Data Science for Business
Lecture #4
Topic Modeling Example

Prof. Alan L. Montgomery

The University of Hong Kong & Carnegie Mellon University, Tepper School of Business

email: alanmontgomery@cmu.edu

Outline

Probabilistic Clustering with Topic Models

An example of Topics Models using the Iris Dataset

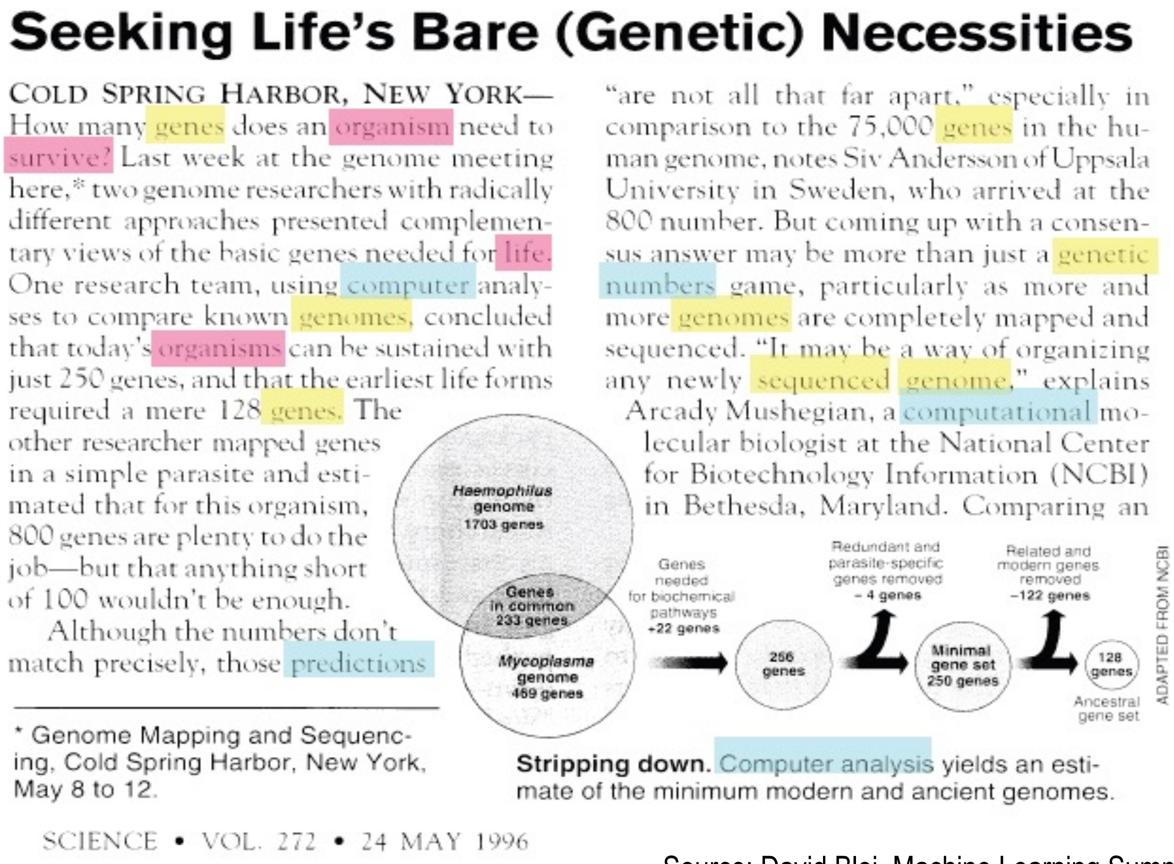
Probabilistic Clustering with Topic Models

Probabilistic Clustering

- K-Means gives one method for a divisive clustering technique, in which all observations are arranged into non-overlapping clusters
- Alternatively we could assign observations into overlapping clusters like with Fuzzy clustering
- Or we could probabilistically assign observations into clusters using probability models like Mixture of Gaussians
- We can also attach more structure to our models to make inferences both about generating process like with Topic Models
- A special type of Topic Models is Latent Dirchlet Allocation (LDA)

Intuition: Documents contain multiple topics

Example article from Science



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed − 4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

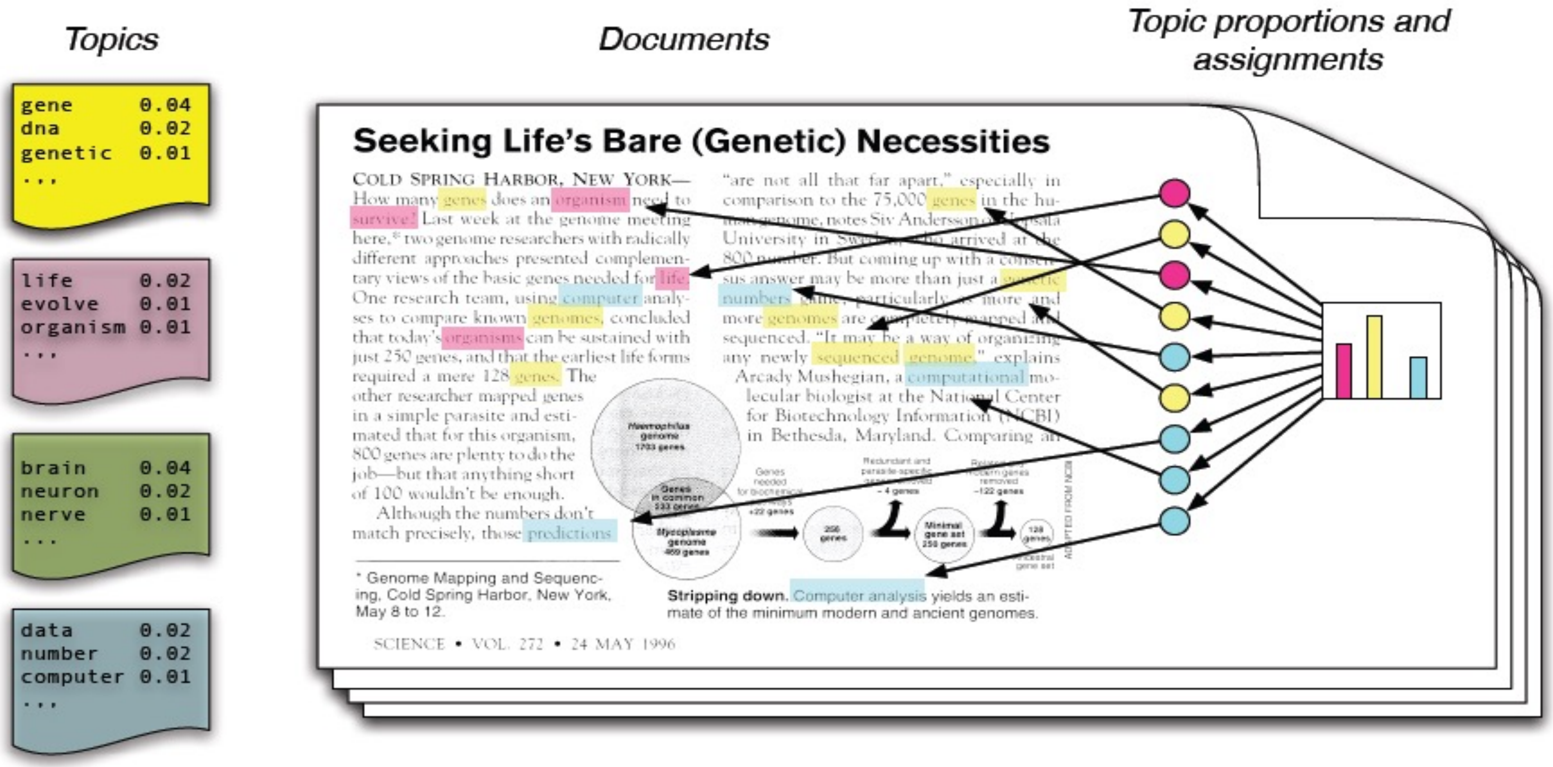
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

Source: David Blei, Machine Learning Summer School (2012)

Intuition: Documents contain multiple topics

Example article from Science and potential topics



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed − 4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

