MACHINE LEARNING BASICS

David Kauchak
CS159 Fall 2024

1

Admin

Assignment 6

2

Quiz #3

45 minutes (plus 20 to scan/upload)

Open book and notes

Text Similarity (10/3) through Machine Translation (10/24)

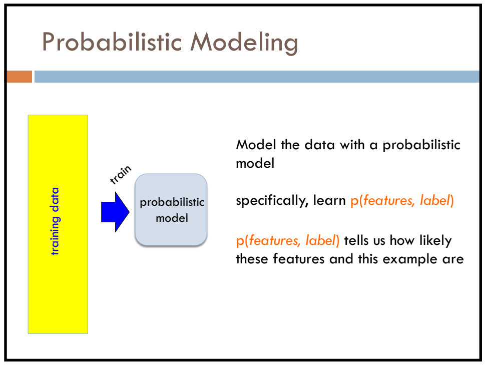
Class will be 30 min discussion about final projects

3

Machine Learning is…

Machine learning is about predicting the future based on the past.
-- Hal Daume III

4



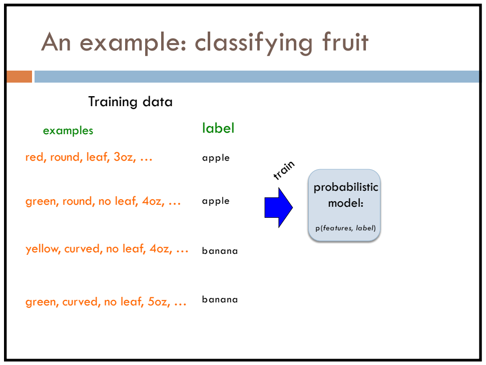
Probabilistic Modeling

Model the data with a probabilistic model

specifically, learn p(*features, label*)

p(*features, label*) tells us how likely these features and this example are

5

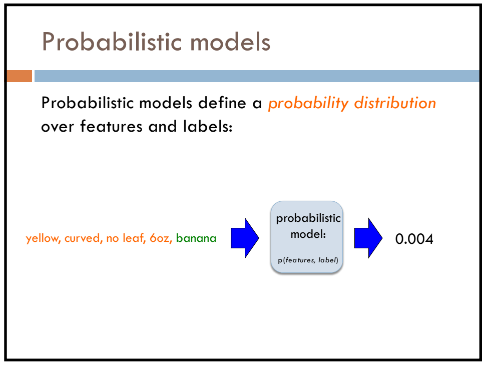


An example: classifying fruit

Training data

examples	label
red, round, leaf, 3oz, …	apple
green, round, no leaf, 4oz, …	apple
yellow, curved, no leaf, 4oz, …	banana
green, curved, no leaf, 5oz, …	banana

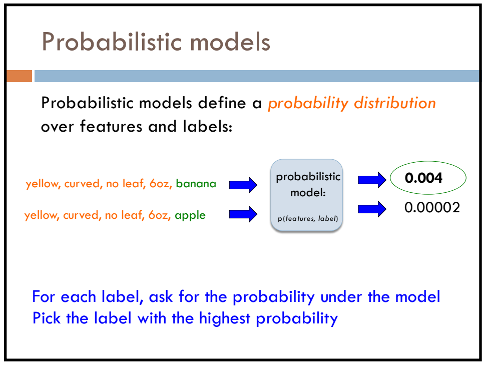
6



Probabilistic models

Probabilistic models define a *probability distribution* over features and labels:

7



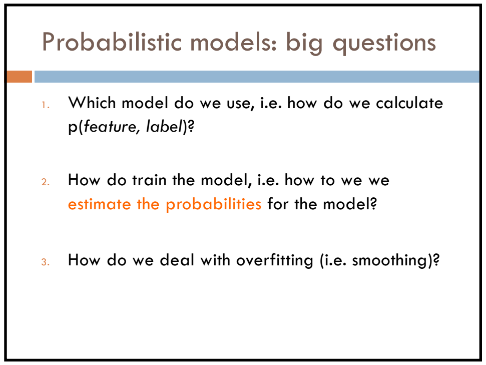
Probabilistic models

Probabilistic models define a *probability distribution* over features and labels:

yellow, curved, no leaf, 6oz, banana
yellow, curved, no leaf, 6oz, apple
probabilistic model:
p(features, label)
0.004
0.00002

For each label, ask for the probability under the model
Pick the label with the highest probability

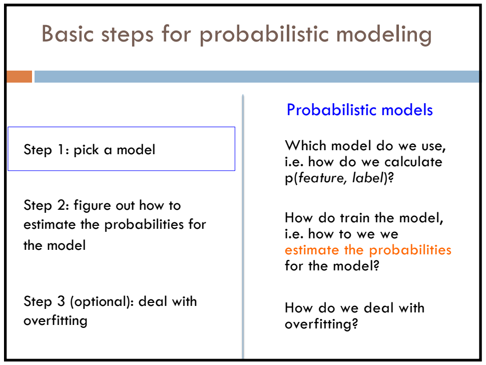
8



Probabilistic models: big questions

1. Which model do we use, i.e. how do we calculate p(*feature, label*)?
2. How do train the model, i.e. how to we we estimate the probabilities for the model?
3. How do we deal with overfitting (i.e. smoothing)?

9



Basic steps for probabilistic modeling

Step 1: pick a model

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

Step 3 (optional): deal with overfitting

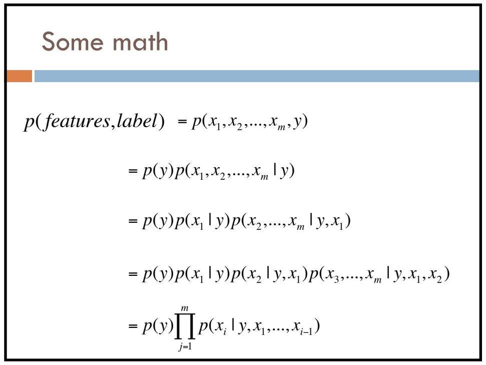
Probabilistic models

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

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

How do we deal with overfitting?

10



Some math

$$p(features,label) = p(x_1,x_2,...,x_m,y)$$

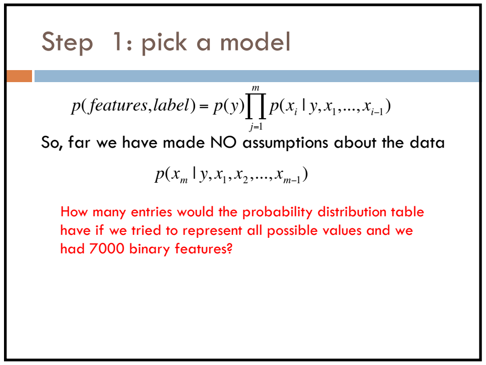
$$= p(y)p(x_1,x_2,...,x_m \mid y)$$

$$= p(y)p(x_1 \mid y)p(x_2,...,x_m \mid y,x_1)$$

$$= p(y)p(x_1 \mid y)p(x_2 \mid y,x_1)p(x_3,...,x_m \mid y,x_1,x_2)$$

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

11



Step 1: pick a model

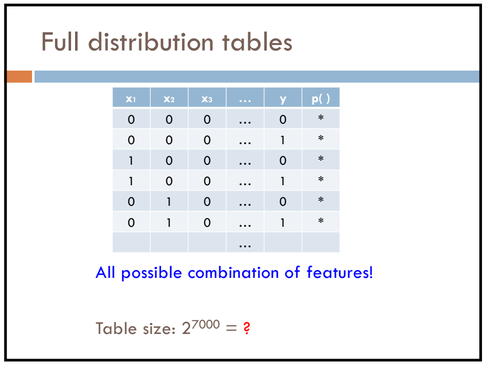
$$p(features,label) = p(y)\prod_{j=1}^{m} p(x_i \mid y,x_1,...,x_{i-1})$$

So, far we have made NO assumptions about the data

$$p(x_m \mid y,x_1,x_2,...,x_{m-1})$$

How many entries would the probability distribution table have if we tried to represent all possible values and we had 7000 binary features?

12



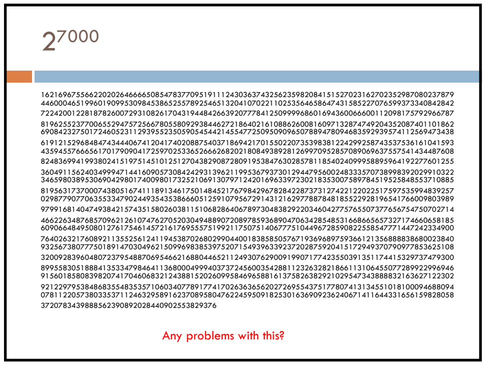
Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

All possible combination of features!

