NAÏVE BAYES CONTINUED

David Kauchak
CS159 Fall 2020

1

Admin

Assignment 6b

No class Tuesday

Assignment 7 out Monday

2

Final project

1. Your project should relate to something involving NLP
2. Your project must include a solid experimental evaluation
3. Your project should be in a group of 2-4. If you'd like to do it solo, please come talk to me.

3

Final project

date	time	description
11/5	in-class	Project proposal presentation
11/11	11:59pm	Project proposal write-up
11/11	11:59pm	Status report
11/23	11:59pm	Paper draft
11/24	in-class	Presentation
11/25	11:59pm	Final paper and code

Read the final project handout ASAP!

Start forming groups and thinking about what you want to do

4

Final project ideas

pick a text classification task
- evaluate different machine learning methods
- implement a machine learning method
- analyze different feature categories

n-gram language modeling
- implement and compare other smoothing techniques
- implement alternative models

parsing
- lexicalized PCFG (with smoothing)
- n-best list generation
- parse output reranking
- implement another parsing approach and compare
- parsing non-traditional domains (e.g. twitter)

EM
- try and implement IBM model 2
- word-level translation models

5

Final project application areas

- spelling correction
- part of speech tagger
- text chunker
- dialogue generation
- pronoun resolution
- compare word similarity measures (more than the ones we looked at)
- word sense disambiguation
- machine translation
- information retrieval
- information extraction
- question answering
- summarization
- speech recognition

6

Basic steps for probabilistic modeling

	Probabilistic models
Step 1: pick a model	Which model do we use, i.e. how do we calculate p(*feature, label*)?
Step 2: figure out how to estimate the probabilities for the model	How do train the model, i.e. how to we we estimate the probabilities for the model?
Step 3 (optional): deal with overfitting	How do we deal with overfitting?

7

Naïve Bayes assumption

$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_i \mid y, x_1, \ldots, x_{i-1})$$

$$p(x_i \mid y, x_1, x_2, \ldots, x_{i-1}) = p(x_i \mid y)$$

Assumes feature i is independent of the the other features *given the label*

8

Generative Story

To classify with a model, we're given an example and we obtain the probability

We can also ask how a given model would *generate* an example

This is the "generative story" for a model

Looking at the generative story can help understand the model

We also can use generative stories to help develop a model

9

Bernoulli NB generative story

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each feature:
   - Flip a *biased* coin:
     - if heads, include the feature
     - if tails, don't include the feature

   What does this mean for text classification, assuming unigram features?

10

Bernoulli NB generative story

$$p(y)\prod_{j=1}^{m} p(w_j \mid y)$$

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a *biased* coin:
     - if heads, include the word in the text
     - if tails, don't include the word

11

Bernoulli NB

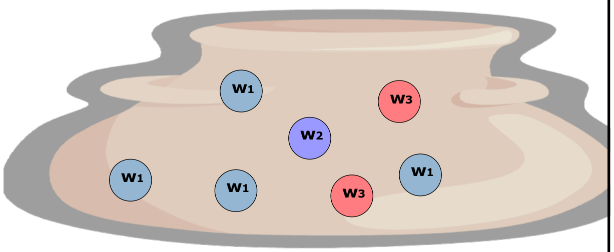
Pros
- Easy to implement
- Fast!
- Can be done on large data sets

Cons
- Naïve Bayes assumption is generally not true
- Performance isn't as good as other models
- For text classification (and other sparse feature domains) the $p(x_i=0 \mid y)$ can be problematic

12

Another generative story

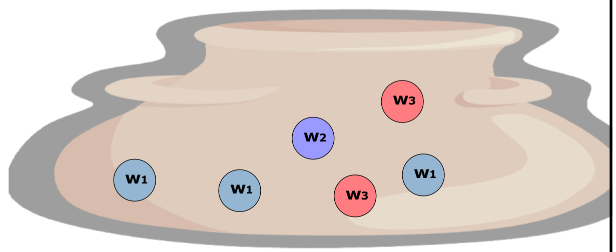
Randomly draw words from a "bag of words" until document length is reached



