

# 343H: Honors AI

Lecture 21: ML: Naïve Bayes

4/8/2014

Kristen Grauman

UT Austin

Slides courtesy of Dan Klein, UC Berkeley

Unless otherwise noted

# Contest

---

- AgentsOrange - wins 6.0 (15.0)
- WickhamBros - wins 5.0 (15.0)
- Eversbots - wins 4.0 (10.0)
- JustDoltAgents - wins 3.0 (8.0)
- OffenseOnlyAgents - wins 2.0 (7.5)
- StaffAgents - wins 1.0 (5.0)
- BaselineAgents - wins 0.0 (2.5)

# Announcements

---

- Contest qualification runs nightly til 4/28
- PS 4 due Monday 4/14
- Thurs 4/8: Guest lecture by Dr. Mugan
  - Perceptrons
- Tues 4/15: Video lecture online
  - Access instructions will be sent on Piazza
- Thurs 4/17: We will meet for CS Colloq by Prof. Deva Ramanan (GDC 2.216)
  - Reading assignment on his lecture

# Recap: Probabilistic reasoning over time

---

- Markov Models
- Hidden Markov Models (HMMs)
  - Forward algorithm (repeated variable elimination) to infer belief state
  - Particle filtering (likelihood weighting with some tweaks)
  - Viterbi algorithm to infer most likely explanation
- Dynamic Bayes Nets
  - Particle filtering

# Machine learning

---

- Up until now: how to use a model to make optimal decisions
- Machine learning: how to acquire a model from data/experience
  - Learning parameters (e.g., probabilities)
  - Learning structure (e.g., BN graphs)
  - Learning hidden concepts (e.g., clustering)
- Today: model-based classification with Naïve Bayes

# Example: Spam Filter

---

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: \$dd, CAPS
  - Non-text: SenderInContacts
  - ...



Dear Sir.

First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES  
FOR ONLY \$99

Ok, I know this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

# Example: Digit Recognition

---

- Input: images / pixel grids
- Output: a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images
- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops
  - ...



0



1



2



1

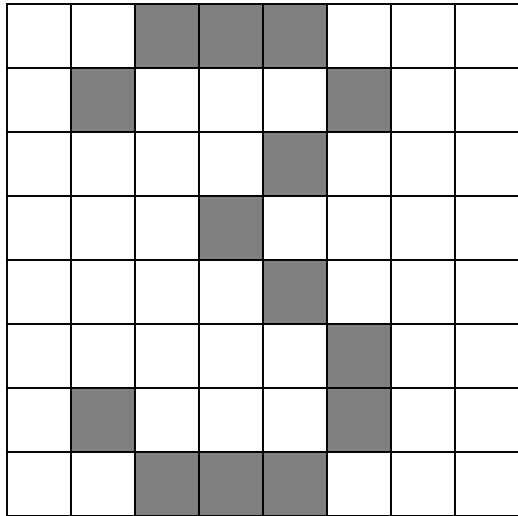


??

# A Digit Recognizer

---

- Input: pixel grids



- Output: a digit 0-9





# Other Classification Tasks

---

- In classification, we predict labels  $y$  (classes) for inputs  $x$
- Examples:
  - Spam detection (input: document, classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, classes: diseases)
  - Automatic essay grader (input: document, classes: grades)
  - Fraud detection (input: account activity, classes: fraud / no fraud)
  - Customer service email routing
  - ... many more
- Classification is an important commercial technology!

# Model-based classification

---

- **Model-based approach**
  - Build a model (e.g., Bayes' net) where both the label and features are random variables
  - Instantiate any observed features
  - Query for the distribution of the label conditioned on the features
- **Challenges**
  - What structure should the BN have?
  - How should we learn its parameters?

