Introduction to Text Analysis

Bryce J. Dietrich

University of Iowa

Text and Political Science

Post 2000, things have changed...

- Massive collections of texts are being increasingly used as a data source:
  - Congressional speeches, press releases, newsletters,...
  - Facebook posts, tweets, emails, text messages...
  - Newspapers, magazines, transcripts...
  - Foreign news sources, treaties, ...
- Why?
  - LOTS of unstructured text data (201 billion emails sent and received every day)
  - LOTS of cheap storage: 1956: $10,000 per megabyte. 2016: $0.0001 per megabyte.
  - LOTS of methods and software to analyze text

Text and Political Science

Ultimately, these trends mean. . .

- Analyzing text has become bigger, faster, and stronger:
  - Generalizable $\rightarrow$ one method can be used across a variety of text
  - Systematic $\rightarrow$ one method can be used again and again
  - Cheap $\rightarrow$ one computer can do a lot, 100 computers can do even more
- Analyzing text is still important:
  - Laws
  - Treaties
  - News media
  - Campaigns
  - Petitions
  - Speeches
  - Press Releases

What Can We Do?

What Can We Do?

There are two things we may want to do with this haystack...

- Understanding the meaning of a sentence or phrase → analyzing a straw of hay
  - Humans = 1
  - Computers = 0
- Classifying text → organizing hay stack
  - Humans = 0
  - Computers = 1

Text = Complex?

This bill has seen a long and winding road, but in the process we have worked together. We have not quit. We have worked across the aisle. The final bill has the support of over 370 groups, and they represent those from all over the country and all over the ideological spectrum.

Text = Complex?

This bill has seen a long and winding road, but in the process we have worked together. We have not quit. We have worked across the aisle. The final bill has the support of over 370 groups, and they represent those from all over the country and all over the ideological spectrum.

Text = Complex?

This bill has seen a long and winding road, but in the process we have worked together. We have not quit. We have worked across the aisle. The final bill has the support of over 370 groups, and they represent those from all over the country and all over the ideological spectrum.

Text = Complex?

This bill has seen a long and winding road, but in the process we have worked together. We have not quit. We have worked across the aisle. The final bill has the support of over 370 groups, and they represent those from all over the country and all over the ideological spectrum.

Text = Complex?

This bill has seen a long and winding road, but in the process we have worked together. We have not quit. We have worked across the aisle. The final bill has the support of over 370 groups, and they represent those from all over the country and all over the ideological spectrum.

Text = Simple?

Speech by Barbara Mikulski (D-MD) delivered on June 28, 2016. . .

word	count
zika	4
million	4
emergency	3
health	3
act	2
response	2
republican	2
report	1
treat	1
world	1
organization	1
affordable	1

Text = Simple?

The Republican conference report also doesn't treat Zika like the emergency it is. The World Health Organization declared the Zika virus a public health emergency on February 1. And Zika meets the Budget Act criteria for emergency spending: It is urgent, unforeseen, and temporary. Yet Republicans insisted that we cut \$750 *million to pay for the response to Zika, including* \$543 *million from the Affordable Care Act,* \$100 *million from the Department of Health and Human Services, HHS, nonrecurring expense fund, and* \$107 *million from Ebola response funds.*

"Big Data"

Suppose we want to categorize 100 documents...

- Consider two documents A and B, how many clusters can we make? $\rightarrow$ (AB, BA) $= 2$
- Consider three documents A, B, and C, how many clusters can we make? $\rightarrow$ (ABC, CBA, ACB, BCA, CAB) $= 5$
- Bell(n) $=$ number of ways to partition *n* objects. Bell(2) $=$ 2, Bell(3) $=$ 5, Bell(5) $=$ 52, etc.
- Bell(100) $= 4.758539 \times 10^{115}$
  - It takes `R` 0.001 seconds to count to 100000
  - It would take `R` $= 4.758539 \times 10^{110}$ seconds to count to Bell(100)
  - There are 3.154×10^{7} seconds in a year.
  - $\frac{4.758539 \times 10^{110}}{3.154 \times 10^{7}} = 1.508731 \times 10^{103}$ years.

Automated methods can help with even small tasks!

Principles of Automated Text Analysis (from Justin Grimmer)

Principle 1: All Quantitative Models of Language are Wrong – But Some are Useful

- Data generation process unknown
- Complexity of language:
  - Time flies like an arrow; fruit flies like a banana
  - Make peace, not war; Make war not peace
- Models necessarily fail to capture language

Principle 2: Quantitative Methods Augment Humans, Not Replace Them

- Computer-Assisted Content Analysis
- Computers suggest, Humans interpret

Principle 3: There is no Globally Best Method for Automated Text Analysis

- Supervised methods = known categories
- Unsupervised methods = discover categories

Principle 4: Validate, Validate, Validate

Principles of Automated Text Analysis (from Justin Grimmer)

Principle 2: Quantitative Methods Augment Humans, Not Replace Them

- Computer-Assisted Content Analysis
- Computers suggest, Humans interpret

Principles of Automated Text Analysis (from Justin Grimmer)

Principle 3: There is no Globally Best Method for Automated Text Analysis

- Supervised methods = known categories
- Unsupervised methods = discover categories

Principles of Automated Text Analysis (from Justin Grimmer)

Principle 4: Validate, Validate, Validate

- Few theorems to guarantee performance
- Apply methods $\rightarrow$ validate
- Do not blinding use methods!

Principles of Automated Text Analysis (from Justin Grimmer)

Principle 1: All Quantitative Models of Language are Wrong – But Some are Useful

Principle 2: Quantitative Methods Augment Humans, Not Replace Them

Principle 3: There is no Globally Best Method for Automated Text Analysis

Principle 4: Validate, Validate, Validate

Types of Classification Problems

Topic: What is this text about?

