SCHOOL OF MATH DEPARTMENT

Statistical Analysis on NIPS Conference Papers

Group memeber: Liu Ping, Guo Feifei
March 13, 2017

1 INTRODUCTION

The Conference and Workshop on Neural Information Processing Systems (NIPS) is a machine learning and computational neuroscience conference held every December. NIPS was designed as a complementary open interdisciplinary meeting for researchers exploring biological and artificial Neural Networks[Wikipedia]. We choose the dataset the NIPS Conference Papers 1987-2015 to do some statistical analysis. The dataset is in the form of 11463×5812 matrix of word counts, containing 11463 words and 5811 NIPS conference papers (the first column contains the list of the words). Each column contains the number of times each word appears in the corresponding document.

Based on our statistical analysis we want to figure the development feature of the conference, some related fields and some specific methods. Besides this theme, we want find some interesting information also.

This report presents some statistical analysis of the dataset which focus on clustering analysis of the words and articles. We choose some statistic methods to pretreat the dataset under the consideration of keeping high frequency and meaningful words. We get a new word-document matrix of order 124 × 5812. Inspired by the topic introduction of NIPS in Wikipedia, we using SVM (support vector machine) method to classify all the articles as 6 different fields. Analysis on the change of quantities of each field and the variation of proportion of different fields every year was presented. Meanwhile, we classify the words by using the index, sum of within square , to get the optimal numbers of cluster and get 5 different categories. Each category reveals specific development process of some methods and theory. In the end, we present the correlation analysis of some specific words and find some applications of this correlation result.

This report is organized as follows. In section 2 a classification of articles based on some specific words related to their field is presented. Analysis on the development tendency of different topic articles exhibits some meaningful facts. Such as the chosen quantities of different topic article every year actually shows some tendency of the development of each field and trade-off made by conference hosts in selecting papers from different field. Some analyses based on clustering of words are presented in section 3. Over these analyses to some specific words we can see a more explicit development process of different methods using in papers, such as kernel method, and deep learning. In section 4, we do some correlation analysis over words, which also indicate interesting facts and applications. Our code is given in the appendices.

2 CLUSTERING ANALYSIS OF ARTICLES

The conference had over 2,500 registered participants in 2014. Besides machine learning and neuroscience, other fields represented at NIPS include cognitive science, psychology, computer vision, statistical linguistics, and information theory[Wikipedia]. Based on this introduction from Wikipedia, we classify all the articles into 5 categories by using SVM (support vector machine) method.

By searching information of the different field we determine some specific words to cluster the articles.

Field	Word
Neuroscience	neutron neutrons
Machine learning	machine learning
Cognitive science	cognition cognitive
Computer vision	vision visual
Information theory	entropy entropies

Table 2.1: Words indicate specific fields.

Each word is usually only contained in the respected field such as neutron, neutrons, cognitive, cognition, entropy, entropies and machine. But vision and visual may also be contained in neuroscience field, so our algorithm concerns words neutron, neutrons prior to vision and visual. And certainly, the word learning will only be considered if other words haven't indicate anything.

More precisely, we have following priority rule.

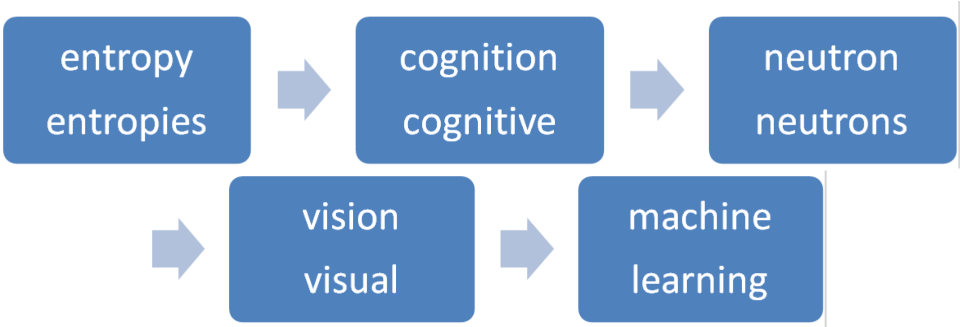

Figure 2.1: Priority rule of words

We determine an algorithm based on above priority rule to classify first 500 articles. Take this result as a training set we use SVM method classify all 5811 articles. Based on the classification we further have some analyses.

