Kevin Knight, http://www.isi.edu/natural-language/people/pictures/ieee-expert-1.gif

Modeling Natural Text

David Kauchak
CS159 – Spring 2019

Admin

Projects
- Status report due Sunday

Schedule for the rest of the semester
- Monday (4/29): text simplification
- Wednesday (5/1): ethics
  - Post 1-2 papers to read
  - Discussion
- Monday (5/6): **1 hr quiz** + presentation info
- Wednesday (5/8): project presentations

Document Modeling

http://whatshaute.com/index.php/2011/05/win-this-limited-edition-silk-scarf-and-inside-book-by-best-selling-author-brenda-novak/brenda-novak-scarf-inside-book-giveaway-model-front/

Modeling natural text

You're goal is to create a probabilistic model of natural (human) text

What are some of the questions you might want to ask about a text?

What are some of the phenomena that occur in natural text that you might need to consider/model?

Modeling natural text

Questions

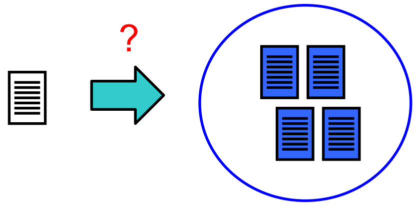
- what are the key topics in the text?
- what is the sentiment of the text?
- who/what does the article refer to?
- what are the key phrases?
- …

Phenomena

- synonymy
- sarcasm/hyperbole
- variety of language (slang), mispellings
- coreference (e.g. pronouns like he/she)
- …

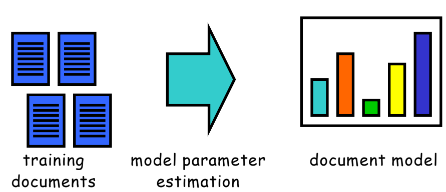
Document modeling: learn a probabilistic model of documents

Predict the likelihood that an unseen document belongs to a set of documents



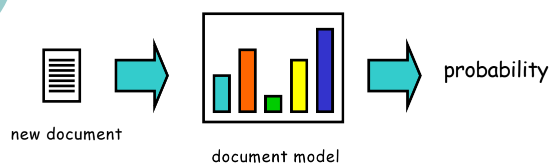
Model should capture text characteristics

Training a document model



Applying a document model

Document model: what is the probability the new document is in the same "set" as the training documents?



Document model applications

?

Applications

search engines

- search
- advertising
- corporate databases

language generation

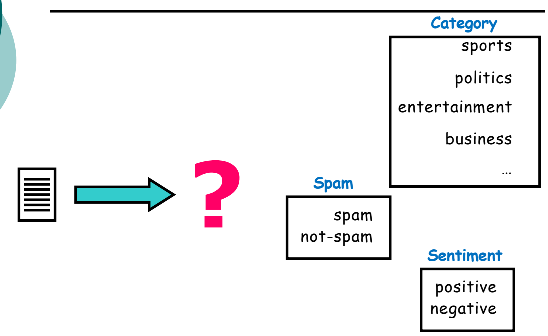
- speech recognition
- machine translation
- text simplification