13

Draw words from a fixed distribution

Selected: w_1



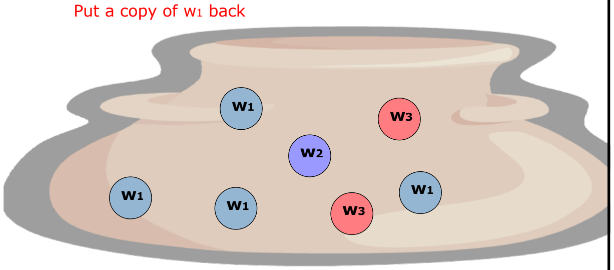
14

Draw words from a fixed distribution

Selected: w_1

Put a copy of w_1 back

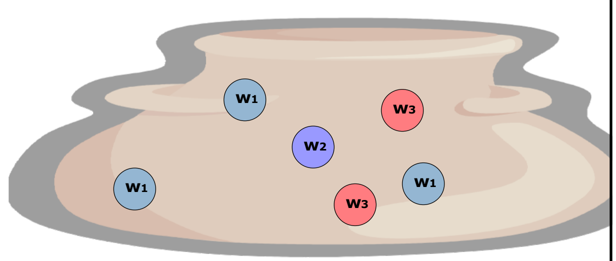
sampling with replacement



15

Draw words from a fixed distribution

Selected: w_1 w_1



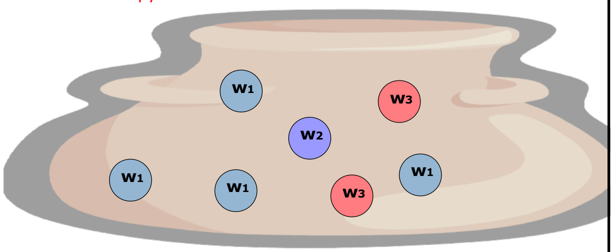
16

Draw words from a fixed distribution

Selected: w_1 w_1

Put a copy of w_1 back

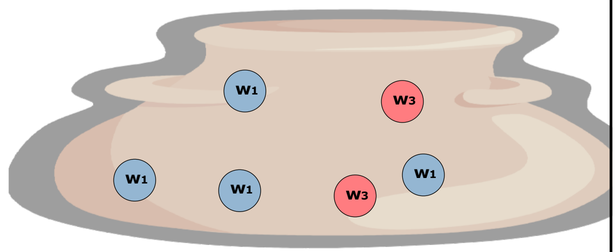
sampling with replacement



17

Draw words from a fixed distribution

Selected: w_1 w_1 w_2



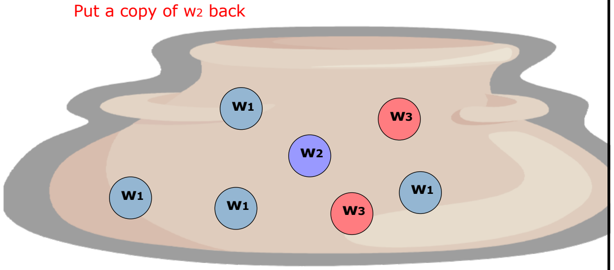
18

Draw words from a fixed distribution

Selected: w_1 w_1 w_2

Put a copy of w_2 back

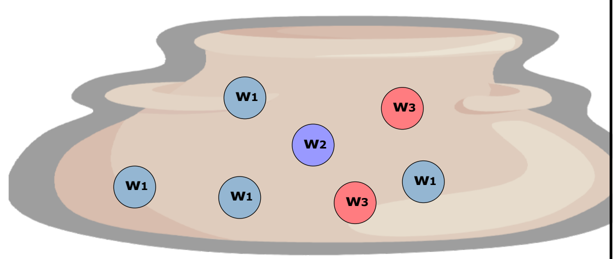
sampling with replacement



19

Draw words from a fixed distribution

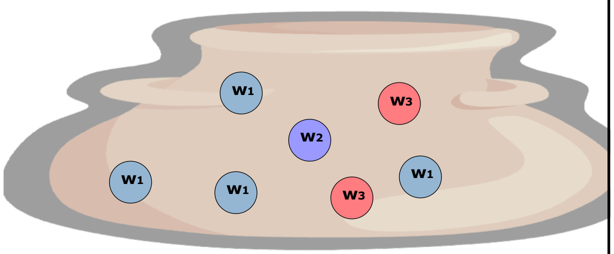
Selected: w_1 w_1 w_2 …



20

Draw words from a fixed distribution

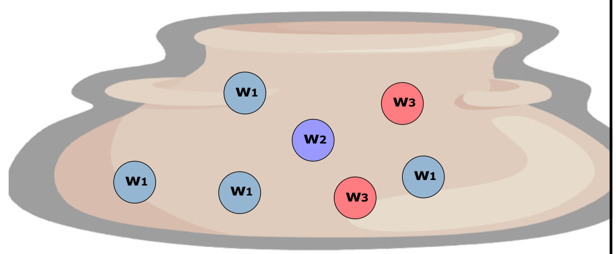
Is this a NB model, i.e. does it assume each individual word occurrence is independent?



21

Draw words from a fixed distribution

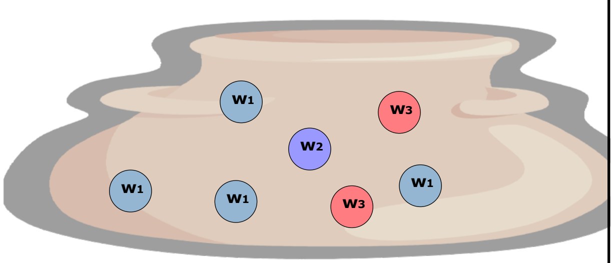
Yes! Doesn't matter what words were drawn previously, still the same probability of getting any particular word



