Project Cover Sheet

Student IDs : 55323707
Student Names : Yu SU
Student Emails : yusu27-c@my.cityu.edu.hk

Student IDs : 55327269
Student Names : Peiyu LI
Student Emails : peiyuli4-c@my.cityu.edu.hk

Student IDs : 55362327
Student Names : Xinpin XU
Student Emails : xinpinxu2-c@my.cityu.edu.hk

Student IDs : 55358897
Student Names : Jianan ZHOU
Student Emails : jianazhou2-c@my.cityu.edu.hk

Title:

Using Text Mining of arXiv Paper Titles to Reveal Research Trends In Computer Science Over the Past Decade

Highlights: (maximum: 5 items)

- Parallel for text processing
- Phrase collocation
- Association rule
- K-means clustering
- Regression
- Visualization

List of Deliverables

File Name	Description
report.pdf	Main Project Report
Code.zip	Source Code
Data.zip	Raw Data and Processing
Group6.ppt	Presentation Slide

------ END ------

Using Text Mining of arXiv Paper Titles to Reveal Research Trends In Computer Science Over the Past Decade Semester A 2018/2019

Yu Su, Xinpin Xu, Peiyu Li, Jianan Zhou

November 29, 2018

Abstract

Objective. To analyse the research heat of the arXiv papers in computer science during last decade and predict the trends in the future.
Methods. A crawler algorithm is wrote to get the data from the internet. The dataset is analysed by association rules, K-means clustering and regression.
Conclusion. In this project, we analysed the existed CS technology data to predict the trend of the future by three different ways. We analyse the data directly, use the clustering model to classify the technology and in terms of the clusters to analyse the holistic rules instead of a single word. Then use the regression model to predict the future trend of the technology.
Keywords: Keywords: arXiv, text mining, computer science

1 Introduction

1.1 Background

We are in an age of rapid change in computer technology. New technology up,old technology drop, what we want is tracking the trend of the technology and have a prediction of the future development of the technology.

1.2 Motivation

We are in an era when the popular technology change rapidly.So get the trend of the technology is important.

For investors, they only want to invest the technology which could developed greatly. For researcher, they should pay more attention on the potential technology.

1.3 Object

Analyse the title of the paper of CS technology.
1. Using the word group in the title represent the technology.
2. Get the word group number data. such as frequency, sum and standard deviation.
3. Making model to analyse the data.
4. Do preparation and get conclusion.

2 Data Preparation

2.1 Data Mining

In this part, we use the scrapy frame to crawl the data from https://arxiv.org I crawl the paper title about computer science in 2009 to 2018 and give it an item structure. The part of code:

```
class ContentSpider(RedisSpider):
    name = "Content"
    redis_key = "links"
```

```
    def parse(self, response):
        host = self.settings['REDIS_HOST']
        item = TitleItem()
        title = response.xpath('//div[@class="meta"]/div[1]/text()').extract()

        item['Title'] = title
yield item
```

2.2 Data Preprocessing(using parallel)

- tokenization: At first, the data format what we get is not usable for analysis. to make the data format is easier to use. So we first tokenize the title using pig in parallel.

```
Lines = LOAD '/Users/xuxinpin/Desktop/bbproject/text_choose_2018.txt' AS (line: chararray);
Groups = GROUP Lines BY ',';
Words = FOREACH Groups GENERATE (TOKENIZE(line)) AS word;
STORE Words INTO '/Users/xuxinpin/Desktop/20183';
```

- merge: the data we get from data mining is divided in each year. Sometimes we need all the data instead of the part, so we also did the merge in python.

```
fout=open("merge.csv","a")

for line in open("text_choose_2009.csv"):
        fout.write(line)
for num in range(2010,2018):
        f = open("text_choose_"+str(num)+".csv")
        for line in f:
                fout.write(line)
        f.close()
fout.close()
```

- Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, removing the step words and punctuation, resolving the inconsistencies in the data. After simple data cleaning, we generate wordcloud graph which size of words is related to frequency to show the words every year.

- Data Transformation: Data is normalized, aggregated and generalized.In this part, we use multi-process to realize the parallel to improve the processing speed.
  Some example:

```
def to_lowercase(x):
    n = []
    for i in x:
        n = n.lower()
        n.append(n)
    return n

def remove_stopwords(x):
    n = []
    for i in x:
        if i not in stopwords.words('english'):
            n.append(i)
    return n
def processing(i):
    titles = []
    csv_rpath = 'data/raw_data/'+str(i)+'.csv'
    csv_wpath = 'data/new_data/new_'+str(i)+'.csv'
    reader = csv.reader(open(csv_rpath,'r'),delimiter=',',quotechar='"')
    for line in reader:
        title = []
        title.append(line)
        title = to_lowercase(title)
        title = remove_stopwords(title)
        titles.append(title)
    title_output = pd.DataFrame(titles)
    title_output.to_csv(csv_wpath_title, index=False, header= False)
```

```
if __name__ == '__main__':
    starttime = time.time()
    pool = multiprocessing.Pool()
    pool.map(processing, range(2009,2018))
    pool.close()
```

And there is other processing like stemming, removing the number, Lemmatize and so on. After all this processing , the data is well structured and neat.

- Data Reduction: The amount of words is extremely large and it will cost time when the file open. In this step, the data which appear rarely, in other words, words with low frequency are removed from data.

2.3 The Data Set

- Raw Data Example

WaveGlow: Flow- Generative Network Speech Synthesis
Deep Net Features Complex Emotion Recognition

Figure 1: Raw Data Example

- Normalized Data Example

deep net feat complex emot recognit
waveglow flow gen network speech synthes

Figure 2: Normalized Data Example

- Word Count Example

word	2009_freq	2010_freq	2011_freq	2012_freq	2013_freq	2014_freq	2015_freq	2016_freq	2017_freq	2018_freq
network	526	781	669	791	783	717	768	853	917	866
system	363	443	439	445	469	487	430	407	402	317
algorithm	327	413	376	406	405	397	334	267	224	189

Figure 3: Word Count Example

- Bigrams Example

phrase	2018
('neur', 'network')	308
('deep', 'learn')	152
('reinforc', 'learn')	95

Figure 4: Bigrams Example

3 METHODS

To better obtain the connection between words, explore the association of various terms of technology and research trends, we use Apriori algorithm to obtain the correlation between various scientific words, and then use K-means algorithm to cluster the words. Finally, regression analysis was used to analyze and predict several major technical categories.

3.1 Association Rules

After the data clean ,what we get is the single word collection, but sometimes only one word can not represent a technology clearly. we want to use association rules to find out the related word, so that we could combine the word to phrase to express the technology or area more specifically. In this section, I use R with apriori algorithm to realize the association rules. The part of the codes:

```
write(itemset_2_n, file="C:/Users/64161/Downloads/association_rules/merge_2.csv",sep=",")
itemset_2_n<-apriori(data, parameter =list(minlen=2,maxlen=2,support=0.001,confidence=0.5))
write(itemset_2_n, file="C:/Users/64161/Downloads/association_rules/merge.csv",sep=",")
plot(itemset_2_n,method = "graph", engine="htmlwidget")
```

when setting the argument, we want take more focus on the association. So we restrict the confidence more strictly instead of the support, so that the rules we got will have more trusted association although the frequency of the word is maybe not too high. At last I generate the graph which could illustrates the strong relation in the titles.