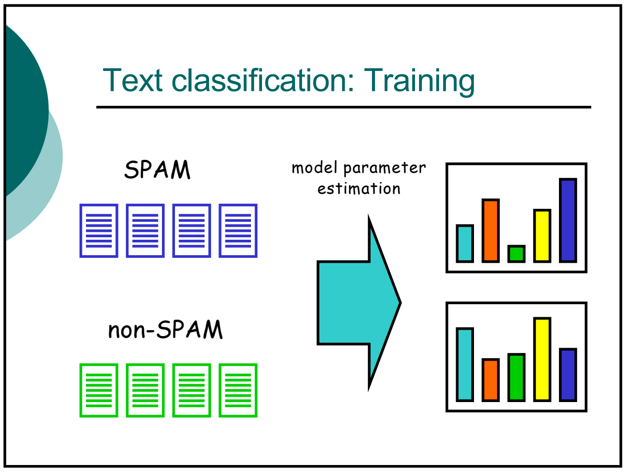
I think, therefore I am → I am

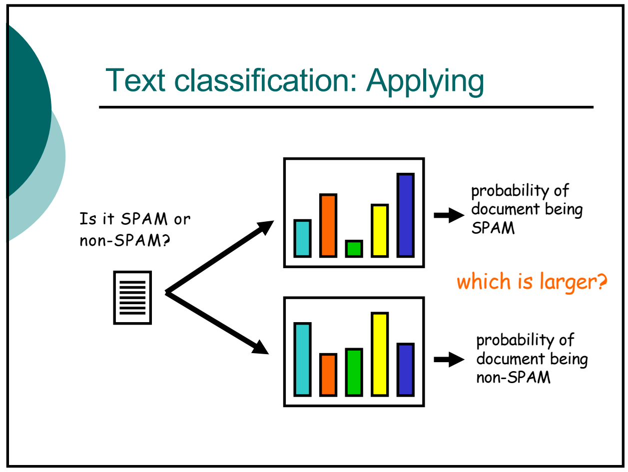
text classification and clustering

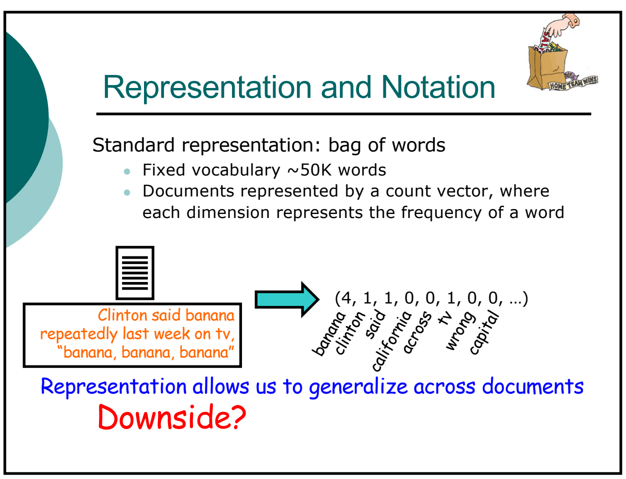
- SPAM
- document hierarchies
- sentiment analysis

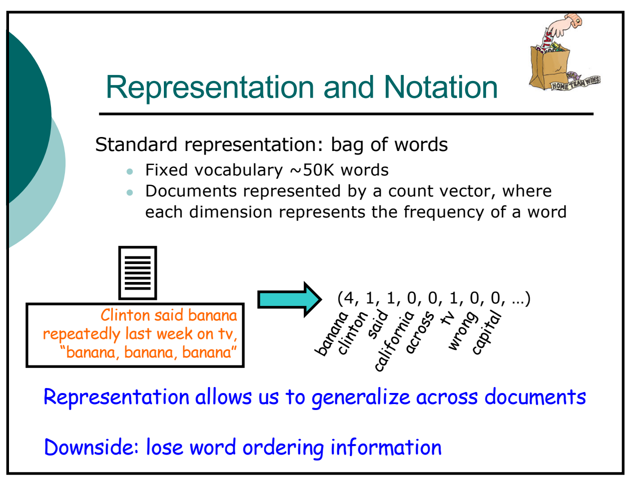
Application: text classification




Text classification: Training
SPAM
non-SPAM
model parameter estimation


Text classification: Applying
Is it SPAM or non-SPAM?
probability of document being SPAM
which is larger?
probability of document being non-SPAM


Representation and Notation
Standard representation: bag of words
Fixed vocabulary ~50K words
Documents represented by a count vector, where each dimension represents the frequency of a word
Clinton said banana repeatedly last week on tv, "banana, banana, banana"
(4, 1, 1, 0, 0, 1, 0, 0, …)
banana clinton said california across tv wrong capital
Representation allows us to generalize across documents
Downside?


Representation and Notation
Standard representation: bag of words
Fixed vocabulary ~50K words
Documents represented by a count vector, where each dimension represents the frequency of a word
Clinton said banana repeatedly last week on tv, "banana, banana, banana"
(4, 1, 1, 0, 0, 1, 0, 0, …)
banana clinton said california across tv wrong capital
Representation allows us to generalize across documents
Downside: lose word ordering information

Word burstiness

What is the probability that a political document contains the word "Clinton" *exactly* once?

The Stacy Koon-Lawrence Powell defense! The decisions of Janet Reno and Bill **Clinton** in this affair are essentially the moral equivalents of Stacy Koon's. …

p("**Clinton**"=1|political)= 0.12

Word burstiness

What is the probability that a political document contains the word "Clinton" *exactly* **twice**?

The Stacy Koon-Lawrence Powell defense! The decisions of Janet Reno and Bill **Clinton** in this affair are essentially the moral equivalents of Stacy Koon's. Reno and **Clinton** have the advantage in that they investigate themselves.

p("**Clinton**"=2|political)= 0.05

Word burstiness in models

p("**Clinton**"=1|political)= 0.12

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_m!} \prod_{j=1}^{m} \theta_j^{x_j}$$

Under the multinomial model, how likely is p("Clinton" = 2 | political)?