Table size: 2^{7000} = ?

13

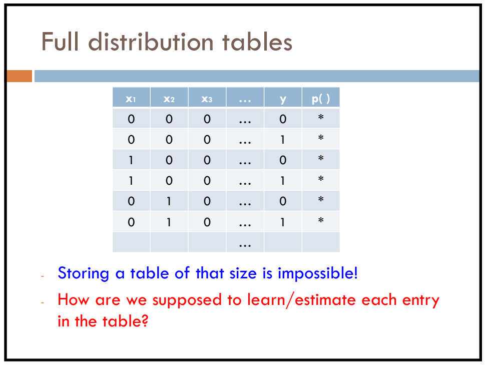


2^{7000}

1621696755662202026466665085478377095191112430363743256235982084151527023162702352987080237879
4460004651996019099530984538652557892546513204107022110253564658647431585227076599373340842842
7224200122818782600729310826170431944842663920777841250999968601694360066600112098175792966787
8196255237700655294757256678055809293844627218640216108862600816097132874749204352087401101862
6908423275017246052311293955235059054544214554772509509096507889478094683592939574112569473438
6191215296848474344406741204174020887540371869421701550220735398381224299258743537536161041593
4359455766656170179090417259702533652666268202180849389281269970952857089069637557541434487608
8248369941993802415197514510125127043829087280919538476302857811854024099958895964192277601255
3604911562403499947144160905730842429313962119953679373012944795600248333570738998392029910322
3465980389530690429801740098017325210691307971242016963397230218353007589784519525848553710885
8195631737000743805167411189134617501484521767984296782842287373127422122022517597535994839257
0298779077063553347902449354353866605125910795672914312162977887848185522928196541766009803989
9799168140474938421574351580260381151068286406789730483829220346042775765507377656754750702714
4662263487685709621261074762705203049488907208978593689047063428548531668665657327174660658185
6090664849508012761754614572161769555751992117507514067775104496728590822558547771447242334900
7640263217608921135525612411945387026802990440018385850576719369689759366121356888838680023840
9325673807775018914703049621509969838539752071549396339237202875920415172949370790977853625108
3200928396048072379548870695466216880446521124930762900919907177423550391351174415329737479300
8995583051888413533479846411368000499940373724560035428811232632821866113106455077289922996946
9156018580839820741704606832124388152026099584696588161375826382921029547343888832163627122302
9212297953848683554835357106034077891774170263636562027269554375177807413134551018100094688094
0781122057380335371124632958916237089580476224595091825301636909236240671411644331656159828058
37207834398885623908920284409025538293376

Any problems with this?

14

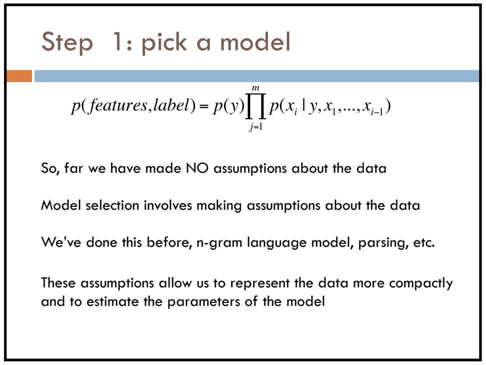


Full distribution tables

x_1	x_2	x_3	…	y	p( )
0	0	0	…	0	*
0	0	0	…	1	*
1	0	0	…	0	*
1	0	0	…	1	*
0	1	0	…	0	*
0	1	0	…	1	*
			…		

- Storing a table of that size is impossible!
- How are we supposed to learn/estimate each entry in the table?

15



Step 1: pick a model

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

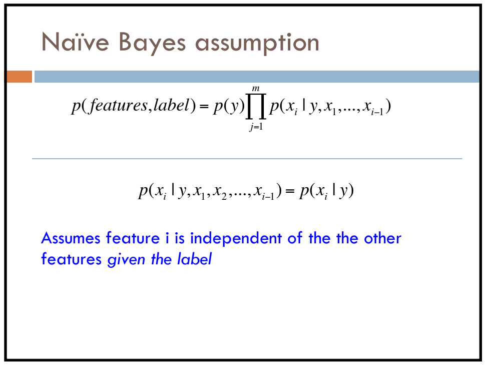
So, far we have made NO assumptions about the data

Model selection involves making assumptions about the data

We've done this before, n-gram language model, parsing, etc.

These assumptions allow us to represent the data more compactly and to estimate the parameters of the model

16



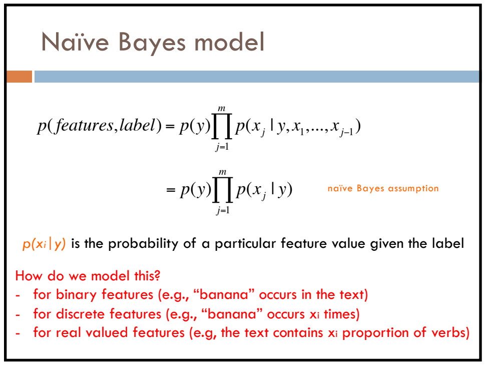
Naïve Bayes assumption

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

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

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

17



Naïve Bayes model

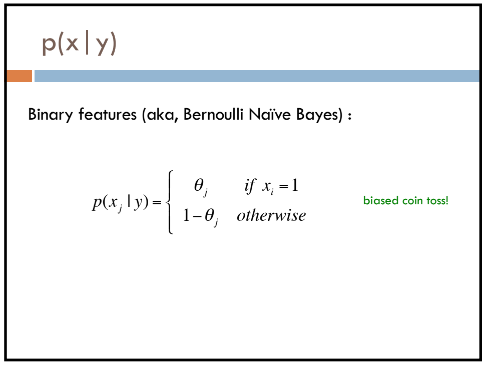
$$p(features, label) = p(y)\prod_{j=1}^{m} p(x_j \mid y, x_1, \ldots, x_{j-1})$$

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

$p(x_i \mid y)$ is the probability of a particular feature value given the label

How do we model this?
- for binary features (e.g., "banana" occurs in the text)
- for discrete features (e.g., "banana" occurs x_i times)
- for real valued features (e.g, the text contains x_i proportion of verbs)

18



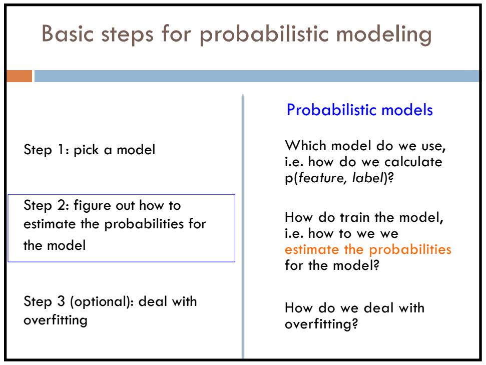
p(x|y)

Binary features (aka, Bernoulli Naïve Bayes) :

$$p(x_j \mid y) = \begin{cases} \theta_j & if\ x_i = 1 \\ 1-\theta_j & otherwise \end{cases}$$

biased coin toss!

19



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

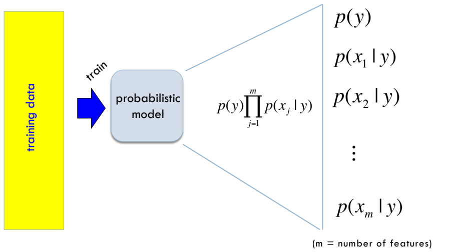
How do train the model, i.e. how to we we estimate the probabilities for the model?

Step 3 (optional): deal with overfitting

How do we deal with overfitting?

20

Obtaining probabilities



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

$p(y)$

$p(x_1 \mid y)$

$p(x_2 \mid y)$

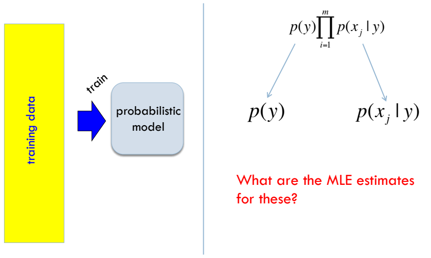
$\vdots$

$p(x_m \mid y)$

(m = number of features)

21

MLE estimation for Bernoulli NB



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

$p(y)$ $\quad$ $p(x_j \mid y)$

What are the MLE estimates for these?

