Data Sampling by applying vsp on Journal Citation Network

Sijia Fang

11/5/2020

Introduction

When people want to learn a new technique, or understand a new field, searching is the most common way to do it. However, this process can be extremely irritating. There could be tremendous amounts of paper related to that techinque, describing how to apply or improve it under centain cases. It's hard to understand these papers before a basic knowledge of the histroy about this techinque.

This project aims to provide a way to analyse the developing process of certain statistical techinques. When a key words such as **lasso** is provided, a summary for it will be produced, including how it was proposed in the first place and when it became widely applied in different field.

To attain this goal, we study **Semantic Scholar Data**. This is a large data set containing 220 million papers from all fields. To study the developing process of statistical techinques, the first step is sampling: we need to focus on mainstream academic journals that develope statistical algorithms or apply them.

This blog post illustrates how to acheive the sampling process via **vsp** algorithm.

Data

(adj matrix A_{ij}=1 if i cite j)

Semantic scholar dataset provides directed network $G_{paper} = (V_{paper}, E_{paper})$ with every node i in V_{paper} represents a paper and $(i, j) \in E_{paper}$ if paper i cite paper j. G_p is extremely large and thus studying it directly is impractical. Luckily, G_p is also sparse and can be aggregated as a weighted directed network $G_{journal} = (V_{journal}, E_{journal})$ with every node i in $V_{journal}$ represents a journal. $(i, j) \in E_{journal}$ if there exist paper in journal i that cites paper in journal j, and (i, j) is weighted by the number of citation from journal i to journal j. $G_{journal}$ is still sparse but with moderate size, it contains approximately 100 thousand nodes. Notice that G_{paper} contains 220 million nodes.

Our data is the adjacency matrix from $G_{journal}$. Our goal is to select top journals in statistics and other fields that frequently apply statistical tools.

Method

vsp is designed to find underlying structures, it fits in many scenarios and here we focus on it's application in Stochastic co-Blockmodel. As $A \approx ZBY^t$, Y and Z record two types of block membership, B_{ij} represents how likely an edge from row-block i to column-block j will exist. Regrading the citation adjacency matrix, Z gives a partition of journals based on how they cite others while Y gives a partition of journals based on how they are cited by others. We will call them citing clusters and cited clusters in the following analysis. B matrix reveals the citation pattern among groups.

Result

1. load the data as a sparse matrix and do some pre-processing.

```
library(Matrix)
library(vsp)
library(dplyr)
library(plotrix)
library(tidyverse)
library(tidytext)
library(tm)

# read data
edge <- read.csv("journalEdgeList.csv")
name <- read.csv("journalNames4EdgeList.csv")
name$y <- name$x

# create sparse adjancy matrix
Adj <- sparseMatrix(i = edge$from, j = edge$to, x = edge$weight,
                    dims = rep(dim(name)[1],2), dimnames = name)

# deal with repeated journal names
journal = str_replace_all(tolower(removePunctuation(rownames(Adj))), "[\r\n]" , " LineBreak ")
compress = sparse.model.matrix(~journal-1)
A = t(compress)%*%Adj%*%compress
A@x=sqrt(A@x)
rownames(A) <- sub("journal","",rownames(A))
colnames(A) <- sub("journal","",colnames(A))
uniqueJournals = rownames(A)
```

2. Simple analysis based on indegree and outdegree of nodes

```
# indegree
indegree <- colSums(A)
names(sort(indegree, decreasing = T)[1:10])
```

```
##  [1] "nature"
##  [2] "science"
##  [3] "proceedings of the national academy of sciences of the united states of america"
##  [4] "the new england journal of medicine"
##  [5] "plos one"
##  [6] "the journal of biological chemistry"
##  [7] "the lancet"
##  [8] "jama"
##  [9] "cell"
## [10] "circulation"
```

The top ten journals being cited are listed above, and they are all well known journals.

```
# outdegree
outdegree <- rowSums(A)
names(sort(outdegree, decreasing = T)[1:10])
```

```
##  [1] "plos one"
##  [2] "scientific reports"
##  [3] "arxiv"
##  [4] "international journal of molecular sciences"
##  [5] "proceedings of the national academy of sciences of the united states of america"
##  [6] "sensors basel  switzerland"
##  [7] "biorxiv"
##  [8] "biomed research international"
##  [9] "nature communications"
## [10] "ieee access"
```

The top ten journals that cite others are more interesting. Actually, “arxiv” and “biorxiv” are not even journals, but they are sources for lots of papers, so it is not suprising that they cite others a lot.