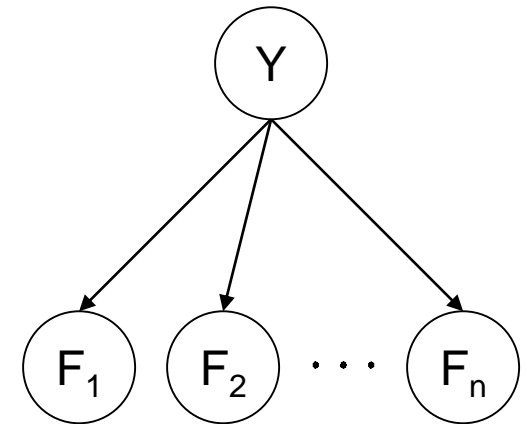
# Naïve Bayes for Digits

- Naïve Bayes: assume all features are independent effects of the label
- Simple version for digits:
  - One feature  $F_{ij}$  for each grid position  $\langle i,j \rangle$
  - Possible feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
  - Each input maps to a feature vector, e.g.

1  $\rightarrow \langle F_{0,0} = 0 \ F_{0,1} = 0 \ F_{0,2} = 1 \ F_{0,3} = 1 \ F_{0,4} = 0 \ \dots F_{15,15} = 0 \rangle$

- Here: lots of features, each is binary valued
- Naïve Bayes model:

$$P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$$



# General Naïve Bayes

- A general *naive Bayes* model:

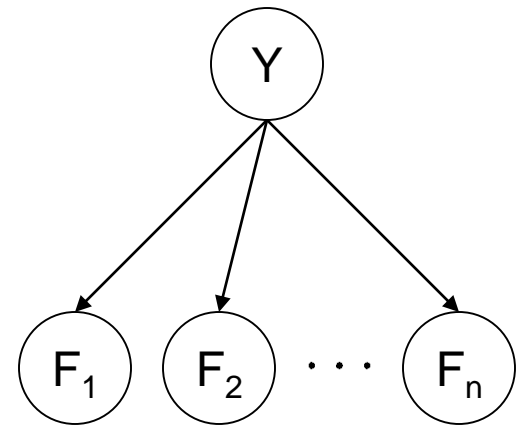
$|Y| \times |F|^n$   
parameters

$$P(Y, F_1 \dots F_n) =$$

$$P(Y) \prod_i P(F_i|Y)$$

$|Y|$  parameters

$n \times |F| \times |Y|$   
parameters



- We only specify how each feature depends on the class
- Total number of parameters is *linear* in  $n$

# Inference for Naïve Bayes

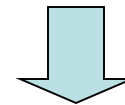
- Goal: compute posterior over label variable  $Y$ 
  - Step 1: get joint probability of causes and evidence

$$P(Y, f_1 \dots f_n) =$$

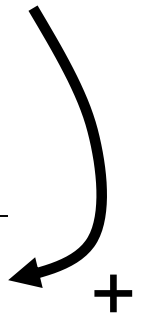
$$\begin{bmatrix} P(y_1, f_1 \dots f_n) \\ P(y_2, f_1 \dots f_n) \\ \vdots \\ P(y_k, f_1 \dots f_n) \end{bmatrix} \Rightarrow \begin{bmatrix} P(f_1) \prod_i P(f_i|y_1) \\ P(f_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(f_k) \prod_i P(f_i|y_k) \end{bmatrix}$$