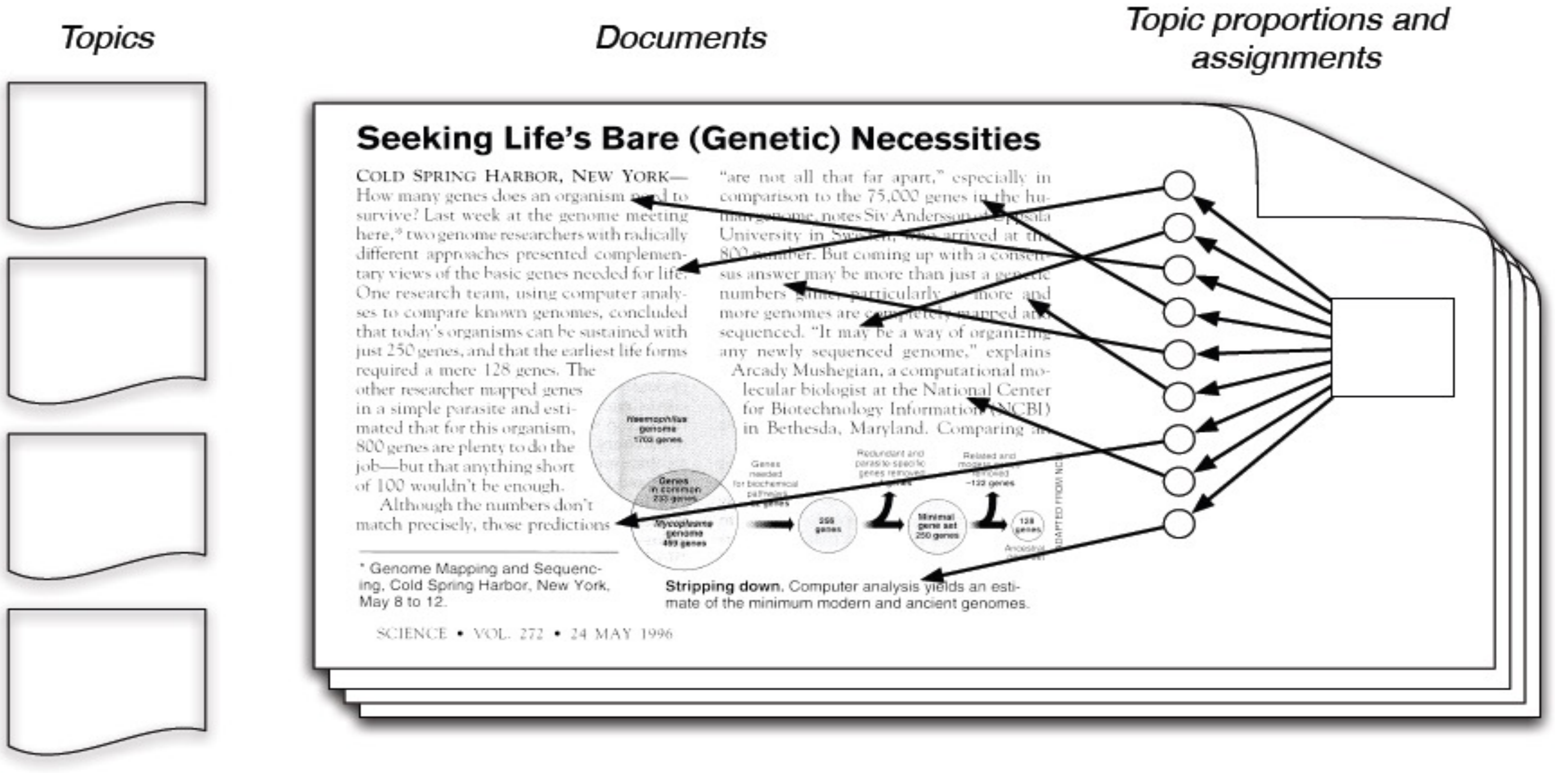
SCIENCE • VOL. 272 • 24 MAY 1996

- Each topic is a distribution over words
- Each document is a mixture over topics
- Each word is drawn from one of those topics

Source: David Blei, Machine Learning Summer School (2012)

Intuition: Documents contain multiple topics

Example article from Science and potential topics


Topics
Documents
Topic proportions and assignments
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 468 genes
Genes needed for biochemical pathways
256 genes
Redundant and parasite-specific genes removed
Minimal gene set 250 genes
Related and modern genes removed ~122 genes
128 genes
Ancestral
ADAPTED FROM NCBI

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

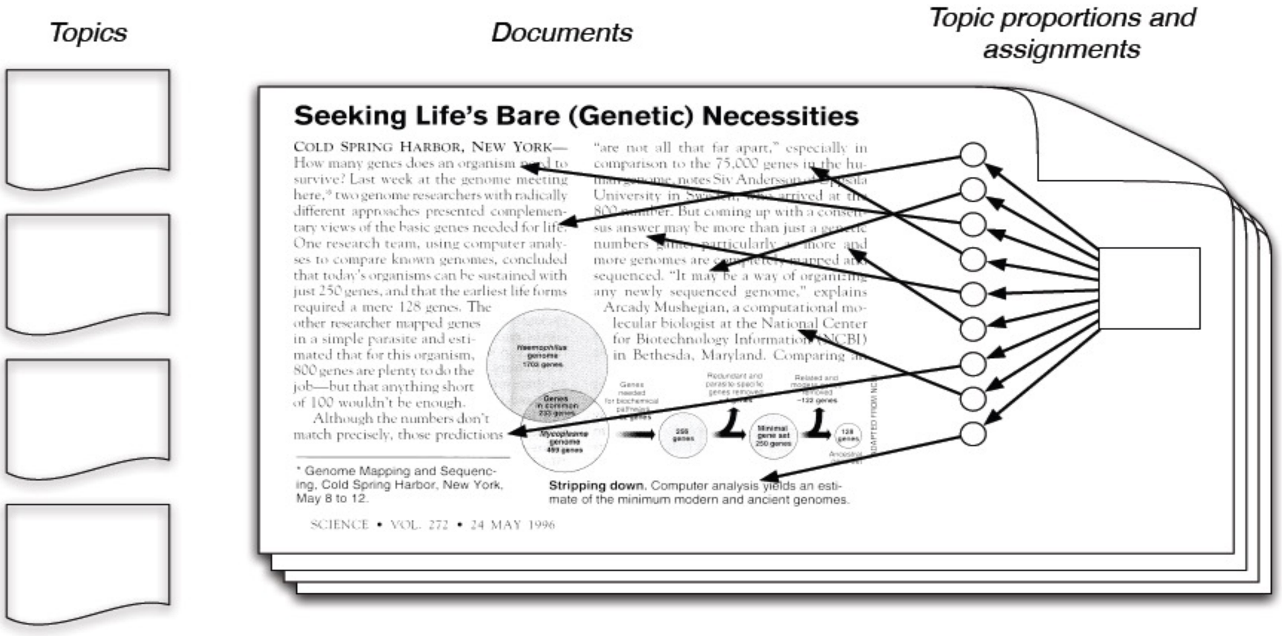
SCIENCE • VOL. 272 • 24 MAY 1996

- Problem is that we only observe the documents – not the topics
- In Machine Learning we think of the topics as *hidden variables* that we want to *infer*

Source: David Blei, Machine Learning Summer School (2012)

Intuition: Documents contain multiple topics

Example article from Science and potential topics



Topics
Documents
Topic proportions and assignments
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

The **observations** are the documents: $\mathbf{w_m}, m \in 1, M$
We need to infer the **model**, i.e the underlying topic structure, i.e. the topic assignments $z_{m,n}$, the topic θ_m, $m \in 1, M$ and word distributions φ_k, $k \in 1, K$

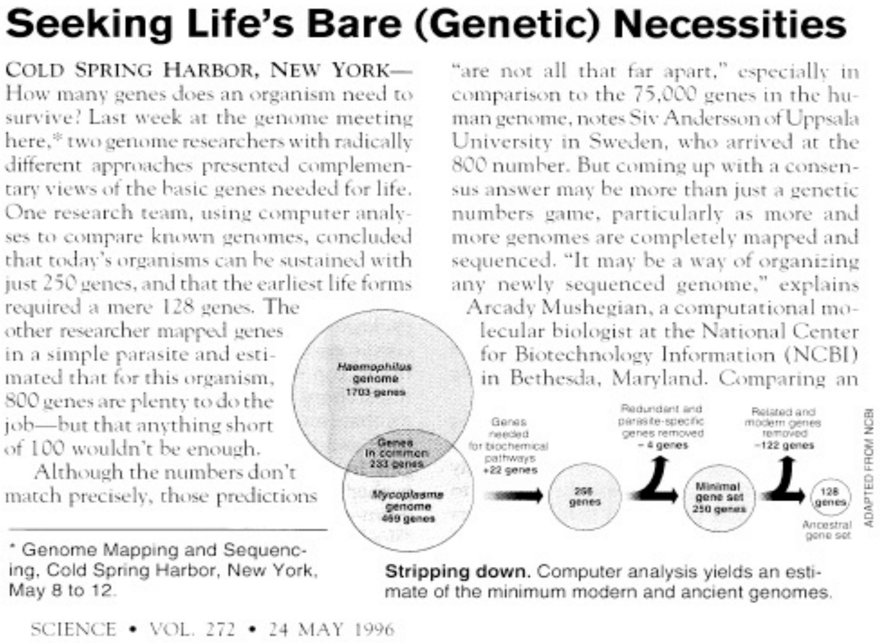
Priors:

$\theta \sim$ distribution with hyperparameter α
$\varphi \sim$ distribution with hyperparameter β

Source: David Blei, Machine Learning Summer School (2012)

Intuition: Documents contain multiple topics

Example article from Science and potential topics



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

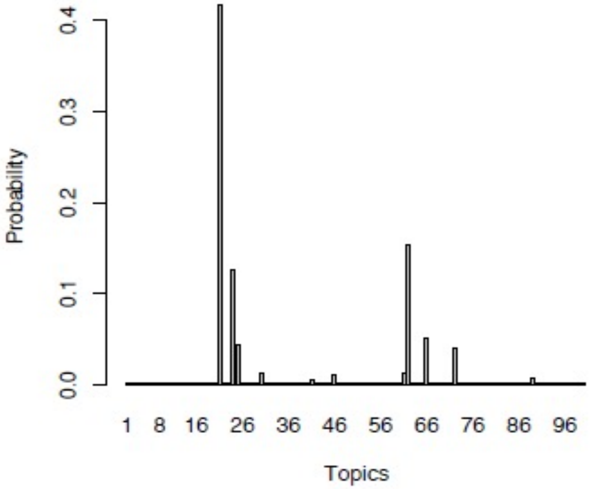
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed – 4 genes
Minimal gene set 250 genes
Related and modern genes removed –122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