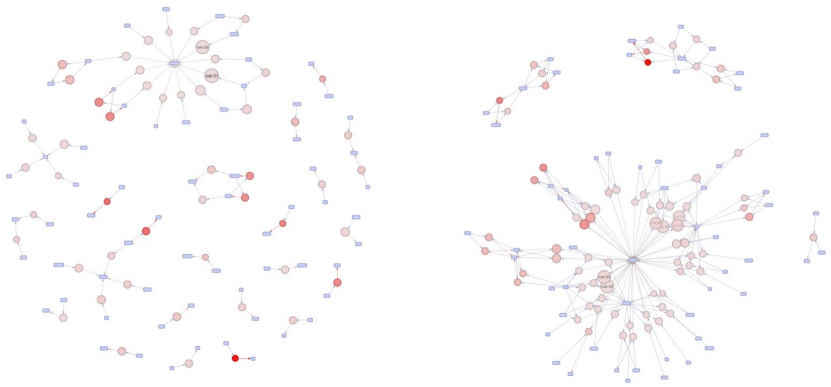


Figure 5: 2-times rules and 3-items rules

I also generated the double-word phrase and triple-word phrase using this association rules for the later analysis.

3.2 Clustering

3.2.1 K-means clustering

To discover the similarity of trends of the ten-year CS technology, we use K-Means Clustering to classify theses phrases. K-means Algorithm iteratively computes the proximity to the center of the k-group to identify the k groups of objects. In this session, we use the package "cluster" in the library of R to identify the phrases, and choose two attributes to describe the different trends of each technology during last decade. The first attribute is the average of each phrases counts in the last ten years, the second attribute is the deviation of the counting phrase as the second attribute.

3.2.2 Determining number of clusters

We set k from 1 to 15, computing the sum of square of the distance between each point and the closest centroid ,respectively. The results of the relation between WSS and k show in Figure 5. The value of WSS dropped sharply untill k increasing to 4, then the changes in the curve tends to be smooth, since we choose the k to be 4.

Listing 1: Estimate the number of centers

```
wordsc=as.data.frame(read.csv('/Users/xuxinpin/Desktop/clusterdata.csv'))
kmdata=as.matrix(wordsc[,c("avg","sub")])
wss <- numeric(15)
for (k in 1:15) wss[k] <- sum(kmeans(kmdata, centers = k,nstart = 30)$withinss)
plot(1:15, wss,type = "b", xlab = "number_of_clusters", ylab = "within_sum_of_squares")
```

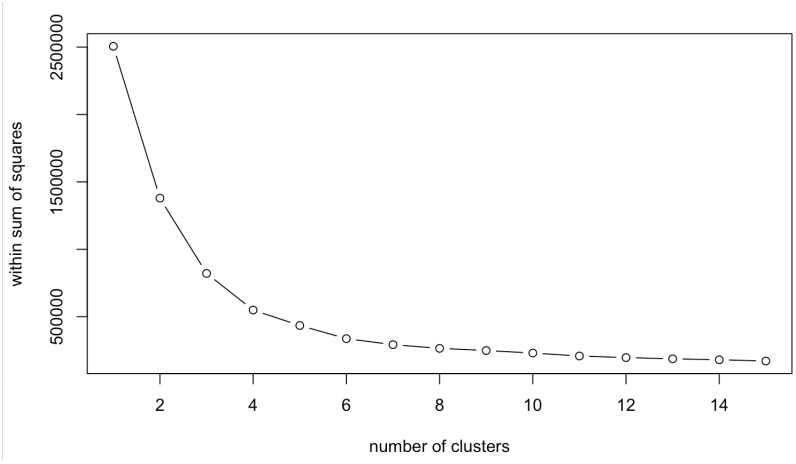
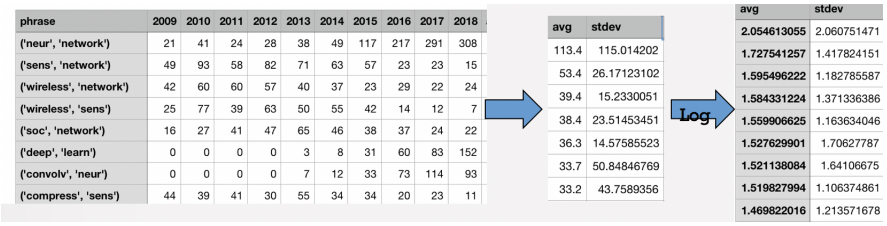


Figure 6: The Number Of Clusters

3.2.3 Generating clusters

Another thing we have to do before clustering it's choosing the attributes of the objects. The first attribute we have choosed is the average counting phrases over last ten years to identify frequency of the phrases appeared in a research topic. Another attribute is the standard deviation of the ten-year phrases counting, it can show the degree of discrete over each year counting phrases. In order to avoid domination and dispersion of clusters, we rescaled the average and standard deviation by logarithm. The figure below shows the data processing that we have done.



phrase	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
('neur', 'network')	21	41	24	28	38	49	117	217	291	308
('sens', 'network')	49	93	58	82	71	63	57	23	23	15
('wireless', 'network')	42	60	60	57	40	37	23	29	22	24
('wireless', 'sens')	25	77	39	63	50	55	42	14	12	7
('soc', 'network')	16	27	41	47	65	46	38	37	24	22
('deep', 'learn')	0	0	0	0	3	8	31	60	83	152
('convolv', 'neur')	0	0	0	0	7	12	33	73	114	93
('compress', 'sens')	44	39	41	30	55	34	34	20	23	11

avg	stdev
113.4	115.014202
53.4	26.17123102
39.4	15.2330051
38.4	23.51453451
36.3	14.57585523
33.7	50.84846769
33.2	43.7589356

Log

avg	stdev
2.054613055	2.060751471
1.727541257	1.417824151
1.595496222	1.182785587
1.584331224	1.371336386
1.559906625	1.163634046
1.527629901	1.70627787
1.521138084	1.64106675
1.519827994	1.106374861
1.469822016	1.213571678

Figure 7: attributes choosing

After choosing the value k and two attributes, we use kmeans() function to generate clusters and set the times of iterative nstart to be 30. Then we add the label column to each phrases and generated a cluster.csv file. The result of average of each clusters and the generated file shows below.

Listing 2: K-means Clustering

```
wordsc=as.data.frame(read.csv('/Users/xuxinpin/Desktop/wordsc.csv'))
kmdata_q=as.matrix(wordsc[,c("words","avg","sub")])
kmdata=kmdata_q[,2:3]
km = kmeans(kmdata,4,nstart = 30)
df=as.data.frame(kmdata_q[,1:3])
df$cluster=factor(km$cluster)
write.csv(df,file="/Users/xuxinpin/Desktop/cluster.csv",row.names = T)
```

```
> km = kmeans(kmdata,4,nstart = 50)

> km
K-means clustering with 4 clusters of sizes 315, 671, 66, 972

Cluster means:
       avg      stdev
1 0.7166292 0.51102352
2 0.3670407 0.29120832
3 1.2086522 1.05324966
4 0.1588468 0.09462321
```

phrase	avg	stdev	Q	cluster
('neur', 'network')	113.4	115.014202	830	1
('sens', 'network')	53.4	26.171231	-172	1
('wireless', 'network')	39.4	15.2330051	-124	4
('wireless', 'sens')	38.4	23.5145345	-124	4
('soc', 'network')	36.3	14.5758552	-29	4
('deep', 'learn')	33.7	50.8484677	331	1
('convolv', 'neur')	33.2	43.7589356	318	1