Field	quantity
Neuroscience	2158
Machine learning	2791
Cognitive science	244
Computer vision	111
Information theory	103
Undetermined	404
Total	5811

Table 2.2: Whole article quantities 1987-2015.

From above figure we can first see that it actually verifies the introduction of NIPS from Wikipedia that besides the major part neuroscience and machine learning, NIPS also includes field cognitive science, computer vision and information theory. And from the total number of neuroscience and machine learning we see that they almost cover all the papers in the conference.

Besides the total quantities of papers over 29 years, we figure out the total number of papers every 10 year and consider the change in proportion of each topic.

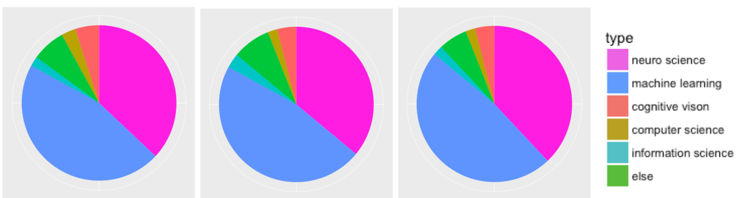

Figure 2.2: distribution of papers 1987-1995 1996-2005 2006-2015

From the figure of distribution of each field over 10 years we see that the proportion of each topic haven't changed too much and the proportion of machine learning and neuroscience are slightly increasing when considering the total number over 10 years.

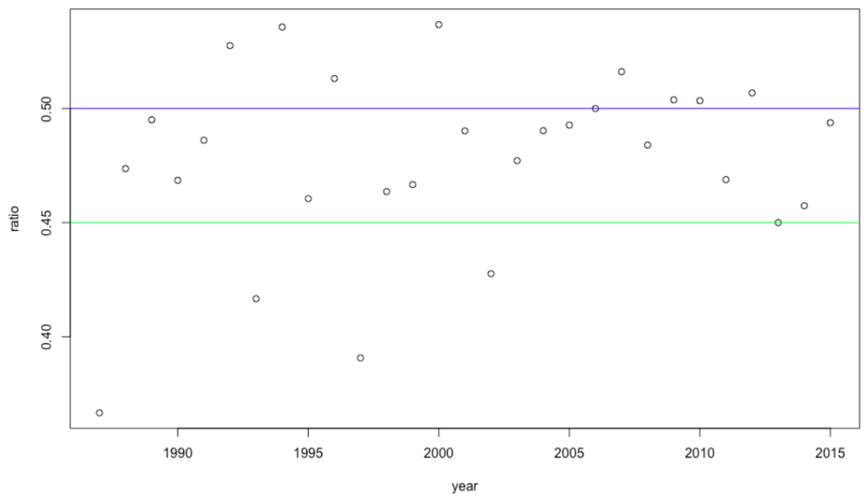

Figure 2.3: Proportion of machine learning

This figure illustrates the papers about machine learning occupied more than 45% almost every year and in many years, it covers over 50% of the whole papers.

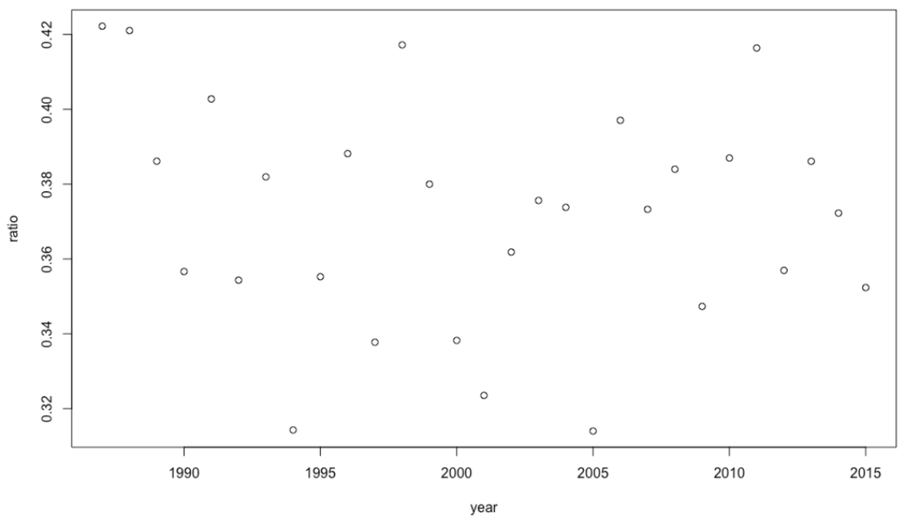

Figure 2.4: Proportion of neuroscience

The neuroscience paper actually has a decreasing tendency from the highest point around year 1987 to the much lower point around year 2015.