22

Maximum likelihood estimates

$$p(y) = \frac{count(y)}{n} \qquad \frac{\text{number of examples with label}}{\text{total number of examples}}$$

$$p(x_j \mid y) = \frac{count(x_j, y)}{count(y)} \qquad \frac{\text{number of examples with the label with feature}}{\text{number of examples with label}}$$

What does training a NB model then involve?
How difficult is this to calculate?

23

Text classification

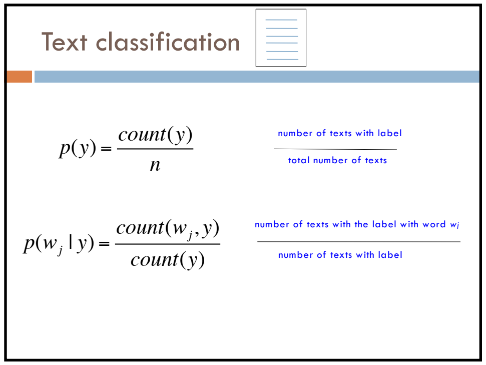
$$p(y) = \frac{count(y)}{n}$$

$$p(w_j \mid y) = \frac{count(w_j, y)}{count(y)}$$

Unigram features:
w_i, whether or not word w_i occurs in the text

What are these counts for text classification with unigram features?

24



Text classification

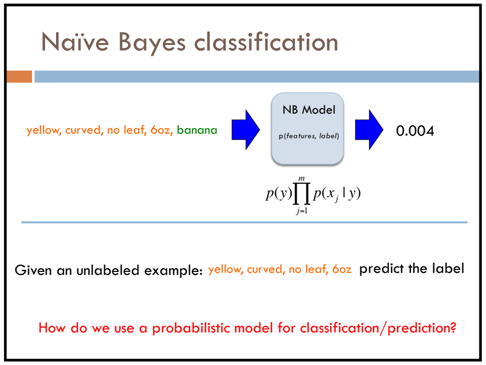
$$p(y) = \frac{count(y)}{n}$$

number of texts with label / total number of texts

$$p(w_j \mid y) = \frac{count(w_j, y)}{count(y)}$$

number of texts with the label with word w_j / number of texts with label

25



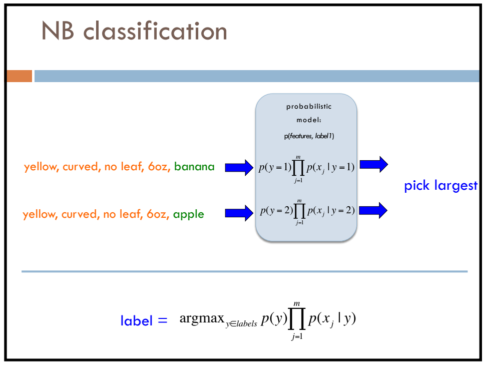
Naïve Bayes classification

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

Given an unlabeled example: yellow, curved, no leaf, 6oz predict the label

How do we use a probabilistic model for classification/prediction?

26

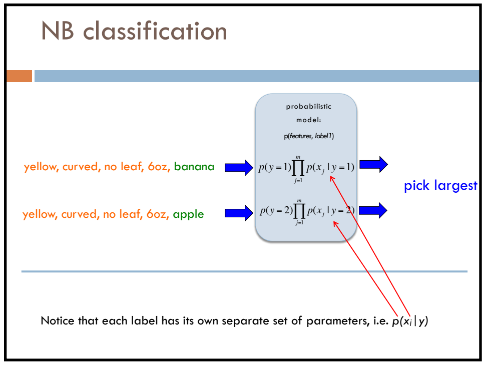


NB classification

probabilistic model:
p(features, label1)
yellow, curved, no leaf, 6oz, banana
$p(y=1)\prod_{j=1}^{m} p(x_j \mid y=1)$
yellow, curved, no leaf, 6oz, apple
$p(y=2)\prod_{j=1}^{m} p(x_j \mid y=2)$
pick largest

label = $\operatorname{argmax}_{y \in labels}\, p(y)\prod_{j=1}^{m} p(x_j \mid y)$

27

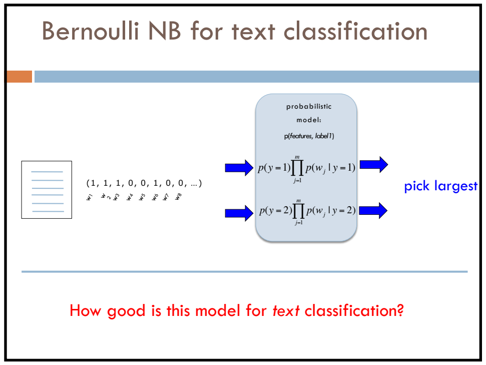


NB classification

probabilistic model:
p(features, label1)
yellow, curved, no leaf, 6oz, banana
$p(y=1)\prod_{j=1}^{m} p(x_j \mid y=1)$
yellow, curved, no leaf, 6oz, apple
$p(y=2)\prod_{j=1}^{m} p(x_j \mid y=2)$
pick largest

Notice that each label has its own separate set of parameters, i.e. $p(x_i \mid y)$

28



Bernoulli NB for text classification

(1, 1, 1, 0, 0, 1, 0, 0, …)

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

probabilistic model:

p(*features, label1*)

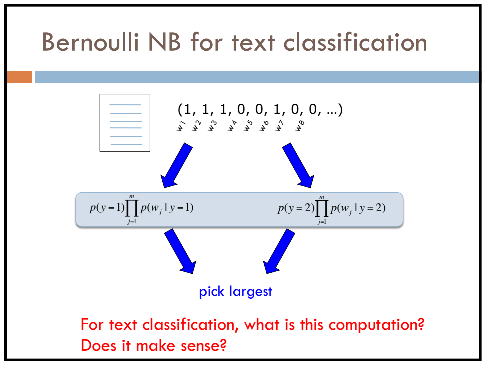
$p(y=1)\prod_{j=1}^{m} p(w_j \mid y=1)$

$p(y=2)\prod_{j=1}^{m} p(w_j \mid y=2)$

pick largest

How good is this model for *text* classification?

29



Bernoulli NB for text classification

(1, 1, 1, 0, 0, 1, 0, 0, …)

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

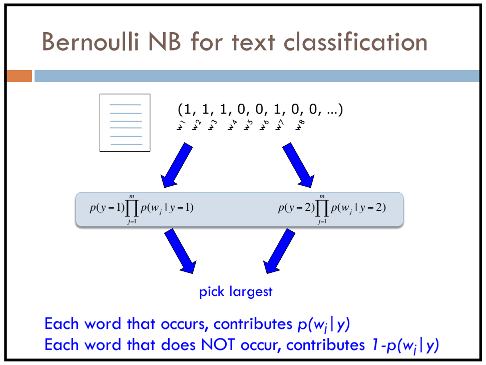
$p(y=1)\prod_{j=1}^{m} p(w_j \mid y=1)$

$p(y=2)\prod_{j=1}^{m} p(w_j \mid y=2)$

pick largest

For text classification, what is this computation?
Does it make sense?

30



Bernoulli NB for text classification

(1, 1, 1, 0, 0, 1, 0, 0, …)

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

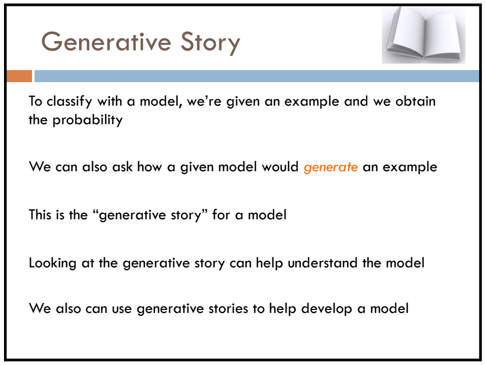
$p(y=1)\prod_{j=1}^{m} p(w_j \mid y=1)$

$p(y=2)\prod_{j=1}^{m} p(w_j \mid y=2)$

pick largest

Each word that occurs, contributes $p(w_j \mid y)$
Each word that does NOT occur, contributes $1-p(w_j \mid y)$

31



Generative Story

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

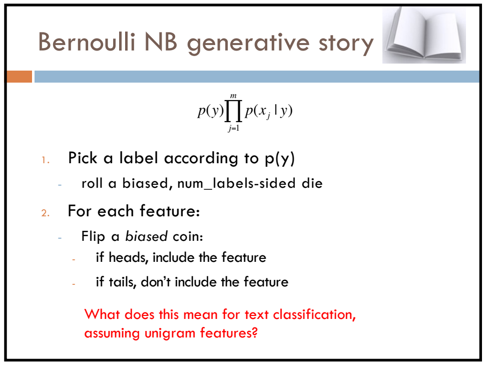
We can also ask how a given model would *generate* an example

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

32



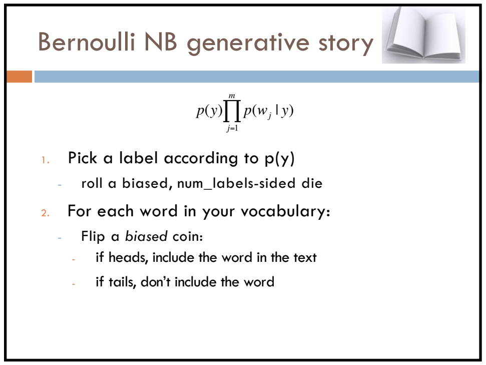
Bernoulli NB generative story

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

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?

