

CS 188: Artificial Intelligence

Naïve Bayes I

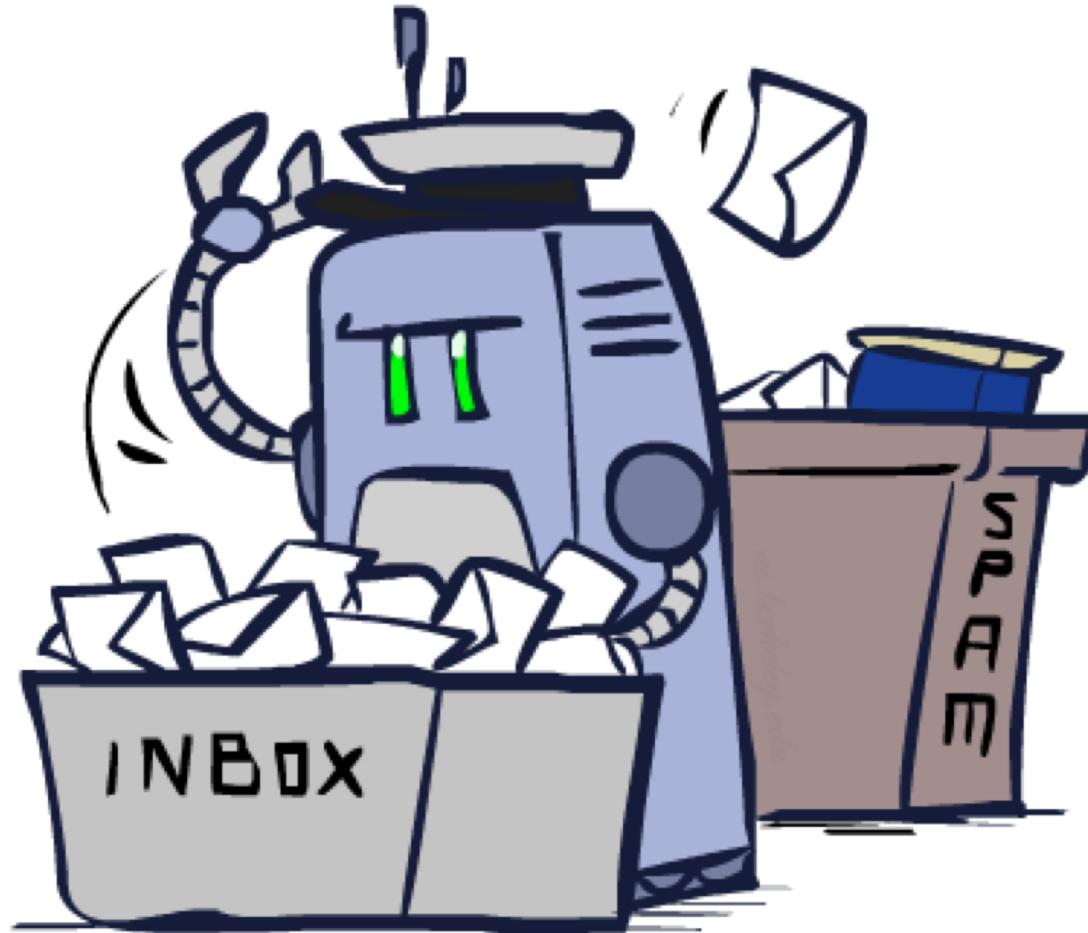


Instructor: Mesut XiaoCheng Yang, Carl Qi--- University of California, Berkeley

Machine Learning

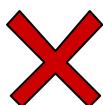
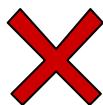
- Up until now: how use a model to make optimal decisions
- Machine learning: how to acquire a model from data / experience
 - Learning parameters (e.g. probabilities)
 - Learning structure (e.g. BN graphs)
 - Learning hidden concepts (e.g. clustering, neural nets)
- Today: model-based classification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

Classification



Example: Spam Filter

- Input: an 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, WidelyBroadcast
 - ...



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: $(6,8)=\text{ON}$
 - Shape Patterns: NumComponents, AspectRatio, NumLoops
 - ...
 - Features are increasingly induced rather than crafted



0



1



2



1



??