- Policy area of legislation
- Party agendas

Sentiment: What is said in this text?

- For or against legislation
- Agree or disagree with an argument
- A liberal/conservative position

Style/Tone: How is it said?

- Positive/Negative Emotion

Weighted Words!

Dictionary Methods

Linguistic Inquiry and Word Count: LIWC2015

Dictionary Methods

Many Dictionary Methods (like LIWC)

1) Proprietary $\rightsquigarrow$ wrapped in GUI
2) Basic tasks:
   a) Count words
   b) Weight some words more than others
   c) Some graphics
3) Expensive!

Other Dictionaries

1) General Inquirer Database (`http://www.wjh.harvard.edu/~inquirer/`)
   - Stone, P.J., Dumphy, D.C., and Ogilvie, D.M. (1966) *The General Inquirer: A Computer Approach to Content Analysis*
   - 1,915 positive words and 2,291 negative words
2) Linguistic Inquiry Word Count (LIWC)
   - Creation process:
     1) Generate word list for categories $\rightsquigarrow$ " We drew on common emotion rating scales...[then] brain-storming sessions among 3-6 judges were held" to generate other words
     2) Judge round $\rightsquigarrow$ (a) Does the word belong? (b) What other categories might it belong to?
   - 406 positive words and 499 negative words

Generating New Words

Three ways to create dictionaries:

- Statistical methods
- Manual generation
  - "Theory"
- "Research Assistants"
  a) Grad Students
  b) Undergraduates
  c) Mechanical Turkers
     - Example: {Happy, Unhappy}
     - Ask Turkers: how happy is `elevator`, `car`, `pretty`, `young`

Applying a Dictionary to NYT Articles

Python!

Document-Term Matrices

$$\boldsymbol{X} = \begin{pmatrix} 1 & 0 & 0 & \dots & 3 \\ 0 & 2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 5 \end{pmatrix}$$

$\boldsymbol{X} = N \times K$ matrix

- $N =$ Number of documents
- $K =$ Number of features

"I Have A Dream"


```python
# import modules
import requests
from bs4 import BeautifulSoup

# create url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# get speech
speech = ""
lines = soup.find_all('p')
for line in lines:
    speech += line.get_text()
```

Installing Beautiful Soup


```
# activate our virtual environment
source activate python-UI

# install beautiful soup
python -m pip install beautifulsoup4
```

Using Beautiful Soup

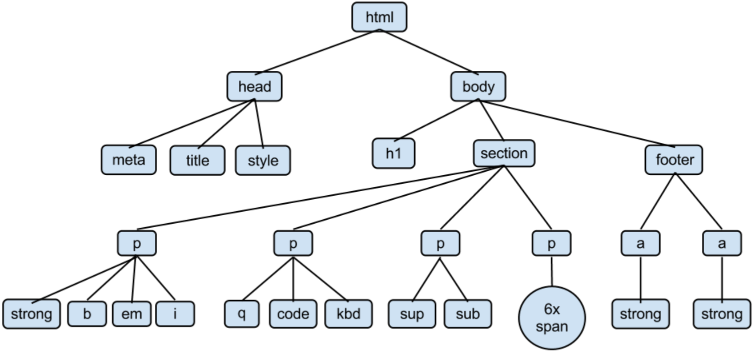
```
# import modules
import requests
from bs4 import BeautifulSoup

# create url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# print soup
print(soup.prettify()[0:300])
```

HTML


```
<html>
 <head>
  <link href=”../css/site.css” rel=”stylesheet” type=”text/css”>
   <title>
    Avalon Project - I have a Dream by Martin Luther King, Jr; August 28, 1963
   </title>
  </link>
 </head>
 <body>
  <div class=”HeaderContainer”>
   <ul class=”HeaderTopTools”>
    <li class=”Search”>
```

HTML

Browsers read in the HTML document, parses it into a DOM (Document Object Model) structure, and then renders the DOM structure.



HTML

Browsers read in the HTML document, parses it into a DOM (Document Object Model) structure, and then renders the DOM structure.

The Document

```
<html>
<body>
<h1>Title</h1>
<p>A <em>word</em></p>
</body>
</html>
```

The DOM Tree

```
DOCUMENT
└ELEMENT: html
  ├TEXT: '\n'
  ├ELEMENT: body
  │ ├TEXT: '\n'
  │ ├ELEMENT: h1
  │ │ └TEXT: 'Title'
  │ ├TEXT: '\n'
  │ ├ELEMENT: p
  │ │ ├TEXT: 'A'
  │ │ └ELEMENT: em
  │ │   └TEXT: word
  │ └TEXT: '\n'
  └TEXT: '\n'
```

“I Have A Dream”

Yale Law School
LILLIAN GOLDMAN LAW LIBRARY
in memory of Sol Goldman

THE AVALON PROJECT *Documents in Law, History and Diplomacy*

Search Avalon

Avalon Home	Document Collections	Ancient 4000bce - 399	Medieval 400 - 1399	15th Century 1400 - 1499	16th Century 1500 - 1599	17th Century 1600 - 1699	18th Century 1700 - 1799	19th Century 1800 - 1899	20th Century 1900 - 1999	21st Century 2000 -

I have a Dream by Martin Luther King, Jr; August 28, 1963

Delivered on the steps at the Lincoln Memorial in Washington D.C. on August 28, 1963

Five score years ago, a great American, in whose symbolic shadow we stand signed the Emancipation Proclamation. This momentous decree came as a great beacon light of hope to millions of Negro slaves who had been seared in the flames of withering injustice. It came as a joyous daybreak to end the long night of captivity.