33

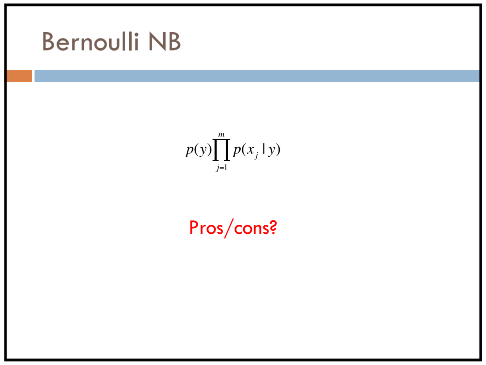


Bernoulli NB generative story

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

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

34

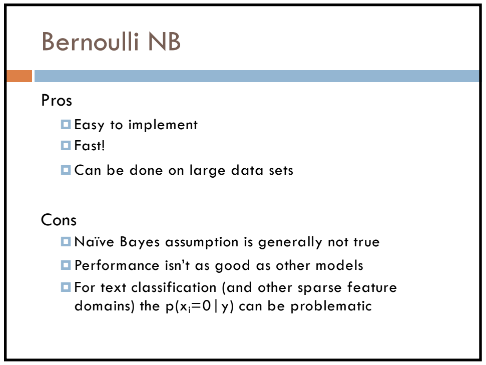


Bernoulli NB

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

Pros/cons?