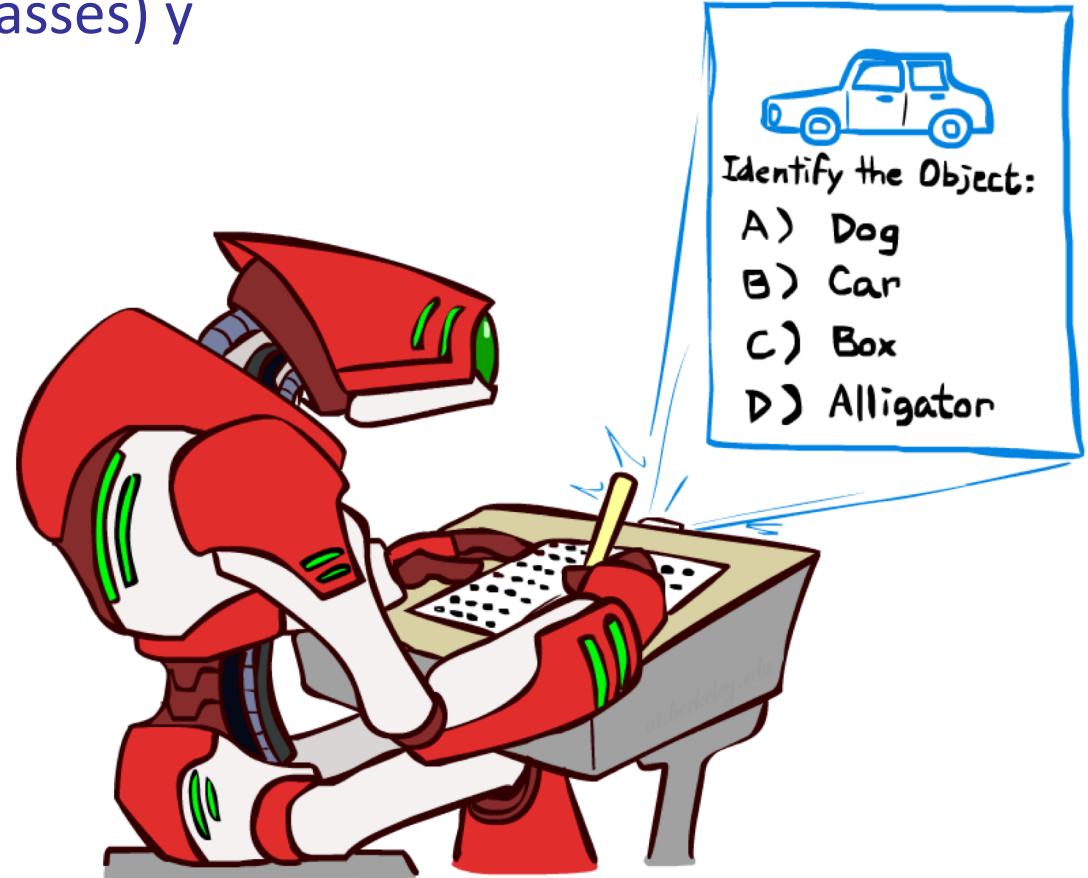
Other Classification Tasks

- Classification: given inputs x , predict labels (classes) y

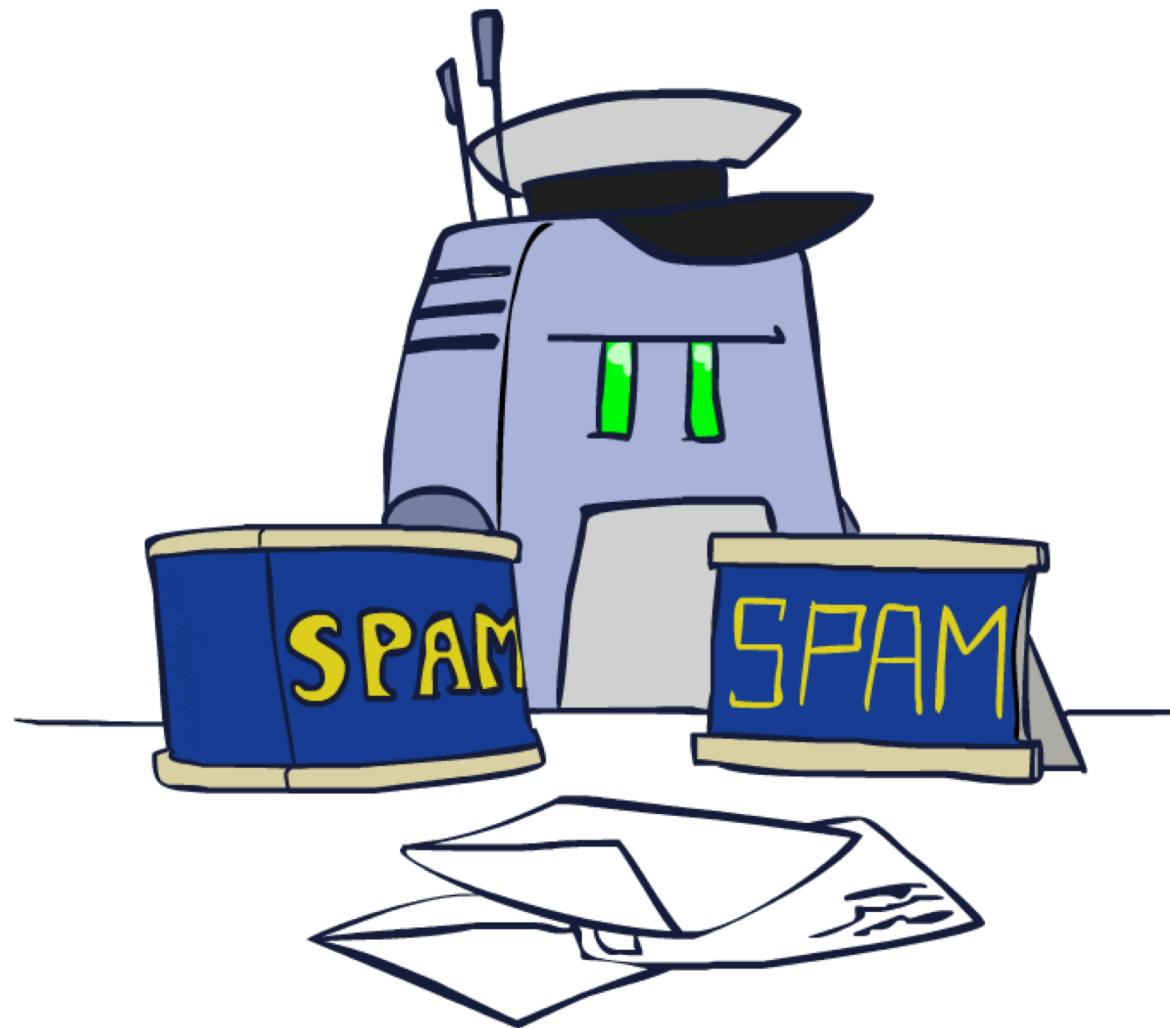
- Examples:

- Medical diagnosis (input: symptoms, classes: diseases)
- Fraud detection (input: account activity, classes: fraud / no fraud)
- Automatic essay grading (input: document, classes: grades)
- Customer service email routing
- Review sentiment
- Language ID
- ... many more

- Classification is an important commercial technology!



