AML_HW7 - Part 1

Vishal Dalmiya (Dalmiya2); Himanshu Shah (Hs8); Deepak Nagarajan (deepakn2)

Mar 23, 2018

EM Topic models The UCI Machine Learning dataset repository hosts several datasets recording word counts for documents here. You will use the NIPS dataset. You will find (a) a table of word counts per document and (b) a vocabulary list for this dataset at the link. You must implement the multinomial mixture of topics model, lectured in class. For this problem, you should write the clustering code yourself (i.e. not use a package for clustering).

Cluster this to 30 topics, using a simple mixture of multinomial topic model, as lectured in class.

Produce a graph showing, for each topic, the probability with which the topic is selected.

Produce a table showing, for each topic, the 10 words with the highest probability for that topic

```
library(readr)
library(matrixStats)
docword_nips <- read_delim(
  "docword.nips.txt",
  " ",
  escape_double = FALSE,
  col_names = FALSE,
  trim_ws = TRUE
)

colnames(docword_nips) = c("docID", "wordID", "count")

# Max number of docs
(D = max(docword_nips$docID))
```

```
## [1] 1500
```

```
# Max number of unique words
(V = max(docword_nips$wordID))
```

```
## [1] 12419
```

```
# number of topics
T = 30

# Word frequency for various documents
X = matrix(rep(0, V * D), nrow = V, ncol = D)

# Initialize X from docword_nips
for (i in 1:D)
{
  temp = docword_nips[docword_nips$docID == i, ]
```

```r
  for (j in 1:nrow(temp))
  {
    entry = as.numeric(temp[j, ])
    X[entry[2], i] = entry[3]
  }
}

# Word probabilities of various topics
P = matrix(nrow = V, ncol = T)

# Initialize P
# P initialize - 4-7
Y = t(X)
samp = Y[sample(nrow(Y), T, replace = FALSE),]
tsamp = t(samp)
csum = apply(tsamp, 2, sum)
P = t(t(tsamp) / csum)

for (j in 1:T)
{
  idx = P[, j] == 0
  if (sum(idx) > 0)
  {
    P[, j] = P[, j] * 0.95
    P[idx, j] = 0.05 / sum(idx)
    P[, j] = P[, j] / sum(P[, j])
  }
}

# Weights
W = matrix(nrow = D, ncol = T)

oldW = matrix(rep(0, D * T), nrow = D, ncol = T)

# Weightage of each topic
pi = rep((1 / T), T)

# Run till convergence
iter = 1
while (1)
{
  ############# E Step #############
  W = t(t(t(X) %*% log(P)) + log(pi))
  for (i in 1:D)
  {
    max_numer = max(W[i, ])
    W[i, ] = W[i, ] - max_numer

    denom = logSumExp(W[i, ])
    W[i, ] = W[i, ] - denom
    W[i, ] = exp(W[i, ])
  }
```

```r
  ############# M Step #############

  # sum of all words in all documents
  # Dim : 1 X D
  # W = D X T
  sum_words = colSums(X)

  # compute P
  # X : V X D
  # W : D X T
  # numer : V X T
  numer = X %*% W

  # sum_words: 1 X D
  # W : D X T
  # den : 1 X T
  den = as.numeric(sum_words %*% W)

  for (j in 1:T)
  {
    P[, j] = (numer[, j]) / (den[j])
    idx = P[, j] == 0
    if (sum(idx) > 0)
    {
      P[, j] = P[, j] * 0.95
      P[idx, j] = 0.05 / sum(idx)
      P[, j] = P[, j] / sum(P[, j])
    }
  }


  # Compute pi
  (pi = colSums(W) / D)

  iter = iter + 1
  temp = max(abs(W - oldW))

  if (temp < 0.0001)
  {
    print(paste("Iteration # ",iter))
    print(paste("Treshold ",temp))
    break
  }
  oldW = W
}
```

```
## [1] "Iteration #  39"
## [1] "Treshold  0.000071327966464918"
```

```r
library(knitr)
library(readr)

# To display the table of top 10 words for each topic
```

```r
vocab_nips <- as.matrix(
  read_delim(
    "vocab.nips.txt",
    " ",
    escape_double = FALSE,
    col_names = FALSE,
    trim_ws = TRUE
  )
)

T = 30
m = matrix(rep(0, 10 * T), nrow = T, ncol = 10)

colnames(m) = (paste(rep("Word", 10), seq(1, 10, 1)))
rownames(m) = (paste(rep("Topic", 30), seq(1, 30, 1)))
for (j in 1:T) {
  temp_sort = sort.int(P[, j], decreasing = TRUE, index.return = TRUE)$ix[1:10]
  for (k in 1:10)
  {
    idx = temp_sort[k]
    m[j, k] = vocab_nips[idx]
  }
}

kable(m[,1:8])
```