Word burstiness in models

p("**Clinton**"=2|political)= 0.05

Many models incorrectly predict:

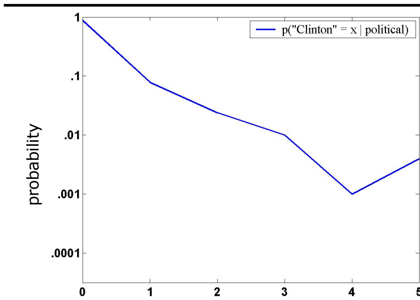
p("Clinton"=2|political) ≈ p("Clinton"=1|political)^{2}

0.05 ≠ ***0.0144*** (0.12^2)

And in general, predict:

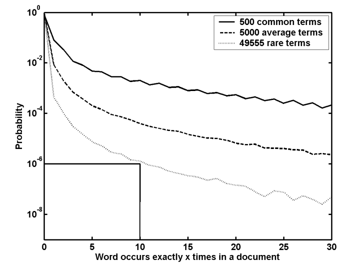
p("Clinton"=**i**|political) ≈ p("Clinton"=1|political)$^{\mathbf{i}}$

p(“Clinton” = x | political)



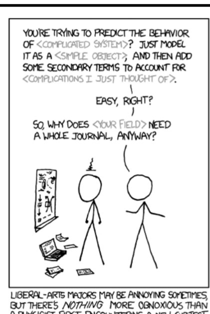
“Clinton” occurs exactly x times in document

Word count probabilities



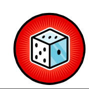
common words – 71% of word occurrences and 1% of the vocabulary
average words – 21% of word occurrences and 10% of the vocabulary
rare words – 8% or word occurrences and 89% of the vocabulary

The models…



https://xkcd.com/793/

Multinomial model



20 rolls of a fair, 6-side die –
each number is equally probable

(1, 10, 5, 1, 2, 1)
ones twos threes fours fives sixes

(3, 3, 3, 3, 4, 4)
ones twos threes fours fives sixes

Which is more probable?

Multinomial model

20 rolls of a fair, 6-side die –
each number is equally probable

(1, 10, 5, 1, 2, 1)
ones twos threes fours fives sixes

(3, 3, 3, 3, 4, 4)
ones twos threes fours fives sixes

How much more probable?

Multinomial model

20 rolls of a fair, 6-side die –
each number is equally probable

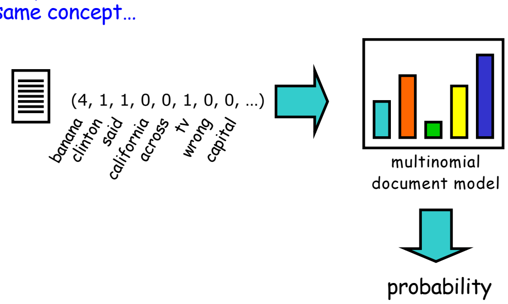
(1, 10, 5, 1, 2, 1)
0.000000764

(3, 3, 3, 3, 4, 4)
0.000891

1000 times more likely

Multinomial model for text

Many more "sides" on the die than 6, but the same concept…



Generative Story

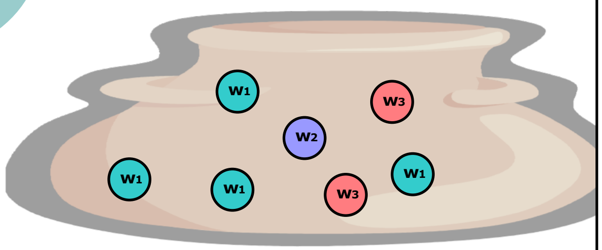
To apply a model, we're given a document and we obtain the probability

We can also ask how a given model would *generate* a document

This is the "generative story" for a model

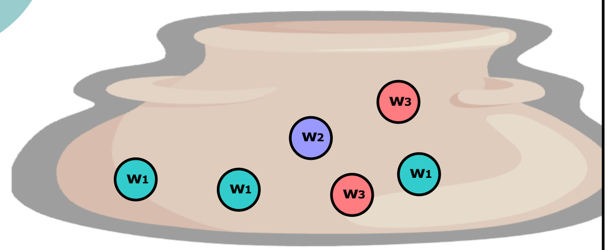
Multinomial Urn: Drawing words from a multinomial

Selected:



Drawing words from a multinomial

Selected: w_1

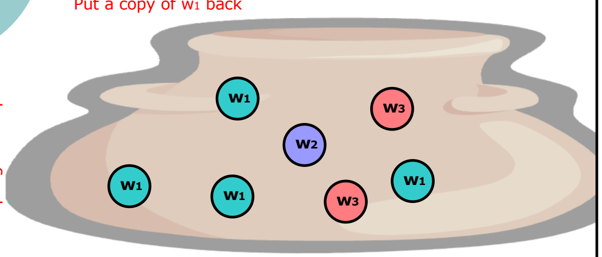


Drawing words from a multinomial

Selected: w_1

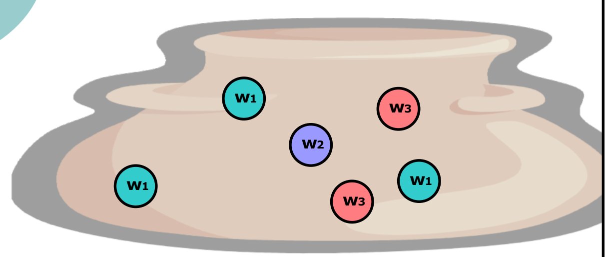
Put a copy of w_1 back

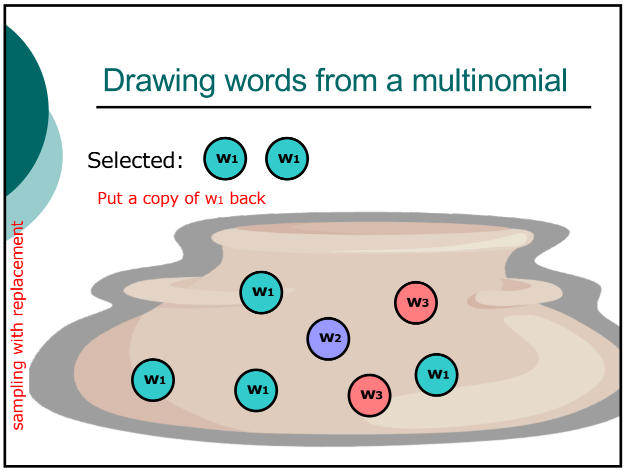
sampling with replacement

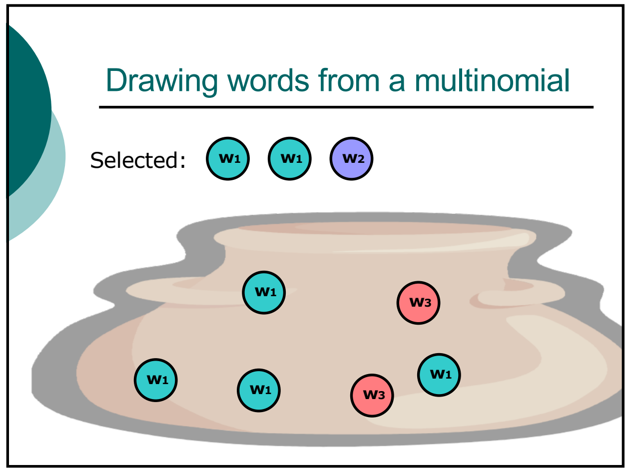


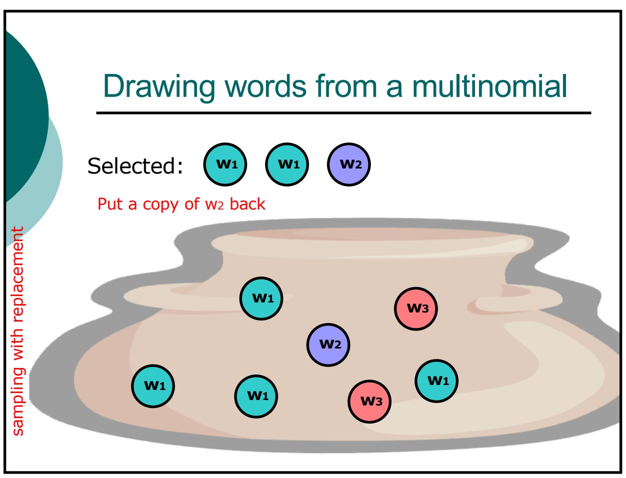
Drawing words from a multinomial

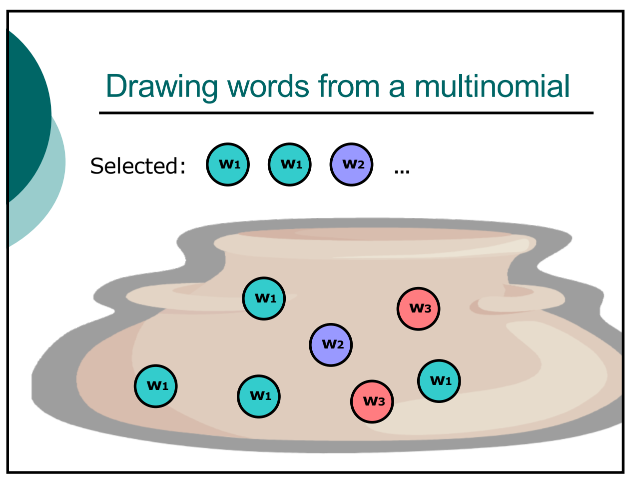
Selected: w_1 w_1




Drawing words from a multinomial
Selected: w_1 w_1
Put a copy of w_1 back
sampling with replacement
w_1 w_3 w_2 w_1 w_1 w_3 w_1

