Statistical Methods of Language Technology

{alacam,yimam}@informatik.uni-hamburg.de

Summerterm 2021

Assignment 8 Due date:

Problem 8.1 Latent Semantic Analysis

a)

```
library('tm')

training = DirSource("./training/",pattern="c.*")
cT=Corpus(training)
dtmT = DocumentTermMatrix(cT)
m = as.matrix(dtmT)

plotLatentVariables<-function(val,files,showFilenames=F){
        X11()
        f = substr(files$Names,0,2);
        plot(val,pch=match(f,unique(f)),col=match(f,unique(f)))
        legend("topleft", legend = unique(f),
        pch=1:length(unique(f)),col=1:length(unique(f) ))
        if(showFilenames){
                text(val, labels = files$Names)
        }
}

k = 2;
r = svd(m,nu=2,nv=2);
u = r$u
s = diag(r$d[1:k])
v = r$v

plotLatentVariables(u,training)
```

Listing 1: Template for Problem 8.1.a)

Figure 1 shows the resulting plot.

Given the topics:

- E: Non-fiction(Skill and Hobbys)
- H: Non-fiction(Government and House organs)
- K: Fiction(General)
- N: Fiction(Adventure)
- P: Fiction(Romance)

One can see that fiction and non-fiction are separated from each other. The subtopics however overlap strongly because of similar words.

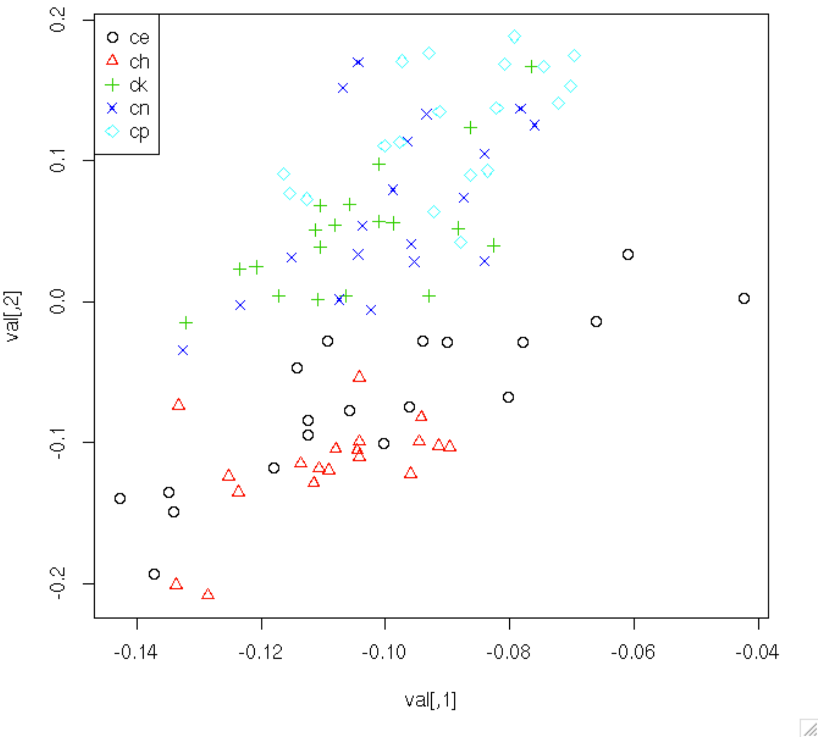

Figure 1: Latent document vectors of training data

b)

```
transformQueryMatrix<-function(dtmQ, dtmT){
        query = matrix(c(0),nrow = dim(dtmQ)[1],ncol=dim( dtmT)[2])
        rownames(query)=rownames(dtmQ)
        colnames(query)=colnames(dtmT)
        columns = colnames(query);
        idx = which(colnames(dtmQ)%in%columns)
        query[,which(columns %in%colnames(dtmQ))] = as.matrix(dtmQ[,idx])
        return(query)
}

pointsLatentVariables<-function(val,files,showFilenames=F){
        f = substr(files$Names,0,2);
        points(val,pch=match(f,unique(f)),col=match(f,unique(f)),lwd=4,bg="black")
        if(showFilenames){
                text(val, labels = files$names)
        }
}

query = DirSource("./test/",pattern="c.*")
cQ=Corpus(query)
dtmQ = DocumentTermMatrix(cQ)
dtmTQ=transformQueryMatrix(dtmQ,dtmT)
mq=as.matrix(dtmTQ)

mqk = mq %*% v %*% solve(s)
pointsLatentVariables(mqk,query)
```

Listing 2: Template for Problem 8.1.b)

In Figure 2 we see that the topic distribution of test- and training data are very similar.