[*] Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

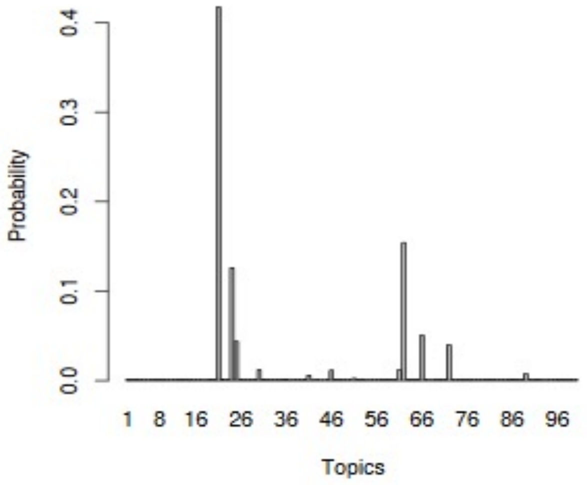


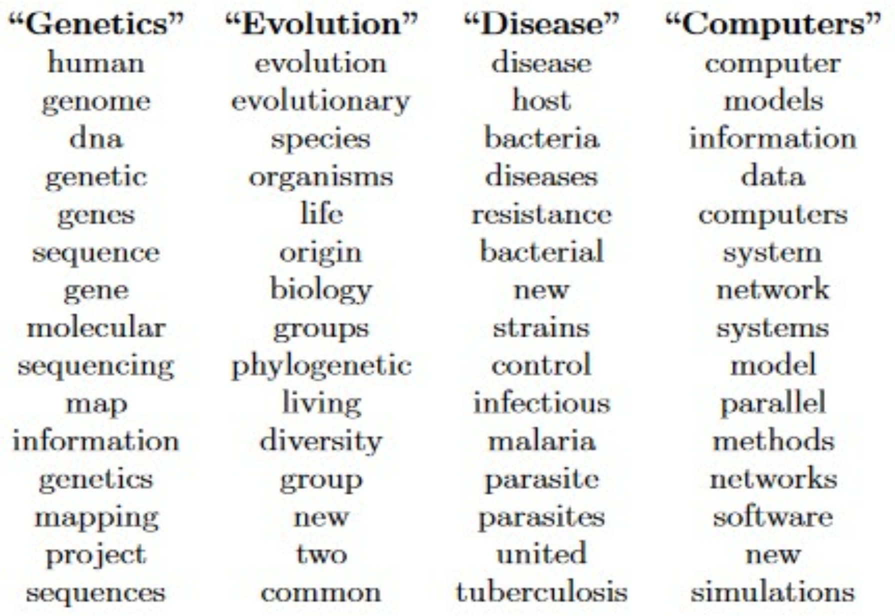
- Notice that there is a high chance that words in this document come from topic #21, and a smaller chance from topic #24 and # 61.

Source: David Blei, Machine Learning Summer School (2012)

Intuition: Documents contain multiple topics

Example article from Science and potential topics





"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

- What is topic #21? It is a "Genetics" topic, but there is also some chance that "Evolution" or "Disease" or "Computers" topics also occur
- The topic labels ("Genetics", "Evolution", "Disease", and "Computers") are assigned by the analyst and meant to be more descriptive than "topic #21"

Source: David Blei (2011), "Introduction to Probabilistic Topic Models"

Latent Dirichlet Allocation

Word selection *s* for document *i* at position *t*

$$w_{it} \sim \mathrm{M}(\beta_{z_{it}})$$

Multinomial Choice across all words based upon selected topic, β defines probabilities across words

Topic selection document *i* at time *t*

$$z_{it} \sim \mathrm{M}(\theta_i)$$

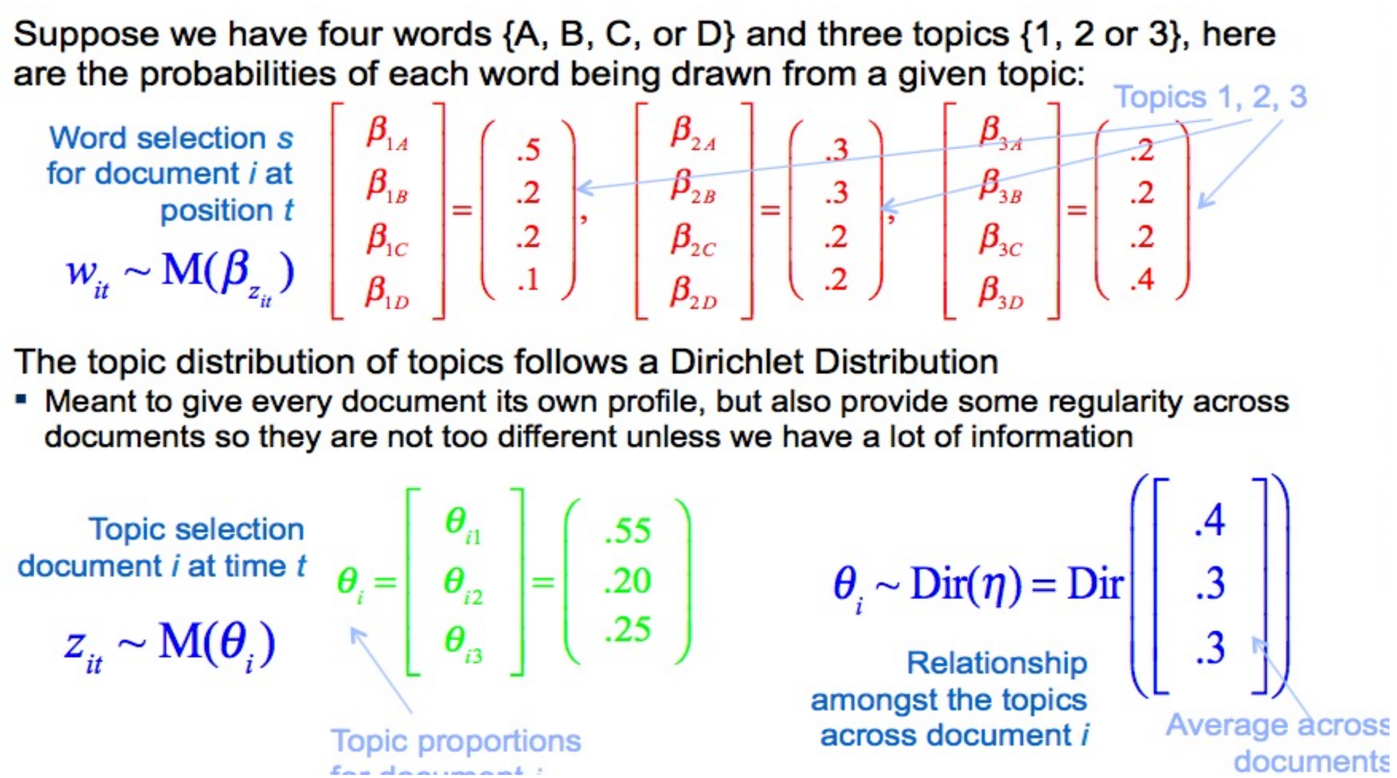
Multinomial Choice across all topics based upon unique document profile. Notice documents may choose more than one topic.

Relationship amongst the topics across document *i*

$$\theta_i \sim \mathrm{Dir}(\eta)$$

Dirichlet gives some structure to relationships between topics. What statisticians refer to as “shrinkage”.

Model Example



Suppose we have four words {A, B, C, or D} and three topics {1, 2 or 3}, here are the probabilities of each word being drawn from a given topic:

Word selection *s* for document *i* at position *t*

$$w_{it} \sim \mathrm{M}(\beta_{z_{it}})$$

$$\begin{bmatrix} \beta_{1A} \\ \beta_{1B} \\ \beta_{1C} \\ \beta_{1D} \end{bmatrix} = \begin{pmatrix} .5 \\ .2 \\ .2 \\ .1 \end{pmatrix}, \begin{bmatrix} \beta_{2A} \\ \beta_{2B} \\ \beta_{2C} \\ \beta_{2D} \end{bmatrix} = \begin{pmatrix} .3 \\ .3 \\ .2 \\ .2 \end{pmatrix}, \begin{bmatrix} \beta_{3A} \\ \beta_{3B} \\ \beta_{3C} \\ \beta_{3D} \end{bmatrix} = \begin{pmatrix} .2 \\ .2 \\ .2 \\ .4 \end{pmatrix}$$

Topics 1, 2, 3

The topic distribution of topics follows a Dirichlet Distribution

- Meant to give every document its own profile, but also provide some regularity across documents so they are not too different unless we have a lot of information

Topic selection document *i* at time *t*

$$z_{it} \sim \mathrm{M}(\theta_i)$$

$$\theta_i = \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \\ \theta_{i3} \end{bmatrix} = \begin{pmatrix} .55 \\ .20 \\ .25 \end{pmatrix}$$