Model-Based Classification



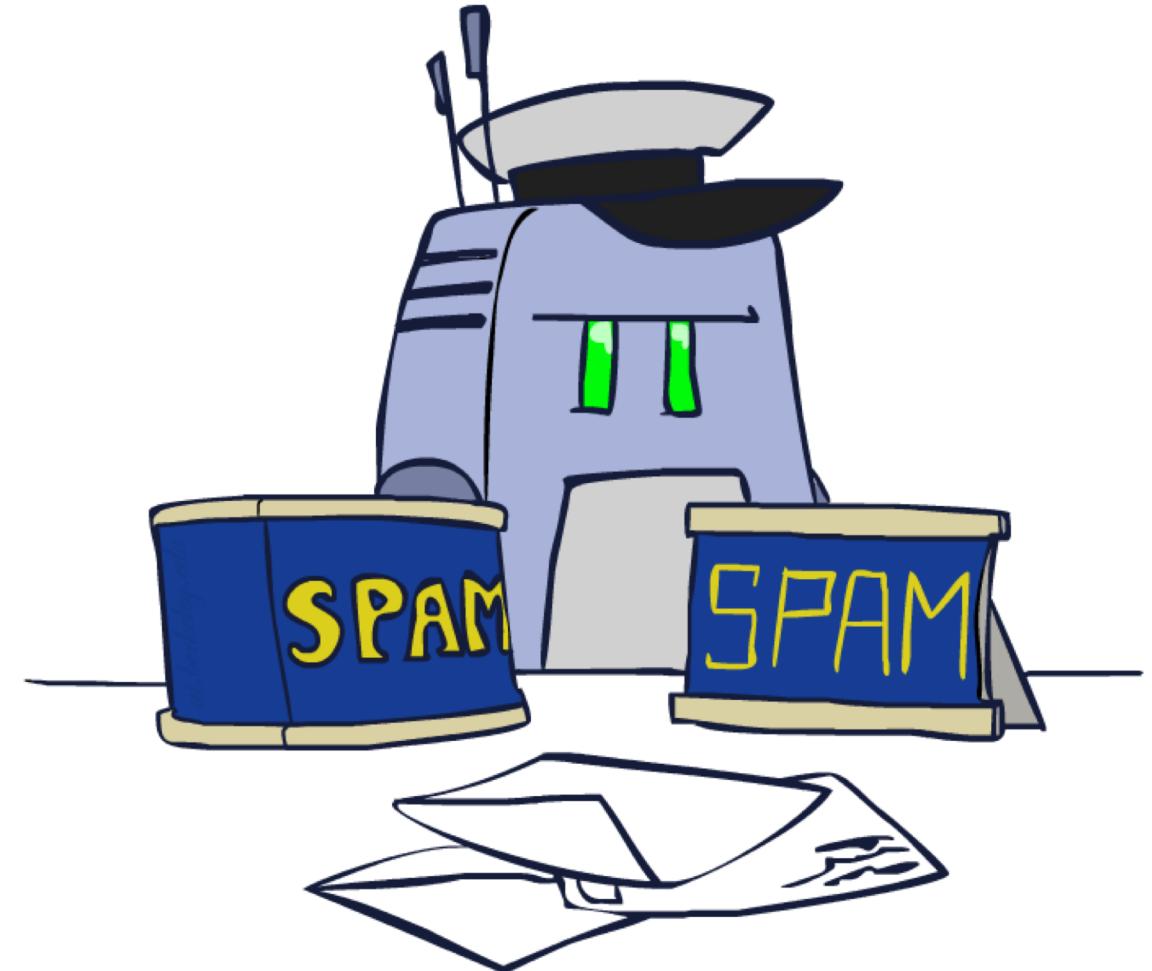
Model-Based Classification

- Model-based approach

- Build a model (e.g. Bayes' net) where both the output label and input 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

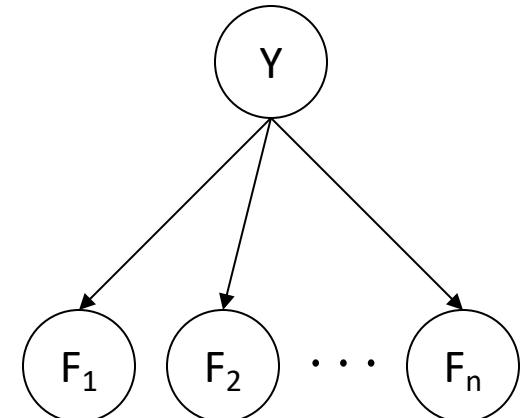
- Simple digit recognition version:

- One feature (variable) F_{ij} for each grid position $\langle i, j \rangle$
- 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.



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

- Here: lots of features, each is binary valued
- Naïve Bayes model: $P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$
- What do we need to learn?



General Naïve Bayes

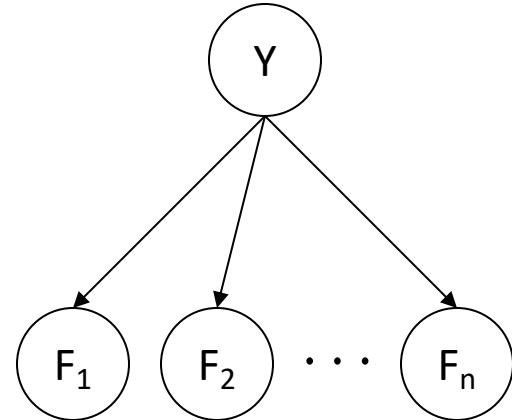
- A general Naive Bayes model:

$|Y|$ parameters

$$P(Y, F_1 \dots F_n) = P(Y) \prod_i P(F_i | Y)$$

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

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



- We only have to specify how each feature depends on the class
- Total number of parameters is *linear* in n
- Model is very simplistic, but often works anyway

Inference for Naïve Bayes

- Goal: compute posterior distribution over label variable Y
 - Step 1: get joint probability of label and evidence for each label

$$P(Y, f_1 \dots f_n) = \begin{bmatrix} P(y_1, f_1 \dots f_n) \\ P(y_2, f_1 \dots f_n) \\ \vdots \\ P(y_k, f_1 \dots f_n) \end{bmatrix} \xrightarrow{\text{ }} \frac{\begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}}{P(f_1 \dots f_n)}$$

- Step 2: sum to get probability of evidence
- Step 3: normalize by dividing Step 1 by Step 2