3. Apply vsp to do anlysis

```
fa =  vsp(A, rank = 50)
```

3.1 Clusters of statistics journals in both citing clusters and cited clusters.

```
apply(fa$Y,2, function(x) uniqueJournals[order(-x)[1:20]])[,37]
```

```
##  [1] "journal of the american statistical association"
##  [2] "annals of statistics"
##  [3] "biometrika"
##  [4] "biometrics"
##  [5] "technometrics"
##  [6] "journal of the royal statistical society series bmethodological"
##  [7] "annals of mathematical statistics"
##  [8] "annals of probability"
##  [9] "journal of the royal statistical society series bstatistical methodology"
## [10] "statistics in medicine"
## [11] "stochastic processes and their applications"
## [12] "statistics  probability letters"
## [13] "journal of statistical planning and inference"
## [14] "journal of econometrics"
## [15] "comput stat data anal"
## [16] "journal of multivariate analysis"
## [17] "econometrica"
## [18] "annals of applied probability"
## [19] "journal of applied probability"
## [20] "probability theory and related fields"
```

```
apply(fa$Z,2, function(x) uniqueJournals[order(-x)[1:20]])[,37]
```

```
##  [1] "arxiv probability"
##  [2] "arxiv statistics theory"
##  [3] "arxiv methodology"
##  [4] "comput stat data anal"
##  [5] "annals of statistics"
##  [6] "journal of statistical computation and simulation"
##  [7] "arxiv"
##  [8] "j multivar anal"
##  [9] "journal of the american statistical association"
## [10] "stochastic processes and their applications"
## [11] "journal of applied statistics"
## [12] "bernoulli"
## [13] "statistics in medicine"
## [14] "journal of econometrics"
## [15] "journal of statistical planning and inference"
## [16] "annals of applied probability"
## [17] "statistics  probability letters"
## [18] "annals of the institute of statistical mathematics"
## [19] "journal of nonparametric statistics"
## [20] "arxiv applications"
```

3.2 Observe B matrix to see relationships between clusters

Statistics clusters in both citing clusters and cited clusters are the 37th, so we focus on 37th column and 37th row.

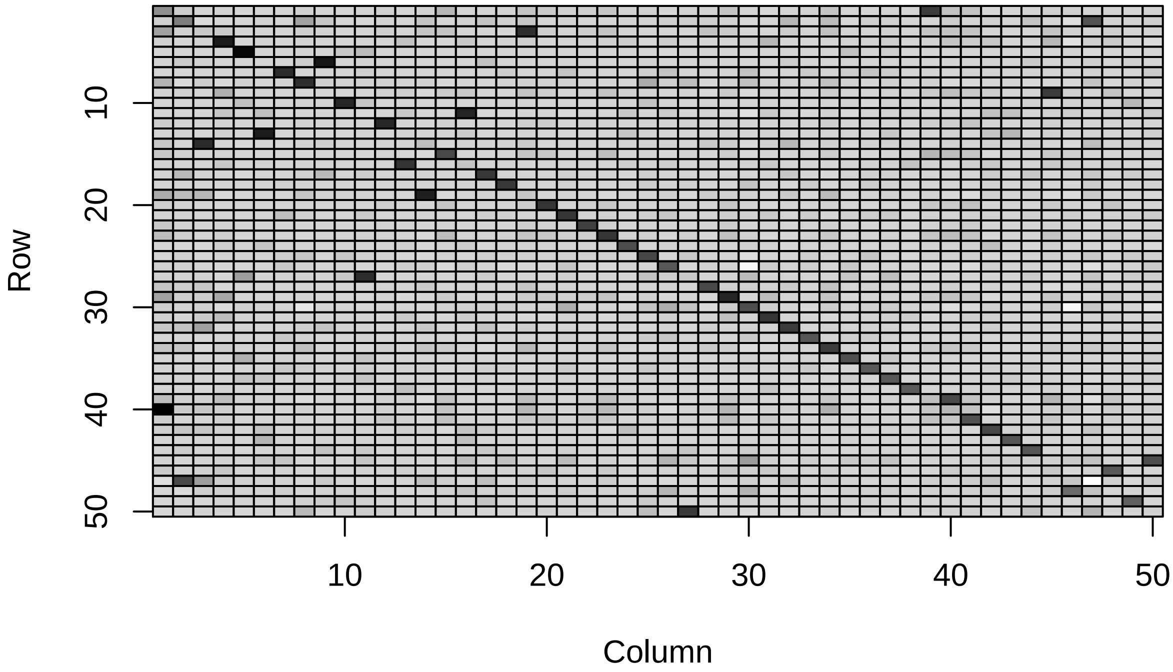
There is a strong diagonal pattern, indicating that if an edge comes from a node in sending block u, then it probably goes to a node in receiving block u.