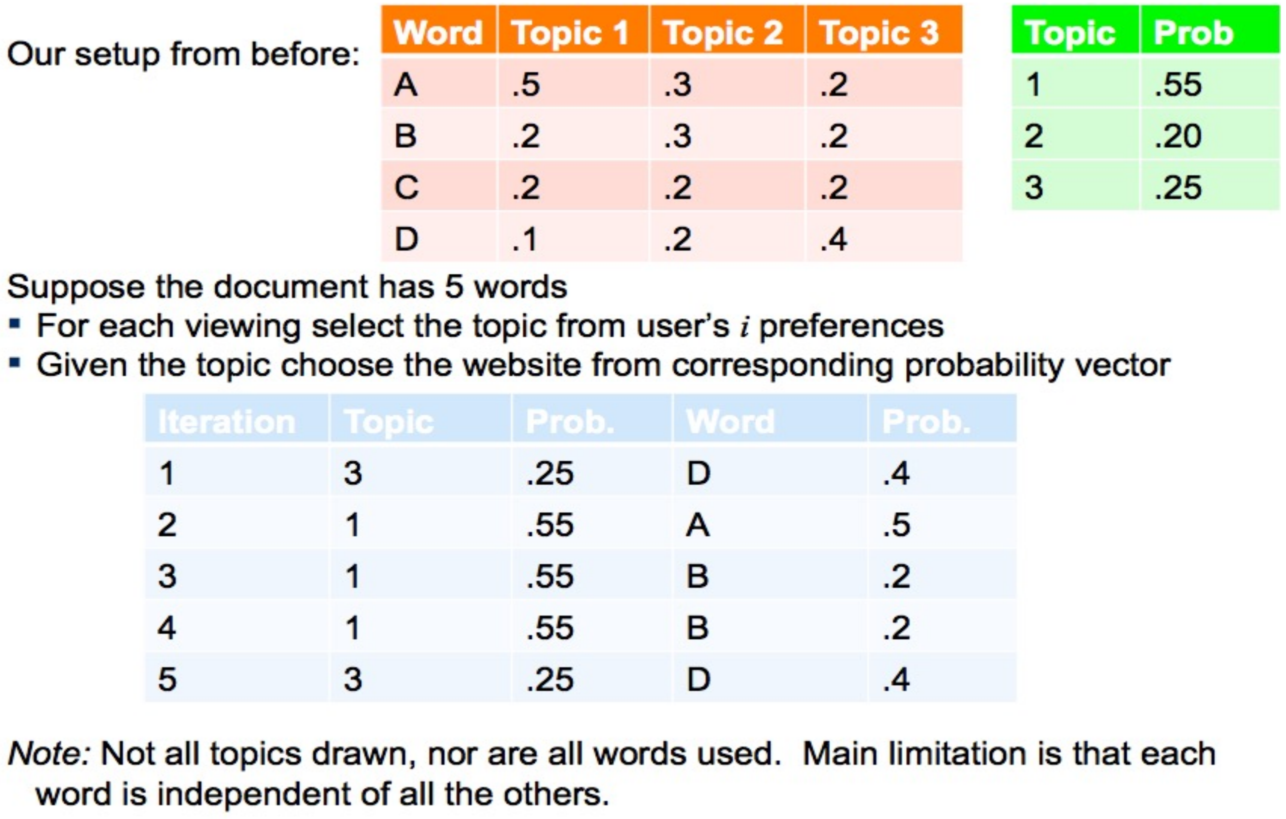
Topic proportions for document *i*

$$\theta_i \sim \mathrm{Dir}(\eta) = \mathrm{Dir}\left(\begin{bmatrix} .4 \\ .3 \\ .3 \end{bmatrix}\right)$$

Relationship amongst the topics across document *i*

Average across documents

Model Example



Our setup from before:

Word	Topic 1	Topic 2	Topic 3
A	.5	.3	.2
B	.2	.3	.2
C	.2	.2	.2
D	.1	.2	.4

Topic	Prob
1	.55
2	.20
3	.25

Suppose the document has 5 words

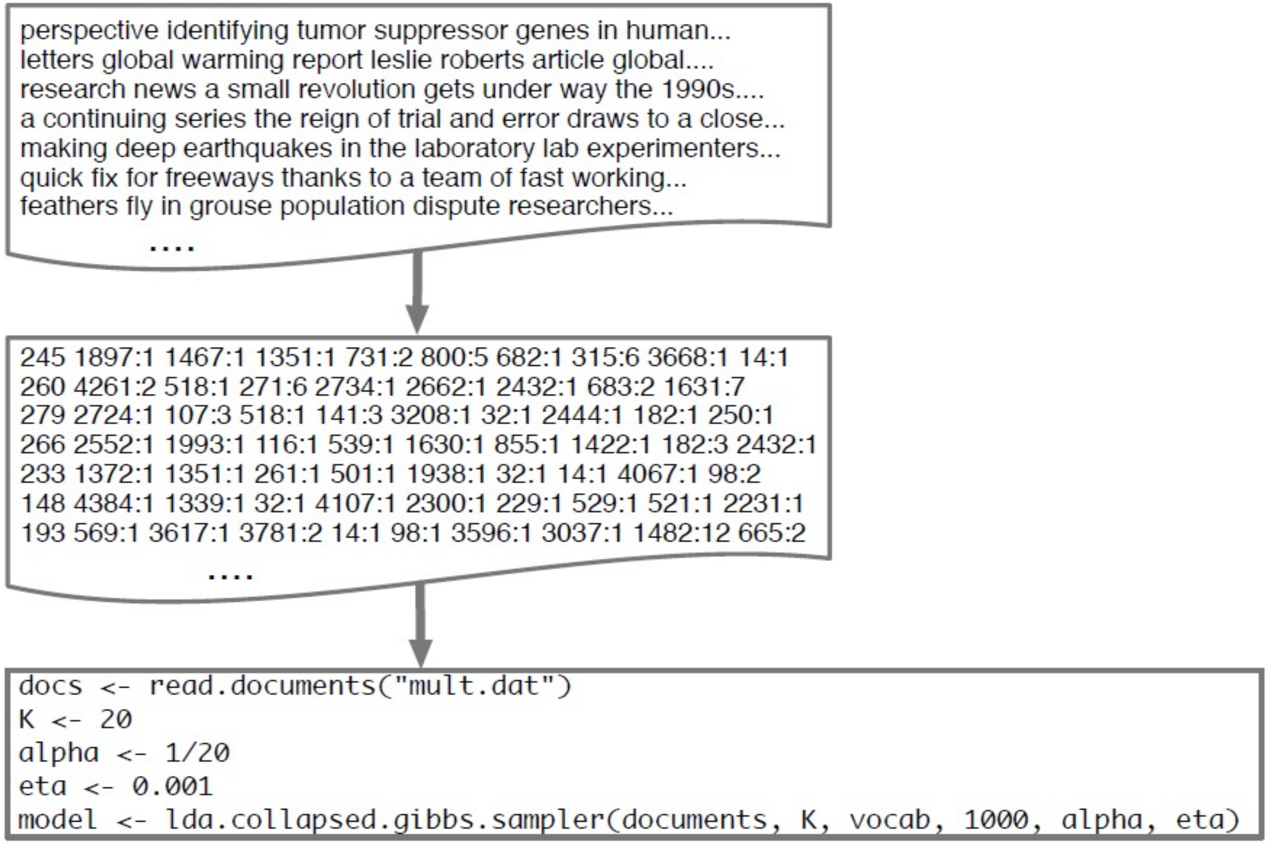
- For each viewing select the topic from user's i preferences
- Given the topic choose the website from corresponding probability vector

Iteration	Topic	Prob.	Word	Prob.
1	3	.25	D	.4
2	1	.55	A	.5
3	1	.55	B	.2
4	1	.55	B	.2
5	3	.25	D	.4

Note: Not all topics drawn, nor are all words used. Main limitation is that each word is independent of all the others.

Clustering using LDA

- LDA is commonly used to cluster documents.
  - Millions of documents with tens of thousands of words
  - LDA assumes documents are made up of latent topics
  - Each document has a “unique” profile of topics that randomly varies across documents. Each word is randomly drawn from a topic. Each topic in turn implies a probability that a word is used.
- LDA can be applied in many contexts: documents/words, emails/words, images/pixels, web browsing/sites.
  - Let’s redo our Ford Ka segmentation exercise:
  - Analogy: “Documents” become “Users”; “Words” become “Questions”. Instead of counting the occurrence of a word, we treat the Likert scales from the question response as word occurrences.



```
perspective identifying tumor suppressor genes in human...
letters global warming report leslie roberts article global....
research news a small revolution gets under way the 1990s....
a continuing series the reign of trial and error draws to a close...
making deep earthquakes in the laboratory lab experimenters...
quick fix for freeways thanks to a team of fast working...
feathers fly in grouse population dispute researchers...
        ....
```

```
245 1897:1 1467:1 1351:1 731:2 800:5 682:1 315:6 3668:1 14:1
260 4261:2 518:1 271:6 2734:1 2662:1 2432:1 683:2 1631:7
279 2724:1 107:3 518:1 141:3 3208:1 32:1 2444:1 182:1 250:1
266 2552:1 1993:1 116:1 539:1 1630:1 855:1 1422:1 182:3 2432:1
233 1372:1 1351:1 261:1 501:1 1938:1 32:1 14:1 4067:1 98:2
148 4384:1 1339:1 32:1 4107:1 2300:1 229:1 529:1 521:1 2231:1
193 569:1 3617:1 3781:2 14:1 98:1 3596:1 3037:1 1482:12 665:2
            ....
```

```
docs <- read.documents("mult.dat")
K <- 20
alpha <- 1/20
eta <- 0.001
model <- lda.collapsed.gibbs.sampler(documents, K, vocab, 1000, alpha, eta)
```

- Use the R package called LDA by Jonathan Chang or the topicmodels package. My example uses topicmodels, but LDA also has supervised LDA.