And compared with above two figures we find that the conference hosts will make some trade-off in selecting papers from different topic to make the proportion changed slowly. Because we can see that the difference of proportion of both machine learning and neuroscience in neighbored year is usually less than 0.1. And the change of neuroscience is much slower.

3 Clustering analysis of words

Because a mass of words are rarely appeared, we firstly utilize the sum of word number over this 29 years to eliminate these words. Then we analyze the variation of remained words to further eliminate meaningless words, such as "abstract" and "also" which appear in every paper. After these pretreatment of the total 11464 words, we finally get around 100 meaningful words. Cause of the order of the first paper and the second paper in one year may not indicate any special different information but they may have enormous difference in word feature. Under this consideration, we sum the number of words in whole year to reduce data dimension and meanwhile obtain more accurate information. We then use clustering method to classify these words.

K-means cluster method is a popular and simple method. However when we use k-means method to do clustering , we should give a numbers of cluster. And we can provide a number according to our experience. This number is subjective. This report we use the sum of the total within sum of square to determine the optimal number of cluster.

Finally we get 5 categories as figure followed shown. Each category represents a specific change in quantity of words and explicit analysis on some special words actually reveal interesting facts.

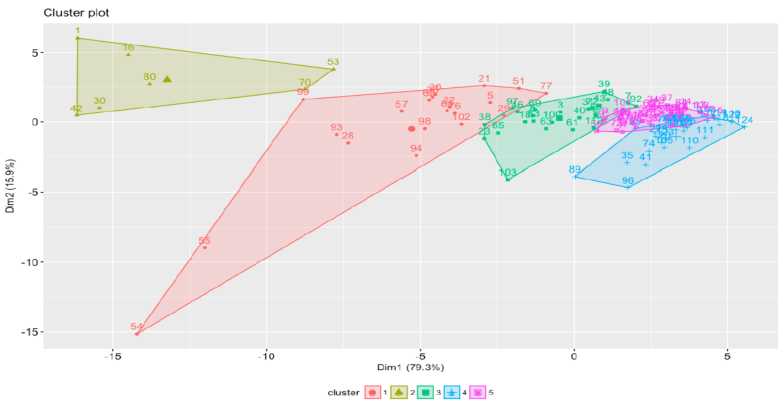

Figure 3.1: Cluster result

Category	Word
I	image neural network training
II	algorithm data function learning
III	optimal gaussian kernel random
IV	deep visual layer recognition
V	convex bayesian sparse machine

Table 3.1: Partial exhibition of classification result

The above table shows some classified words in different category. Each category stands for a specific change in number of words over 29 years.

Firstly we present the figure of paper quantity every year to show some information contained.

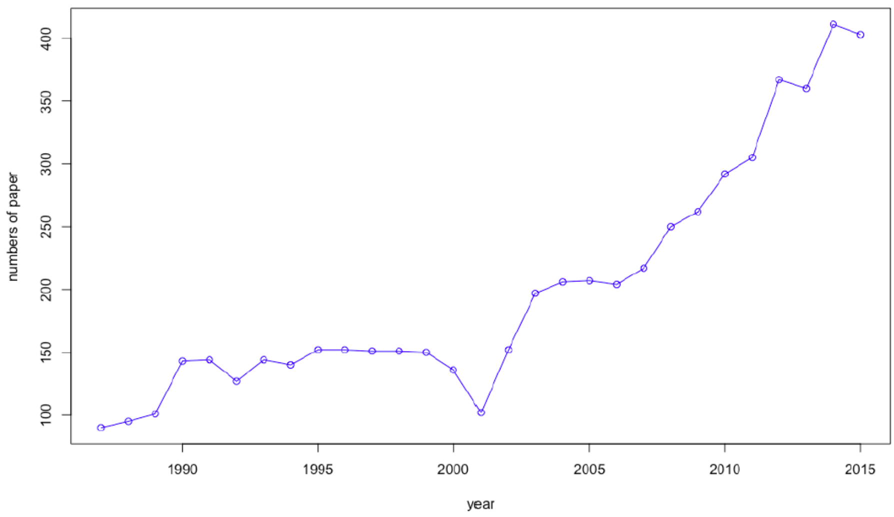

Figure 3.2: The change of paper number

From the above figure we can see that the paper number every has an steadily and large increase from year 2001 to 2015 which indicates the popularity of research on neuroscience and machine learning recent 15 years.