Figure 8: Results of K-means with R

We use ggplot() function to visualize the clusters(Figure 7). The x-axis is attribute1 average, and the y-axis is attribute2 deviation, different colour represent different clusters. In this figure, we can clearly distinguish the distribution of each cluster.

```
centers=as.data.frame(km$centers)
gl=ggplot(data = df, mapping=aes(x=avg,y=sub,color=cluster))+
  geom_point()+theme(legend.position = "right")+
  geom_point(data = centers,
             mapping=aes(x=avg,y=sub,color=as.factor(c(1,2,3,4))),
             size=5,alpha=.4,show.legend = FALSE)
ggplot_gtable(ggplot_build(gl))
e=grid.arrange(arrangeGrob(gl+theme(legend.position = "none"),
                           main="clusters"))
grid.draw(e)
```

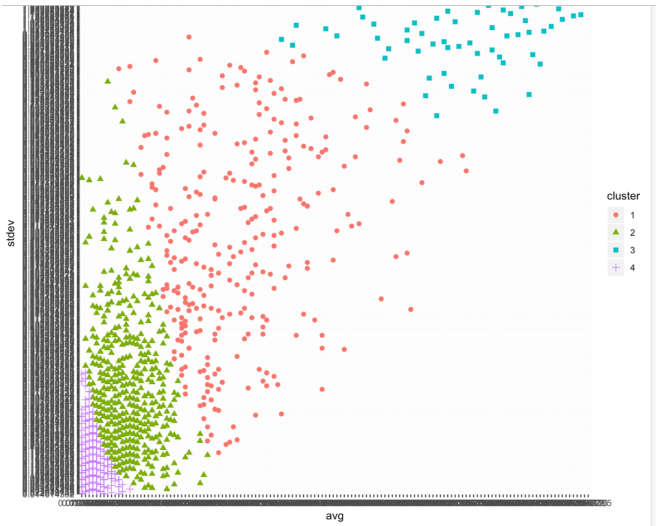


Figure 9: Visualization of clustering

In order to better analyse the trends between these four clusters, we divided the cluster file into four sub set based on the cluster label. Then we use ggplot() to draw a bar chart. An other attribute Q added to each counting phrases is the difference between the first five years and last five years, so that we can find the positive trend and the negative trend of the topic over the ten years. The figures below illustrate that the difference trend based on the attributes of average ,standard deviation and difference between each year.

```
library(ggplot2)
wordsc=read.csv('/Users/xuxinpin/Desktop/cluster1.csv',row.names = 1)
library(reshape2)
data=t(wordsc)
gg <- melt(data,id.vars="phrase")
ggplot(gg,aes(x=Var2,y=value,fill=factor(Var1)))+
  geom_bar(stat="identity",position="dodge")+
  scale_fill_discrete(name="Var1",
                      breaks=c("avg", "stdev", "Q"),
                      labels=c("avg", "stdev", "Q"))+
  xlab("phrase")+ylab("value")+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

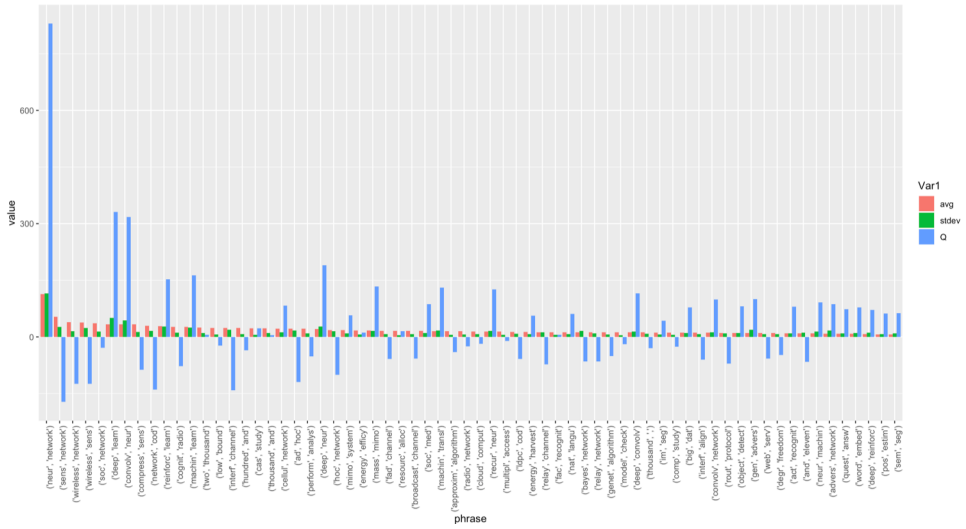


Figure 10: Cluster 1

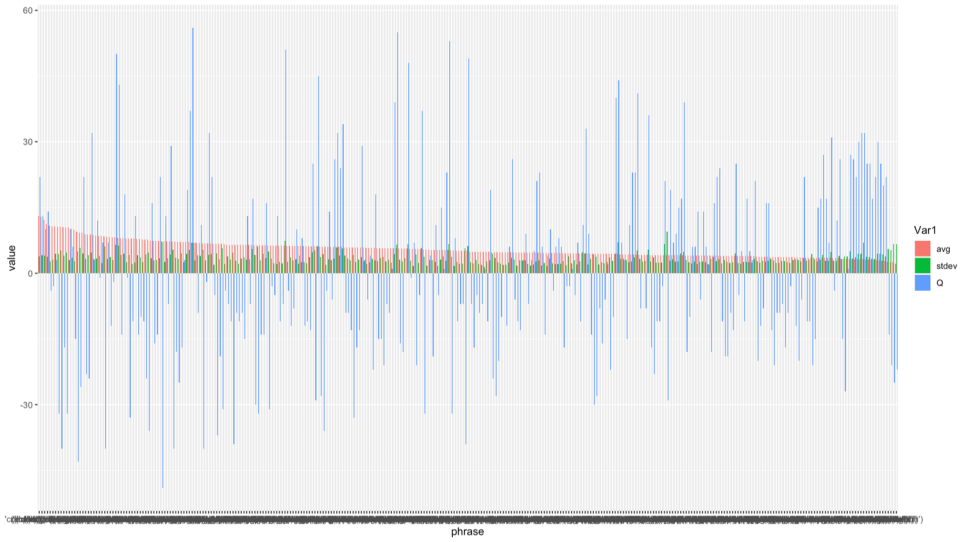


Figure 11: Cluster 2