Model Features

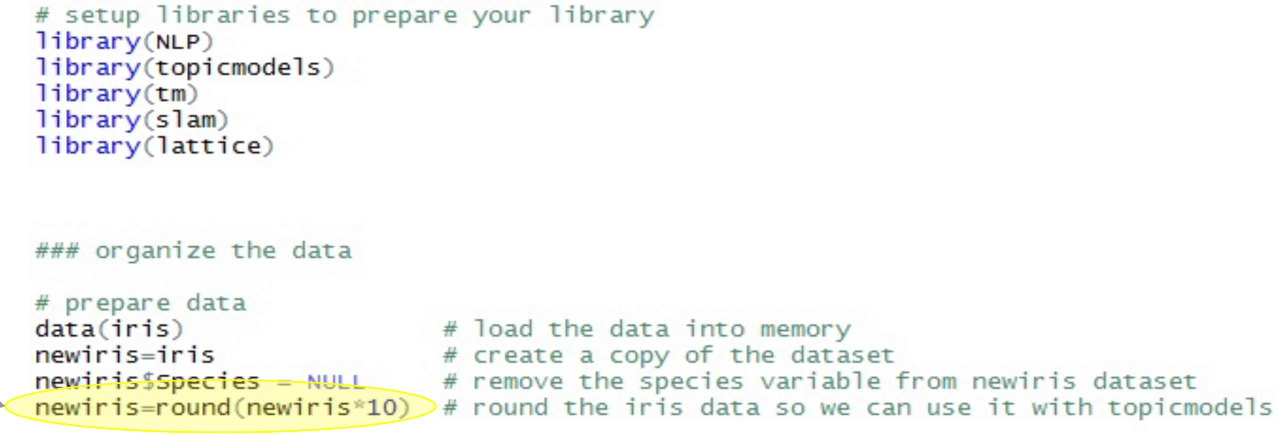
- Latent topics drive probability of choosing a word
- Probability of using a word given a topic is common across all users (e.g., “money” is the same for every document)
- Each document’s topics are unique (but if there are not too many words in the document then the topics will look similar across documents)
- Estimation using variational EM algorithm or Monte-Carlo Markov Chain (simulation approach)

Topic Modeling applied to the Iris Data

See “iris_LDAcluster.R”

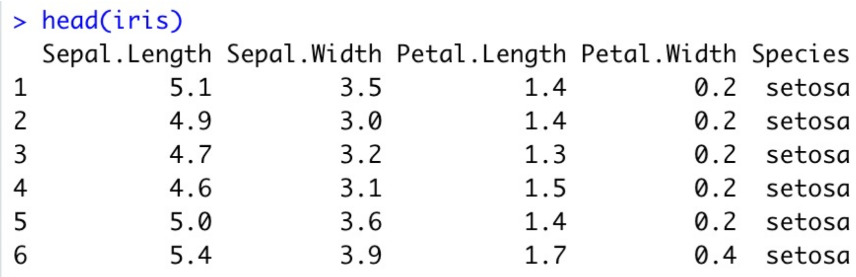
Implementing with R
using the Package topicmodels
Example using Iris Data

LDA expects the input data to be counts (since it uses a Multinomial assumption), So we multiply all measurements by 10 and round (e.g., “3.8” becomes “38”).



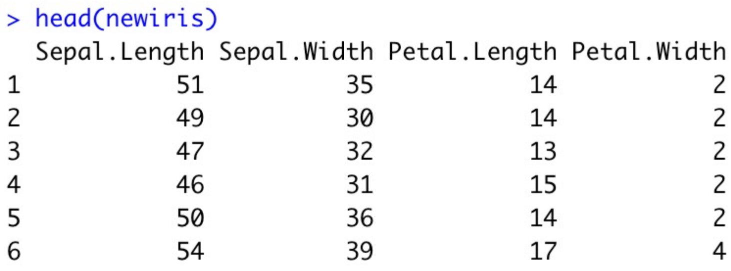
```
# setup libraries to prepare your library
library(NLP)
library(topicmodels)
library(tm)
library(slam)
library(lattice)


### organize the data

# prepare data
data(iris)                    # load the data into memory
newiris=iris                  # create a copy of the dataset
newiris$Species = NULL        # remove the species variable from newiris dataset
newiris=round(newiris*10)     # round the iris data so we can use it with topicmodels
```

Original Data



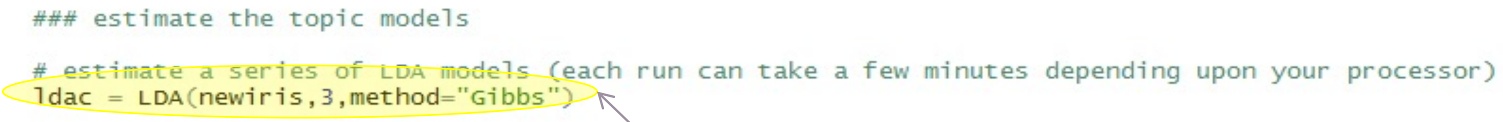
```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

“Count” Style Data



```
> head(newiris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1           51          35           14           2
2           49          30           14           2
3           47          32           13           2
4           46          31           15           2
5           50          36           14           2
6           54          39           17           4
```

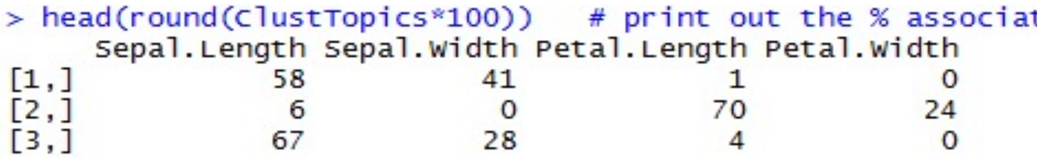
Implementing with R
using the Package topicmodels
Example using Iris Data



```
### estimate the topic models

# estimate a series of LDA models (each run can take a few minutes depending upon your processor)
ldac = LDA(newiris,3,method="Gibbs")
```

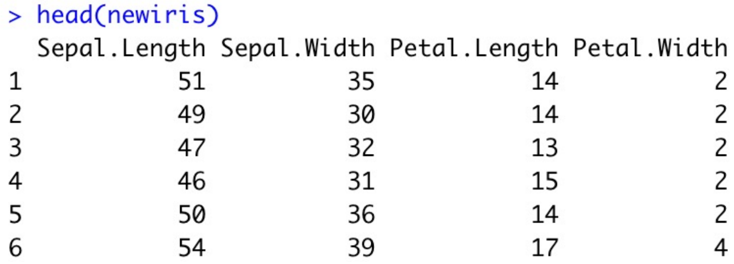
The “3” means three topics using all four measures that are given in the newiris dataset. There are two Methods “Gibbs” or “VEM” which stands for Variation EM and is an approximate technique but is faster

Output: “Clusters” or Topics



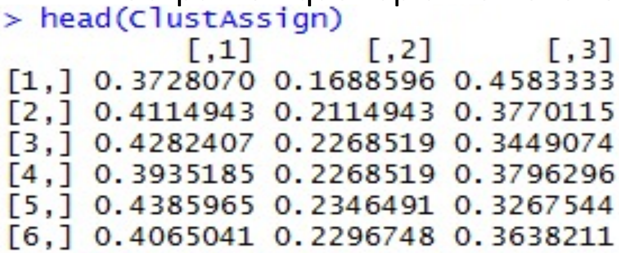
```
> head(round(clustTopics*100))    # print out the % associat
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]           58          41            1           0
[2,]            6           0           70          24
[3,]           67          28            4           0
```

Input: “Count” Style Data



```
> head(newiris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1           51          35           14           2
2           49          30           14           2
3           47          32           13           2
4           46          31           15           2
5           50          36           14           2
6           54          39           17           4
```

Output: Unique Topic Profile for each Flower