35



Bernoulli NB

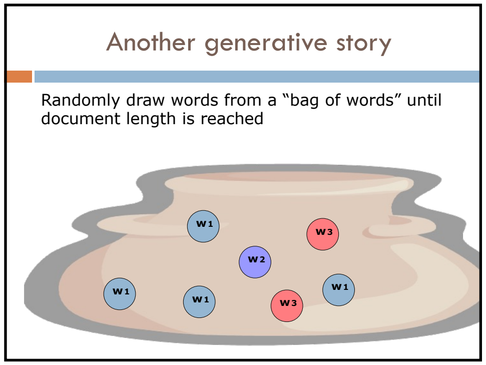
Pros

- 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

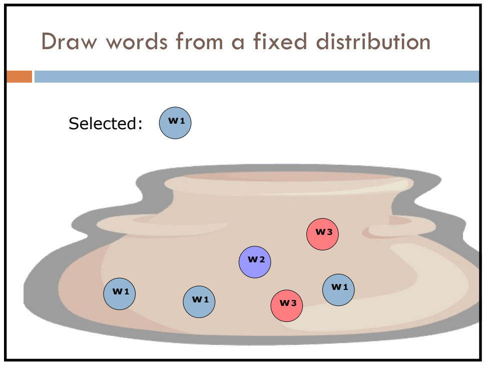
36



Another generative story

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

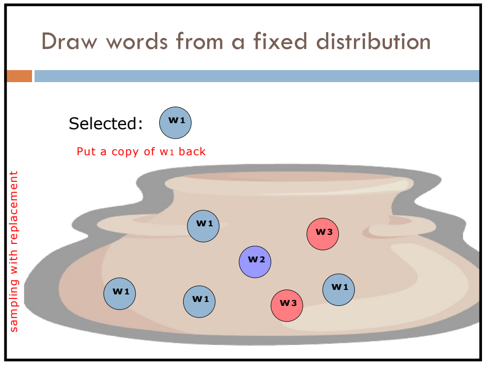
37



Draw words from a fixed distribution

Selected: w1

38



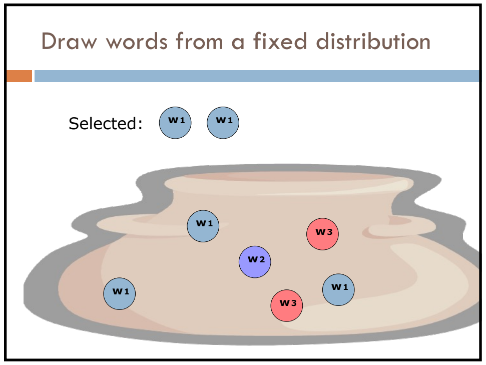
Draw words from a fixed distribution

Selected: w1

Put a copy of w1 back

sampling with replacement

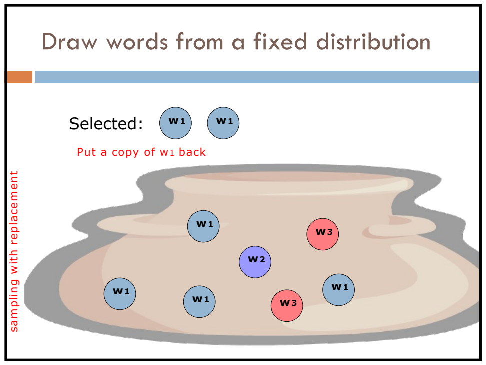
39



Draw words from a fixed distribution

Selected: w1 w1

40



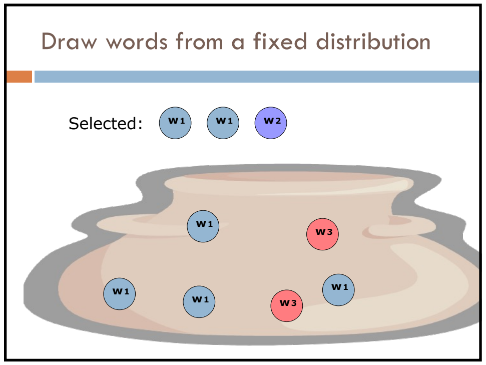
Draw words from a fixed distribution

Selected: w_1 w_1

Put a copy of w_1 back

sampling with replacement

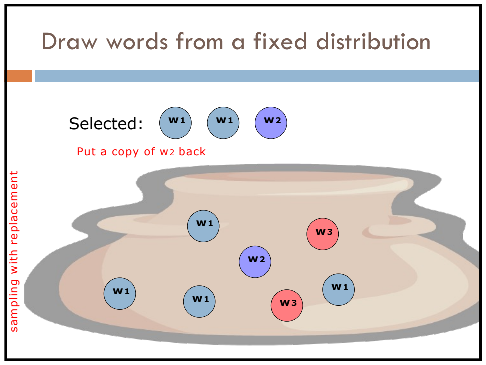
41



Draw words from a fixed distribution

Selected: w_1 w_1 w_2

42



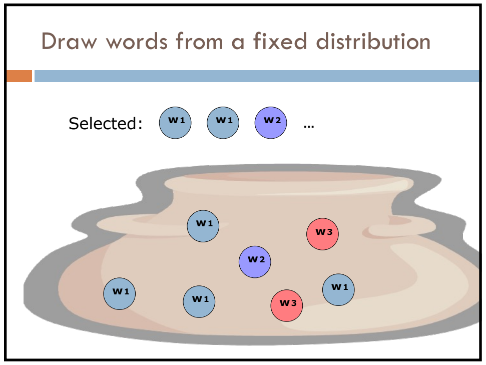
Draw words from a fixed distribution

Selected: w_1 w_1 w_2

Put a copy of w_2 back

sampling with replacement

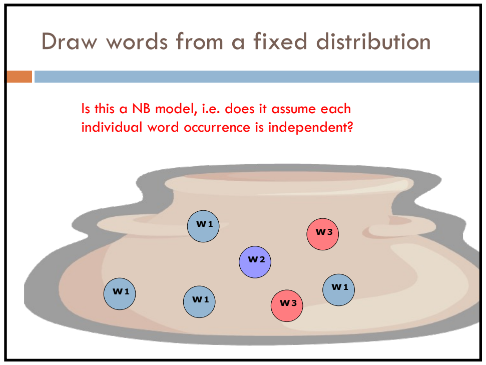
43



Draw words from a fixed distribution

Selected: w_1 w_1 w_2 …

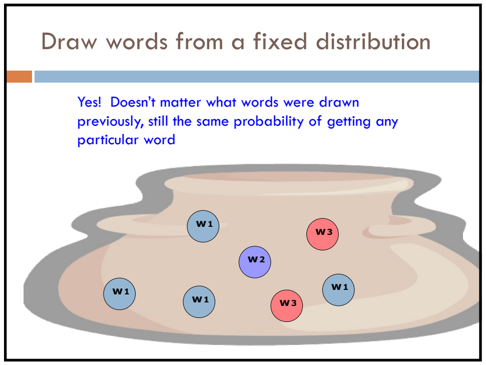
44



Draw words from a fixed distribution

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

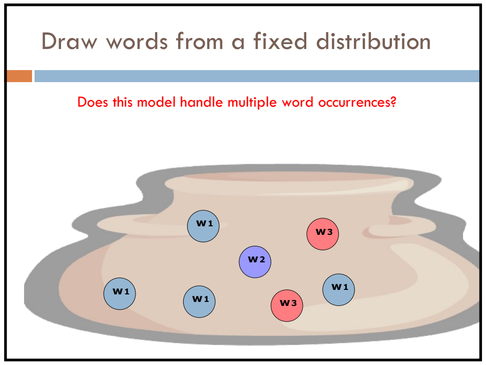
45



Draw words from a fixed distribution

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

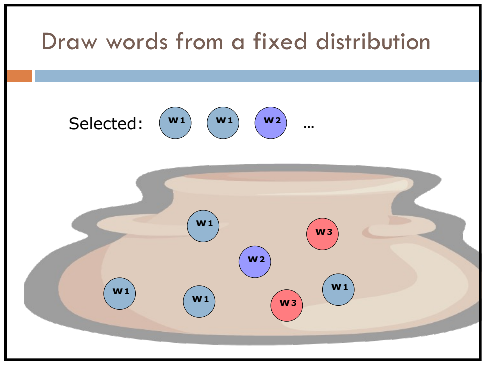
46



Draw words from a fixed distribution

Does this model handle multiple word occurrences?

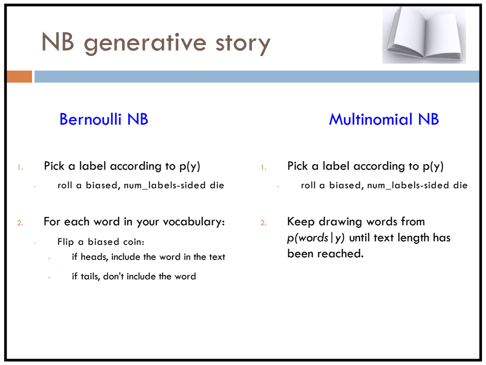
47



Draw words from a fixed distribution

Selected: w1 w1 w2 ...

48



NB generative story

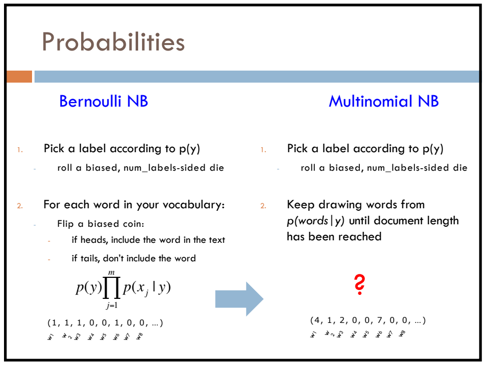
Bernoulli NB

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.

49



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, …)

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

Multinomial NB

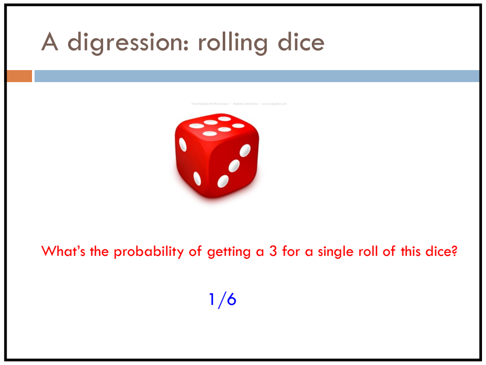
1. Pick a label according to p(y)
   - roll a biased, num_labels-sided die
2. Keep drawing words from *p(words|y)* until document length has been reached

?

(4, 1, 2, 0, 0, 7, 0, 0, …)

w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8

50

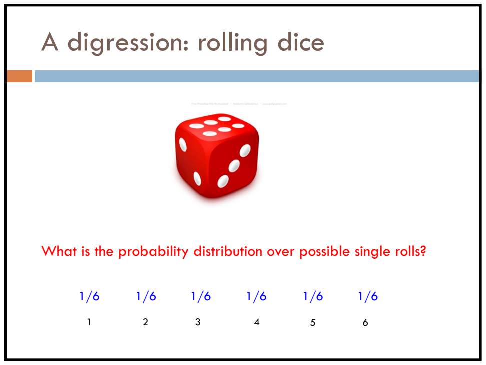


A digression: rolling dice

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

1/6

51

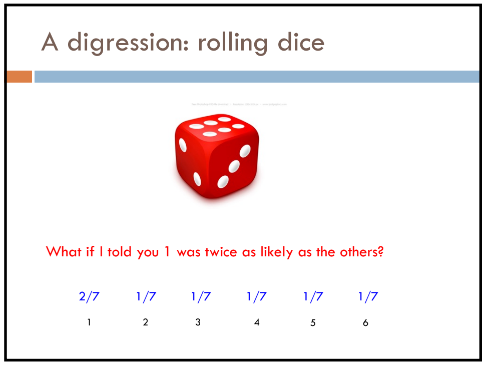


A digression: rolling dice

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

52

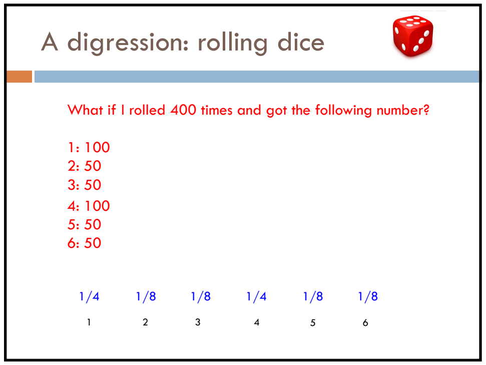


A digression: rolling dice

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

53



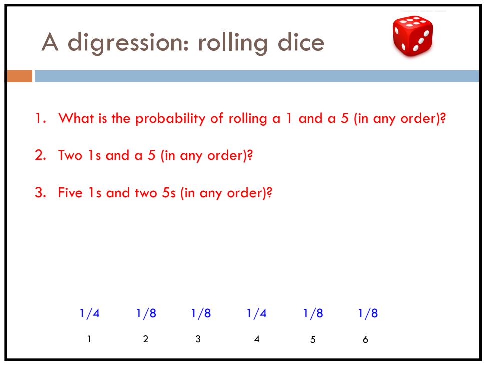
A digression: rolling dice

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

54

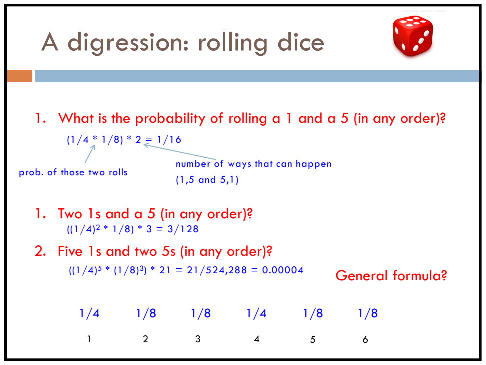


A digression: rolling dice

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

55



A digression: rolling dice

1. What is 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)?

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

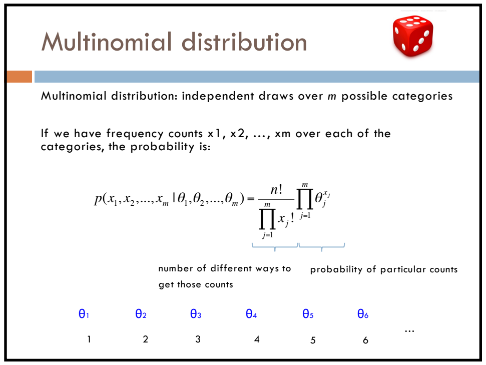
2. Five 1s and two 5s (in any order)?