- Step 2: get probability of evidence
- Step 3: renormalize

$$P(f_1 \dots f_n)$$



$$P(Y|f_1 \dots f_n)$$



# General Naïve Bayes

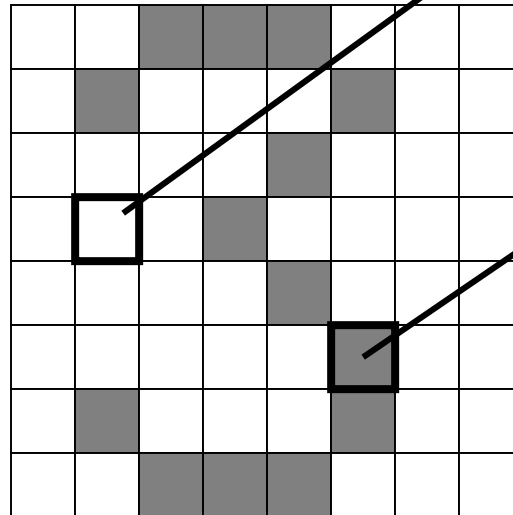
---

- What do we need in order to use naïve Bayes?
  - Inference (we just saw this part)
    - Start with a bunch of probabilities:  $P(Y)$  and the  $P(F_i|Y)$  tables
    - Use standard inference to compute  $P(Y|F_1 \dots F_n)$
    - Nothing new here
  - Estimates of local conditional probability tables
    - $P(Y)$ , the prior over labels
    - $P(F_i|Y)$  for each feature (evidence variable)
    - These probabilities are collectively called the *parameters* of the model and denoted by  $\theta$
    - Up until now, we assumed these appeared by magic, but...
    - ...they typically come from training data: we'll look at this now

# Examples: Conditional probabilities

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$     $P(F_{5,5} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

# Naïve Bayes for Text


---

- Bag-of-Words Naïve Bayes:

- Features:  $W_i$  is the word at position  $i$
- Predict unknown class label (spam vs. ham)
- Assume evidence features (e.g. the words) are independent
- New: each  $W_i$  is identically distributed.

*Word at position  
 $i$ , not  $i^{\text{th}}$  word in  
the dictionary!*

- Generative model

$$P(C, W_1 \dots W_n) = P(C) \prod_i P(W_i | C)$$


- “Tied” distributions and bag-of-words

- Usually, each variable gets its own conditional probability distribution  $P(F|Y)$
- In a bag-of-words model
  - Each position is identically distributed
  - All positions share the same conditional probs  $P(W|C)$
  - Why make this assumption?



# Example: Spam Filtering

---

- **Model:**  $P(C, W_1 \dots W_n) = P(C) \prod_i P(W_i|C)$
- What are the parameters?

$P(C)$

ham : 0.66
spam: 0.33

$P(W|\text{spam})$

the : 0.0156
to : 0.0153
and : 0.0115
of : 0.0095
you : 0.0093
a : 0.0086
with: 0.0080
from: 0.0075
...

$P(W|\text{ham})$

the : 0.0210
to : 0.0133
of : 0.0119
2002: 0.0110
with: 0.0108
from: 0.0107
and : 0.0105
a : 0.0100
...

# Spam Example

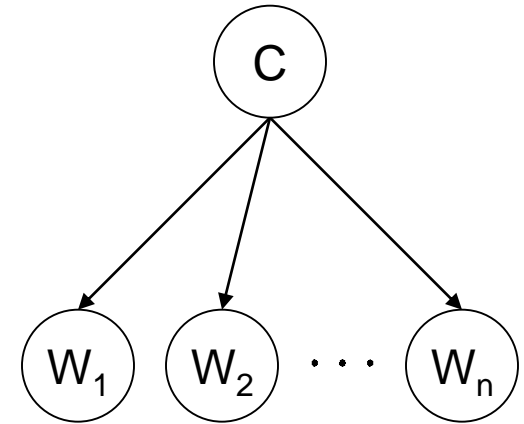
---

Word	P(w spam)	P(w ham)	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4
Gary	0.00002	0.00021	-11.8	-8.9
would	0.00069	0.00084	-19.1	-16.0
you	0.00881	0.00304	-23.8	-21.8
like	0.00086	0.00083	-30.9	-28.9
to	0.01517	0.01339	-35.1	-33.2
lose	0.00008	0.00002	-44.5	-44.0
weight	0.00016	0.00002	-53.3	-55.0
while	0.00027	0.00027	-61.5	-63.2
you	0.00881	0.00304	-66.2	-69.0
sleep	0.00006	0.00001	-76.0	-80.5

---

$$P(\text{spam} | w) = 98.9$$

# Image classification with Naïve Bayes



$$c^* = \arg \max_c p(c | w) \propto p(c) p(w | c) = p(c) \prod_{n=1}^N p(w_n | c)$$