```
> head(clustAssign)
          [,1]      [,2]      [,3]
[1,] 0.3728070 0.1688596 0.4583333
[2,] 0.4114943 0.2114943 0.3770115
[3,] 0.4282407 0.2268519 0.3449074
[4,] 0.3935185 0.2268519 0.3796296
[5,] 0.4385965 0.2346491 0.3267544
[6,] 0.4065041 0.2296748 0.3638211
```

Implementing with R using the Package topicmodels

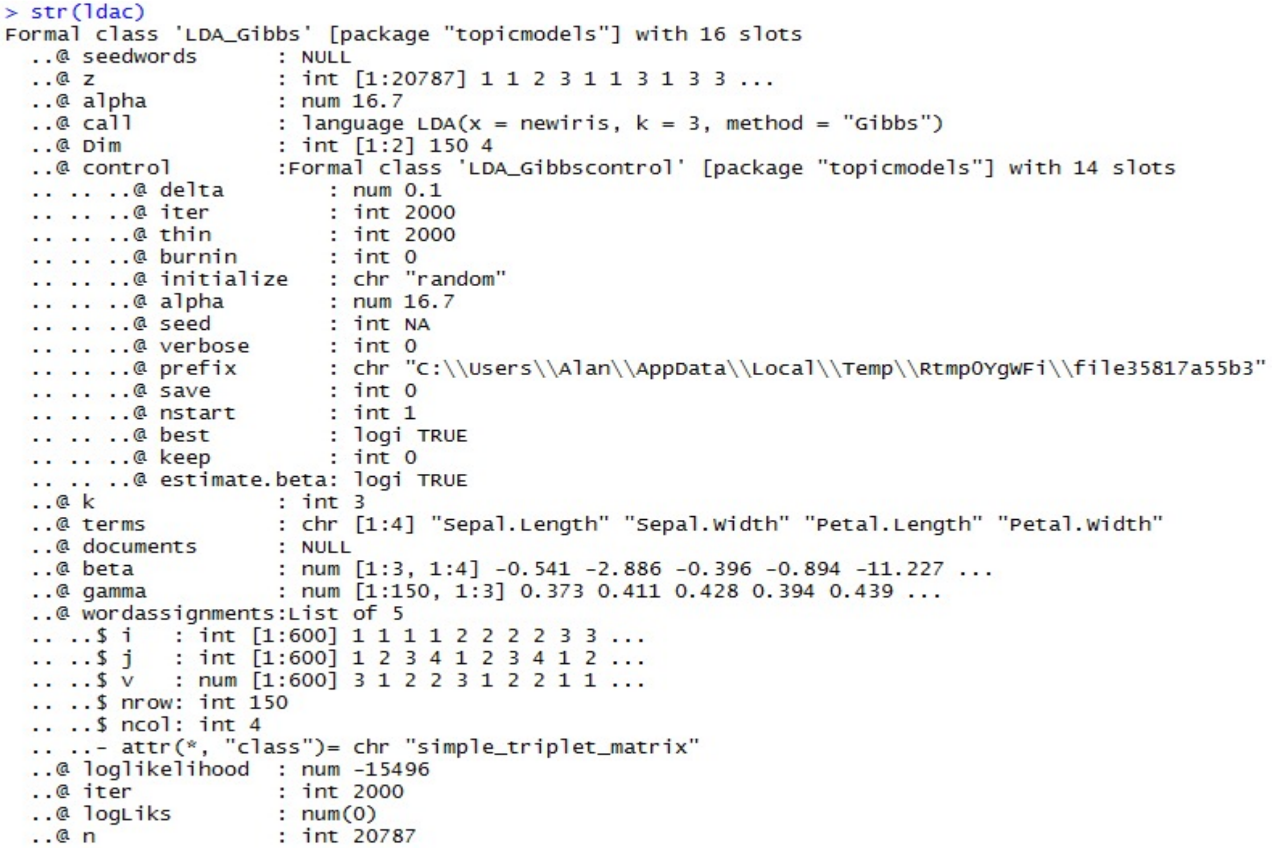
Example using Iris Data

R has three different
classes of objects:
S3, S4, and R6.

The default is S3
and '$' to access elements, S4 requires initialization and '@' to access slots. R6 is not used frequently but behaves much more like
Python or C++

An LDA object contains:

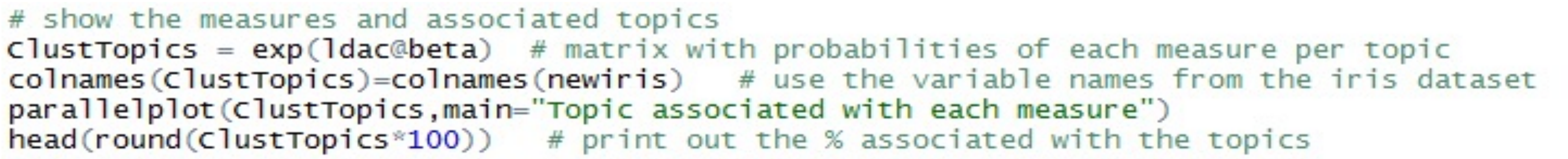
- k = # of topics
- beta = log of word distribution for each topic
- gamma = posterior topic distribution for each document
- wordassignments = most probable topic for each word in document
- Loglikelihood = measure of fit



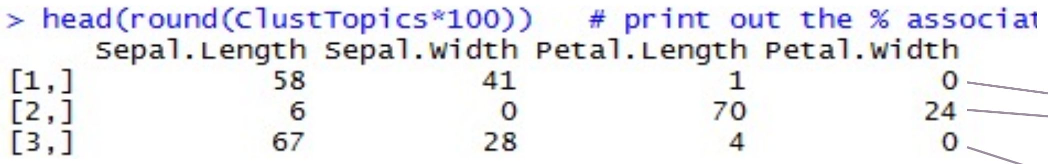
```
> str(ldac)
Formal class 'LDA_Gibbs' [package "topicmodels"] with 16 slots
  ..@ seedwords       : NULL
  ..@ z               : int [1:20787] 1 1 2 3 1 1 3 1 3 3 ...
  ..@ alpha           : num 16.7
  ..@ call            : language LDA(x = newiris, k = 3, method = "Gibbs")
  ..@ Dim             : int [1:2] 150 4
  ..@ control         :Formal class 'LDA_Gibbscontrol' [package "topicmodels"] with 14 slots
  .. .. ..@ delta         : num 0.1
  .. .. ..@ iter          : int 2000
  .. .. ..@ thin          : int 2000
  .. .. ..@ burnin        : int 0
  .. .. ..@ initialize    : chr "random"
  .. .. ..@ alpha         : num 16.7
  .. .. ..@ seed          : int NA
  .. .. ..@ verbose       : int 0
  .. .. ..@ prefix        : chr "C:\\Users\\Alan\\AppData\\Local\\Temp\\Rtmp0YgWFi\\file35817a55b3"
  .. .. ..@ save          : int 0
  .. .. ..@ nstart        : int 1
  .. .. ..@ best          : logi TRUE
  .. .. ..@ keep          : int 0
  .. .. ..@ estimate.beta: logi TRUE
  ..@ k               : int 3
  ..@ terms           : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
  ..@ documents       : NULL
  ..@ beta            : num [1:3, 1:4] -0.541 -2.886 -0.396 -0.894 -11.227 ...
  ..@ gamma           : num [1:150, 1:3] 0.373 0.411 0.428 0.394 0.439 ...
  ..@ wordassignments:List of 5
  .. ..$ i   : int [1:600] 1 1 1 1 2 2 2 2 3 3 ...
  .. ..$ j   : int [1:600] 1 2 3 4 1 2 3 4 1 2 ...
  .. ..$ v   : num [1:600] 3 1 2 2 3 1 2 2 1 1 ...
  .. ..$ nrow: int 150
  .. ..$ ncol: int 4
  .. ..- attr(*, "class")= chr "simple_triplet_matrix"
  ..@ loglikelihood  : num -15496
  ..@ iter            : int 2000
  ..@ logLiks         : num(0)
  ..@ n               : int 20787
```

Implementing with R using the Package topicmodels

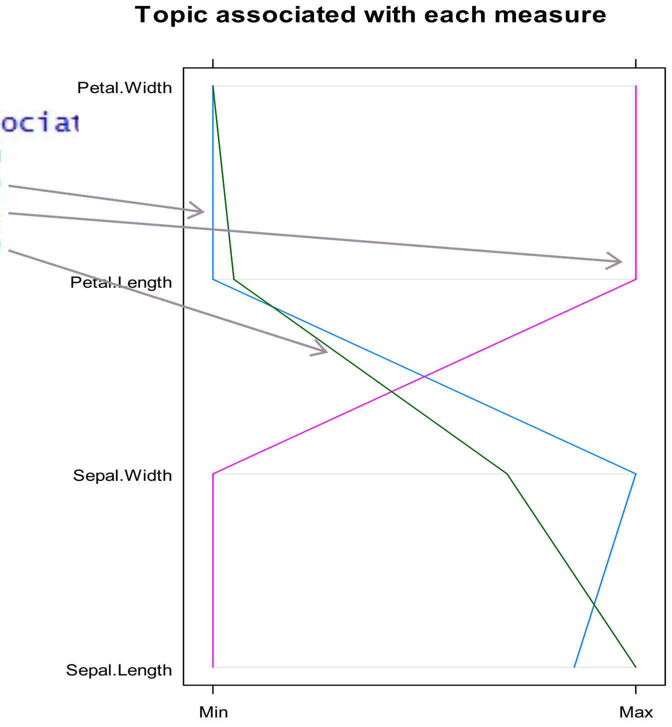
Example using Iris Data: What do our topics mean?