But one hundred years later, we must face the tragic fact that the Negro is still not free. One hundred years later, the life of the Negro is still sadly crippled by the manacles of segregation and the chains of discrimination. One hundred years later, the Negro lives on a lonely island of poverty in the midst of a vast ocean of material prosperity. One hundred years later, the Negro is still languishing in the corners of American society and finds himself an exile in his own land. So we have come here today to dramatize an appalling condition.

Finding Text

```
<p>
I have a dream that one day on the red hills of
    Georgia the sons of former slaves and the sons
    of former slaveowners will be able to sit down
    together at a table of brotherhood.
</p>
```

Finding Text


```
# import modules
import requests
from bs4 import BeautifulSoup

# create url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# get lines
lines = soup.find_all('p')

# get type
print(type(lines))
```

Finding Text

This a `bs4.element.ResultSet` object which is simply a collection of tags. Note, this is a `bs4` object, so most standard functions will not work.

```
# inspect first element
print(lines[0])

# output
'<p>Delivered on the steps at the Lincoln Memorial
    in Washington D.C. on August 28, 1963 </p>'

# get text
print(lines[0].get_text())

# output
'Delivered on the steps at the Lincoln Memorial in
    Washington D.C. on August 28, 1963'
```

"I Have A Dream"


```python
# import modules
import requests
from bs4 import BeautifulSoup

# create url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# get speech
speech = " "
lines = soup.find_all('p')
for line in lines:
    speech += line.get_text()
```

Preprocessing

Preprocessing $\rightsquigarrow$ Simplify text in order to make it useful.

1. Remove capitalization, punctuation
2. Discard word order (Bag of Words Assumption)
3. Discard stop words
4. Create equivalence class: Stem, lemmatize, or synonym
5. Discard less useful features
6. Other reduction, specialization

Step 1: Removing Capitalization and Punctuation

Removing capitalization:

- Python : `string.lower()`
- R : `tolower('string')`

Removing punctuation

- Python: `re.sub('\W', ' ', string)`
- R : `gsub('\\W', ' ', string)`

Step 1: Removing Capitalization and Punctuation


```
# import modules
import re

# create sentence
sentence = 'Five score years ago, a great American, in whose symbolic shadow we stand signed the Emancipation Proclamation.'
sentence2 = sentence.lower()
sentence3 = re.sub('\W', ' ', sentence2)

# output
'five score years ago a great american in whose symbolic shadow we stand signed the emancipation proclamation'
```

Step 1: Removing Capitalization and Punctuation

```
#import modules
import re

#create sentence
sentence = 'Five score years ago, a great American,
    in whose symbolic shadow we stand signed the
    Emancipation Proclamation.'
sentence = re.sub('\W', ' ', sentence.lower())
print(sentence)

five score years ago a great american in whose
    symbolic shadow we stand signed the emancipation
     proclamation
```

Step 2: "Bag of Words Assumption"

Assumption: Discard Word Order

Five score years ago, a great American, in whose symbolic shadow we stand signed the Emancipation Proclamation. five score years ago a great american in whose symbolic shadow we stand signed the emancipation proclamation Unigrams

Unigram	Count
the	101
of	96
to	57
and	44
a	36
be	31
will	26
that	24
is	21
freedom	19
in	19
from	18

Bigrams

Bigram	Count
five score	1
score years	1
years ago	1
ago a	1
a great	1
great american	1
american in	1
in whose	1
whose symbolic	1
symbolic shadow	1
shadow we	1
we stand	1

Trigrams

Step 2: "Bag of Words Assumption"


```
# activate virtual environment
source activate python-UI

# install nltk
python -m pip install nltk

# download stop words
nltk.download('stopwords')

# download wordnet
nltk.download('wordnet')
```

Step 2: "Bag of Words Assumption"

```
# import modules
import requests
import re
from bs4 import BeautifulSoup
from nltk import FreqDist
from nltk import word_tokenize
from nltk import bigrams
from nltk import trigrams
```

Step 2: “Bag of Words Assumption”

```
# url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# get text
speech = ""
lines = soup.find_all('p')
for line in lines:
    speech += line.get_text()

# remove punctuation
speech = re.sub('\W', ' ', speech.lower())
```

Step 2: "Bag of Words Assumption"

```
# get unigrams
words = word_tokenize(speech.lower())
words_frequency = FreqDist(words)
words_frequency.most_common(10)

# get bigrams
list(bigrams(words))

# get trigrams
list(trigrams(words))
```

How Could This Possibly Work?

Three answers

1) It might not: Validation is critical (task specific)
2) Central Tendency in Text: Words often imply what a text is about `war, civil, union` or tone `consecrate, dead, died, lives`.

   Likely to be used repeatedly: create a theme for an article
3) Human supervision: Inject human judgement (coders): helps methods identify subtle relationships between words and outcomes of interest

   Dictionaries
   Training Sets

Step 3: Discard Stopwords

Stop Words: English Language place holding words

- `the, it, if, a, able, at, be, because...`
- Add "noise" to documents (without conveying much information)
- Discard stop words: focus on substantive words

Be Careful!

- `she, he, her, his`
- You may need to customize your stop word list $\rightsquigarrow$ abbreviations, titles, etc.