If we focus on statistics block, it is clearly strongly connected to itself, but there are also other blocks that cites it or being cited by it.

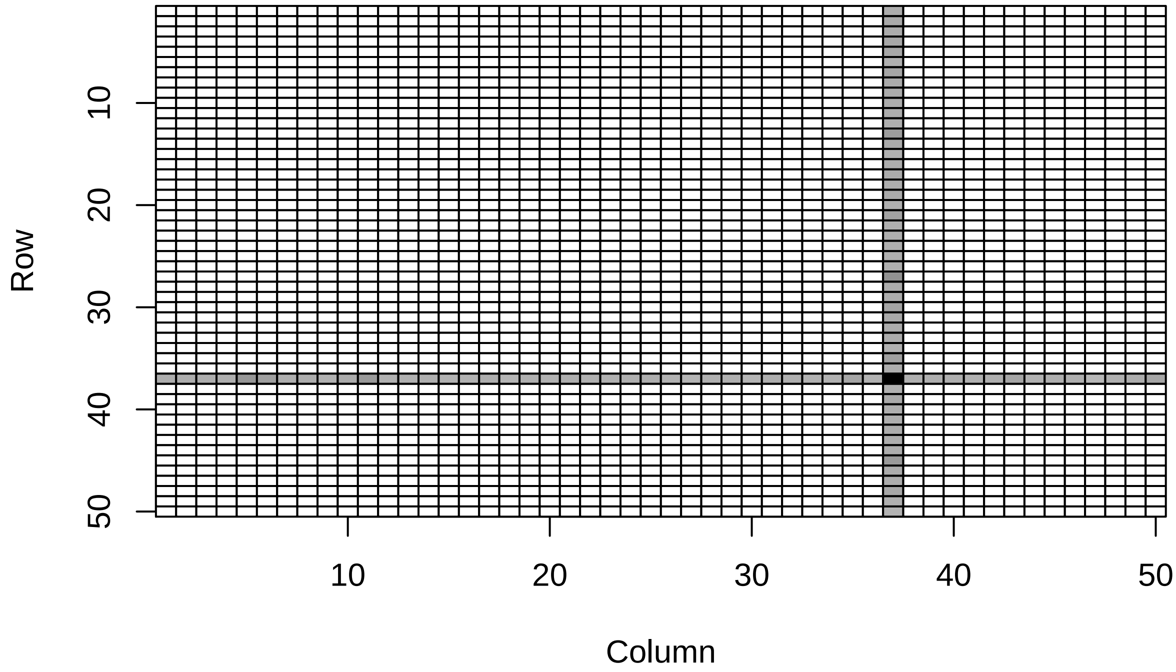
```
B <- as.matrix(fa$B)
range(B)
```

```
## [1] -9.875978e-07  4.949031e-06
```

```
B1 <- matrix(-1.299*10^(-6),50,50)
B1[,37] <- B[,37]
B1[37,] <- B[37,]
color2D.matplot(B, cs1=c(1,0),cs2=c(1,0),cs3=c(1,0))
```