```
# show the measures and associated topics
ClustTopics = exp(ldac@beta)  # matrix with probabilities of each measure per topic
colnames(ClustTopics)=colnames(newiris)   # use the variable names from the iris dataset
parallelplot(ClustTopics,main="Topic associated with each measure")
head(round(ClustTopics*100))    # print out the % associated with the topics
```



```
> head(round(ClustTopics*100))    # print out the % associat
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]           58          41            1           0
[2,]            6           0           70          24
[3,]           67          28            4           0
```

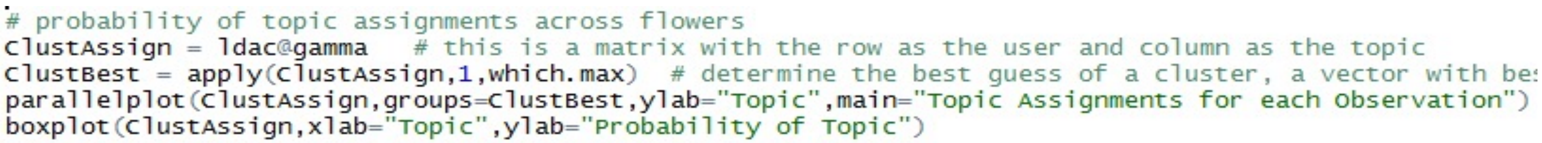


What do the topics mean?

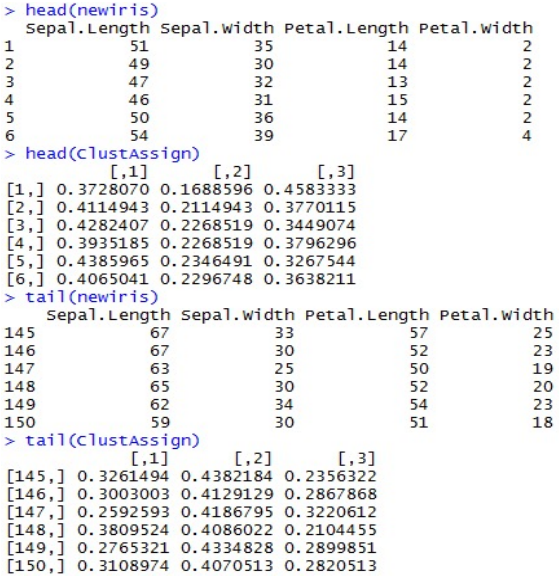
- Topic #1 captures flowers in which the sepal length and width are roughly proportional
- Topic #2 captures the petals lengths
- Topic #3 captures flowers with long sepals and narrower sepals

Implementing with R using the Package topicmodels

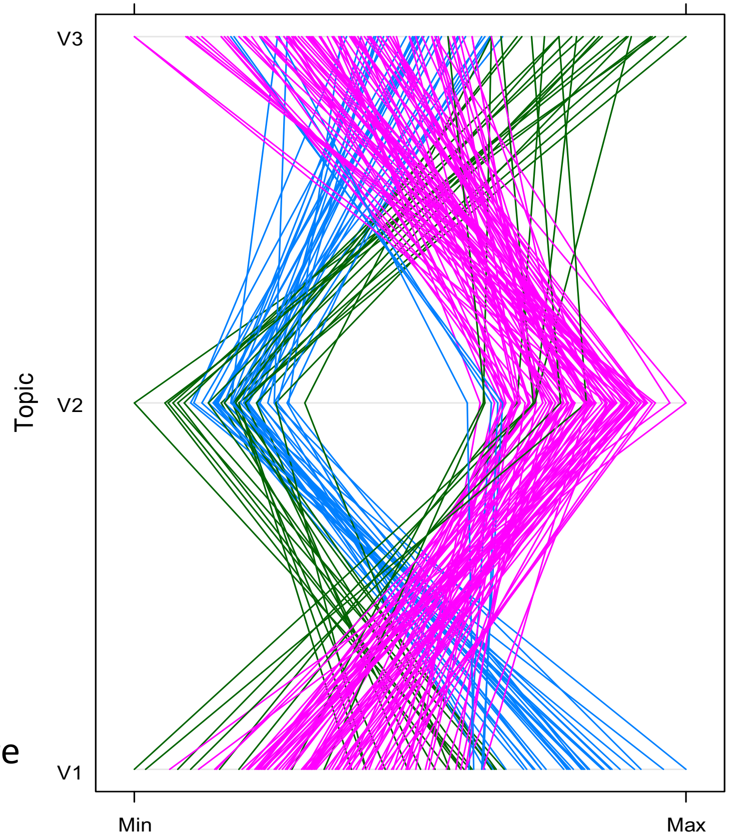
Example using Iris Data: What are the topics for each flower?



```
# probability of topic assignments across flowers
ClustAssign = ldac@gamma   # this is a matrix with the row as the user and column as the topic
ClustBest = apply(ClustAssign,1,which.max)  # determine the best guess of a cluster, a vector with bes
parallelplot(ClustAssign,groups=ClustBest,ylab="Topic",main="Topic Assignments for each Observation")
boxplot(ClustAssign,xlab="Topic",ylab="Probability of Topic")
```



```
> head(newiris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1           51          35           14           2
2           49          30           14           2
3           47          32           13           2
4           46          31           15           2
5           50          36           14           2
6           54          39           17           4
> head(ClustAssign)
          [,1]      [,2]      [,3]
[1,] 0.3728070 0.1688596 0.4583333
[2,] 0.4114943 0.2114943 0.3770115
[3,] 0.4282407 0.2268519 0.3449074
[4,] 0.3935185 0.2268519 0.3796296
[5,] 0.4385965 0.2346491 0.3267544
[6,] 0.4065041 0.2296748 0.3638211
> tail(newiris)
    Sepal.Length Sepal.Width Petal.Length Petal.Width
145           67          33           57          25
146           67          30           52          23
147           63          25           50          19
148           65          30           52          20
149           62          34           54          23
150           59          30           51          18
> tail(ClustAssign)
            [,1]      [,2]      [,3]
[145,] 0.3261494 0.4382184 0.2356322
[146,] 0.3003003 0.4129129 0.2867868
[147,] 0.2592593 0.4186795 0.3220612
[148,] 0.3809524 0.4086022 0.2104455
[149,] 0.2765321 0.4334828 0.2899851
[150,] 0.3108974 0.4070513 0.2820513
```



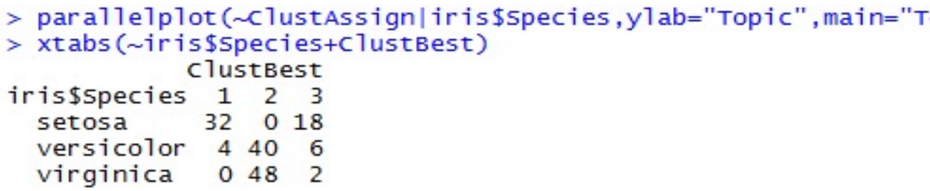
Notice that variability in topics across flowers.

The early ones (Setosa) have relative long sepals (topic #1 and 3) and small petals (topic #2)

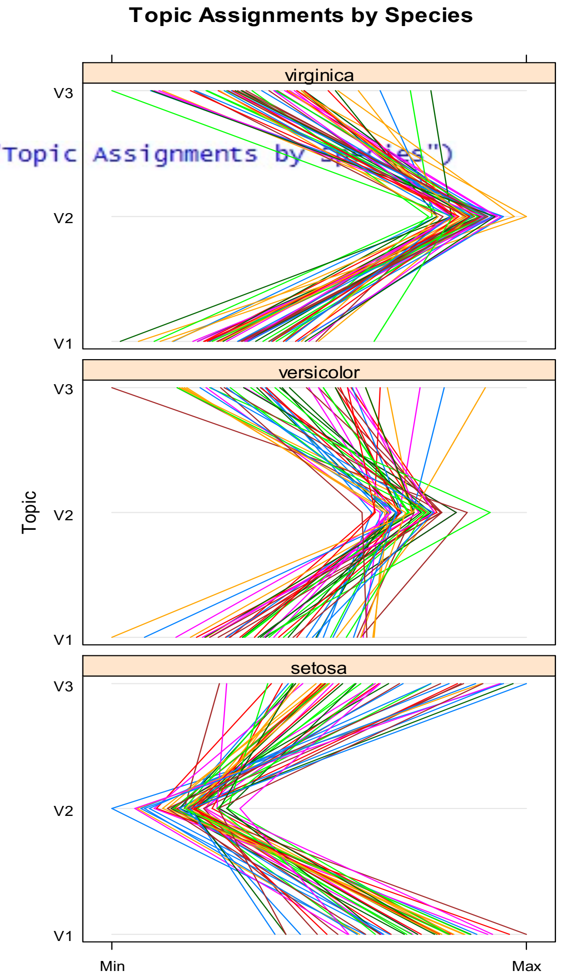
The latter ones (Virginica) have long petals and are more likely to have "topic #2",

Implementing with R using the Package topicmodels

Example using Iris Data: What are the topics for each flower?

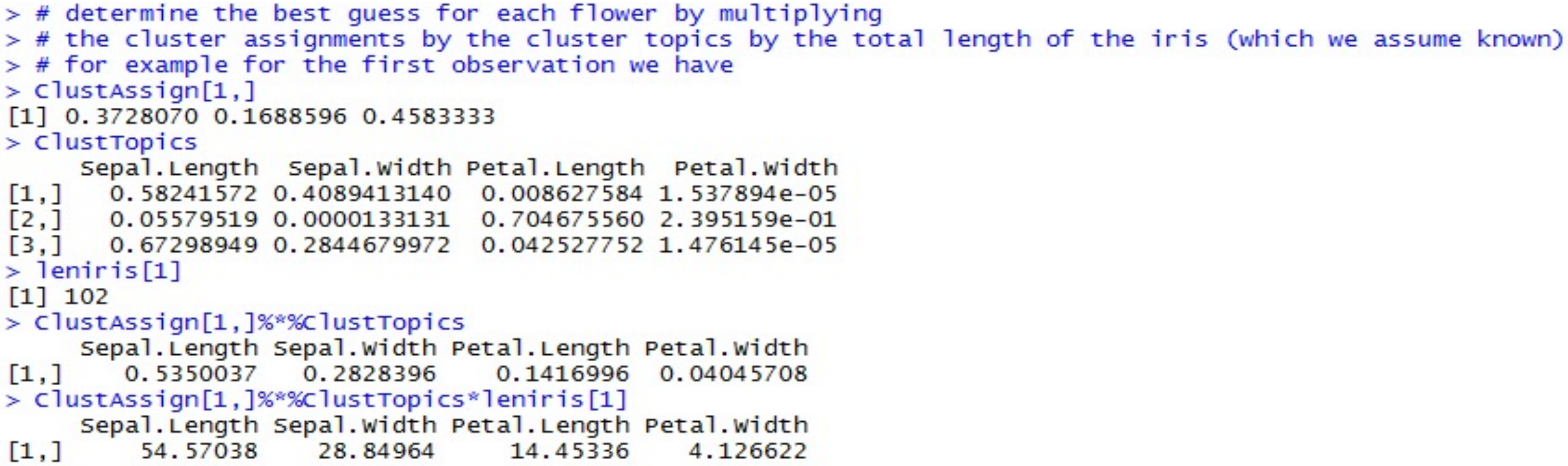