Step 3: Discard Stopwords


```python
# import modules
import requests
import re
import string
from bs4 import BeautifulSoup
from nltk import FreqDist
from nltk import word_tokenize
from nltk import bigrams
from nltk import trigrams
from nltk.corpus import stopwords
```

Step 3: Discard Stopwords


```
# url
url = 'http://avalon.law.yale.edu/20th_century/mlk01.asp'

# create soup
response = requests.get(url)
contents = response.content
soup = BeautifulSoup(contents, 'html.parser')

# get text
speech = ""
lines = soup.find_all('p')
for line in lines:
    speech += line.get_text()

# remove punctuation
speech = re.sub('\W', ' ', speech.lower())
```

Step 3: Discard Stopwords

```
# import modules
import string

# create word list
words = word_tokenize(speech.lower())

# remove stopwords
stopwords = stopwords.words('english')
clean_speech = filter(lambda x: x not in stopwords,
    words)
clean_speech2 = [word for word in words if word not
    in stopwords]
```

Step 4: Create an Equivalence Class of Words

Reduce dimensionality further

Comparing Stemming and Lemmatizing

Stemming algorithm:

- $\rightarrow$ Porter – most commonly used stemmer.
- $\rightarrow$ Lancaster – very aggressive, stems may not be interpretable.
- $\rightarrow$ Snowball (Porter 2) – essentially a better version of Porter.

Comparing Stemming Algorithms


```
# import modules
from nltk.corpus import stopwords
from nltk.stem.porter import *
from nltk.stem.lancaster import *
from nltk.stem.snowball import SnowballStemmer
from nltk.stem import WordNetLemmatizer
```

Comparing Stemming Algorithms

```
# sample_text
'five score years ago a great american in whose
    symbolic shadow we stand today signed the
    emancipation proclamation this momentous decree
    came as a great beacon light of hope to millions
     of negro slaves who had been seared in the
    flames of withering injustice it came as a
    joyous daybreak to end the long night of their
    captivity'
```

Porter


```
# use porter
stemmer = PorterStemmer()
porter_text = [stemmer.stem(word) for word in
    sample_words]

# output
'five score year ago a great american in whose
    symbol shadow we stand today sign the emancip
    proclam thi moment decre came as a great beacon
    light of hope to million of negro slave who had
    been sear in the flame of wither injustic it
    came as a joyou daybreak to end the long night
    of their captiv'
```

Lancaster


```
# use lancaster
stemmer = LancasterStemmer()
lancaster_text = [stemmer.stem(word) for word in
    sample_words]

# output
'fiv scor year ago a gre am in whos symbol shadow we
    stand today sign the emancip proclam thi mom
    decr cam as a gre beacon light of hop to mil of
    negro slav who had been sear in the flam of with
    injust it cam as a joy daybreak to end the long
    night of their capt'
```

Snowball


```
# use snowball
stemmer = SnowballStemmer('english')
snowball_text = [stemmer.stem(word) for word in
    sample_words]

# output
'five score year ago a great american in whose
    symbol shadow we stand today sign the emancip
    proclam this moment decre came as a great beacon
     light of hope to million of negro slave who had
     been sear in the flame of wither injustic it
    came as a joyous daybreak to end the long night
    of their captiv'
```

Finding Lower Dimensional Embeddings of Text

1) Task:
   - Embed our documents in a lower dimensional space
   - Visualize our documents
   - Inference about similarity
   - Inference about behavior
2) Supervised Learning:
   - Predict the values of one or more outputs or response variables $Y = (Y_1, \ldots, Y_m)$ for a given set of input or predictor variables $X^T = (X_1, \ldots, X_p)$
   - $x_i^T = (x_{i1}, \ldots, X_{ip})$ denotes the inputs for the i^{th} training case and $\hat{y}_i$ is the response measure.
   - "Student" presents an answer $\hat{y}_i$ for each x_i in the training sample, and the supervisor or "teacher" grades the answer.
   - Usually this requires some loss function $L(y, \hat{y})$, for example, $L(y, \hat{y}) = (y - \hat{y})^2$.

Finding Lower Dimensional Embeddings of Text

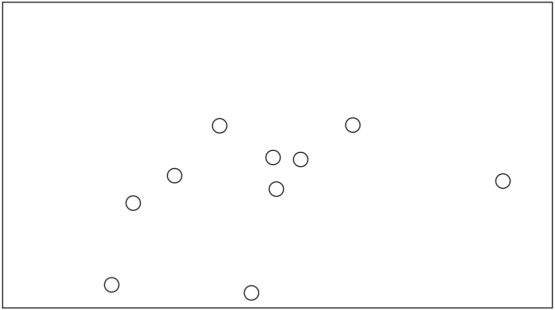
3) Unsupervised Learning:
   - In this case one has a set of N observations $(x_1, x_2, \ldots, x_N)$ of a random p-vector X having joint density $Pr(X)$
   - The goal is to directly infer the properties of this probability density without the help of a supervisor or teacher.
   - The dimension of X is sometimes much higher than in supervised learning, and the properties of interest are often more complicated than simple point estimates.

k-Means

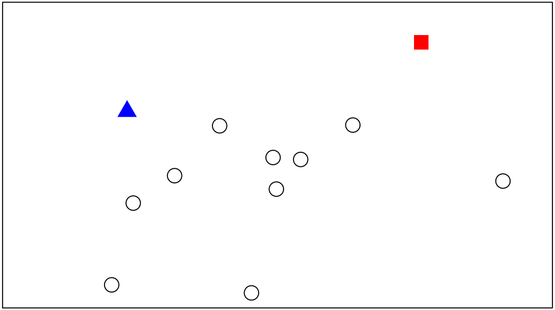
Suppose, we have a dataset with two variables $x = (1, 2, 3, 4, 5)$ and $y = (1, 2, 3, 4, 5)$:

1. Randomly place k centroids inside the two-dimensional space (X,Y).
2. For each point (x_i, y_i) find the nearest centroid by minimizing some distance measure.
3. Assign each point (x_i, y_i) to cluster j.
4. For each cluster $j = 1 \ldots K$:
    - Create a new centroid c_j using the average across all points x_i and y_i