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

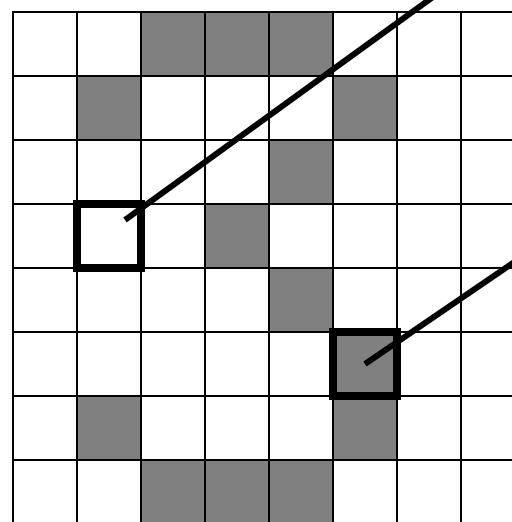
General Naïve Bayes

- What do we need in order to use Naïve Bayes?
 - Inference method (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 θ
 - Up until now, we assumed these appeared by magic, but...
 - ...they typically come from training data counts: we'll look at this soon

Example: 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 |



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

| | |
|---|------|
| 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 |

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

| | |
|---|------|
| 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
 - As before: predict label conditioned on feature variables (spam vs. ham)
 - As before: assume features are conditionally independent given label
 - New: each W_i is identically distributed
- Generative model: $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$

Word at position i, not i^{th} word in the dictionary!
- “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 $P(W|Y)$
 - Why make this assumption?
 - Called “bag-of-words” because model is insensitive to word order or reordering

Example: Spam Filtering

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

$P(Y)$

| |
|------------|
| 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 |
| ... |

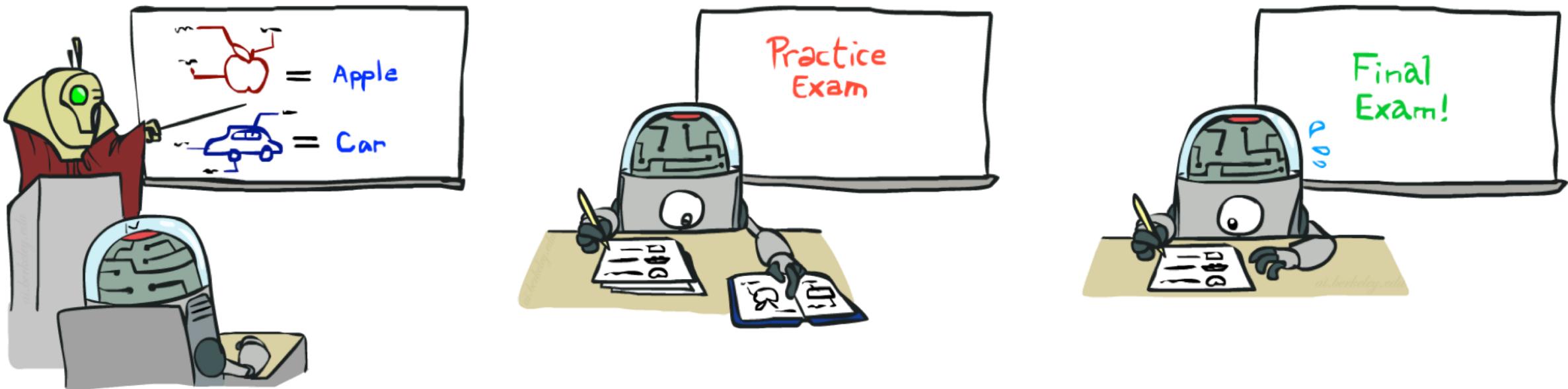
- Where do these tables come from?

Spam Example

| Word | P(w spam) | P(w ham) | Tot Spam | Tot Ham |
|---------|-----------|----------|----------|---------|
| (prior) | 0.33333 | 0.66666 | -1.1 | -0.4 |

