-01", "2021-01-01")
#creating palette for visualization
cc<-
c("orange","red","green3","blue","cyan","magenta","yellow","gray","brown","purple","beige"
)
color<-data.frame(factor=levels(top$Sector),color=as.character(cc))
#graph creation
mg<-list()
tree<-list()
palette(cc)

for (i in (1:(length(firstdates)-1))){
  #calculating correlation and distance
  correlation<-
cor(returns[rownames(returns)>firstdates[i]&rownames(returns)<firstdates[i+1],])
  distance<-1-correlation
```

```
 #creating graph from distance matrix ir order to obtain edgelist
 g  <- graph.adjacency(distance,weighted=TRUE,diag=F,mode="upper")
 #getting edgelist from graph
 el <- get.data.frame(g)
 #creating increasing order in distance.
 el<-el[order(el$weight,decreasing = F),]
 #creating new graph
 mg[[i]]<-graph.data.frame(el,directed=F)
 #creating attribute to nodes in order to draw them in differenct colors
 industry<-sapply(seq(1,50),function(x) top[top$Name==V(mg[[i]])$name[x],3])
 mg[[i]]<-set_vertex_attr(mg[[i]],name="sector",value=industry)
 V(mg[[i]])$color<-sapply(1:gorder(mg[[i]]), function(x) color[color$factor==industry[x],2])
 #calculating minimum spanning tree
 tree[[i]]<-mst(mg[[i]],algorithm = "prim")

}

#plotting minimum spanning trees
par(bg="black")
for (i in (1:(length(firstdates)-1))){
 plot(tree[[i]],
      #general
      palette=cc,
      layout=layout_with_fr,
      #vertex attribuites
      vertex.size=5,
      vertex.label.font=1,
      vertex.label.color=V(mg[[i]])$color,
      vertex.label.dist=1,
      vertex.label.cex=0.75,

      #edge attributes
      edge.color="orange",
      edge.curved=0.3)
 title(paste("Minimum spanning tree of top",samp_size, "SP500 companies
in",unlist(strsplit(firstdates[i],"-"))[1]),col.main="azure",font.main=2)
 legend("topleft",
legend=color$factor,pch=21,pt.bg=color$color,text.col="white",cex=0.75,bg="transparent",pt
.cex=2,bty="o",title='Sector')
}

#comparing minimum spanning trees by calculating Jaccard measure
jaccard<-c()
jaccard$date<-sapply(seq(1,(length(firstdates)-2)),function(x)
paste(unlist(strsplit(firstdates[x],"-"))[1],"-",unlist(strsplit(firstdates[x+1],"-"))[1]))
jaccard$jaccard<-sapply(seq(1,(length(firstdates)-2)), function(x)
gsize(intersection(tree[[x]],tree[[x+1]]))/gsize(union(tree[[x]],tree[[x+1]])))
jaccard<-as.data.frame(jaccard)
#the higher the jaccard measure the higher the similarity
```

```
#plotting jaccard measure over the years
dev.off()
gg<-
ggplot(data=jaccard,aes(x=date))+geom_point(aes(y=jaccard),col="red",size=3)+ggtitle("Jacc
ard similarity measure over the years")+theme(plot.title = element_text(hjust = 0.5))
#gg_full<-gg #once calculated and saved
gg_full
gg2<-
ggplot(data=jaccard,aes(x=seq(1:20)))+geom_line(aes(y=jaccard),col="red",size=2)+ggtitle("
Jaccard similarity measure over the years")+theme(plot.title = element_text(hjust = 0.5))
gg2
#gg2_full<-gg2 #once calculated and saved
```