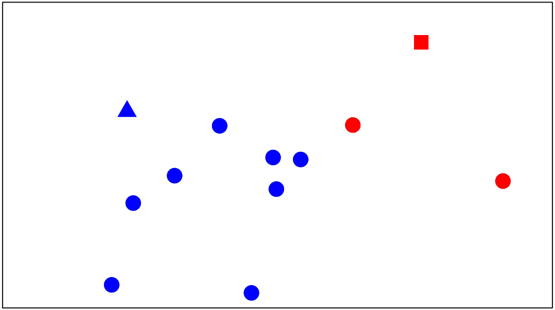
k-Means



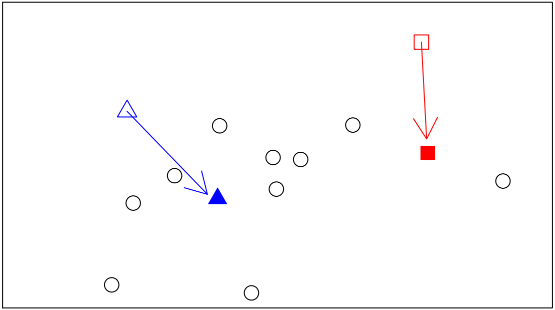
k-Means



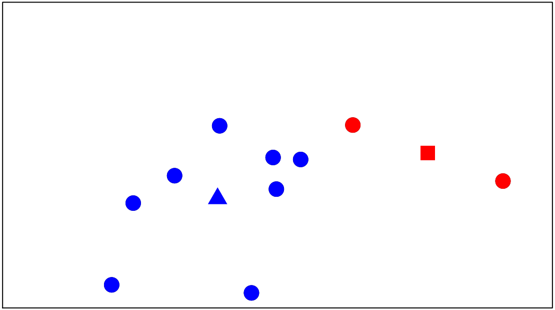
k-Means



k-Means

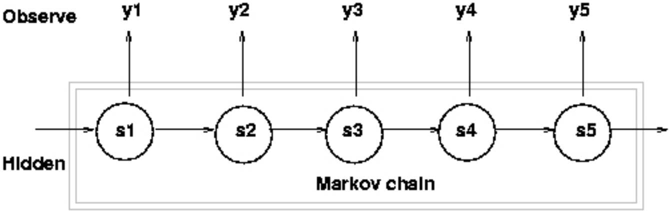


k-Means



Expectation-Maximization Algorithm

The **E**xpectation **M**aximization algorithm enables parameter estimation in probabilistic models with incomplete data.



Expectation-Maximization Algorithm

The Expectation **M**aximization algorithm enables parameter estimation in probabilistic models with incomplete data.

- exponential family of distributions:
  - Normal
  - Exponential
  - Gamma
  - Chi-Squared
  - Beta
  - Dirichlet (Der-rick-let)
  - Bernoulli
  - Poisson
- $P_{\theta_{t+1}}(X) \geq P_{\theta_t}(X)$

Expectation-Maximization Algorithm

The Expectation **M**aximization algorithm enables parameter estimation in probabilistic models with incomplete data.

- exponential family of distributions:
- Not guaranteed to give θ_{MLE}
- Overfitting
- Slow
- Generally, it can't be used for non-exponential distributions.

Gaussian Mixture Model

Let's assume we have observations $x_1 \ldots x_n$:

- Each x_i is drawn from one of two normal distributions.
- One of these distributions (red) has a mean of μ_{red} and a variance of σ^2_{red}.
- The other distribution (blue) has a mean of μ_{blue} and a variance of σ^2_{blue}.
- If we know the source of each observation, then estimating μ_{red}, μ_{blue}, σ^2_{red}, and σ^2_{blue} is trivial.

Gaussian Mixture Model

Let's assume we have observations $x_1 \ldots x_n$ drawn from the *red* distribution:

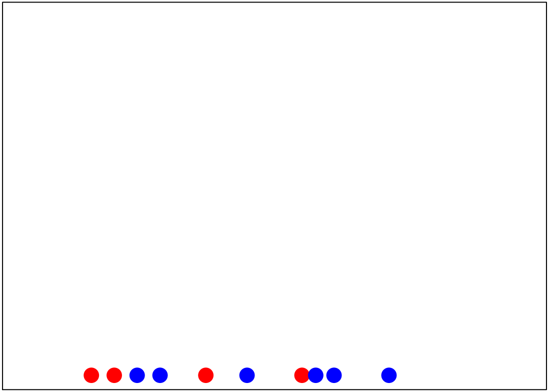
- $\mu_{red} = \frac{x_1 + x_2 + \cdots + x_n}{n_{red}}$
- $\sigma^2_{red} = \frac{(x_1 - \mu_{red})^2 + \cdots + (x_n - \mu_{red})^2}{n_{red}}$

Gaussian Mixture Model

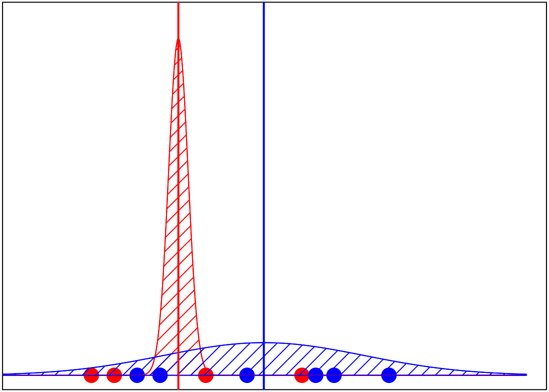
Let's assume we have observations $x_1 \dots x_n$ drawn from the *blue* distribution:

- $\mu_{blue} = \frac{x_1 + x_2 + \cdots + x_n}{n_{blue}}$
- $\sigma^2_{blue} = \frac{(x_1 - \mu_{blue})^2 + \cdots + (x_n - \mu_{blue})^2}{n_{blue}}$