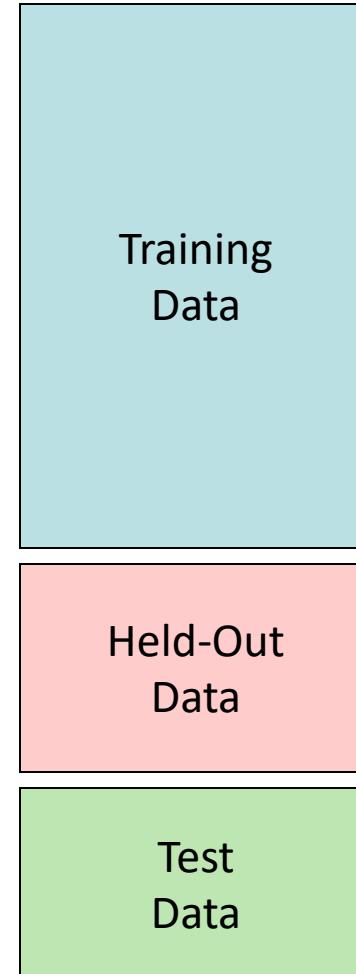
$$P(\text{spam} | w) = 98.9$$

Training and Testing

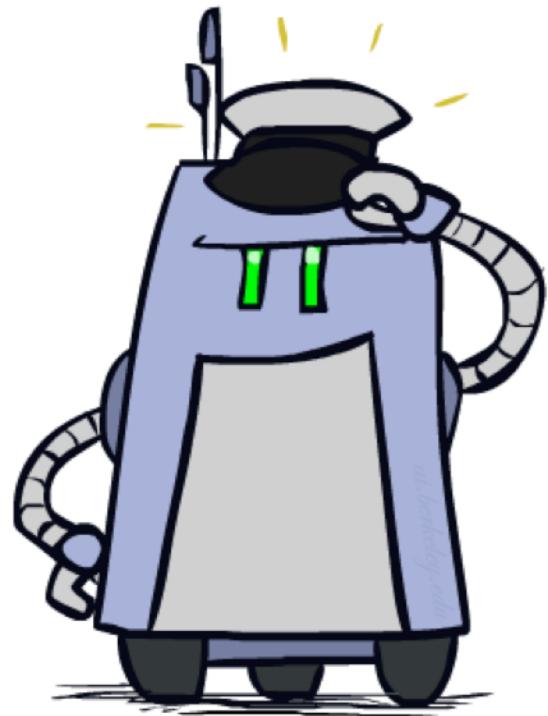
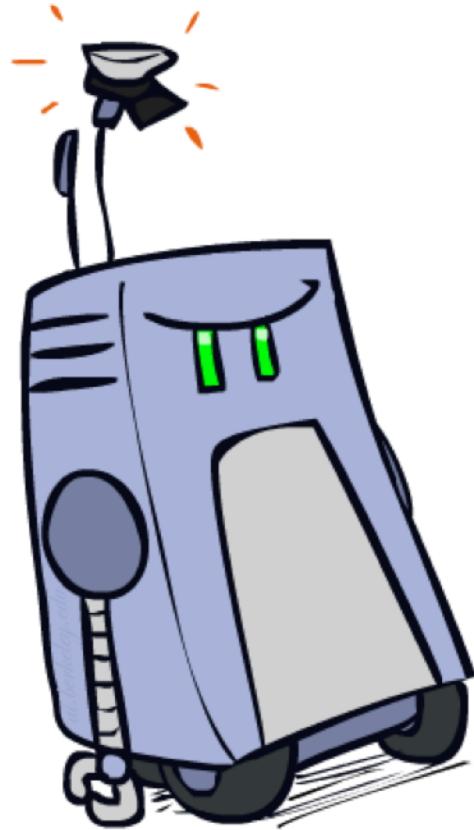