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

General formula?

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

56



Multinomial distribution

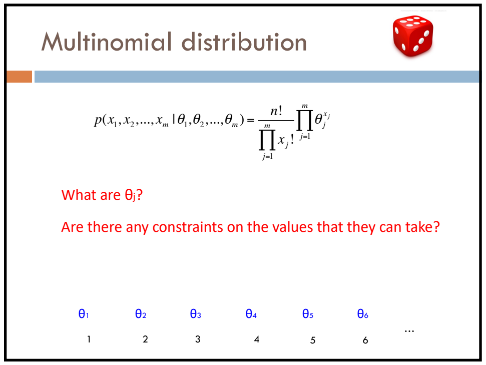
Multinomial distribution: independent draws over *m* possible categories

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

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

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

57



Multinomial distribution

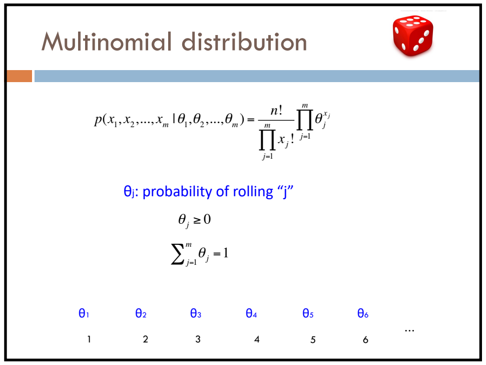
$$p(x_1, x_2, \dots, x_m \mid \theta_1, \theta_2, \dots, \theta_m) = \frac{n!}{\prod_{j=1}^{m} x_j!} \prod_{j=1}^{m} \theta_j^{x_j}$$

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	

58



Multinomial distribution

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

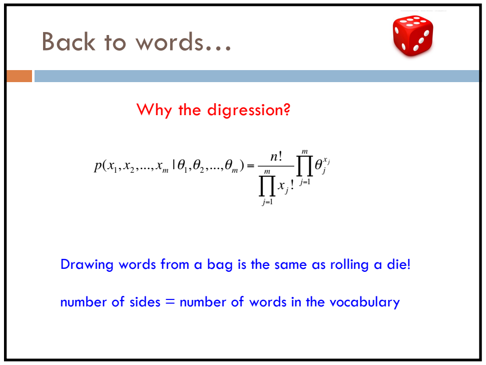
θ_j: probability of rolling "j"

$$\theta_j \geq 0$$

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

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

59



Back to words…

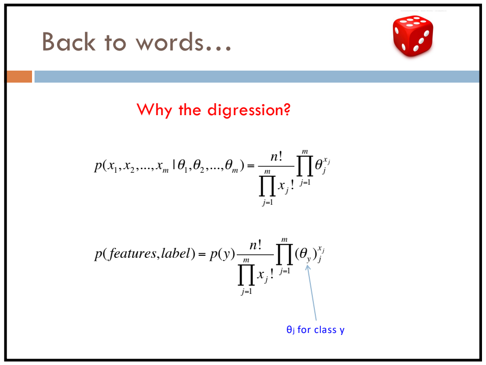
Why the digression?

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

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

number of sides = number of words in the vocabulary

60



Back to words…

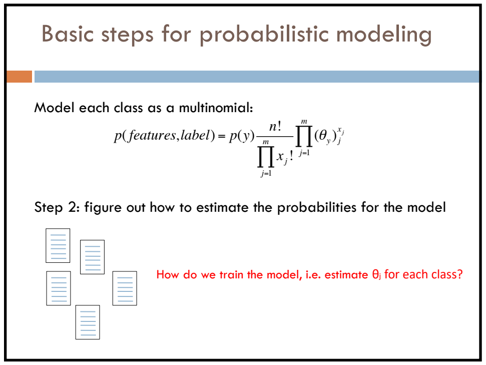
Why the digression?

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

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

θ_j for class y

61



Basic steps for probabilistic modeling

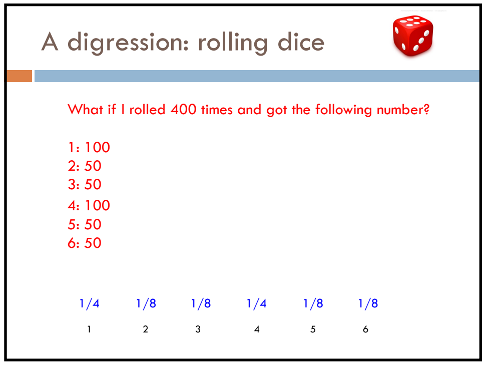
Model each class as a multinomial:

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

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?

62



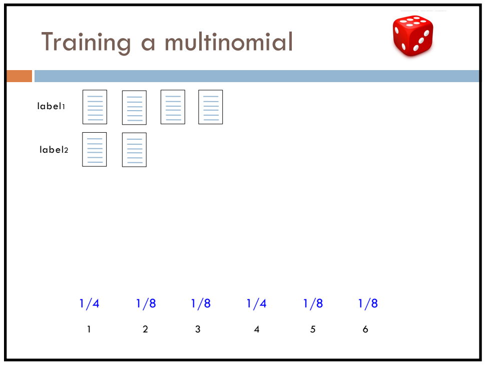
A digression: rolling dice

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

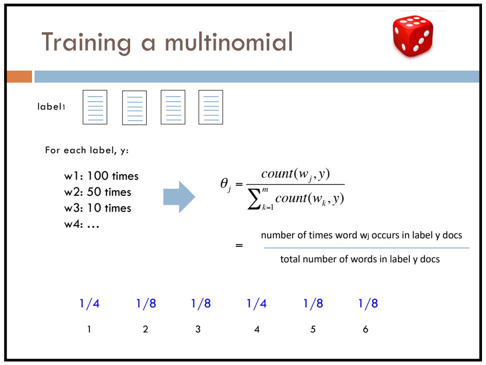
63



Training a multinomial

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

64



Training a multinomial

label1

For each label, y:

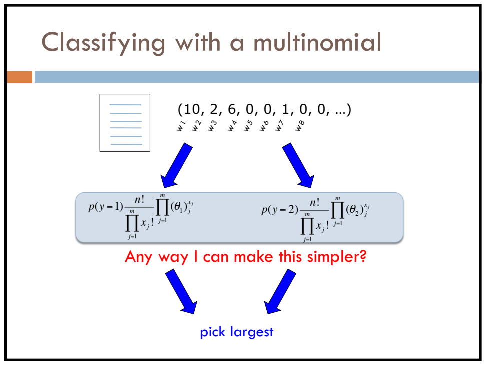
w1: 100 times
w2: 50 times
w3: 10 times
w4: …

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

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

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

65



Classifying with a multinomial

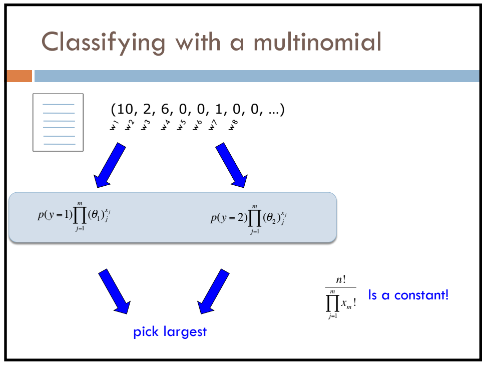
(10, 2, 6, 0, 0, 1, 0, 0, …)

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

Any way I can make this simpler?

pick largest

66



Classifying with a multinomial

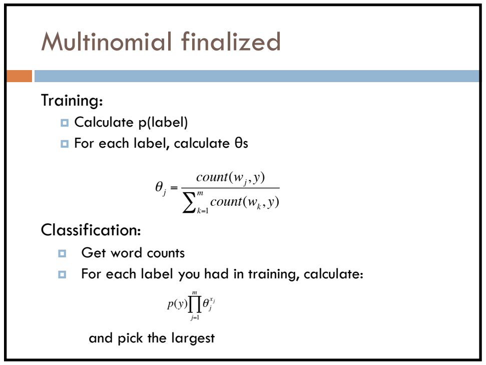
(10, 2, 6, 0, 0, 1, 0, 0, …)

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

pick largest

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

67



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:

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

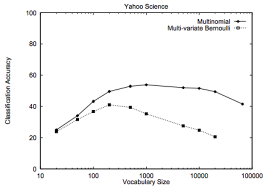
and pick the largest

68

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



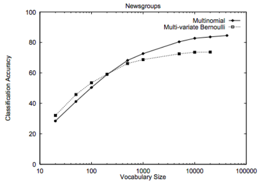
http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

69

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



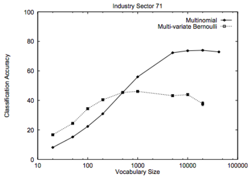
http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

70

Multinomial vs. Bernoulli?