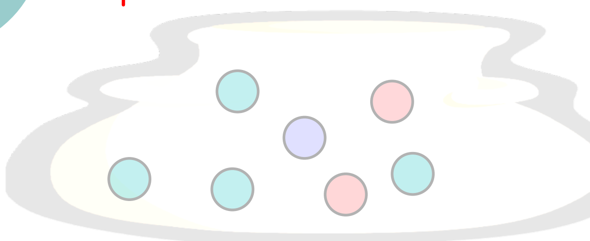

Drawing words from a multinomial
Selected: w_1 w_1 w_2
w_1 w_3 w_1 w_1 w_3 w_1


Drawing words from a multinomial
Selected: w_1 w_1 w_2
Put a copy of w_2 back
sampling with replacement
w_1 w_3 w_2 w_1 w_1 w_3 w_1


Drawing words from a multinomial
Selected: w_1 w_1 w_2 …
w_1 w_3 w_2 w_1 w_1 w_3 w_1

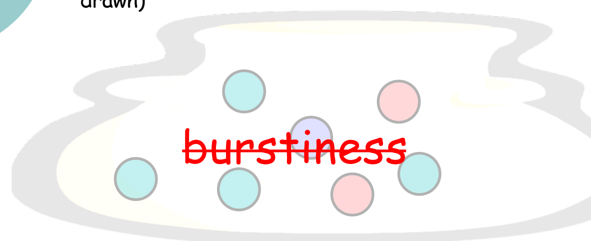
Drawing words from a multinomial

Does the multinomial model capture burstiness?



Drawing words from a multinomial

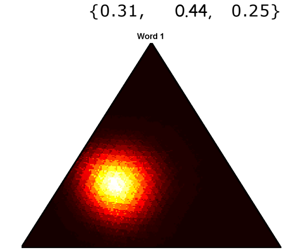
p(word) remains constant, independent of which words have already been drawn (in particular, how many of this particular word have been drawn)



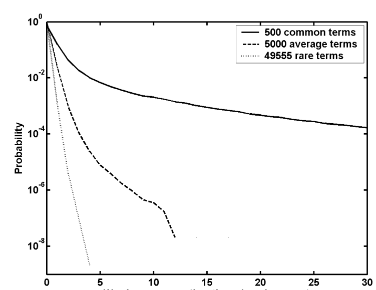
Multinomial probability simplex

Generate documents containing 100 words from a multinomial with just 3 possible words

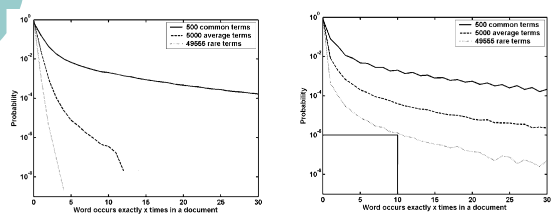
word 1	word 2	word 3
{0.31,	0.44,	0.25}



Multinomial word count probabilities



Multinomial does not model burstiness of average and rare words



Better model of burstiness: DCM

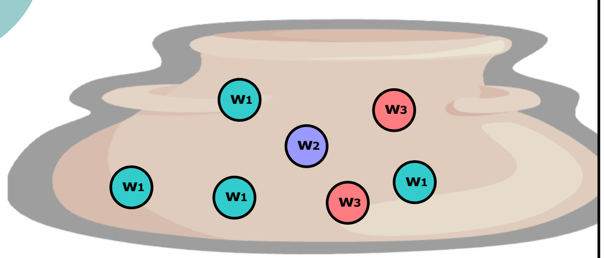
Dirichlet Compound Multinomial

Polya Urn process

- **KEY**: Urn distribution changes based on previous words drawn
- Generative story:
  - Repeat until document length hit
    - Randomly draw a word from urn – call it ***wi***
    - Put **2** copies of ***wi*** back in urn