Gaussian Mixture Model

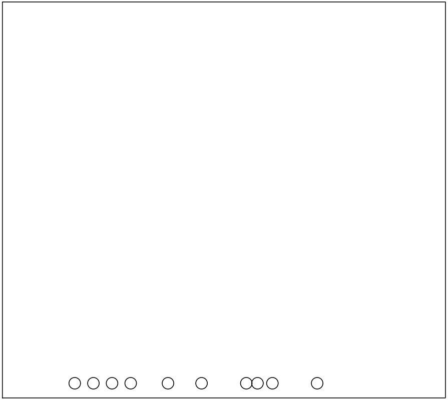


Gaussian Mixture Model



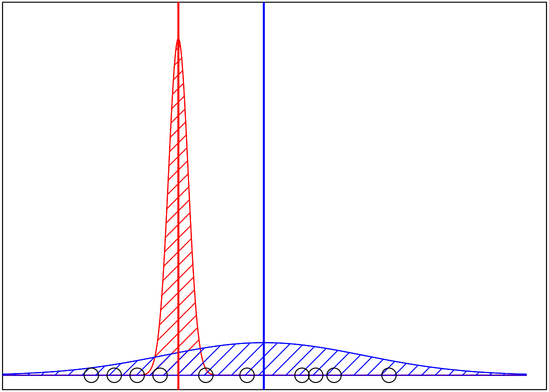
Gaussian Mixture Model

However, what if we do not know the source?



Gaussian Mixture Model

Why don't we just guess?



Gaussian Mixture Model

Let's assume someone gave you the parameters μ_{red} and σ^2_{red}, what is the probability a given point, x_i, is from that distribution? (Normal PDF)

$$P(x_i|red) = \frac{1}{\sqrt{2\pi\sigma^2_{red}}} \exp\left(-\frac{(x_i - \mu_{red})^2}{2\sigma^2_{red}}\right) \quad (1)$$

Gaussian Mixture Model

What is the probability the parameters μ_{red} and σ^2_{red} are correct, given point x_i? (Bayes' Rule)

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

$$P(Yes|x_i) = \frac{P(x_i|Yes)P(Yes)}{P(x_i|Yes)P(Yes) + P(x_i|No)P(No)} \tag{3}$$

Gaussian Mixture Model

What is the probability the parameters μ_{red} and σ^2_{red} are correct, given point x_i? (Bayes' Rule)

$$P(red|x_i) = \frac{P(x_i|red)P(red)}{P(x_i|red)P(red) + P(x_i|blue)P(blue)} \tag{4}$$

$$P(blue|x_i) = 1 - P(red|x_i) \tag{5}$$

Gaussian Mixture Model

If you knew where the points came from, you could estimated μ and σ^2 easily. Unfortunately, you do not know where the points came from.

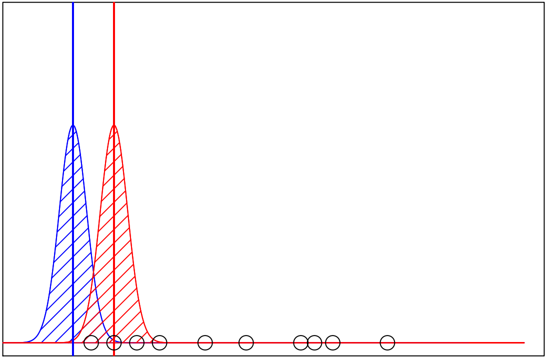
- If we knew μ_{red}, σ^2_{red}, μ_{blue}, and σ^2_{blue} we could figure out which distribution the points came from.
- EM Algorithm
  - Start with two randomly placed normal distributions (μ_{red}, σ^2_{red}) and (μ_{blue}, σ^2_{blue}).
  - For each x_i, determine $P(red|x_i) =$ the probability that the point was drawn from the *red* distribution.
  - This is a soft assignment, meaning that each x_i with have two probabilities: $P(red|x_i)$ and $P(blue|x_i)$.
  - Once this is done, re-estimate (μ_{red}, σ^2_{red}) and (μ_{blue}, σ^2_{blue}), given what we learned.
  - Iterate until it convergence.

Gaussian Mixture Model

If you knew where the points came from, you could estimated μ and σ^2 easily. Unfortunately, you do not know where the points came from.

- If we knew μ_{red}, σ^2_{red}, μ_{blue}, and σ^2_{blue} we could figure out which distribution the points came from.
- EM Algorithm
  - Start with two randomly placed normal distributions (μ_{red}, σ^2_{red}) and (μ_{blue}, σ^2_{blue}).
  - For each x_i, determine $P(red|x_i) =$ the probability that the point was drawn from the *red* distribution (E-STEP).
  - This is a soft assignment, meaning that each x_i with have two probabilities: $P(red|x_i)$ and $P(blue|x_i)$.
  - Once this is done, re-estimate (μ_{red}, σ^2_{red}) and (μ_{blue}, σ^2_{blue}), given what we learned (M-STEP).
  - Iterate until it convergence.