22

Draw words from a fixed distribution

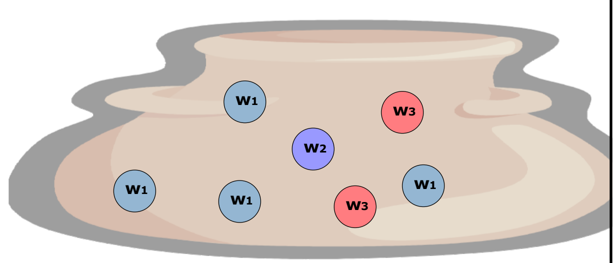
Does this model handle multiple word occurrences?



23

Draw words from a fixed distribution

Selected: w_1 w_1 w_2 ...



24

NB generative story

Bernoulli NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - if heads, include the word in the text
     - if tails, don't include the word

Multinomial NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. Keep drawing words from *p(words|y)* until text length has been reached.

25

Probabilities

Bernoulli NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. For each word in your vocabulary:
   - Flip a biased coin:
     - if heads, include the word in the text
     - if tails, don't include the word

$$p(y)\prod_{j=1}^{m} p(x_j \mid y)$$

(1, 1, 1, 0, 0, 1, 0, 0, …)

w1 w2 w3 w4 w5 w6 w7 w8

Multinomial NB

1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. Keep drawing words from *p(words|y)* until document length has been reached

?

(4, 1, 2, 0, 0, 7, 0, 0, …)

w1 w2 w3 w4 w5 w6 w7 w8

26

A digression: rolling dice

What's the probability of getting a 3 for a single roll of this dice?

1/6

27

A digression: rolling dice

What is the probability distribution over possible single rolls?