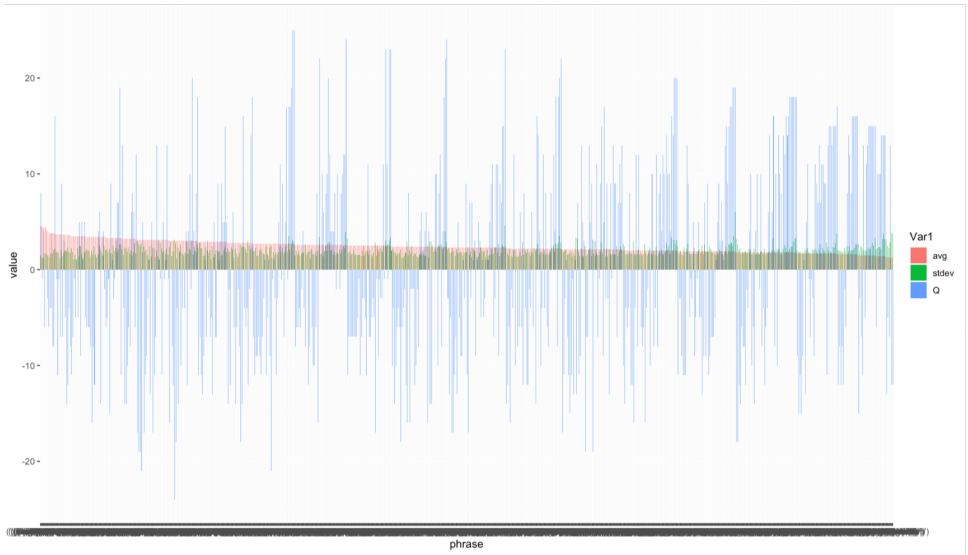


Figure 12: Cluster 3

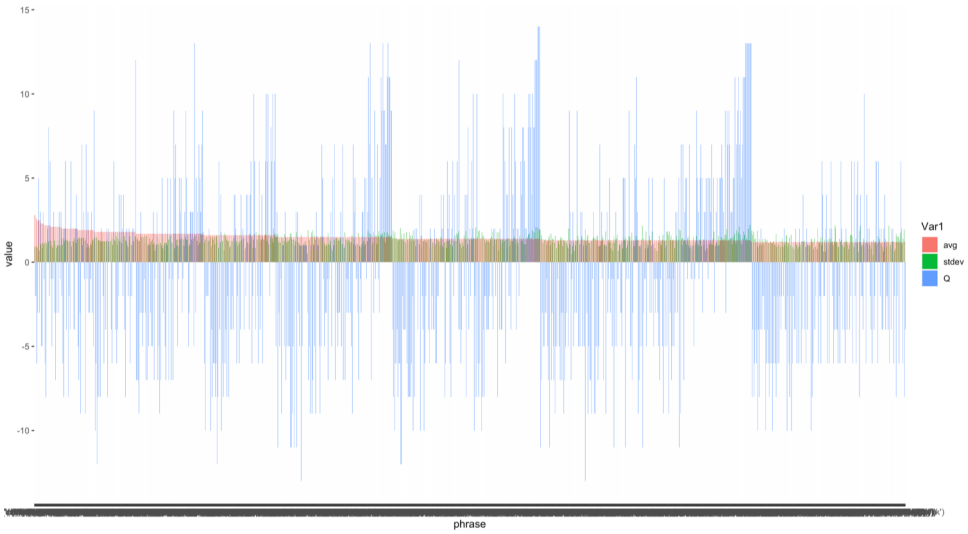


Figure 13: Cluster 4

3.2.4 Analysis of clustering data

According to the results of classified clusters, we can easily identify the trend of the frequency of a topic appearing in a research over ten years. Cluster 1 contains high average and standard deviation data, these data represent that the topic is innovative if the Q is positive, and obsolescent if Q is negative. The topics in cluster 2 are some popular topics and still in change, and cluster 3 represent some unstable and minority topic. The topics in cluster 4 are some useless and minority topics. The figure below shows the distribution of each clusters.

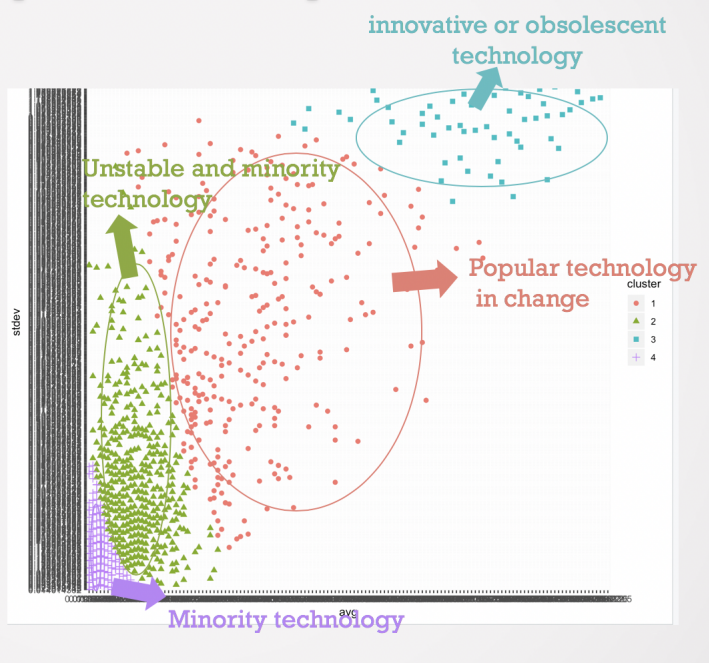


Figure 14: Distribution of Clusters

3.3 Regression

3.3.1 Choose Category

According to the results of data classification, it is clear that there are several obvious large categories, such us the category about neural network, the category about distributed systems. After considering the frequency and correlation, a set of representative categories are been chosen to do regression analysis.

In order to show the changes about the research direction of each category, we use words rather than phrases. There are the basic 7 categories:

- Category 1: Learning, Machine, Neural, Network, Deep, Convolutional, Recurrent
- Category 2: Wireless, Sensor, Networks, Radio, Cognitive, Ad, Hoc
- Category 3: Channel, Fading, Interface, Broadcast, Capacity, Gaussion
- Category 4: Systems, Distributed, Communication, Networks, Control, MIMO
- Category 5: Resource, Power, Allocation
- Category 6: Workshop, International, Proceeding
- Category 7: Big, Data

3.3.2 Prediction

In the beginning, we use linear regression and predict function to predict development trends of the words, and predict their values in the confidence interval of 0.95 for the next three years.

Listing 3: predict2019.R