EM Example



EM Example

To use the EM algorithm, we have to answer several questions:

1. How Likely is Each of the Points to Come From the Red Distribution?

$$P(x_i|red) = \frac{1}{\sqrt{2\pi\sigma^2_{red}}} \exp\left(-\frac{(x_i - \mu_{red})^2}{2\sigma^2_{red}}\right)$$

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?

x_i	$P(x_i \mid red)$
5	0.03316
10	0.13298
15	0.03316
20	0.00051
30	0
39	0
51	0
54	0
58	0
70	0

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?

$$P(red_i|x_i) = \frac{P(x_i|red)P(red)}{P(x_i|red)P(red) + P(x_i|blue)P(blue)}$$

$$P(red_i|x_i) = \frac{P(x_i|red).50}{P(x_i|red).50 + P(x_i|blue).50}$$

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?

x_i	$P(red_i \mid x_i)$
5	0.01658
10	0.06649
15	0.01658
20	0.00026
30	0
39	0
51	0
54	0
58	0
70	0

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?
3. How Likely is it the Blue Distribution is specified correctly, given x_i?

$$P(blue_i|x_i) = 1 - P(red_i|x_i)$$

EM Example

How Likely is it the Blue Distribution is specified correctly, given x_i?

x_i	$P(blue_i \mid x_i)$
5	0.98342
10	0.93351
15	0.98342
20	0.99974
30	1
39	1
51	1
54	1
58	1
70	1

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?
3. How Likely is it the Blue Distribution is specified correctly, given x_i?
4. Given what we know what is the likely *red* mean (μ_{red}) and variance (σ^2_{red})?

$$\mu_{red} = \frac{red_1 x_1 + red_2 x_2 + \cdots + red_n x_n}{red_1 + red_2 + \ldots red_n}$$

$$\sigma^2_{red} = \frac{red_1(x_1 - \mu_{red})^2 + \cdots + red_n(x_n - \mu_{red})^2}{red_1 + red_2 + \ldots red_n}$$

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?
3. How Likely is it the Blue Distribution is specified correctly, given x_i?
4. Given what we know what is the likely *red* mean (μ_{red}) and variance (σ^2_{red})?

$$\hat{\mu}_{red} = 10.02573$$

$$\hat{\sigma}^2_{red} = 8.554147$$

EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?
3. How Likely is it the Blue Distribution is specified correctly, given x_i?
4. Given what we know what is the likely *red* mean (μ_{red}) and variance (σ^2_{red})?
5. Given what we know what is the likely *blue* mean (μ_{blue}) and variance (σ^2_{blue})?

$$\mu_{blue} = \frac{blue_1 x_1 + blue_2 x_2 + \cdots + blue_n x_n}{blue_1 + blue_2 + \ldots blue_n}$$

$$\sigma^2_{blue} = \frac{blue_1 (x_1 - \mu_{blue})^2 + \cdots + blue_n (x_n - \mu_{blue})^2}{blue_1 + blue_2 + \ldots blue_n}$$

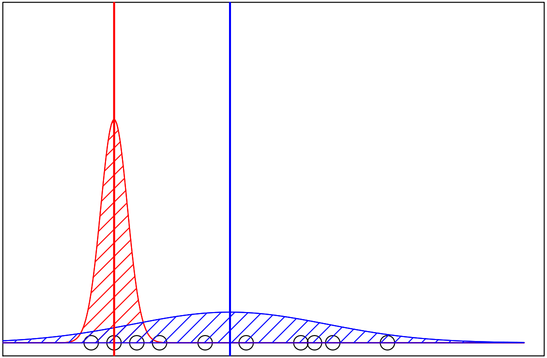
EM Example

1. How Likely is Each of the Points to Come From the Red Distribution ($\mu_{red} = 10$, $\sigma^2_{red} = 9$)?
2. How Likely is it the Red Distribution is specified correctly, given x_i?
3. How Likely is it the Blue Distribution is specified correctly, given x_i?
4. Given what we know what is the likely *red* mean (μ_{red}) and variance (σ^2_{red})?
5. Given what we know what is the likely *blue* mean (μ_{blue}) and variance (σ^2_{blue})?

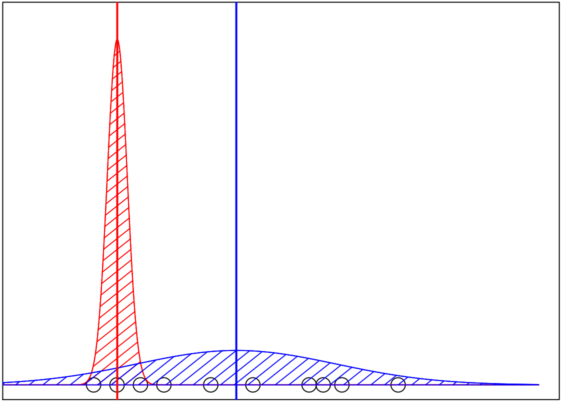
$$\mu_{blue} = 35.45405$$

$$\sigma^2_{blue} = 454.2171$$

EM Example



EM Example



Topic and Mixed Membership Models (Grimmer)

Doc 1

Doc 2

Doc 3

$\vdots$

Doc N

Cluster 1

Cluster 2

$\vdots$

Cluster K

Latent Dirichlet Allocation (LDA)

Suppose, we have the following sentences:

- Donald Trump is running for president.
- Donald Trump debated last night.
- Hillary Clinton is running for president.
- Hillary Clinton debated last night.
- Hillary Clinton debated better than Donald Trump last night.

Latent Dirichlet Allocation (LDA)

Suppose, we have the following sentences:

- Donald Trump is running for president. (50% Topic A, 50% Topic C)
- Donald Trump debated last night. (50% Topic A, 50% Topic D)
- Hillary Clinton is running for president. (50% Topic B, 50% Topic C)
- Hillary Clinton debated last night. (50% Topic B, 50% Topic D)
- Hillary Clinton debated better than Donald Trump last night. (33% Topic A, 33% Topic B, 33% Topic D)

Latent Dirichlet Allocation (LDA)

Suppose, we have the following sentences:

- Donald Trump is running for president. (50% Topic A, 50% Topic C)
- Donald Trump debated last night. (50% Topic A, 50% Topic D)
- Hillary Clinton is running for president. (50% Topic B, 50% Topic C)
- Hillary Clinton debated last night. (50% Topic B, 50% Topic D)
- Hillary Clinton debated better than Donald Trump last night. (33% Topic A, 33% Topic B, 33% Topic D)

