Natural Language Processing (NLP) application.

Document Clustering (unsupervised learning)

- You have a bunch of documents
- A document is text. Short docs: tweets, longer docs: newspaper articles, book chapters.
- algorithm will "read" documents and try to group similar ones together
- no predefined clusters. The algorithm must discover clusters on its own.
- we do need to specify how many clusters to search for.

Document clustering is generally a harder problem than document classification (supervised learning)

Training data has classes labeled. We want to classify a new test document.

Article: "Gibbs sampling for the uninitiated."

Simple example: We will search for 2 clusters in the documents.

Notation:

$\mathbb{C}$ = corpus = all the documents we have

$\mathbb{C}_0$ = all documents with label 0.

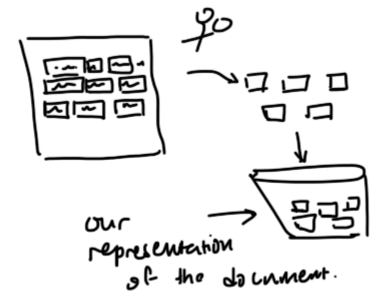
C_0 = count of documents labeled 0.

$\mathbb{C}_1$ = all documents labeled 1

C_1 = count of documents labeled 1.

We record the documents using the bag of words model.

Imagine the article printed on paper. You cut out each word and put them in a bag.



- order of words do not matter.
- documents are treated as a list of words and frequencies.

shortcoming: The following documents have different meanings but have the same representation.

"I am not upset. I am happy." → am: 2, happy: 1, I: 2, not: 1, upset: 1 ← doc representation.

"I am not happy. I am upset." →

Bag of words model works best if there is a strong link between words and subjects/topics.

A very simplified document creation model. (documents with 2 possible classes: 0, 1)

Bernoulli Trial
π

π is prob of 1 (or 0)

Class 0: fruit
class 1: sports

If class = 0, we have a word-probability lexicon associated with class 0.

If class = 1, we have a different word-prob lexicon.

lexicon for class 0

- apple: .015
- banana: .018
- ball: .0001
- goal: .0002
- ⋮

multinomial distribution to generate a bag of words

lexicon for Class 1

- apple: .0001
- banana: .00015
- ball: .016
- goal: .013
- ⋮

multinomial dist to generate a bag of words.

Binomial dist: n trials. Each trial has prob θ of success (or failure)

A binomial dist w/ $n=10$ and $p=0.5$. A random draw could be 6.

"how many times will a fair coin land heads if we flip it 10 times?"

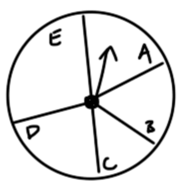
Conjugate prior for θ in the binomial is the Beta distribution.

The Beta distribution produces a value between 0 and 1 that can be used in binomial.

Multinomial generalizes to more than 2 categories.

- n trials. A vector $\vec{\theta}$ of length k of probabilities of the possible classes.

$\vec{\theta} = (\theta_1, \theta_2, \ldots \theta_k)$ requirement: $\sum_{i=1}^{k} \theta_i = 1$



$\vec{\theta} = (.2, .18, .12, .24, .26)$

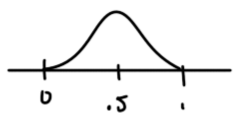
Spin the wheel 20 times. How many times did it land on each space.

One possible random draw: A: 4, B: 3, C: 3, D: 6, E: 4

Conjugate prior for the Multinomial dist is the Dirichlet distribution.

The Dirichlet dist produces a random vector $\vec{\theta}$ (that sums to 1) that can be used in the multinomial dist.

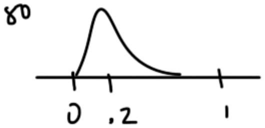
Beta dist has 2 parameters: α, β. The parameters α and β can be thought of as pseudo counts of success and failure



$\alpha = 5$
$\beta = 5$

$\alpha = 500$
$\beta = 500$

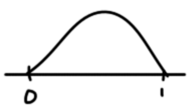
$\alpha = 20$
$\beta = 80$



Dirichlet has a vector $\vec{\alpha}$ of length k of pseudo counts for each category.

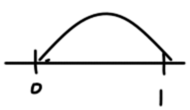
$\vec{\alpha} = (2, 2, 2)$

cat 1



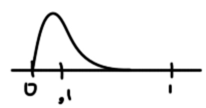
cat 2

cat 3

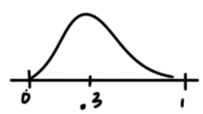


$\vec{\alpha} = (2, 6, 12)$

cat 1



cat 2



cat 3

a random draw from Dirichlet (2, 6, 12)
Could be $\vec{\theta} = (.12, .31, .57)$

another random draw
Could be $\vec{\theta} = (.09, .28, .63)$