Then we analyze some specific words from different categories.

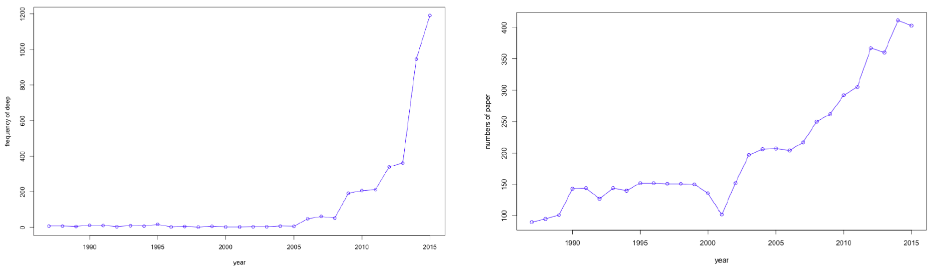

Figure 3.3: The change in number of "deep" and paper number

Compared with the increasing rate of paper quantity every year in the right figure, we can directly see from the sharp change in the word "deep" that the research of deep learning explosive increases in year 2014 and 2015. And around year 2008 the research begins to be paid more attention to.

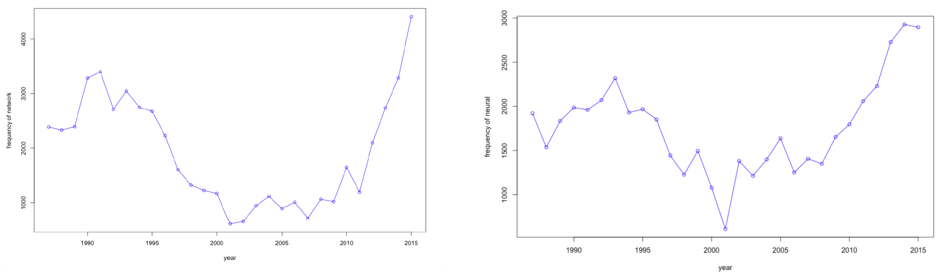

Figure 3.4: The change in number of "network" and "neural"

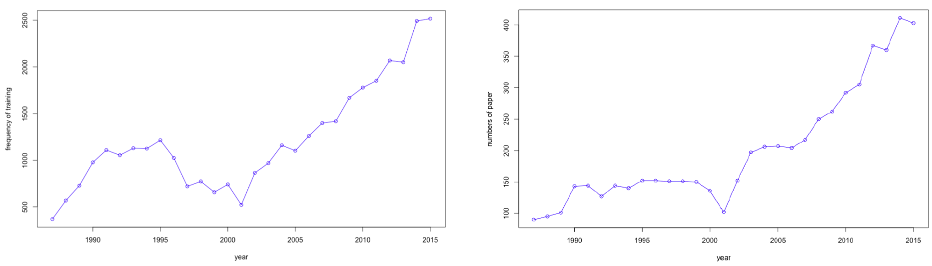

Figure 3.5: The change in number of "training" and paper number

Cause training, network and neural both belong to the category I, we present them together with the paper number figure to illustrute some interesting information. From the above figure we directly see the special decreasing of the words network, neural and training over year 1993 to year 2000 when the paper number seems doesn't change. So this may indicate these years is a bottleneck of the development of neural network. And there may be a great breakthrough around year 2002 cause of the increasing of both paper number and words "network" and "training".

And we also see the increasing of three words over year 1987 to 1990. The three year may be a short popular period of neural network or the rise of it. But later scientists encountered a bottleneck. Besides the same tendency of the three words, we can see the special explosive growth of "network" around recent year while the number of paper just increases approximately linearly. This may related with the explosive growth of deep learning we have shown.