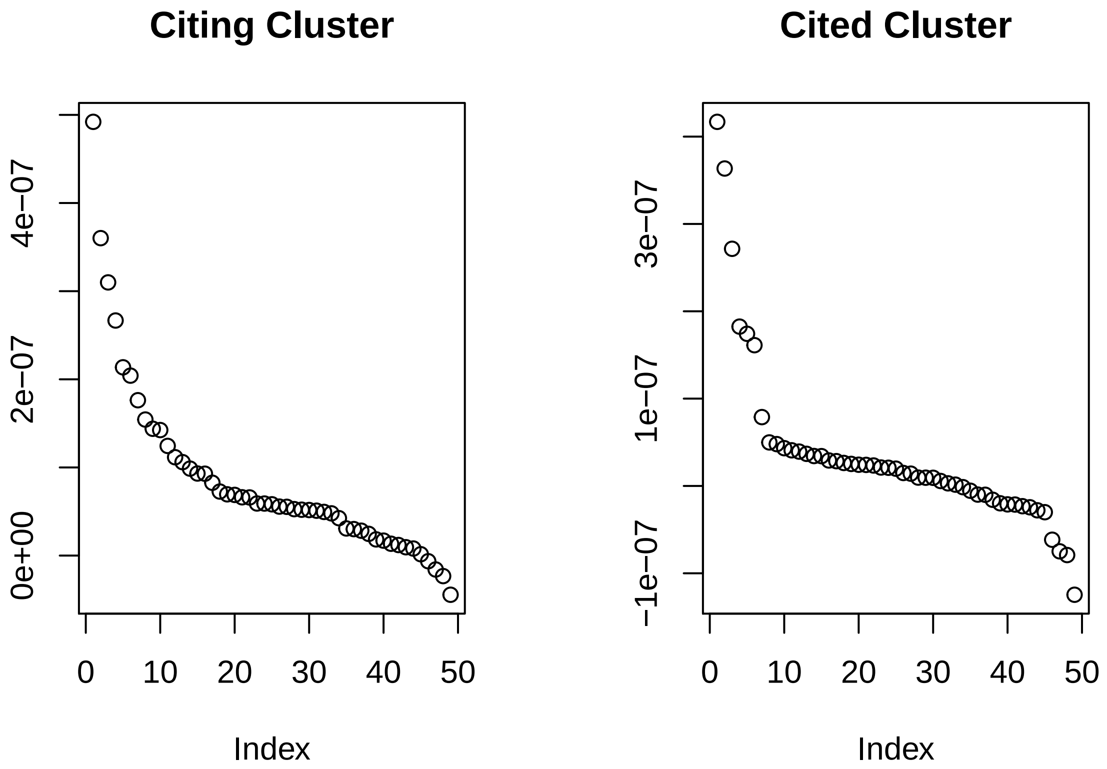
```
color2D.matplot(B1, cs1=c(1,0),cs2=c(1,0),cs3=c(1,0))
```



Coefficient are ranked and ploted without the first one, since the first one is way larger than the others.

For cited cluster, it's clear that we should choose the first 6 clusters, for citing cluster, we could choose the first 4 or 6 clusters. Since we choose 6 clusters for cired cluster and it won't harm to sample more data, we also choose the first 6 clusters for citing cluster.

```r
CiteStat <- sort(fa$B[,37], decreasing = T)
StatCite <- sort(fa$B[37,], decreasing = T)
par(mfrow = c(1,2))
plot(CiteStat[-1], ylab = '', main = 'Citing Cluster') # first 7 (including itself)
plot(StatCite[-1], ylab = '', main = 'Cited Cluster') # first 7 (including itself)
```



```
CiteStatSample <- CiteStat[2:7]
StatCiteSample <- StatCite[2:7]
```

Now let’s find out the topics (like statistics for the 37th cluster) of these clusters and top journals in them.