1/6	1/6	1/6	1/6	1/6	1/6
1	2	3	4	5	6

28

A digression: rolling dice

What if I told you 1 was twice as likely as the others?

2/7	1/7	1/7	1/7	1/7	1/7
1	2	3	4	5	6

29

A digression: rolling dice

What if I rolled 400 times and got the following number?

1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

30

A digression: rolling dice

1. What is the probability of rolling a 1 and a 5 (in any order)?
2. Two 1s and a 5 (in any order)?
3. Five 1s and two 5s (in any order)?

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

31

A digression: rolling dice

1. What it the probability of rolling a 1 and a 5 (in any order)?

   $(1/4 * 1/8) * 2 = 1/16$

   prob. of those two rolls

   number of ways that can happen (1,5 and 5,1)

1. Two 1s and a 5 (in any order)?

   $((1/4)^2 * 1/8) * 3 = 3/128$

2. Five 1s and two 5s (in any order)?

   $((1/4)^5 * (1/8)^3) * 21 = 21/524{,}288 = 0.00004$

General formula?

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

32

Multinomial distribution

Multinomial distribution: independent draws over *m* possible categories

If we have frequency counts x1, x2, …, xm over each of the categories, the probability is:

$$p(x_1, x_2, \ldots, x_m \mid \theta_1, \theta_2, \ldots, \theta_m) = \underbrace{\frac{n!}{\prod_{j=1}^{m} x_j!}}_{\text{number of different ways to get those counts}} \underbrace{\prod_{j=1}^{m} \theta_j^{x_j}}_{\text{probability of particular counts}}$$

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

33

Multinomial distribution

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

What are θ_j?

Are there any constraints on the values that they can take?

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

34

Multinomial distribution

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

θ_j: probability of rolling "j"

$$\theta_j \geq 0$$

$$\sum_{j=1}^{m} \theta_j = 1$$

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	
1	2	3	4	5	6	…

35

Back to words…

Why the digression?

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

Drawing words from a bag is the same as rolling a die!

number of sides = number of words in the vocabulary

36

Back to words…

Why the digression?

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

$$p(features, label) = p(y) \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} (\theta_y)_j^{x_j}$$

θ_j for class y

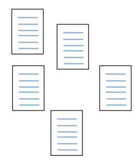
37

Basic steps for probabilistic modeling

Model each class as a multinomial:

$$p(features, label) = p(y) \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} (\theta_y)_j^{x_j}$$

Step 2: figure out how to estimate the probabilities for the model



How do we train the model, i.e. estimate θ_j for each class?

38

A digression: rolling dice

What if I rolled 400 times and got the following number?

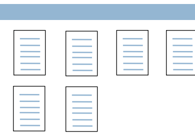
1: 100
2: 50
3: 50
4: 100
5: 50
6: 50

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

39

Training a multinomial

label1
label2



1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

40

Training a multinomial

label1

For each label, y:

w1: 100 times
w2: 50 times
w3: 10 times
w4: …

$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

$$= \frac{\text{number of times word } w_j \text{ occurs in label y docs}}{\text{total number of words in label y docs}}$$

1/4	1/8	1/8	1/4	1/8	1/8
1	2	3	4	5	6

41

Classifying with a multinomial

(10, 2, 6, 0, 0, 1, 0, 0, …)
w1 w2 w3 w4 w5 w6 w7 w8

$$p(y=1)\frac{n!}{\prod_{j=1}^{m} x_j!}\prod_{j=1}^{m}(\theta_1)_j^{x_j} \qquad p(y=2)\frac{n!}{\prod_{j=1}^{m} x_j!}\prod_{j=1}^{m}(\theta_2)_j^{x_j}$$

Any way I can make this simpler?

pick largest

42

Classifying with a multinomial

(10, 2, 6, 0, 0, 1, 0, 0, …)
w1 w2 w3 w4 w5 w6 w7 w8

$$p(y=1)\prod_{j=1}^{m}(\theta_1)_j^{x_j} \qquad p(y=2)\prod_{j=1}^{m}(\theta_2)_j^{x_j}$$

pick largest

$\frac{n!}{\prod_{j=1}^{m} x_m!}$ Is a constant!