Important Concepts

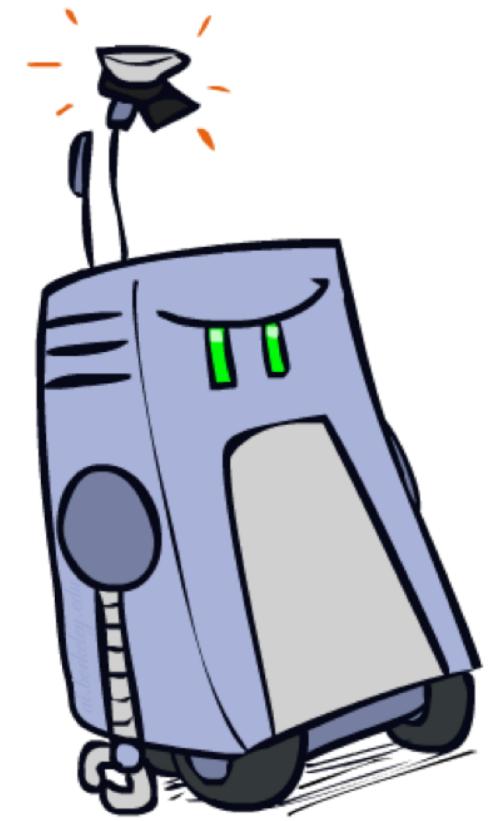
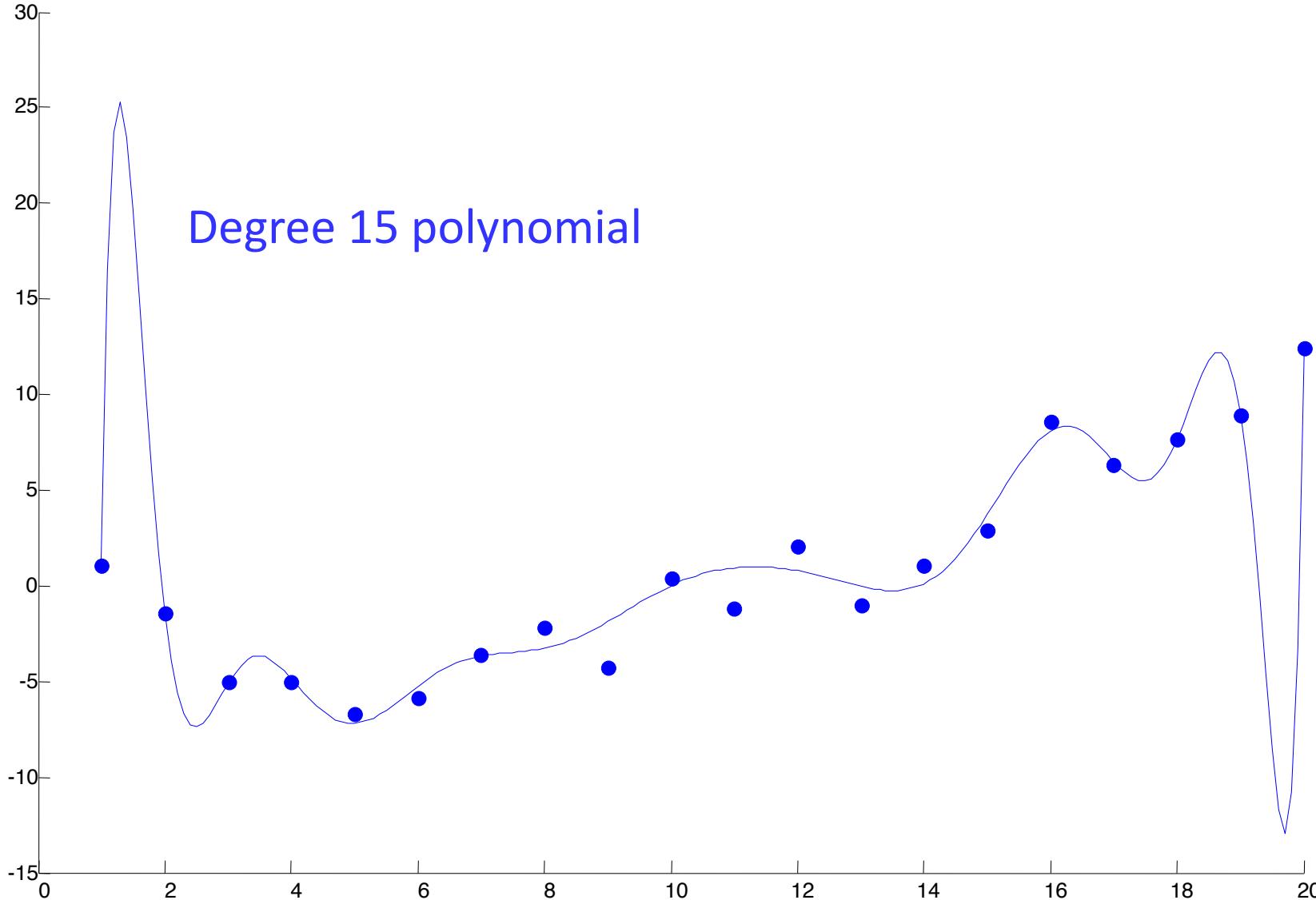
- Data: labeled instances (e.g. emails marked spam/ham)
 - Training set
 - Held out set
 - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
 - Learn parameters (e.g. model probabilities) on training set
 - (Tune hyperparameters on held-out set)
 - Compute accuracy of test set
 - Very important: never “peek” at the test set!
- Evaluation (many metrics possible, e.g. accuracy)
 - 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