We use bff to extract best feature of these functions

need some analysis here, but my writing skill sucks

In short, statistical journals are highly connected with journals in math, computer science and economics. Now we can get a list of journal names and sample papers based on them.

```
CiteStatIndex <- as.numeric(sub("z","",names(CiteStatSample)))
StatCiteIndex <- as.numeric(sub("y","",names(StatCiteSample)))
text_df <- tibble(id = 1:length(uniqueJournals),
                  text = uniqueJournals)
# this does a lot of processing!
#  to lower, remove @ # , .
#  often these make sense on a first cut.
#  worth revisiting before "final results"!
tt  = text_df %>% unnest_tokens(word, text)
dt = cast_sparse(tt, id, word)
cs = colSums(dt)
dt = dt[,cs>3]
# bff is the "best feature function"
#  it is my favorite way to contextualize clusters.
bff(fa$Z, dt,10)[,CiteStatIndex]
```

```
##       [,1]           [,2]           [,3]           [,4]
##  [1,] "mathematical" "control"      "economics"    "j"
##  [2,] "arxiv"        "engineering"  "economic"     "comput"
##  [3,] "analysis"     "transactions" "review"       "math"
##  [4,] "mathematics"  "j"            "finance"      "arxiv"
##  [5,] "numerical"    "syst"         "policy"       "theory"
##  [6,] "mechanics"    "comput"       "financial"    "discret"
##  [7,] "equations"    "fuzzy"        "journal"      "comb"
##  [8,] "siam"         "management"   "econometrics" "combinatorics"
##  [9,] "math"         "cybernetics"  "economy"      "sci"
## [10,] "comput"       "oper"         "of"           "appl"
##       [,5]           [,6]
##  [1,] "processing"   "robotics"
##  [2,] "signal"       "control"
##  [3,] "vision"       "automation"
##  [4,] "image"        "robots"
##  [5,] "on"           "ieee"
##  [6,] "ieee"         "intelligent"
##  [7,] "pattern"      "ieeersj"
##  [8,] "recognition"  "conference"
##  [9,] "conference"   "on"
## [10,] "transactions" "transactions"
```

```
bff(fa$Y, dt,10)[,StatCiteIndex]
```

```
##       [,1]            [,2]           [,3]           [,4]           [,5]
##  [1,] "mathematical"  "mathematical" "economics"    "j"            "accounting"
##  [2,] "mathematics"   "mathematics"  "economic"     "comput"       "finance"
##  [3,] "arxiv"         "siam"         "review"       "math"         "financial"
##  [4,] "geometry"      "physics"      "journal"      "theory"       "economics"
##  [5,] "mathematica"   "analysis"     "finance"      "comb"         "journal"
##  [6,] "algebra"       "mechanics"    "of"           "mathematical" "business"
##  [7,] "topology"      "numerical"    "econometrics" "discret"      "review"
##  [8,] "differential"  "applied"      "economy"      "foundations"  "of"
##  [9,] "mathematische" "equations"    "policy"       "mathematics"  "management"
## [10,] "mathã"         "math"         "political"    "siam"         "banking"
##       [,6]
##  [1,] "oper"
##  [2,] "comput"
##  [3,] "res"
##  [4,] "syst"
##  [5,] "transactions"
##  [6,] "j"
##  [7,] "management"
##  [8,] "transportation"
##  [9,] "appl"
## [10,] "fuzzy"
```

code to get journal list

```r
# n is the number of journals we want in each clusters
n = 50
Jname <- as.vector(apply(fa$Z,2, function(x) uniqueJournals[order(-x)[1:n]])[,c(37,CiteStatIndex)])
Jname <- c(Jname,as.vector(apply(fa$Y,2, function(x) uniqueJournals[order(-x)[1:n]])[,c(37,StatCiteInde
Jname <- unique(Jname)
```