```
> parallelplot(~ClustAssign|iris$Species,ylab="Topic",main="Topic Assignments by Species")
> xtabs(~iris$Species+ClustBest)
            ClustBest
iris$Species  1  2  3
  setosa     32  0 18
  versicolor  4 40  6
  virginica   0 48  2
```



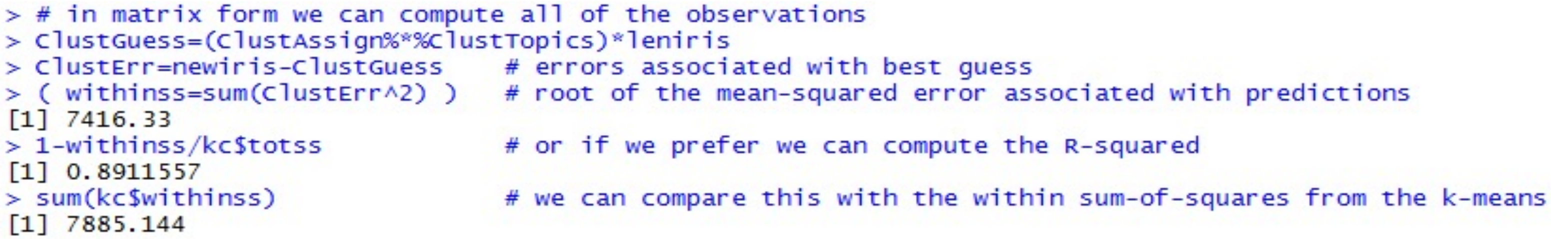
We can see the “clusters” by observing:

- Virginica seems to have a relatively high probability of topic#2
- Versicolor is high on topic #2
- Setosa has a low probability of topic #2, but high probability of topic #1 and #3



```
> # determine the best guess for each flower by multiplying
> # the cluster assignments by the cluster topics by the total length of the iris (which we assume known)
> # for example for the first observation we have
> ClustAssign[1,]
[1] 0.3728070 0.1688596 0.4583333
> ClustTopics
     Sepal.Length  Sepal.Width Petal.Length  Petal.Width
[1,]   0.58241572 0.4089413140  0.008627584 1.537894e-05
[2,]   0.05579519 0.0000133131  0.704675560 2.395159e-01
[3,]   0.67298949 0.2844679972  0.042527752 1.476145e-05
> leniris[1]
[1] 102
> ClustAssign[1,]%*%ClustTopics
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]    0.5350037   0.2828396    0.1416996  0.04045708
> ClustAssign[1,]%*%ClustTopics*leniris[1]
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]     54.57038    28.84964     14.45336    4.126622
```

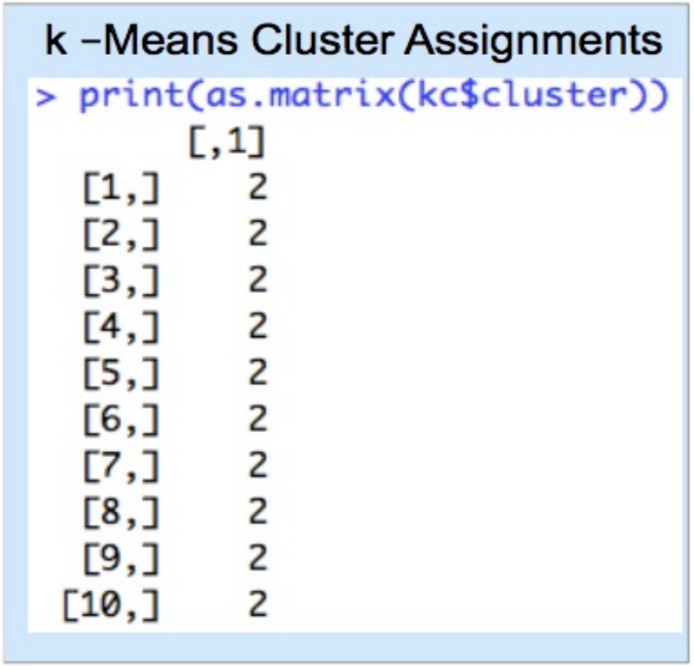
To get the predictions we have to multiply the Topics Assigned (=ClustAssign) by the weights for each topic (=ClustTopics) by the total length of the flower (which is assumed known in topic models, since the word count usually doesn't have much information)



```
> # in matrix form we can compute all of the observations
> ClustGuess=(ClustAssign%*%ClustTopics)*leniris
> ClustErr=newiris-ClustGuess    # errors associated with best guess
> ( withinss=sum(ClustErr^2) )   # root of the mean-squared error associated with predictions
[1] 7416.33
> 1-withinss/kc$totss            # or if we prefer we can compute the R-squared
[1] 0.8911557
> sum(kc$withinss)               # we can compare this with the within sum-of-squares from the k-means
[1] 7885.144
```

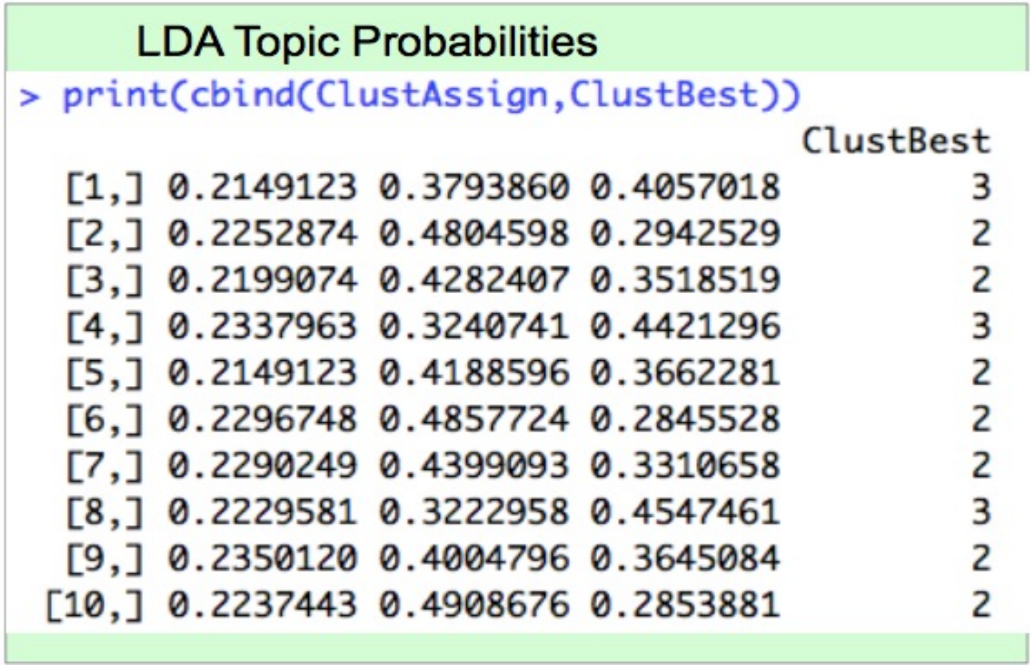
Notice the R-squared of the predictions for our LDA model is 89% which is a little better than our k-Means cluster analysis with k=3

Comparing our Topic Model Output with k-Means



k –Means Cluster Assignments

```
> print(as.matrix(kc$cluster))
      [,1]
 [1,]    2
 [2,]    2
 [3,]    2
 [4,]    2
 [5,]    2
 [6,]    2
 [7,]    2
 [8,]    2
 [9,]    2
[10,]    2
```



LDA Topic Probabilities

```
> print(cbind(ClustAssign,ClustBest))
                                        ClustBest
 [1,] 0.2149123 0.3793860 0.4057018         3
 [2,] 0.2252874 0.4804598 0.2942529         2
 [3,] 0.2199074 0.4282407 0.3518519         2
 [4,] 0.2337963 0.3240741 0.4421296         3
 [5,] 0.2149123 0.4188596 0.3662281         2
 [6,] 0.2296748 0.4857724 0.2845528         2
 [7,] 0.2290249 0.4399093 0.3310658         2
 [8,] 0.2229581 0.3222958 0.4547461         3
 [9,] 0.2350120 0.4004796 0.3645084         2
[10,] 0.2237443 0.4908676 0.2853881         2
```

Conclusion

Summary

Some types of clusters are well represented by k-Means

- “Clumpy” data when the clusters are about the same size

Probabilistic clustering is an alternative scheme

- Instead of every observation being assigned to one cluster
- Every observation will have a probability of being in each cluster

Topic Modeling is a specific type of probabilistic clustering that has been found to work well for modeling text data

- Topics can be thought of as underlying groupings or frequencies of words
- Every document has a unique profile of topics
- Every word in a document has a topic which gives the probability that a word will occur