```
countdata <- as.data.frame(read.csv("/Users/lipeiyu/Desktop/words.csv"))
kmdata <- as.matrix(countdata[1:24382,])
count <- 1
year <- c(2009:2018)
nextyear <- data.frame(year=2019)
while(count<24382){
  number <- as.vector(kmdata[count,2:11])
  lm.reg <- lm(number~year)
  result <- predict(lm.reg,nextyear,interval = "prediction",level = 0.95)
```

```
  rowname <- as.vector(kmdata[count,1])
  write.table(result, file = "/Users/lipeiyu/Desktop/predict[U+FFFD]2019.csv",sep =",",
  row.names = rowname,append = TRUE, col.names =FALSE)
  count=count+1
}
```

Then we draw the prediction line for the seven categories. The results are in Figure 15, Figure 16, and Figure 17.

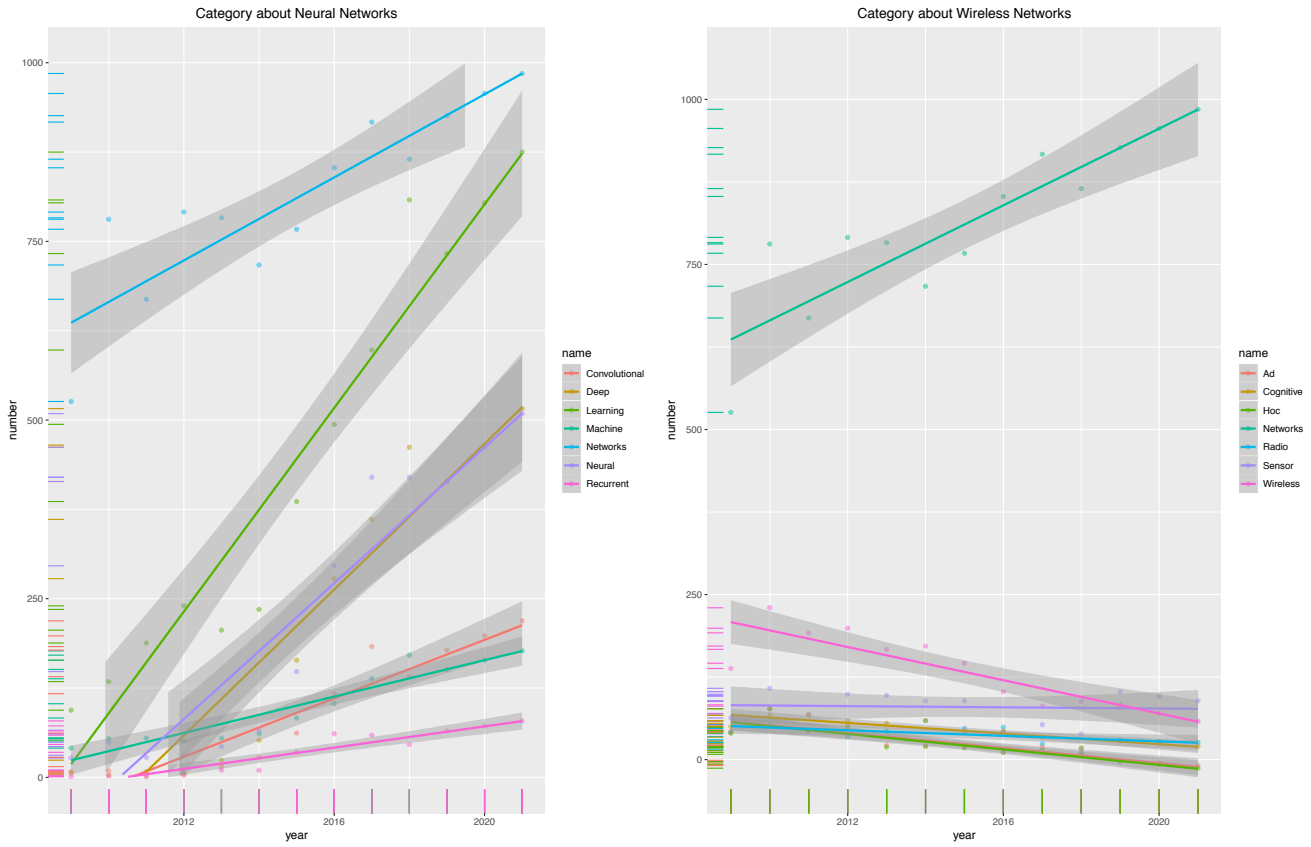


Figure 15: Prediction results of category 1 and 2

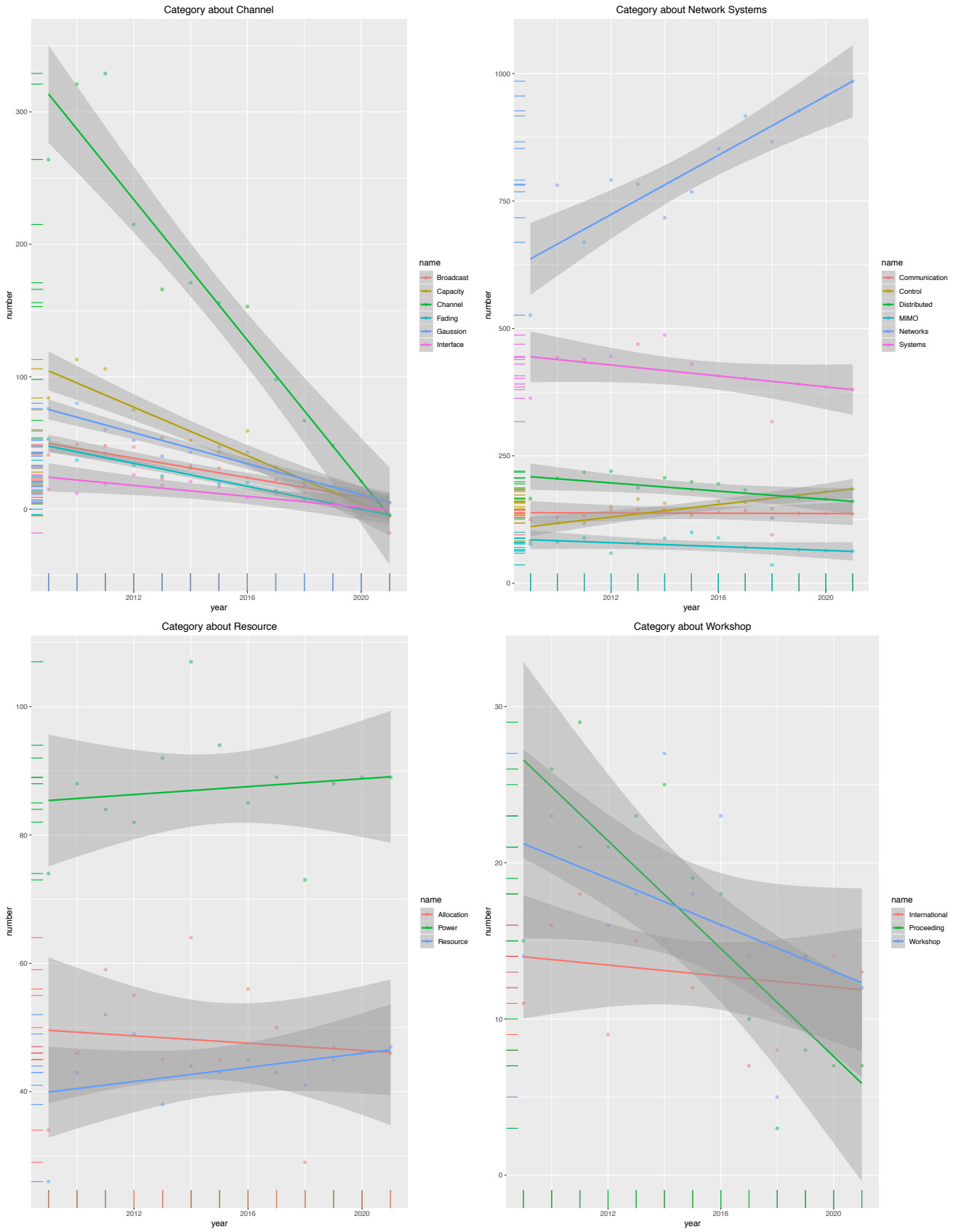


Figure 16: Prediction results of category 3 to 6

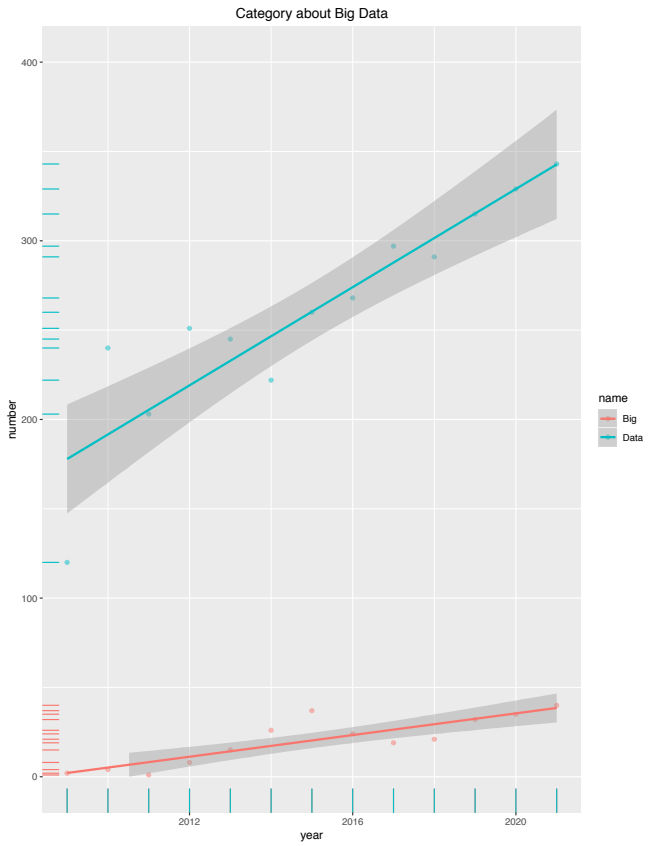


Figure 17: Prediction results of category 7

3.3.3 Regression with R

However, using linear regression does not reflect the connection between different technologies, and will lose some important information. For example, we will ignore that some technologies have been popular for some time. Since the attributes of the dataset are a bit small, in order to clearly show the trends of various technologies over the past decade, we have chosen the simplest fit curve.

As the codes show below, for each category, we adjust the value of ylim and generate the curve by geom_smooth().

Listing 4: loess7.R