Object class decision

Prior prob. of the object classes

Image likelihood given the class

$N$  patches

# Important Concepts

---

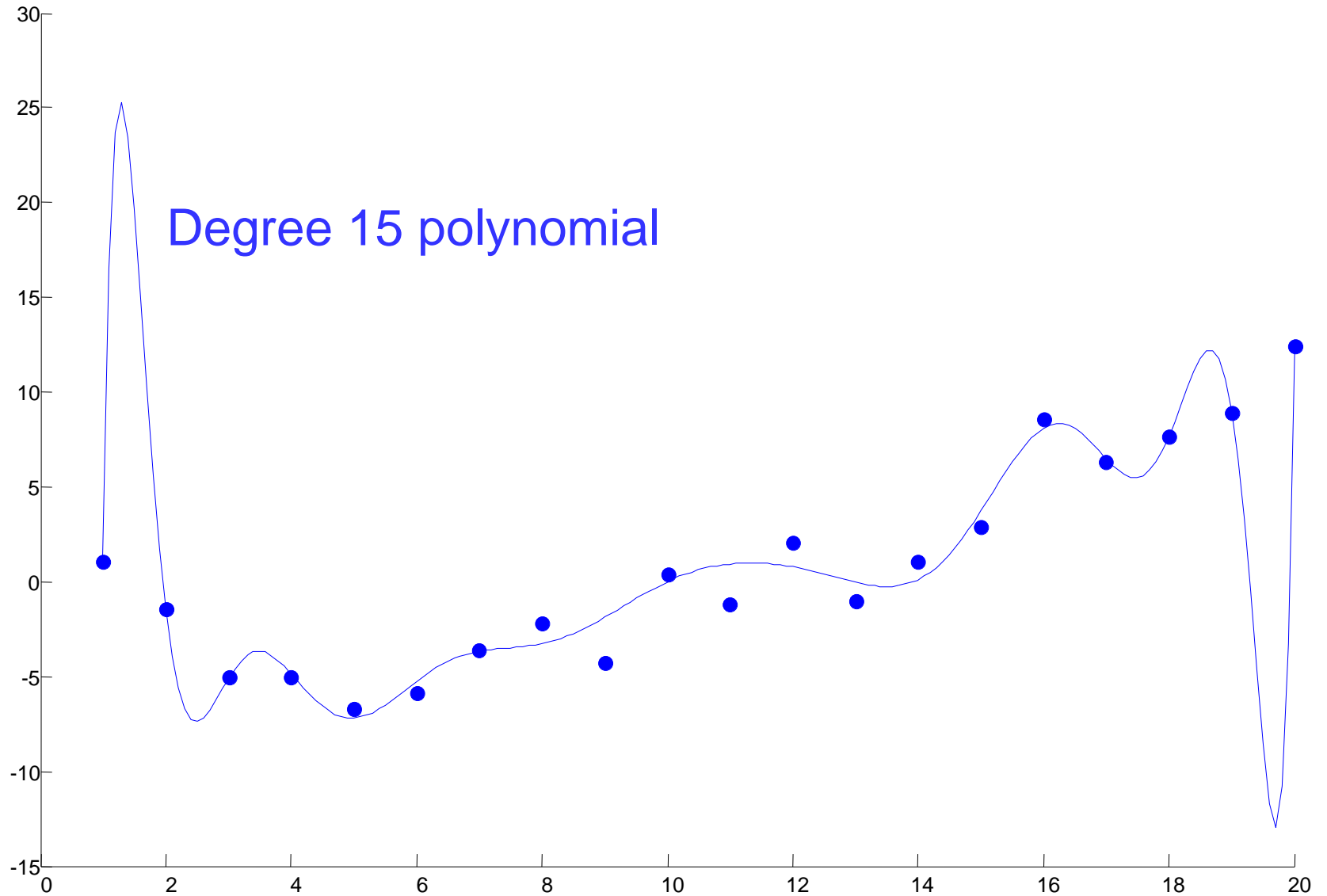
- **Data:** labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out set
  - Test set
- **Features:** attribute-value pairs which characterize each  $x$
- **Experimentation cycle**
  - Learn parameters (e.g. model probabilities) on training set
  - (Tune hyperparameters on held-out set)
  - Compute accuracy of test set
  - Very important: never “peek” at the test set!
- **Evaluation**
  - Accuracy: fraction of instances predicted correctly
- **Overfitting and generalization**
  - Want a classifier which does well on *test* data
  - Overfitting: fitting the training data very closely, but not generalizing well
  - We’ll investigate overfitting and generalization formally in a few lectures