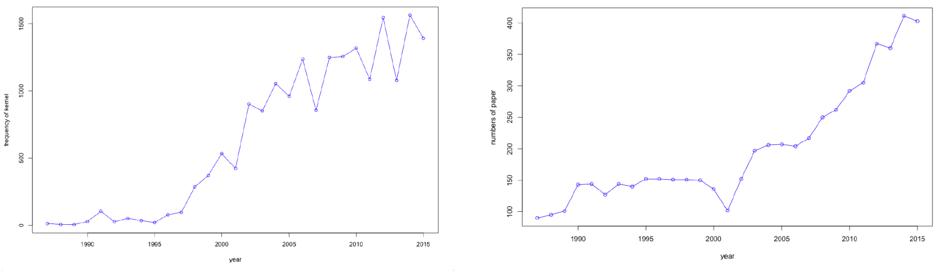

Figure 3.6: The change in number of "kernel" and paper number

Kernel may represent the kernel method and it start to increase in year 1997, while the paper number doesn't change or somehow decreases. This indicates that around year 1997, there must be some breakthrough in the kernel method or in applying it into neuroscience and machine learning. And this method must be very useful and important cause of the increasing rate is much greater than the growth of paper number for a long period.

4 Correlation analysis of words

Based on the common sense that articles from a specific field may show some common feature in related words, we try to find some internal correlation between some words. Certainly, we do the correlation analysis only to our chosen meaningful words and no matter how many times the word appears in the paper, we just care about whether this word appears in paper or not. Therefore first we will change out data into 0-1 data. And then we analyze the correlation of these words.

Word	Correlation
likelihood	0.7299
prior	0.7243
regression	0.6169
stochastic	0.6047
estimation	0.6033

Table 4.1: Correlation with "Bayesian"

Word	Correlation
learning	0.6148
solution	0.6138
performance	0.6050
value	0.6048
linear	0.6038
bound	0.593

Table 4.2: Correlation with "optimal"

As the above table showing, we give two examples to show our analysis results. When we look at the word "Bayesian", we will find some words, such as likelihood, prior and regression, have high correlation with it. And also we will find some words, such as learning , solution and bound, have high correlation with word "optimal". Obviously, these results are also consistent with the actual.

In the process of searching the actual paper, the keyword search will be particularly important. Some keywords may appear in the title of the paper, but some keywords will be hidden in the paper. At this point, the use of analysis of the correlation of the words which have highly relative degree, may also be more comprehensive and more accurate to find the information we need.

Therefore, based on our result, we can use the correlation between each words to provide some guides for literature retrieval.

5 Summary

We use some methods in statistic and machine learning to study some useful information hidden in the dataset. Our analysis over the classification of specific words and papers shows some interesting and meaningful facts or information. These analyses can help to understand more about the development of NIPS conference and further the development of some field and some specific methods. In our last section we study the correlation between some specific words which may helpful for literature retrieval. Beyond just only get information from the data we find some real application after statistical analysis.