```
library("ggplot2")
loess7 <- read.csv("/Users/lipeiyu/Desktop/loess7.csv")
png("/Users/lipeiyu/Desktop/classified1.png")
ggplot(loess7,aes(x = year, y = number, shape = name))+geom_point(colour="black",size=3.5)+geom_point(c
+scale_size_area()+geom_smooth(colour="black")+ggtitle("Category about Big Data")
+theme(plot.title = element_text(hjust = 0.5))+geom_rug()
dev.off()
```

The results of loess regression about the 7 categories are in Figure 18 and Figure 19

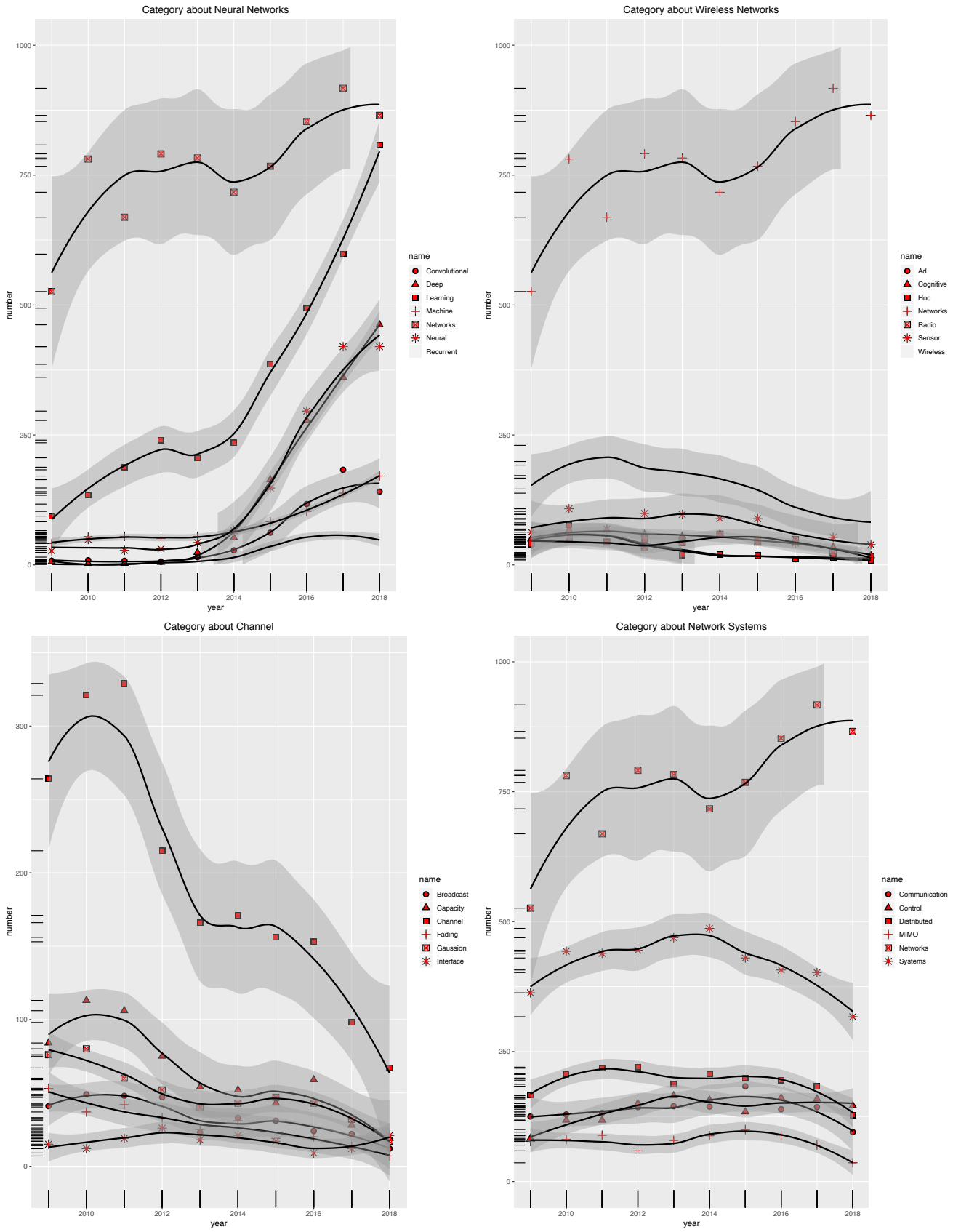


Figure 18: Loess regression results of category 1 to 4

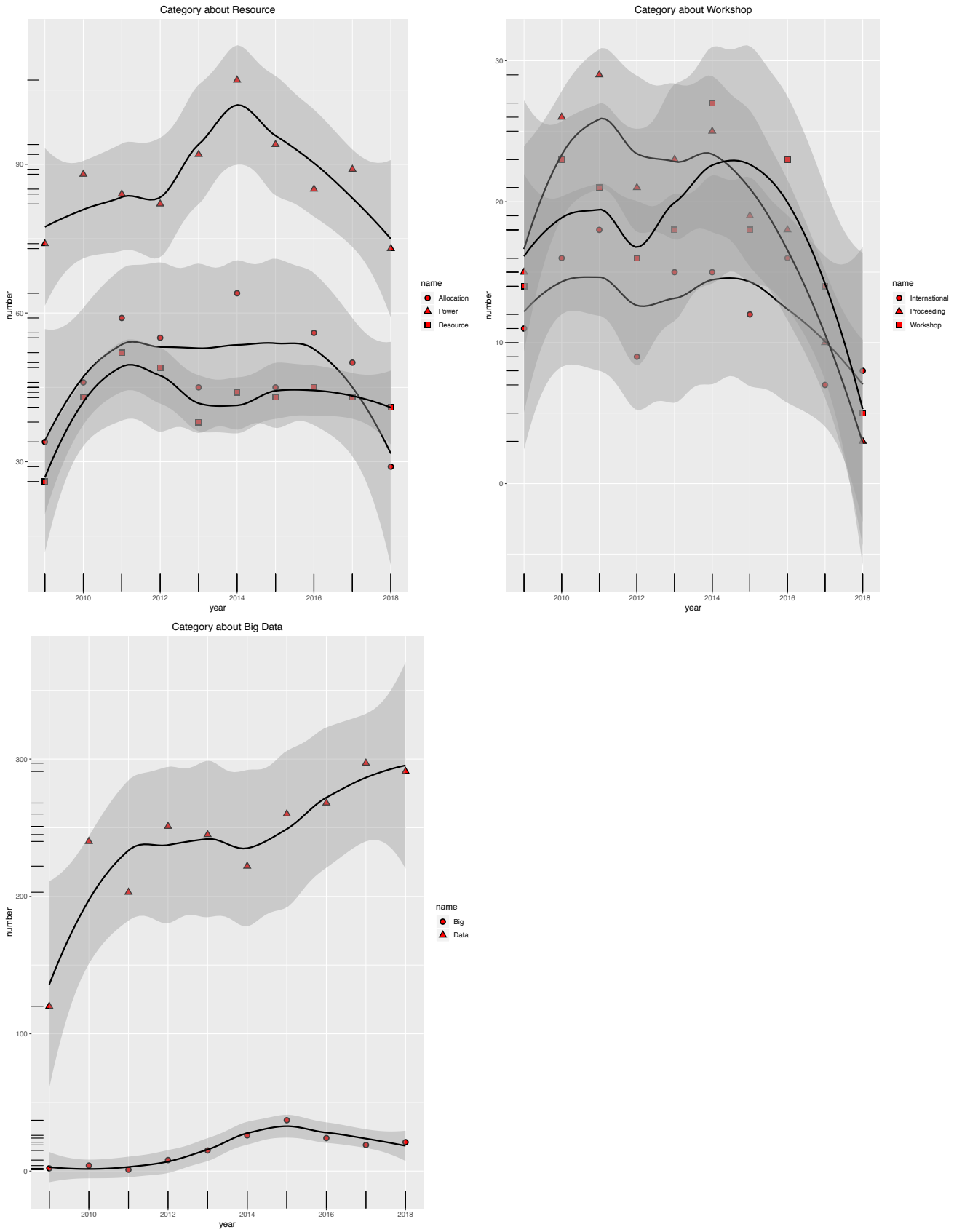


Figure 19: Loess regression result of category 5 to 7

As can be seen from the above five figures, research topics related to the networks in the past decade have always been popular. However, the research heat of wireless networks and network systems is

declining year by year, and the research heat for neural networks, deep learning and machine learning has risen rapidly since 2013. The attention for big data field is gradually rising.

In addition, it is clear that the research heat of power allocation and resource allocation is rising from 2009 to 2012, and then is declining from 2015 to 2018. The research heat of proceeding international workshop is fluctuating during 2009 to 2014 and then declining year by year.

These trends indicate that the research heat for category 1, such as neural networks, deep learning, machine learning and convolutional networks is going to rise in the future. And machine learning and deep learning should be the main aspects of research. The focus of channel research maybe become the gaussion and the wireless networks research category is still focus on wireless network and sensor network. Futhermore, the research heat of network systems is going to be stable.

4 Data Visualization

4.1 Word Cloud In Ten Years

To visually show the distribution of the number of topics over the past decade, word cloud(see Listing 5) is a great tool to achieve this goal. The data is read from the generated csv file, which contains the bigrams and the word frequency of the phrase.

Listing 5: WordCloud.ipynb