Training  
Data

Held-Out  
Data

Test  
Data

# Overfitting



# Example: Overfitting

$P(\text{features}, C = 2)$

$$P(C = 2) = 0.1$$

$$P(\text{on}|C = 2) = 0.8$$

$$P(\text{on}|C = 2) = 0.1$$

$$P(\text{off}|C = 2) = 0.1$$

$$P(\text{on}|C = 2) = 0.01$$

$P(\text{features}, C = 3)$

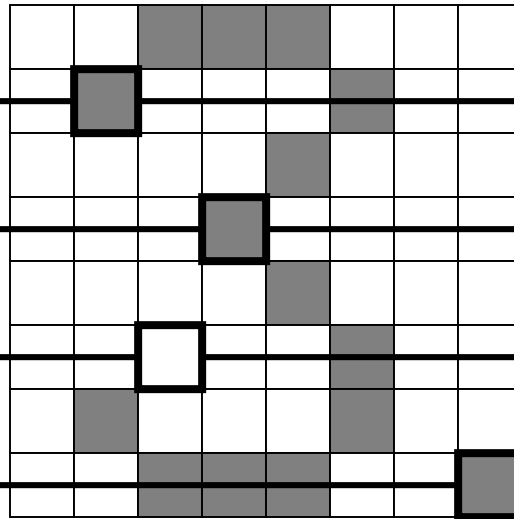
$$P(C = 3) = 0.1$$

$$P(\text{on}|C = 3) = 0.8$$

$$P(\text{on}|C = 3) = 0.9$$

$$P(\text{off}|C = 3) = 0.7$$

$$P(\text{on}|C = 3) = 0.0$$



*2 wins!!*

# Example: Overfitting

- Posterior determined by *relative* probabilities (odds ratios):

$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

south-west	:	inf
nation	:	inf
morally	:	inf
nicely	:	inf
extent	:	inf
seriously	:	inf
...		

$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

screens	:	inf
minute	:	inf
guaranteed	:	inf
\$205.00	:	inf
delivery	:	inf
signature	:	inf
...		

*What went wrong here?*

# Generalization and Overfitting

---

- Relative frequency parameters will **overfit** the training data!
  - Just because we never saw a 3 with pixel (15,15) on during training doesn't mean we won't see it at test time
  - Unlikely that every occurrence of "minute" is 100% spam
  - Unlikely that every occurrence of "seriously" is 100% ham
  - What about all the words that don't occur in the training set at all?
  - In general, we can't go around giving unseen events zero probability
- As an extreme case, imagine using the entire email as the only feature
  - Would get the training data perfect (if deterministic labeling)
  - Wouldn't *generalize* at all
  - Just making the bag-of-words assumption gives us some generalization, but isn't enough
- To generalize better: we need to **smooth** or **regularize** the estimates

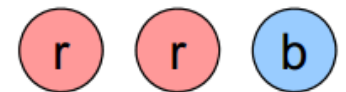


# Parameter estimation

---

- Estimating the distribution of a random variable
- **Elicitation**: ask a human (why is this hard?)
- **Empirically**: use training data (learning!)
  - E.g., for each outcome  $x$ , look at the **empirical rate** of that value

$$P_{\text{ML}}(x) = \frac{\text{count}(x)}{\text{total samples}}$$



$$P_{\text{ML}}(r) = 2/3$$

- This is the estimate that maximizes the **likelihood of the data**

$$L(x, \theta) = \prod_i P_{\theta}(x_i)$$

---

# Maximum likelihood?

---

- Relative frequencies are the maximum likelihood estimates

$$\begin{aligned}\theta_{ML} &= \arg \max_{\theta} P(\mathbf{X}|\theta) \\ &= \arg \max_{\theta} \prod_i P_{\theta}(X_i)\end{aligned} \quad \Rightarrow \quad P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