Handles word frequency

Given enough data, tends to performs better



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

71

Maximum likelihood estimation

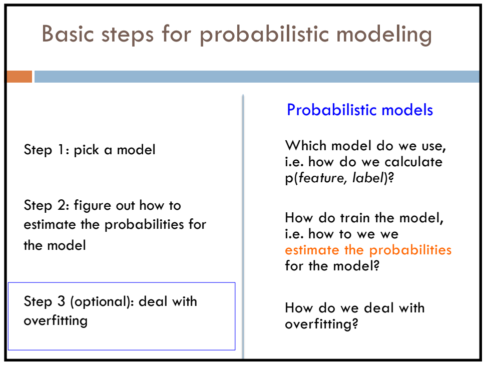
Intuitive

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

72



Basic steps for probabilistic modeling

Step 1: pick a model

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

Step 3 (optional): deal with overfitting

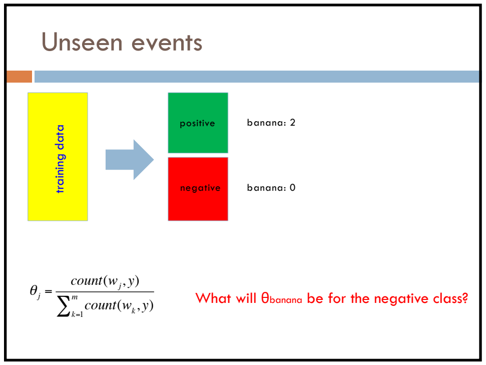
Probabilistic models

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

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

How do we deal with overfitting?

73



Unseen events

training data

positive

negative

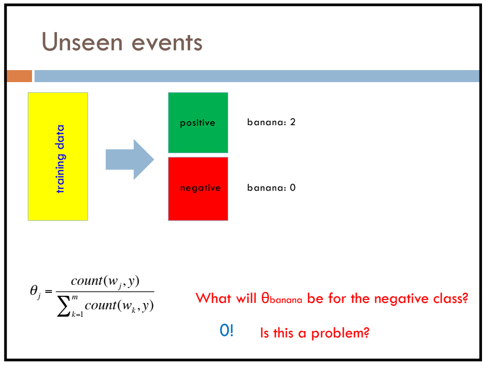
banana: 2

banana: 0

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

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

74



Unseen events

training data

positive

negative

banana: 2

banana: 0

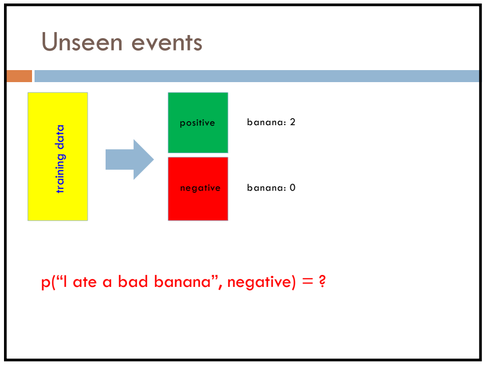
$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

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

0!

Is this a problem?

75



Unseen events

training data

positive

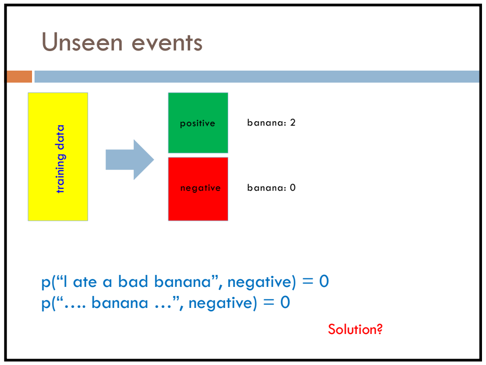
negative

banana: 2

banana: 0

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

76



Unseen events

training data

positive

banana: 2

negative

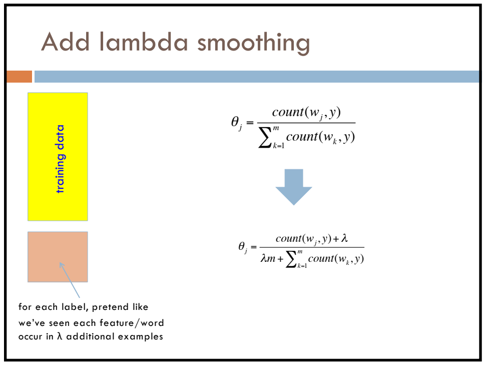
banana: 0

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

p("…. banana …", negative) = 0

Solution?

77



Add lambda smoothing

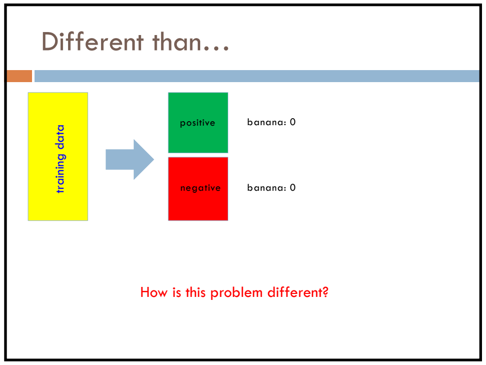
training data

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

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

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

78



Different than…

training data

positive

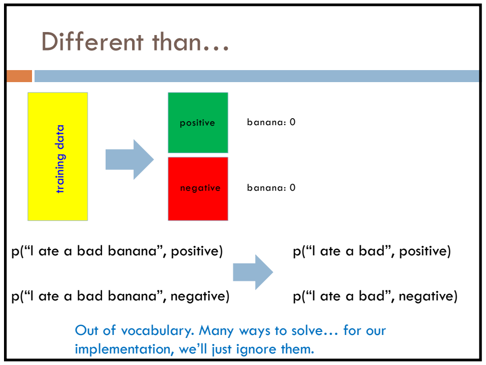
banana: 0

negative

banana: 0

How is this problem different?

79



Different than…

training data

positive

banana: 0

negative

banana: 0

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

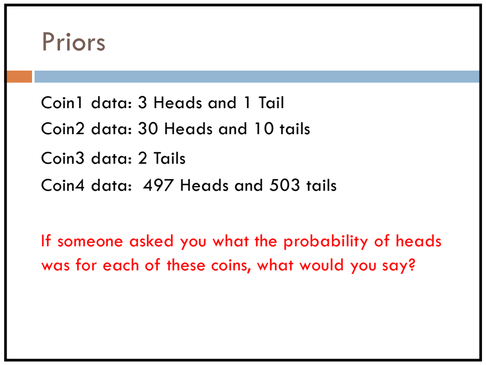
p("I ate a bad banana", negative)

p("I ate a bad", positive)

p("I ate a bad", negative)

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

80



Priors

Coin1 data: 3 Heads and 1 Tail

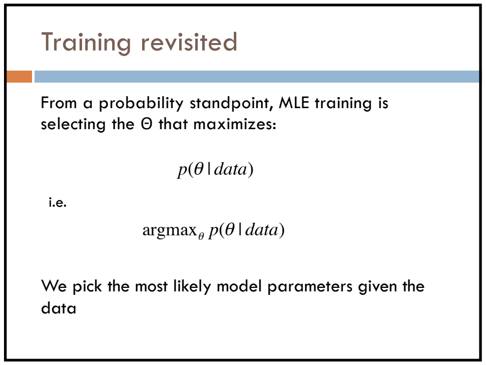
Coin2 data: 30 Heads and 10 tails

Coin3 data: 2 Tails

Coin4 data: 497 Heads and 503 tails

If someone asked you what the probability of heads was for each of these coins, what would you say?