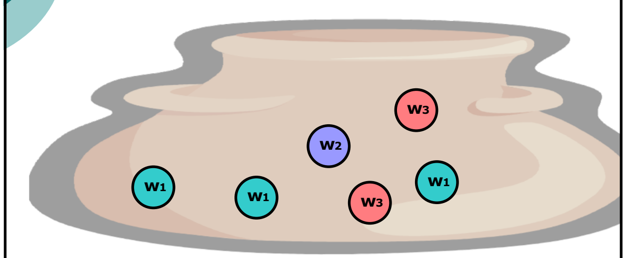
Drawing words from a Polya urn

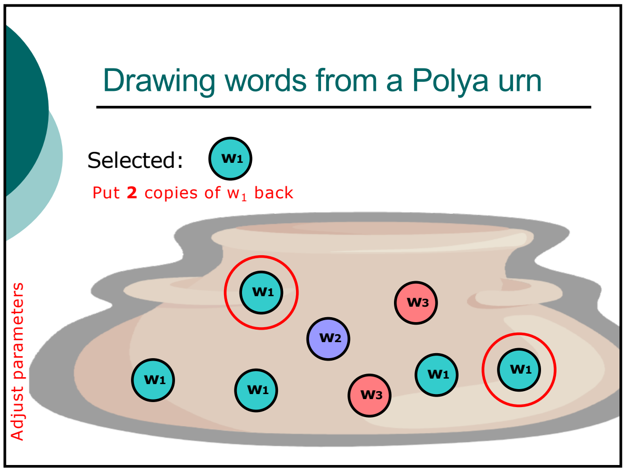
Selected:

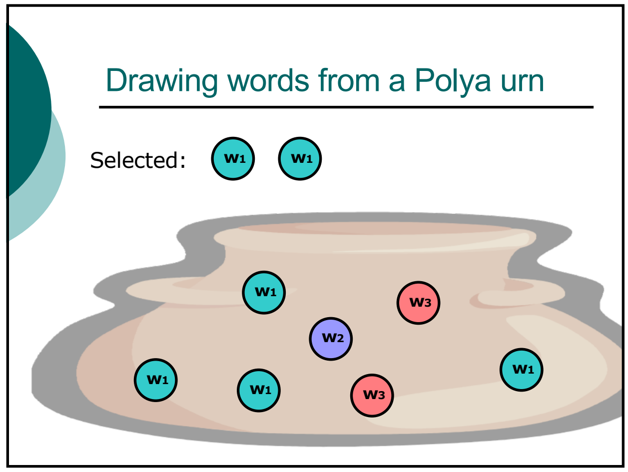


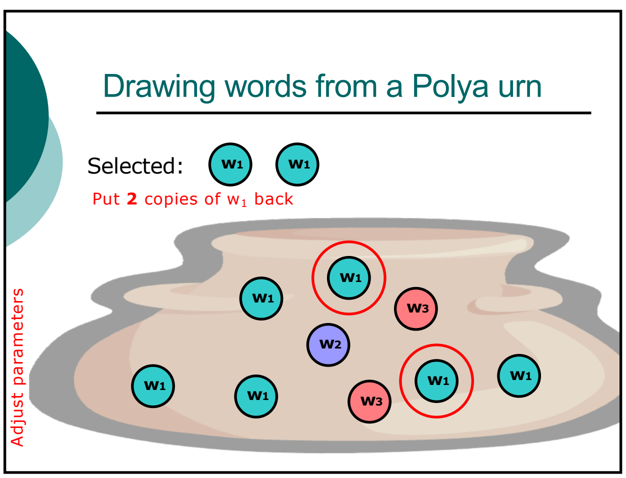
Drawing words from a Polya urn

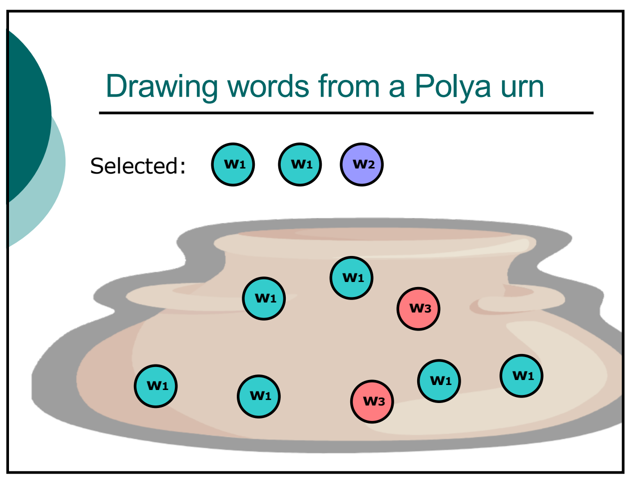
Selected: 




Drawing words from a Polya urn
Selected: w1
Put 2 copies of w1 back
Adjust parameters