Appendices

```
    ##input data(R)
setwd("/Users")
data1=read.csv("NIPS_1987-2015.csv");
##Data preprocessing
data11=data1[,2:11464];
data2=colSums(data1[,2:11464]);
data7=data11[,which(data2>10000)];
data12=data7[,apply(var(data7)>16,2,any)]
data8=data11[,which(data2>2000 & data2<10000)];
data9=data8[,apply(var(data8)>25,2,any)];
data10=cbind(data12,data9);
data13=data10[,-c(2,28,33,40,43,59,60,61,63,65,72,74,77,84,92,96,119,121,130,132,141,145)]
data14=cbind(data13,data1$deep);
data111=data.frame(X=data1$X,data14);
install.packages("stringr")
library(stringr)
d=matrix(0,nrow=5811,ncol=2);

for (i in 1:5811){
a=substr(data111$X[i],1,4);
a1=as.numeric(a);
b=substr(data111$X[i],6,stop=str_length(data111$X[i]));
b1=as.numeric(b);
c=cbind(a1,b1);
d[i,]=c;
}

id=data.frame(id.year=d[,1],id.num=d[,2]);
data.final=cbind(data111,id);
w=matrix(0,nrow=29,ncol=124);
for(i in 1987:2015){
w[i-1986,]=colSums(data111[which(data.final$id.year==i),2:125]);
}
datayear=data111[1:29,];
datayear[1:29,2:125]=w;
datay=datayear[,2:125];
data101= t(data.frame(datay))
data102=as.data.frame(data101,row.names=F)

##clustering
install.packages("factoextra")
```

```
install.packages("ggplot2")
library(factoextra)
library(ggplot2)
set.seed(1234)
cluster2=fviz_nbclust(data102,kmeans,method="wss")
geom_vline(xintercept = 5,linetype=2)
km.res=kmeans(data102,5)
fviz_cluster(km.res,data=data102)
datatest=data14[,which(km.res$cluster==1)];
a1=names(datatest)
datatest2=data14[,which(km.res$cluster==2)];
a2=names(datatest2)
datatest3=data14[,which(km.res$cluster==3)];
a3=names(datatest3)
datatest4=data14[,which(km.res$cluster==4)];
a4=names(datatest4)
datatest5=data14[,which(km.res$cluster==5)];
a5=names(datatest5)
datayl=data.frame(year=unique(data.final$id.year),deep=datay$data1.deep);
datayl=data.frame(year=unique(data.final$id.year),network=datay$kernel);
datayl=data.frame(year=unique(data.final$id.year),network=datay$network);
datayl=data.frame(year=unique(data.final$id.year),network=datay$training);
datayl=data.frame(year=unique(data.final$id.year),network=datay$deep);
plot(x=datayl[,1],y=datayl[,2],type="o",xlab="year",ylab="frequency of deepâĂİ,col="blue")
plot(x=datayl[,1],y=datayl[,2],type="o",xlab="year",ylab="frequency of training",col="blue
plot(x=datayl[,1],y=datayl[,2],type="o",xlab="year",ylab="frequency of networkâĂİ,col="blu
plot(x=datayl[,1],y=datayl[,2],type="o",xlab="year",ylab="frequency of kernelâĂİ,col="blue
future=paste("network",".png");
jpeg(file=future);
plot(x=datayl[,1],y=datayl[,2],type="o",xlab="year",ylab="frequency of network",col="blue"
dev.off();

library(fpc)
pamk.best=pamk(data102);
pamk.best$nc
pamk.best$pamobject$clustering
datatest=data13[,which(pamk.best$pamobject$clustering==1)];
names(datatest)
datatest2=data13[,-which(pamk.best$pamobject$clustering==1)];
names(datatest2)
library(cluster)
clusplot(pam(data102,pamk.best$nc))
```

```
##correlation
data_replace=read.csv(file)
data_replace[data_replace>0]=1
data_replace_delete=data_replace[,-1]
a=cor(data_replace_delete)
t=ncol(data_replace_delete)
m=nrow(data_replace_delete)
corr_data=a-diag(t)
for (i in 1:t) {
h=data_replace_delete-data_replace_delete[,i]
h[h!=0]=-1
h[h==0]=1
h[h==-1]=0
for (j in 1:t) {
corr_data[i,j]=sum(h[,j])/m
}
}
corr_data=corr_data-diag(t)
write.csv (corr_data, file ="corr_data.csv")


##analysis of the result of svm
da=read.table(âĂIJdata6.txt",sep=".");
dat=da[,1:18];
datanew=matrix(0,nrow=1,ncol=1);
for(i in 1:18){
datanew=rbind(datanew,as.matrix(dat[,i]));
}
datanew1=datanew[2:5812];
ab=read.csv(file="datayear.csv");
data.yc=data.frame(id=data.final$X,year=data.final$id.year,num=data.final$id.num,datac=dat
w=matrix(0,nrow=29,ncol=2);
for(i in 1987:2015){

c=nrow(data.final[which(data.final$id.year==i),]);
w[i-1986,]=rbind(i,c);

}
ynum=data.frame(year=w[,1],num=w[,2]);

data.y0=data.yc[which(data.yc$datac==0),];
data.y1=data.yc[which(data.yc$datac==1),];
data.y2=data.yc[which(data.yc$datac==2),];