Latent Dirichlet Allocation (LDA)

Suppose, we have the following sentences:

- Donald Trump is running for president. (50% Topic A, 50% Topic C)
- Donald Trump debated last night. (50% Topic A, 50% Topic D)
- Hillary Clinton is running for president. (50% Topic B, 50% Topic C)
- Hillary Clinton debated last night. (50% Topic B, 50% Topic D)
- Hillary Clinton debated better than Donald Trump last night. (33% Topic A, 33% Topic B, 33% Topic D)

Latent Dirichlet Allocation (LDA)

Suppose, we have the following sentences:

- Donald Trump is running for president. (50% Topic A, 50% Topic C)
- Donald Trump debated last night. (50% Topic A, 50% Topic D)
- Hillary Clinton is running for president. (50% Topic B, 50% Topic C)
- Hillary Clinton debated last night. (50% Topic B, 50% Topic D)
- Hillary Clinton debated better than Donald Trump last night. (33% Topic A, 33% Topic B, 33% Topic D)

Latent Dirichlet Allocation (LDA)

LDA represents documents as mixtures of topics that produce words with certain probabilities. Imagine you are writing an article:

1. How many N words will your article have? (Let's assume this follows a Poisson distribution.)
2. What is the mixture of K topics? (Let's assume this follows a Dirichlet distribution.)
3. For each word in your article:
   - First, pick a topic from the distribution outlined above.
   - Second, given the topic you have selected, choose the word that appears the most.

Latent Dirichlet Allocation (LDA)

LDA represents documents as mixtures of topics that produce words with certain probabilities. Imagine you are writing an article:

1.) Let's assume your article will have 5 words.

2.) Let's assume our topic distribution will be 75% Topic B and 25% Topic A.

3.) Let's assume "Hillary Clinton," "president," and "win," appears in 75%, 50%, and 25% of the documents that are included in Topic B, respectively.

4.) Let's assume the "Donald Trump" and "president" appears in 75% and 50% of the documents that are included in Topic A, respectively.

Latent Dirichlet Allocation (LDA)

LDA represents documents as mixtures of topics that produce words with certain probabilities. Imagine you are writing an article:

4.) Let's assume the "Donald Trump" and "president" appears in 75% and 50% of the documents that are included in Topic A, respectively.
  - 1^{st} word comes from Topic B, which then gives you "Hillary Clinton."
  - 2^{nd} word comes from Topic A, which then gives you "Donald Trump."
  - 3^{rd} word comes from Topic A, which then gives you "president."
  - 4^{th} word comes from Topic B, which then gives you "president."
  - 5^{th} word comes from Topic B, which then gives you "win."
  - "Hillary Clinton Donald Trump president president win."

Harmonic Mean

Wallach et al. (2009) suggest the harmonic mean could be considered a goodness-of-fit test for the LDA model:

$$\frac{1}{M}\sum_{i=1}^{M}\left(\frac{1}{K}\sum_{i=1}^{K}\theta_{m,k}\right)^{-1} \tag{6}$$

, where $\theta_{m,k}$ is the topic distribution for document m and topic k.

- The degree of differentiation of a distribution θ over all topics is theoretically captured by the harmonic mean.
- If the topic distribution has a high probability for only a few topics, then it would have a lower harmonic mean value $\rightsquigarrow$ the model has a better ability to separate documents into different topics.

Perplexity

Heinrich (2005) suggests perplexity as a way to prevent overfitting an LDA model. Perplexity is equivalent to the geometric mean per-word likelihood:

$$Perplexity(w) = exp\left\{-\frac{log(p(w))}{\sum_{d=1}^{D}\sum_{j=1}^{V} n^{jd}}\right\} \tag{7}$$

, where n_{jd} denotes how often the j^{th} term occurred in the d^{th} document.

- Perplexity is essentially the reciprocal geometric mean of the likelihood of testing data given the trained model M.
- Therefore, the lower perplexity value indicates that the model could fit the testing data better.

Entropy

The entropy measure can also be used to indicate how the topic distributions differ across LDA models. Higher values indicate that the topic distributions are more evenly spread over the topics.

```
sapply(fitted_models, function(x) mean(apply(
    posterior(x)$topics,1, function(z) - sum(z * log
    (z)))))
```

Entropy

The entropy measure can also be used to indicate how the topic distributions differ across LDA models. Higher values indicate that the topic distributions are more evenly spread over the topics.

```
entropy<-NULL
for(i in 1:length(fitted_models)){
  temp_topics<-posterior(fitted_models[[i]])$topics
  temp_sums<-NULL
  for(j in 1:NROW(temp_topics)){
    temp_sums<-c(temp_sums,sum(temp_topics[j,]*log(temp_topics[j,])))
  }
  entropy<-c(entropy,mean(-temp_sums))
}
```

Applying LDA to NYT Articles

R!

Cluster Quality (Grimmer and King 2011)

Assessing Cluster Quality with experiments

- Goal: group together similar documents
- Who knows if similarity measure corresponds with semantic similarity

$\rightsquigarrow$ Inject human judgement on pairs of documents

Design to assess cluster quality

- Sample pairs of documents
- Scale: (1) unrelated, (2) loosely related, (3) closely related
- Cluster Quality = mean(within cluster) - mean(between clusters)
- Select clustering with highest cluster quality

What Can We Do?

Additional Resources

- Webscraping – `http://web.stanford.edu/~zlotnick/TextAsData/Web_Scraping_with_Beautiful_Soup.html`
- Machine Learning – *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* by Hastie et al. 2009
- Text Analysis – `https://aeshin.org/textmining/`

Questions?

Questions?