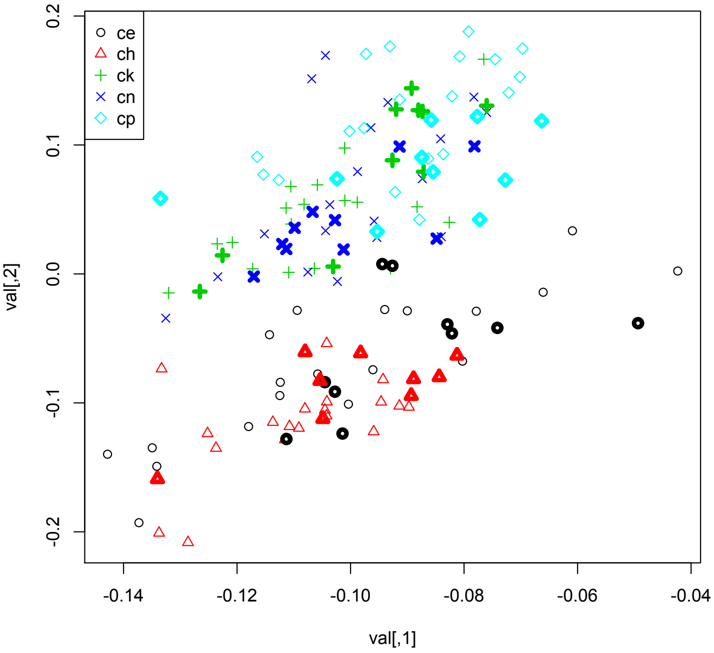

Figure 2: Latent document vectors distribution of training- and test data

c)

```
cT = tm_map(cT, removeWords, stopwords("english"))
cQ = tm_map(cQ, removeWords, stopwords("english"))
dtmT_clean = DocumentTermMatrix(cT)
dtmQ_clean = DocumentTermMatrix(cQ)
dtmTQ=transformQueryMatrix(dtmQ_clean,dtmT_clean)

mt=as.matrix(dtmT_clean)

k = 2;
r = svd(mt,nu=k,nv=k);
u = r$u
s = diag(r$d[1:k])
v = r$v

mq=as.matrix(dtmTQ)
plotLatentVariables(u,training)
mqk = mq %*% v %*% solve(s)
pointsLatentVariables(mqk,query)
```

In Figure 3 we see that the non-fictional topic E (skill and hobbies) and the fictional topics overlap more after the removal of the stopwords.

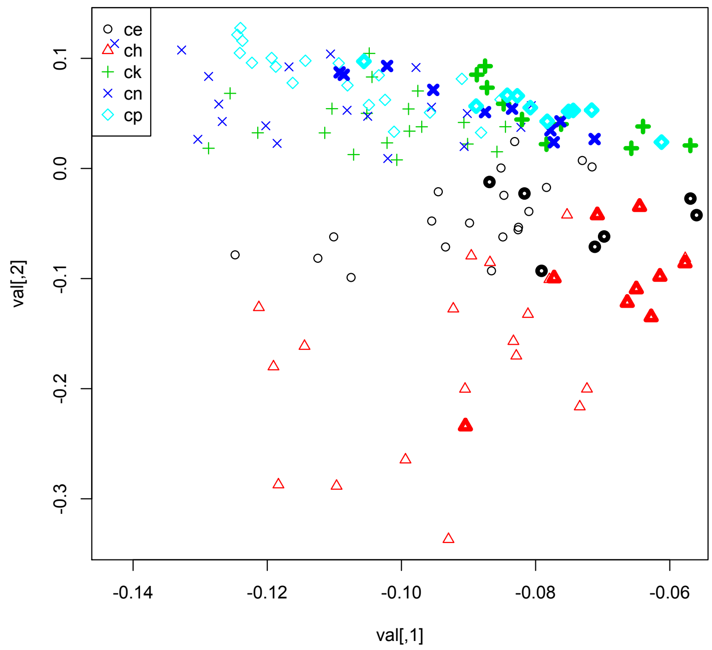

Figure 3: Latent document vectors distribution of training and test data without stopwords

Problem 8.2 LDA

a) First convert the training data into the format used by GibbsLDA++ (first line: number of documents, then one document per line).

Then train LDA model with GibbsLDA++:

```
lda -est -beta 0.01 -ntopics 1 -niters 200 -twords 30 -dfile lda-input-train
```

Since we train with one topic only, all documents and words are assigned the same (single) topic. The `twords` file contains word and their probabilities (frequency/number of all words) that can also be interpreted as a unigram model.

b) Train the LDA model with GibbsLDA++ with settings according to the task:

```
lda -est -beta 0.01 -ntopics 5 -niters 200 -savestep 10 -twords 100 -dfile lda-input-train
```

Calculating the overlap should produce results like this:

T	model-010	model-020	model-030	model-040	model-050	model-060	model-070	model-080	model-090	model-100
1	0	0.83	0.92	0.90	0.94	0.94	0.94	0.96	0.94	0.92
2	0	0.73	0.90	0.86	0.87	0.89	0.90	0.90	0.90	0.90
3	0	0.77	0.87	0.91	0.93	0.93	0.91	0.92	0.92	0.94
4	0	0.69	0.84	0.84	0.89	0.91	0.98	0.94	0.90	0.96
5	0	0.84	0.89	0.92	0.92	0.91	0.90	0.89	0.91	0.85

T	model-110	model-120	model-130	model-140	model-150	model-160	model-170	model-180	model-190	model-200
1	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.98	0.98	0.98
2	0.94	0.90	0.85	0.90	0.92	0.88	0.91	0.93	0.91	0.89
3	0.91	0.93	0.93	0.96	0.94	0.95	0.95	0.94	0.93	0.94
4	0.92	0.94	0.95	0.92	0.93	0.92	0.94	0.89	0.93	0.92
5	0.88	0.93	0.93	0.90	0.89	0.92	0.93	0.92	0.94	0.94

We can see that the overlap of the words for each topic converges fast. The changes that happen after iteration 110 are expected, as Gibbs Sampling is a random process.

Results, obtained analogously for training data with removed stopwords show similar properties.

c) LDA inference is the application of the model trained in b) to unseen data. Thus, resulting topics can be mapped easily to the topics from b).

However, without removing stopwords, all topics share their top 30 words and are therefore look very similar to undistinguishable.

With stopwords removed, the distinction is clear and topics usually contain words from coherent domains.