```
df = pd.read_csv('data/phrase/wordcloud.csv')
d = {}
for a, x in df.values:
    d[a] = x
wordcloud = WordCloud(background_color ='white')
wordcloud.generate_from_frequencies(frequencies=d)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

The graph(see Figure 20) of word cloud result shows that neural networks, sensor networks, wireless networks, etc. are very popular paper themes in the past decade.

Figure 20: WordCloud

4.2 Top 10 Topics In Each Year

In order to show the difference between the hot topics of 2009-2018, horizontal histograms(see Listing 6) were used to show the most frequently occurring topics in 2009, 2013, 2016 and 2018. Because the data with an interval of 1-2 years is not much different, only the data differences between 3 and 4 years are reflected in the report.

Listing 6: Barh.ipynb

```
plt.barh(y, x, align='center', color=(0.2, 0.4, 0.6, 0.6))
plt.ylabel('%s Topic' % year)
plt.xlabel('Frequency')
my_x_ticks = np.arange(0, 350, 50)
plt.xticks(my_x_ticks)
```

The result(see Figure 21) shows that, in 2009, the Sensor Network was the most popular topic of thesis, and even in 2013, it was still the most frequently studied subject. But in 2016 and 2018, neural networks are the hottest topic, and the number of papers on this topic is much larger than the number of sensor networks, even topics related to artificial intelligence are welcomed by researchers, such as deep learning, reinforcement learning, etc.

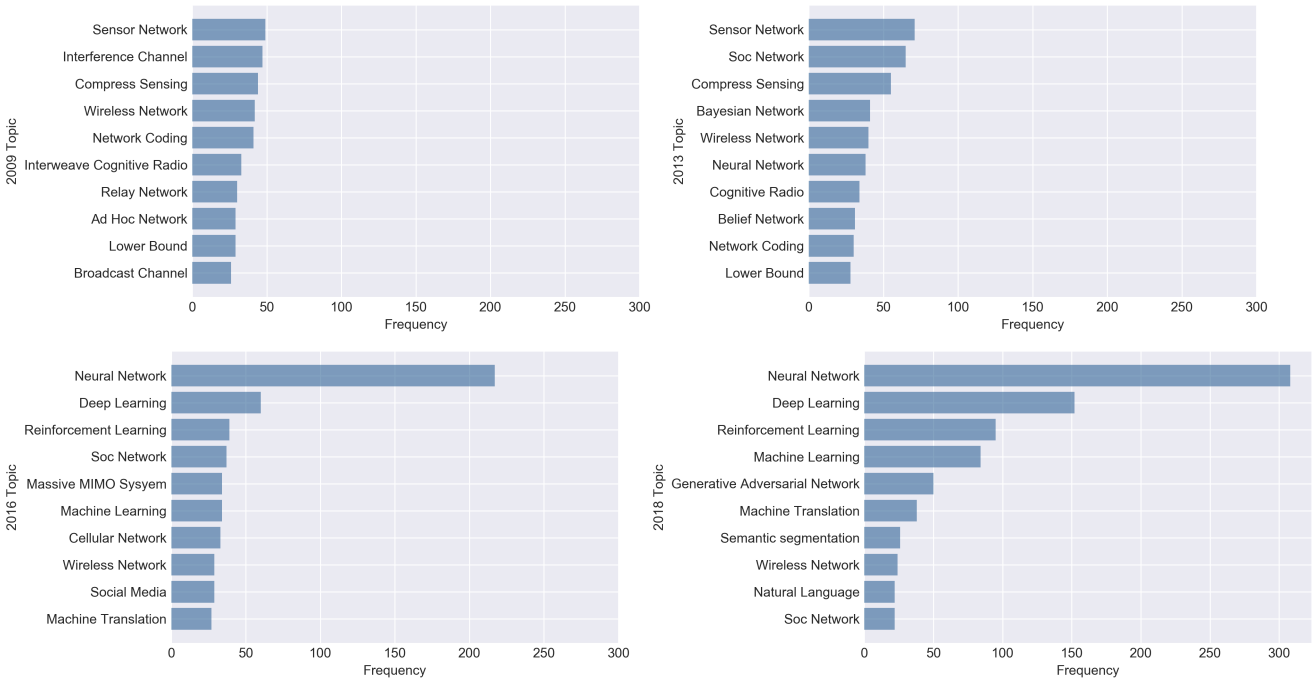


Figure 21: Topics in 2009,2013,2016,2018

4.3 Top 12 Topics Changed In Ten Years

In order to see the changes in the top 12 topics in the decade, choose to display data in small multiple line charts(see Listing 7).