Drawing words from a Polya urn
Selected: w1 w1

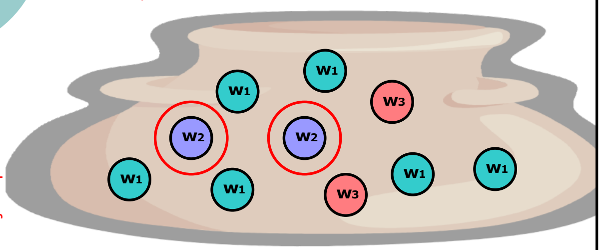

Drawing words from a Polya urn
Selected: w1 w1
Put 2 copies of w1 back
Adjust parameters


Drawing words from a Polya urn
Selected: w1 w1 w2

Drawing words from a Polya urn

Selected: w_1 w_1 w_2

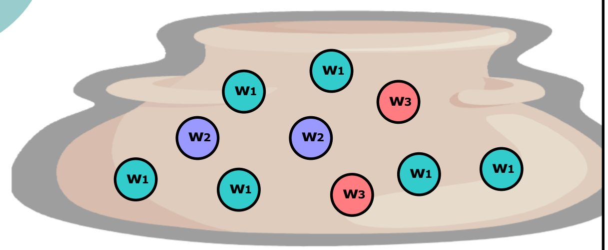
Put **2** copies of w_2 back



Adjust parameters

Drawing words from a Polya urn

Selected: w_1 w_1 w_2 …



Polya urn

- Words already drawn are more likely to be seen again

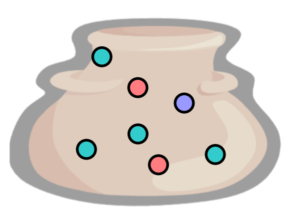
Results in the *Dirichlet Compound Multinomial (DCM) distribution*

Controlling burstiness

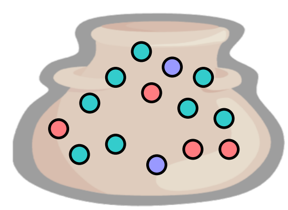
Same distribution of words

Which is more bursty?

more bursty



less bursty



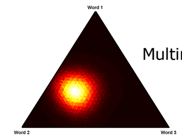
Polya urn

- Words already drawn are more likely to be seen again

Results in the *DCM distribution*

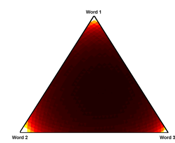
We can modulate burstiness by increasing/decreasing the number of words in the urn while keeping distribution the same

Burstiness with DCM

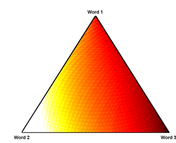


Multinomial

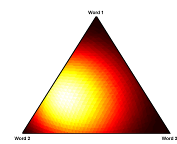
DCM



Down scaled
{.31, .44, .25}

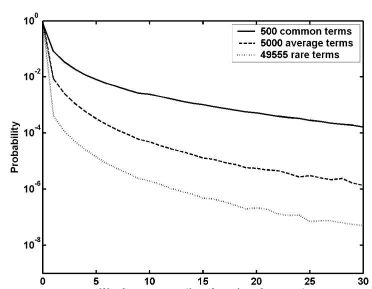


Medium scaled
{.93, 1.32, .75}

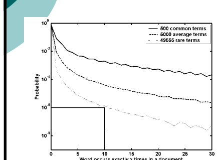


Up scaled
{2.81,3.94, 2.25}

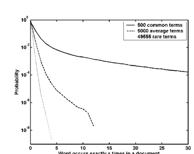
DCM word count probabilities



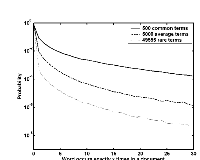
Reminder…



Data



Multinomial



DCM

DCM Model: another view

$$p(x_1, x_2, ..., x_m \mid \theta_1, \theta_2, ..., \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_m!} \prod_{j=1}^{m} \theta_j^{x_j}$$ Multinomial

$$p(x_1, x_2, ..., x_m \mid \alpha_1, \alpha_2, ..., \alpha_m) = \frac{|\mathrm{x}|!}{\prod_{w=1}^{m} x_w!} \frac{\Gamma\left(\sum_{w=1}^{m} \alpha_w\right)}{\prod_{w=1}^{m} \Gamma(\alpha_w)} \prod_{w=1}^{m} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)}$$ DCM

DCM Model: another view

$$p(x_1 x_2 ... x_m \mid \alpha) = \int_\theta p(\mathrm{x} \mid \theta) p(\theta \mid \alpha) d\theta$$

p(x|θ) ~ multinomial

p(θ|α) ~ Dirichlet

document is drawn from a multinomial

Dirichlet distribution over the types of multinomials that are generated per class