- Another option is to consider the most likely parameter value given the data

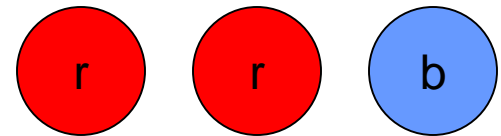
$$\begin{aligned}\theta_{MAP} &= \arg \max_{\theta} P(\theta|\mathbf{X}) \\ &= \arg \max_{\theta} P(\mathbf{X}|\theta)P(\theta)/P(\mathbf{X}) \quad \Rightarrow \quad \text{????} \\ &= \arg \max_{\theta} P(\mathbf{X}|\theta)P(\theta)\end{aligned}$$

# Estimation: Laplace Smoothing

---

- Laplace's estimate:
  - Pretend you saw every outcome once more than you actually did

$$\begin{aligned} P_{LAP}(x) &= \frac{c(x) + 1}{\sum_x [c(x) + 1]} \\ &= \frac{c(x) + 1}{N + |X|} \end{aligned}$$



$$P_{ML}(X) = \left\langle \frac{2}{3}, \frac{1}{3} \right\rangle$$

$$P_{LAP}(X) = \left\langle \frac{3}{5}, \frac{2}{5} \right\rangle$$

# Estimation: Laplace Smoothing

---

- Laplace's estimate (extended):

- Pretend you saw every outcome  $k$  extra times

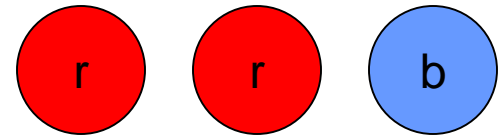
$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

- What's Laplace with  $k = 0$ ?
- $k$  is the **strength** of the prior

- Laplace for conditionals:

- Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x, y) + k}{c(y) + k|X|}$$



$$P_{LAP,0}(X) = \left\langle \frac{2}{3}, \frac{1}{3} \right\rangle$$

$$P_{LAP,1}(X) = \left\langle \frac{3}{5}, \frac{2}{5} \right\rangle$$

$$P_{LAP,100}(X) = \left\langle \frac{102}{203}, \frac{101}{203} \right\rangle$$

# Real NB: Smoothing

---

- For real classification problems, smoothing is critical
- New odds ratios:

$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

helvetica	:	11.4
seems	:	10.8
group	:	10.2
ago	:	8.4
areas	:	8.3
...		

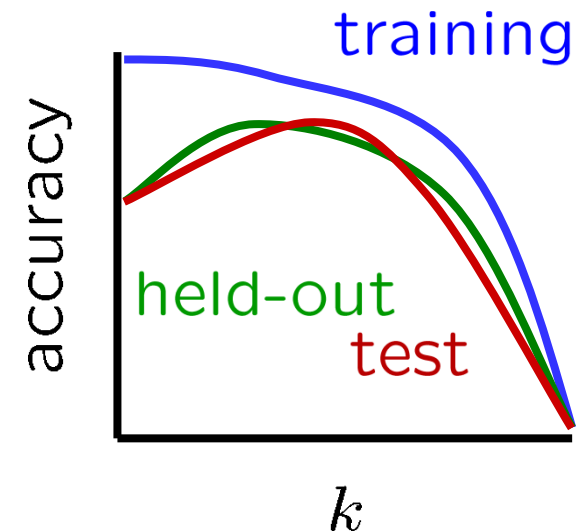
$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

verdana	:	28.8
Credit	:	28.4
ORDER	:	27.2
<FONT>	:	26.9
money	:	26.5
...		

*Do these make more sense?*

# Tuning on Held-Out Data

- Now we've got two kinds of unknowns
  - Parameters: the probabilities  $P(X|Y)$ ,  $P(Y)$
  - Hyperparameters, like the amount of smoothing to do:  $k$
- Where to learn?
  - Learn parameters from training data
  - Tune hyperparameters on different data
    - Why?
  - For each value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data



# Summary

---

- Model-based classification
- Naïve Bayes
  - Spam and digits examples
- Generalization and overfitting
  - Data splits, held-out data, hyperparameter tuning
  - Laplace smoothing