81



Training revisited

From a probability standpoint, MLE training is selecting the Θ that maximizes:

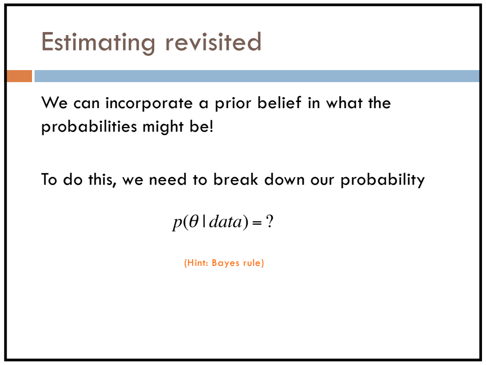
$$p(\theta \mid data)$$

i.e.

$$\operatorname{argmax}_{\theta} p(\theta \mid data)$$

We pick the most likely model parameters given the data

82



Estimating revisited

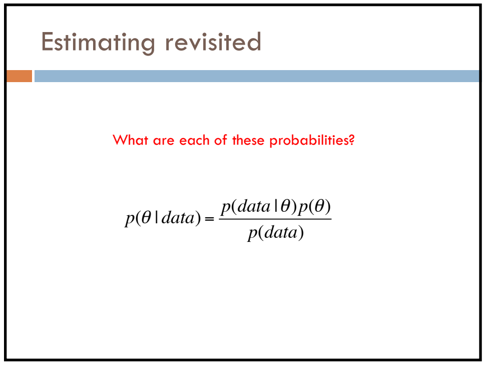
We can incorporate a prior belief in what the probabilities might be!

To do this, we need to break down our probability

$$p(\theta \mid data) = ?$$

(Hint: Bayes rule)

83

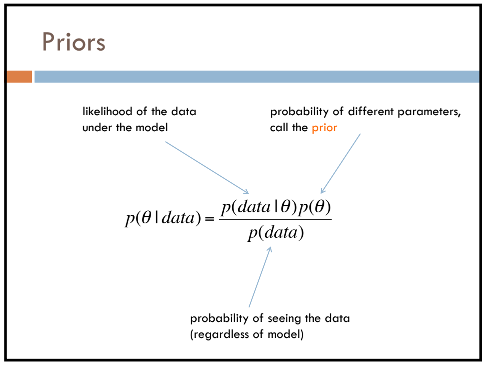


Estimating revisited

What are each of these probabilities?

$$p(\theta \mid data) = \frac{p(data \mid \theta) p(\theta)}{p(data)}$$

84



Priors

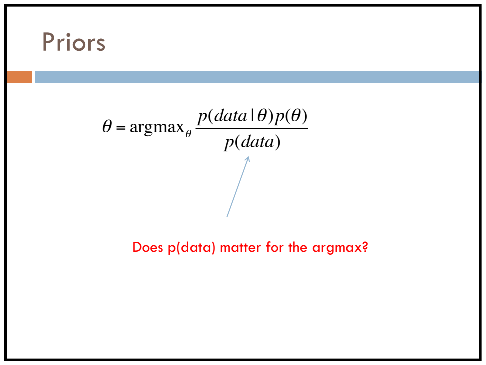
likelihood of the data under the model

probability of different parameters, call the prior

$$p(\theta \mid data) = \frac{p(data \mid \theta)p(\theta)}{p(data)}$$

probability of seeing the data (regardless of model)

85

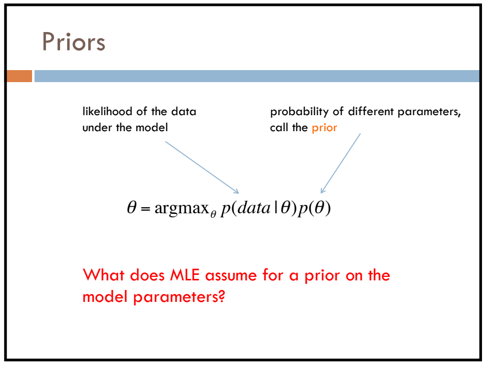


Priors

$$\theta = \text{argmax}_\theta \frac{p(data \mid \theta)p(\theta)}{p(data)}$$

Does p(data) matter for the argmax?

86



Priors

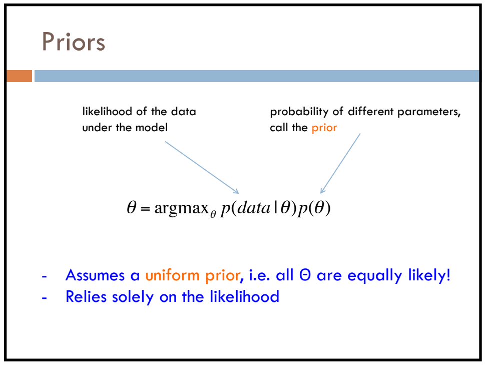
likelihood of the data under the model

probability of different parameters, call the prior

$$\theta = \text{argmax}_\theta \, p(data \mid \theta)p(\theta)$$

What does MLE assume for a prior on the model parameters?

87



Priors

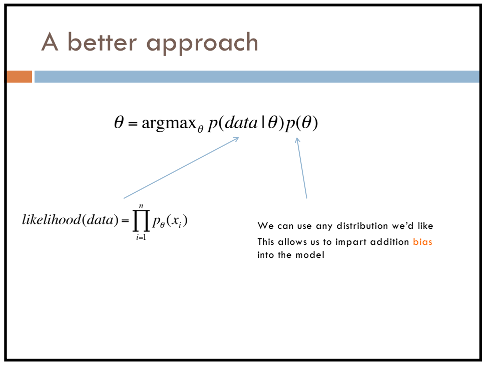
likelihood of the data under the model

probability of different parameters, call the prior

$$\theta = \text{argmax}_\theta \, p(data \mid \theta)p(\theta)$$

- Assumes a uniform prior, i.e. all Θ are equally likely!
- Relies solely on the likelihood

88



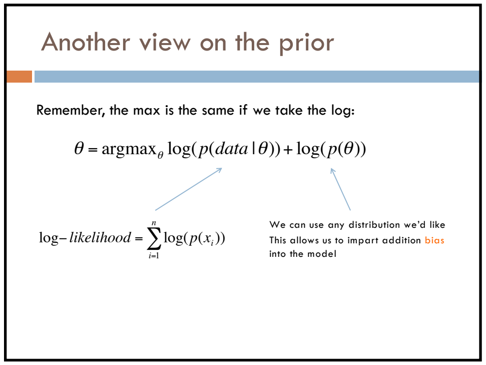
A better approach

$$\theta = \text{argmax}_\theta\, p(data \mid \theta) p(\theta)$$

$$likelihood(data) = \prod_{i=1}^{n} p_\theta(x_i)$$

We can use any distribution we'd like
This allows us to impart addition bias
into the model

89



Another view on the prior

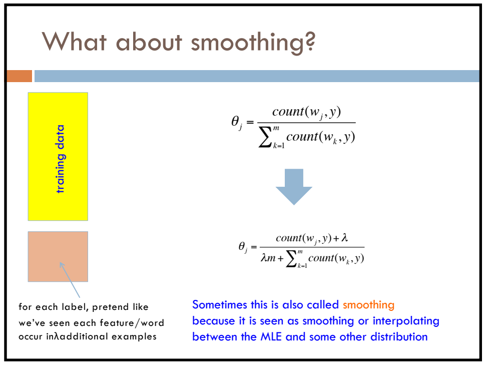
Remember, the max is the same if we take the log:

$$\theta = \text{argmax}_\theta \log(p(data \mid \theta)) + \log(p(\theta))$$

$$\log\text{-}likelihood = \sum_{i=1}^{n} \log(p(x_i))$$

We can use any distribution we'd like
This allows us to impart addition bias
into the model

90



What about smoothing?

training data

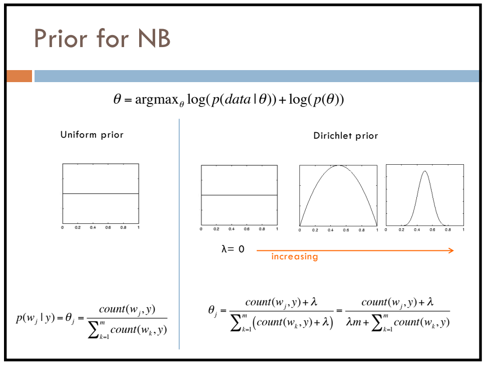
$$\theta_j = \frac{count(w_j, y)}{\sum_{k=1}^{m} count(w_k, y)}$$

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

for each label, pretend like we've seen each feature/word occur inλadditional examples

Sometimes this is also called smoothing because it is seen as smoothing or interpolating between the MLE and some other distribution

91



Prior for NB

$$\theta = \text{argmax}_\theta \log(p(data \mid \theta)) + \log(p(\theta))$$

Uniform prior

Dirichlet prior

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

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

92