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8
Topic 1	model	network	algorithm	data	learning	parameter	tree	mean
Topic 2	model	distribution	field	point	gaussian	component	method	data
Topic 3	part	processing	handwriting	signal	speech	vii	visual	control
Topic 4	function	algorithm	genetic	basis	population	model	wavelet	problem
Topic 5	model	algorithm	data	recognition	set	problem	network	word
Topic 6	model	data	function	set	algorithm	vector	training	learning
Topic 7	network	neural	processor	system	instruction	block	data	weight
Topic 8	network	unit	learning	input	training	set	neural	output
Topic 9	orientation	model	ocular	dominance	map	pattern	eye	correlation
Topic 10	network	pattern	input	unit	stress	filter	training	output
Topic 11	model	network	cell	input	unit	neuron	visual	system
Topic 12	model	system	learning	movement	network	field	control	motor
Topic 13	model	image	network	object	images	system	recognition	set
Topic 14	neuron	model	cell	input	network	synaptic	spike	firing
Topic 15	neuron	circuit	model	network	input	system	neural	output
Topic 16	network	data	set	algorithm	vector	model	input	neural
Topic 17	network	model	learning	function	neural	data	system	input
Topic 18	network	unit	input	hidden	learning	output	net	componen
Topic 19	learning	action	function	algorithm	policy	reinforcement	problem	control
Topic 20	network	algorithm	model	learning	function	data	neural	input
Topic 21	function	network	neural	input	weight	bound	learning	result
Topic 22	chip	network	neuron	neural	weight	synapse	analog	performan
Topic 23	algorithm	function	learning	data	set	problem	point	result
Topic 24	error	network	learning	training	weight	function	set	generalizat
Topic 25	network	input	circuit	function	learning	output	neural	weight
Topic 26	network	training	word	speech	recognition	classifier	set	neural

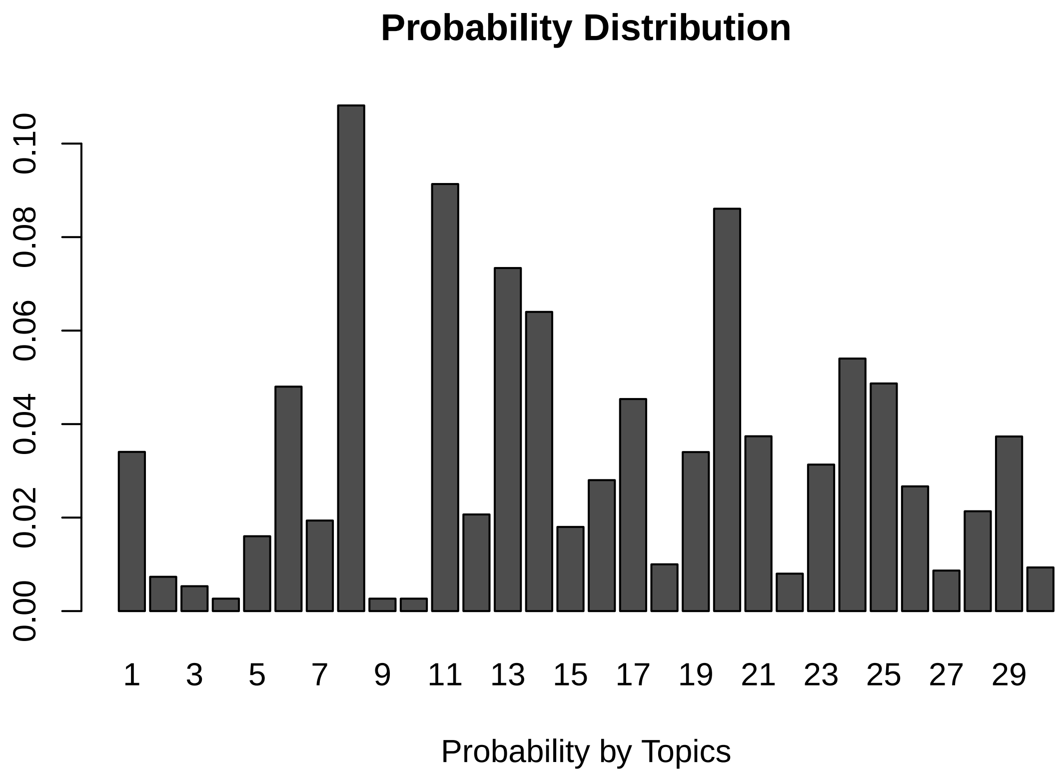
	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8
Topic 27	network	learning	input	system	weight	classifier	node	neural
Topic 28	network	neural	system	data	input	memory	pattern	output
Topic 29	model	data	network	speech	system	training	hmm	parameter
Topic 30	algorithm	cell	class	classification	system	network	pattern	image

```r
kable(m[,9:10])
```

	Word 9	Word 10
Topic 1	distribution	variables
Topic 2	parameter	function
Topic 3	navigation	planning
Topic 4	number	vector
Topic 5	system	training
Topic 6	classifier	problem
Topic 7	chip	algorithm
Topic 8	weight	hidden
Topic 9	cortex	cortical
Topic 10	syllable	learning
Topic 11	pattern	direction
Topic 12	dynamic	robot
Topic 13	neural	point
Topic 14	neural	function
Topic 15	spike	current
Topic 16	training	tangent
Topic 17	control	set
Topic 18	model	set
Topic 19	system	step
Topic 20	set	error
Topic 21	algorithm	set
Topic 22	input	current
Topic 23	method	distribution
Topic 24	input	parameter
Topic 25	analog	chip
Topic 26	error	system
Topic 27	set	unit
Topic 28	set	algorithm
Topic 29	algorithm	vector
Topic 30	vector	learning

```r
# To display the graph showing, for each topic, the probability with which the topic is selected.

pd = matrix(pi, nrow = 1)
colnames(pd) = seq(1, 30, 1)
barplot(pd, main = "Probability Distribution", xlab = "Probability by Topics")
```



- The EM model converges at about 39 iterations
- From the above bar plot of the probability distribution, the Topic #8 seems to be the most selected topic for this run.
- From the above table, it seems like Model and Network are the two most commonly used words across all topics, and there is some distinction between the topics that can be seen easily.