data.y3=data.yc[which(data.yc$datac==3),];
```

```
data.y4=data.yc[which(data.yc$datac==4),];
data.y5=data.yc[which(data.yc$datac==5),];

c=unique(data.y0$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y0[which(data.y0$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
datanew=rbind(datanew,m);
}
datanew1=datanew[2:nrow(datanew),];
plot(datanew1,xlab="year",ylab="ratio")

c=unique(data.y1$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y1[which(data.y1$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
datanew=rbind(datanew,m)
}
datanew1=datanew[2:nrow(datanew),];
plot(datanew1,xlab="year",ylab="ratio")



c=unique(data.y2$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y2[which(data.y2$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
datanew=rbind(datanew,m)
}
datanew1=datanew[2:nrow(datanew),];
plot(datanew1,xlab="year",ylab="ratio")

c=unique(data.y3$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y3[which(data.y3$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
```

```
datanew=rbind(datanew,m)
}
datanew1=datanew[2:nrow(datanew),];
plot(datanew1,xlab="year",ylab="ratio")

c=unique(data.y4$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y4[which(data.y4$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
datanew=rbind(datanew,m)
}
datanew1=datanew[2:nrow(datanew),];
mean(datanew1[,2])
plot(datanew1,xlab="year",ylab="ratio")
abline(h=0.5,lwd=1,col="blue")
abline(h=0.45,lwd=1,col="green")

c=unique(data.y5$year)
datanew=matrix(0,nrow=1,ncol=2);
for(i in 1:length(c)){
l=nrow(data.y5[which(data.y5$year==c[i]),]);
n=ynum[which(ynum$year==c[i]),2];
m=cbind(c[i],l/n);
datanew=rbind(datanew,m)
}
datanew1=datanew[2:nrow(datanew),];
mean(datanew1[1:9,2])
mean(datanew1[10:19,2])
mean(datanew1[20:29,2])
plot(datanew1,xlab="year",ylab="ratio")


##pie chart 1987-1995
type <- c('neuro science','machine learning','cognitive vison','computer science','informa
nums <- c(0.37,0.46,0.05,0.03,0.02,0.07)
df <- data.frame(type = type, nums = nums)
label_value <- paste('(', round(df$nums/sum(df$nums) * 100, 1), '%)', sep = '')
label <- paste(df$type, label_value, sep = '')
p<-ggplot(df, aes(x = "",  y = nums, fill = type)) + geom_bar(stat = "identity", position
theme(axis.text = element_blank()) + scale_fill_discrete(breaks = df$type, labels = label)
p
```

```
##pie chart 1996-2005
library(ggplot2)
type <- c('neuro science','machine learning','cognitive vison','computer science','informa
nums <- c(0.36,0.47,0.04,0.02,0.03,0.08)
df <- data.frame(type = type, nums = nums)
label_value <- paste('(', round(df$nums/sum(df$nums) * 100, 1), '%)', sep = '')
label <- paste(df$type, label_value, sep = '')
p<-ggplot(df, aes(x = "",  y = nums, fill = type)) + geom_bar(stat = "identity", position
theme(axis.text = element_blank()) + scale_fill_discrete(breaks = df$type, labels = label)
p

##pie chart 2006-2015
type <- c('neuro science','machine learning','cognitive vison','computer science','informa
nums <- c(0.38,0.48,0.04,0.02,0.02,0.06)
df <- data.frame(type = type, nums = nums)
label_value <- paste('(', round(df$nums/sum(df$nums) * 100, 1), '%)', sep = '')
label <- paste(df$type, label_value, sep = '')
label <- df$type
p<-ggplot(df, aes(x = "",  y = nums, fill = type)) + geom_bar(stat = "identity", position
theme(axis.text = element_blank()) + scale_fill_discrete(breaks = df$type, labels = label)
p
}

#SVM(python)
import numpy as np
import pandas as pd
import math
data = pd.read_csv("sample1.csv",header= None)
A=data.T
from sklearn import svm
#training 500
train = A.iloc[0:500, :]
X = np.empty([500,12])
y = np.empty([500,1])
for i in range(500):
X[i,:]=train.values[i, :]

if X[i,2]+X[i,3]>1:
y[i,0]=1
elif X[i,6]+X[i,7]>1:
y[i,0]=2
elif X[i,0]+X[i,1]>2:
y[i,0]=0
elif X[i,8]+X[i,9]+X[i,10]+X[i,11]>1  :
```

```
y[i,0]=4
elif X[i,4]+X[i,5]>1:
y[i,0]=3

else:
y[i,0]=5
#svm
clf=svm.SVC()
clf.fit(X,y)

res=clf.predict(A)
np.set_printoptions(threshold='nan')
print(res)
```