DCM Model: another view

p(x|θ) ~ multinomial

p(θ|α) ~ Dirichlet

$$p(x_1 x_2 ... x_m \mid \alpha) = \int_\theta p(\mathrm{x} \mid \theta) p(\theta \mid \alpha) d\theta$$

Generative story for a single class

- A class is represented by a Dirichlet distribution
- Draw a multinomial based on class distribution
- Draw a document based on the drawn multinomial distribution

Dirichlet Compound Multinomial

$$p(x_1 x_2 ... x_m \mid \alpha) = \int_\theta p(\mathrm{x} \mid \theta) p(\theta \mid \alpha) d\theta$$

$$= \int_\theta \frac{|\mathrm{x}|!}{\prod_{w=1}^{W} x_w!} \left(\prod_{w=1}^{W} \theta_w^{x_w} \right) \frac{\Gamma\left(\sum_{w=1}^{W} \alpha_w\right)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \theta_w^{\alpha_w - 1} d\theta$$

p(x|θ) ~ multinomial

p(θ|α) ~ Dirichlet

Dirichlet Compound Multinomial

$$p(\mathbf{x} \mid \alpha) = \int_\theta \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!}\left(\prod_{w=1}^{W}\theta_w^{x_w}\right)\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\prod_{w=1}^{W}\theta_w^{\alpha_w-1}d\theta$$

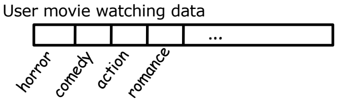
$$= \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!}\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\int_\theta\prod_{w=1}^{W}\theta_w^{\alpha_w+x_w-1}d\theta$$

$$= \frac{|\mathbf{x}|!}{\prod_{w=1}^{W} x_w!}\frac{\Gamma\left(\sum_{w=1}^{W}\alpha_w\right)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\prod_{w=1}^{W}\frac{\Gamma(x_w+\alpha_w)}{\Gamma(\alpha_w)}$$

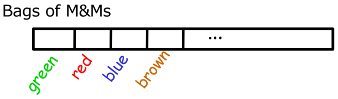
Modeling burstiness in other applications

Which model would be better: multinomial, DCM, other?

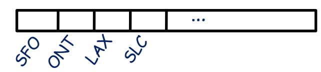
- User movie watching data



- Bags of M&Ms



- Daily Flight delays



Experiments

Modeling one class: document modeling

Modeling alternative classes: classification

Two standard data sets

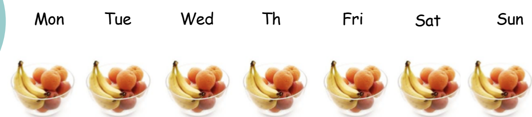
Industry sector (web pages)

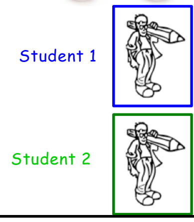
- More classes
- Less documents per class
- Longer documents

20 newsgroups (newsgroup posts)

- Fewer classes
- More documents per class
- Shorter documents

Modeling a single class: the fruit bowl

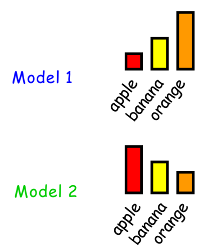


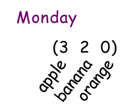


Goal: predict what the fruit mix will be for the following Monday (assign probabilities to options)

Modeling a single class/group

How well does a model predict unseen data?



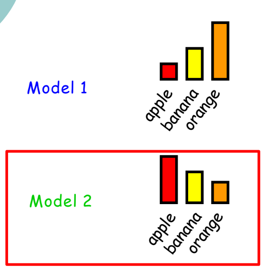


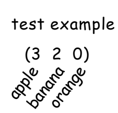
Which model is better?

How would you quantify how much better?

Modeling evaluation: perplexity

Perplexity is the average of the negative log of the model probabilities (likelihood) on test data





Use the same idea to measure the performance of the different models for modeling one set of documents

Perplexity results

20 newsgroups data set

Multinomial	**92.1**
DCM	**58.7**

Lower is better

- ideally the model would have a perplexity of 0!

Significant increase in modeling performance!

Classification results

Accuracy = number correct/ number of documents

	Industry	20 Newsgroups
Multinomial	0.600	0.853
DCM	**0.806**	**0.890**

(results are on par with state of the art discriminative approaches!)

Next steps in text modeling

Modeling textual phenomena like burstiness in text is important

Better grounded models like DCM **ALSO** perform better in applications (e.g. classification)

Better models

- text substitutability
- relax bag of words constraint (model co-occurrence)
- handling short phrases (tweets, search queries)

hierarchical models

Applications of models

- multi-class data modeling (e.g. clustering)
- text similarity
- language generation applications (speech recognition, translation, summarization)