Listing 7: lineChart.ipynb

```
num = 0
for v in topics:
    num += 1
    '''row = 3,column = 4,num is small chart position in the whole chart'''
    plt.subplot(3, 4, num)
    plt.plot(x, df[v], marker='', color=palette(num), linewidth=1.9, alpha=0.9, label=v)
    my_y_ticks = np.arange(10, 100, 20)
    plt.yticks(my_y_ticks)
    if num in range(9):
        plt.tick_params(labelbottom='off')
    if num not in [1, 5, 9]:
        plt.tick_params(labelleft='off')
    plt.title(v, loc='left', fontsize=7, fontweight=0, color=palette(num))
```

The results(see Figure 22) of the visualization show that in 2009-2018, only neural networks, reinforcement learning and machine learning is on the rise, and other topics are almost declining. It can be seen that the research of artificial intelligence is getting more and more attention.

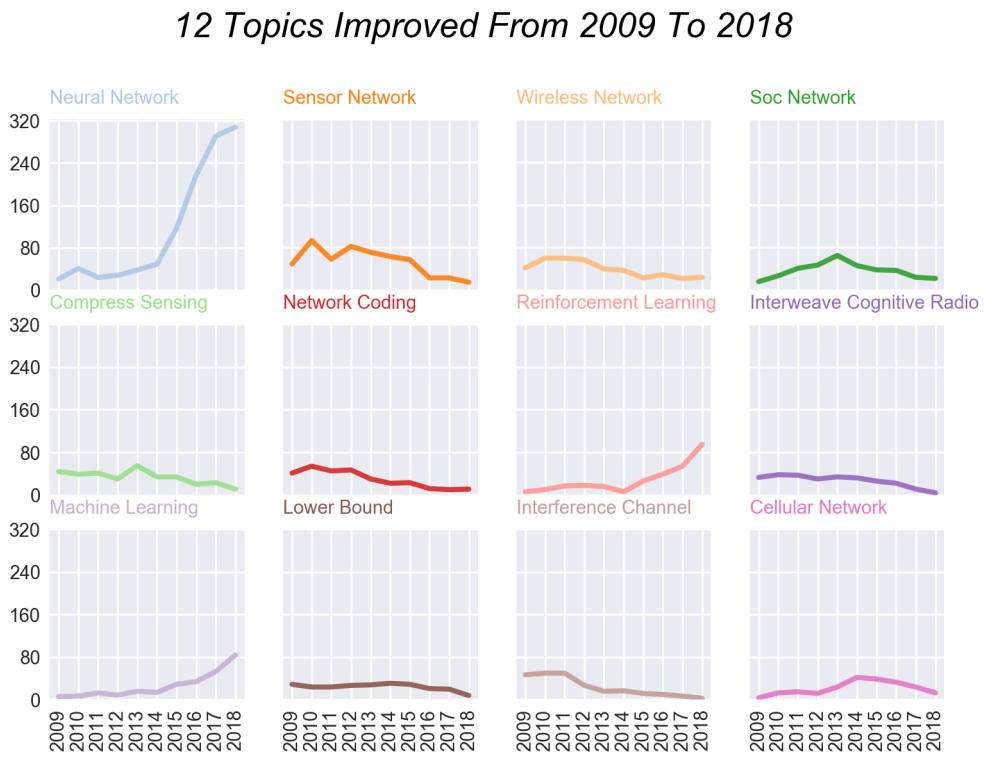


Figure 22: Topics Improved 2009-2018

4.4 Topic About AI Changed In Ten Years

This section is about the topics related to artificial intelligence. The method of analysis is to observe the frequency of occurrence of these topics in 2009-2018, and use Stacked bar chart(see Listing 8) to realize.

Listing 8: StackedBarChart.ipynb

```
'''Ten year'''
N = 10
Neural_Network = df['Neural_Network']
...
ind = np.arange(N)
width = 0.35
p1 = plt.bar(ind, Neural_Network, width, color=palette(0))
p2 = plt.bar(ind, Deep_Learning, width, color = palette(1),
             bottom=Neural_Network)
...
plt.legend((p1[0], p2[0], p3[0], p4[0], p5[0]), ('Neural_Network', 'Deep_Learning',
'Reinforcement_Learning', 'Machine_Learning', 'Machine_Translation'))
```

In the visualization results graph(see Figure 23), the number of AI-related papers is on the rise and has grown very rapidly since 2014. The neural network always occupies a very large proportion, showing the importance of neural networks to artificial intelligence. The website did not publish papers about deep learning before 2013.

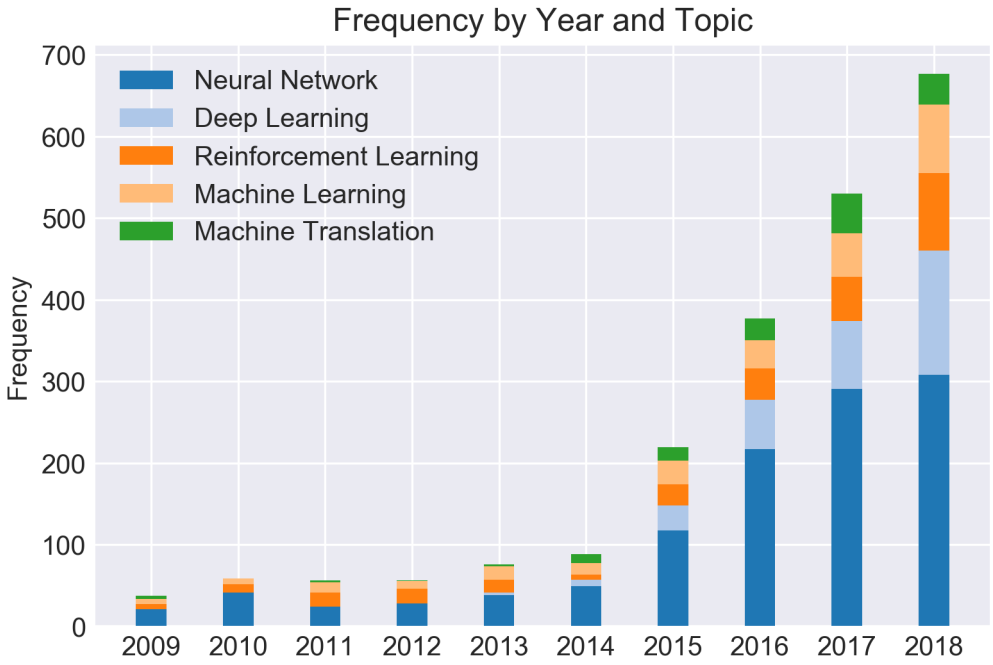


Figure 23: Topics About AI

Conclusions

In this project, we analysed the existed CS technology data to predict the trend of the future. In detail, we realize this goal in three ways.

- Analyse the data directly to get some rules or result.
- Use the clustering model to classify the technology and in terms of the clusters to analyse the holistic rules instead of a single word.
- Use the regression model to predict the future trend of the technology.

Future Work

Using the clustering, we could only classify the technologies in the frequency relation. However, we also want to get to classify the word by their meaning. In the future, we will try to create training set and using machine learning to classify the word in different area, such as data engineering, artificial intelligence and so on. In this way, we could get better and clearer analysis about the trend of technology in the future.