43

Multinomial finalized

Training:

- Calculate p(label)
- For each label, calculate θs

$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

Classification:

- Get word counts
- For each label you had in training, calculate:

$$p(y)\prod_{j=1}^{m}\theta_j^{x_j}$$

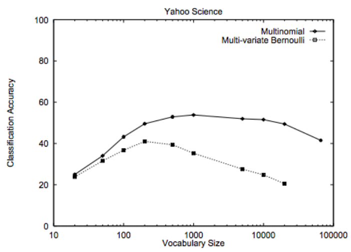
and pick the largest

44

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



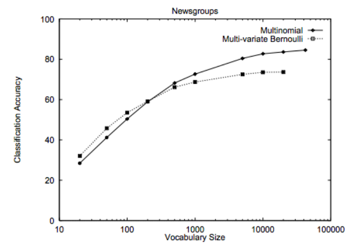
http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

45

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



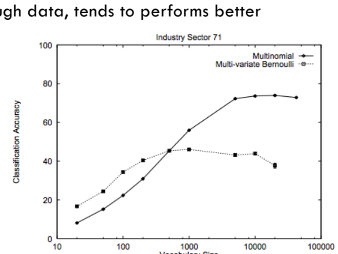
http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

46

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

47

Maximum likelihood estimation

Intuitive

Sets the probabilities so as to maximize the probability of the training data

Problems?

- Overfitting!
- Amount of data
  - particularly problematic for rare events
- Is our training data representative

48

Basic steps for probabilistic modeling

Probabilistic models

Step 1: pick a model

Which model do we use, i.e. how do we calculate p(*feature, label*)?

Step 2: figure out how to estimate the probabilities for the model

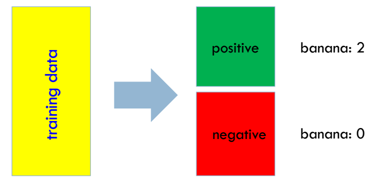
How do train the model, i.e. how to we we estimate the probabilities for the model?

Step 3 (optional): deal with overfitting

How do we deal with overfitting?

49

Unseen events

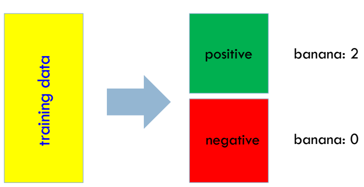


$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

What will θ_{banana} be for the negative class?

50

Unseen events



$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

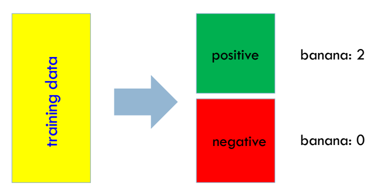
What will θ_{banana} be for the negative class?

0!

Is this a problem?

51

Unseen events



p("I ate a bad banana", negative) = ?

52

Unseen events

training data

positive — banana: 2

negative — banana: 0

p("I ate a bad banana", negative) = 0
p("…. banana …", negative) = 0

Solution?

53

Add lambda smoothing

training data

$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

$$\theta_j = \frac{count(w_j, y) + \lambda}{\lambda m + \sum_{k=1}^{m} count(w_k, y)}$$

for each label, pretend like we've seen each feature/word occur in λ additional examples

54

Different than…

training data

positive — banana: 0

negative — banana: 0

How is this problem different?

55

Different than…

training data

positive — banana: 0

negative — banana: 0

p("I ate a bad banana", positive) → p("I ate a bad", positive)

p("I ate a bad banana", negative) → p("I ate a bad", negative)

Out of vocabulary. Many ways to solve… for our implementation, we'll just ignore them.

56