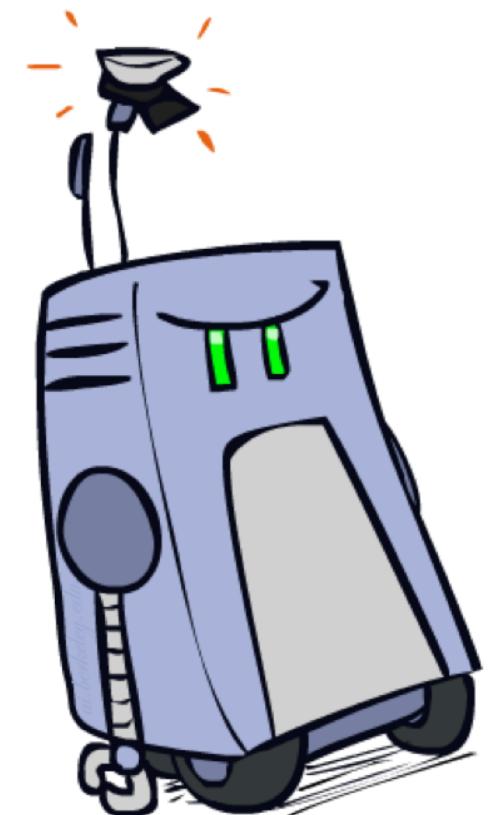
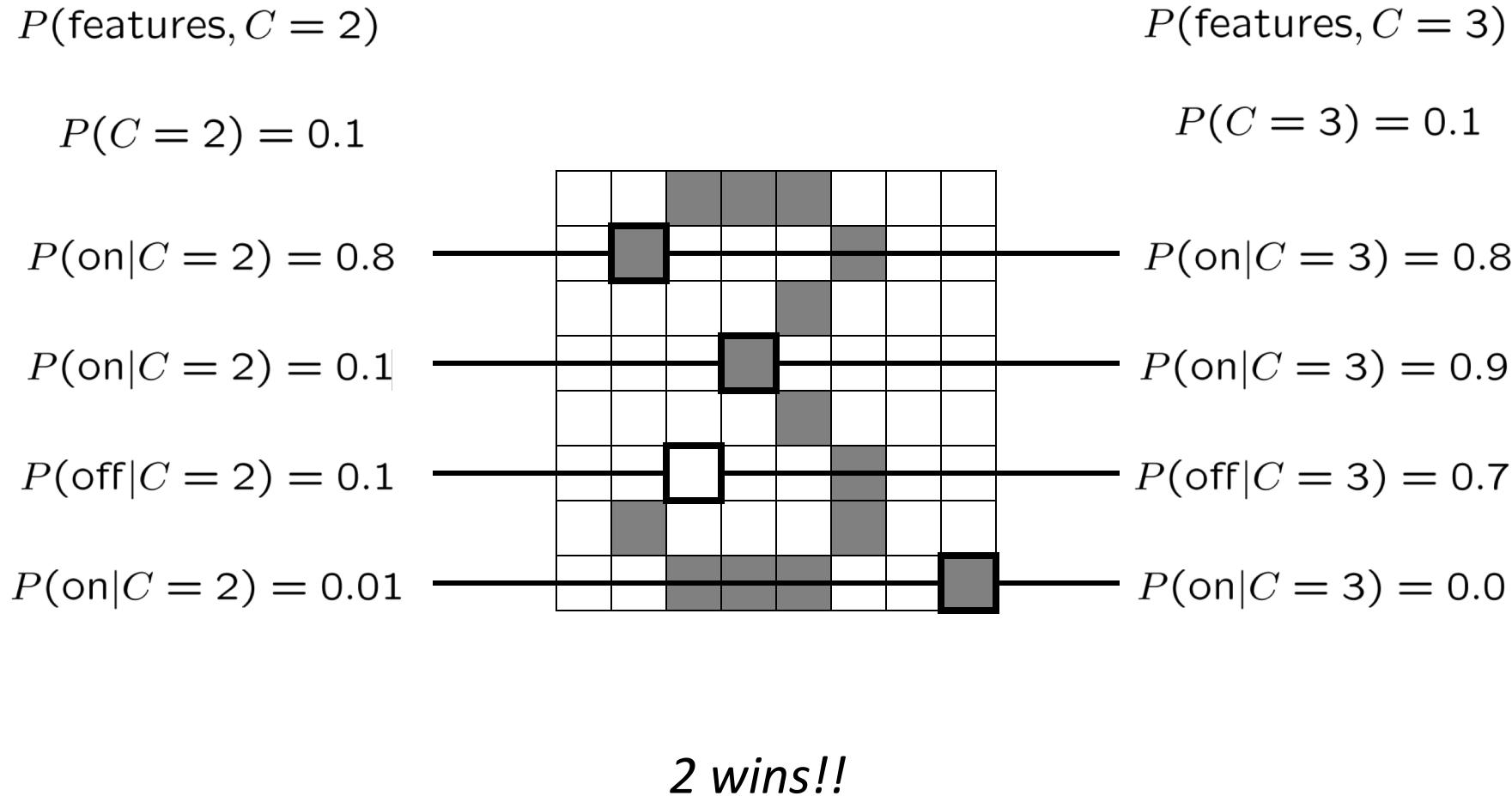
Generalization and Overfitting



Overfitting



Example: Overfitting



Example: Overfitting

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

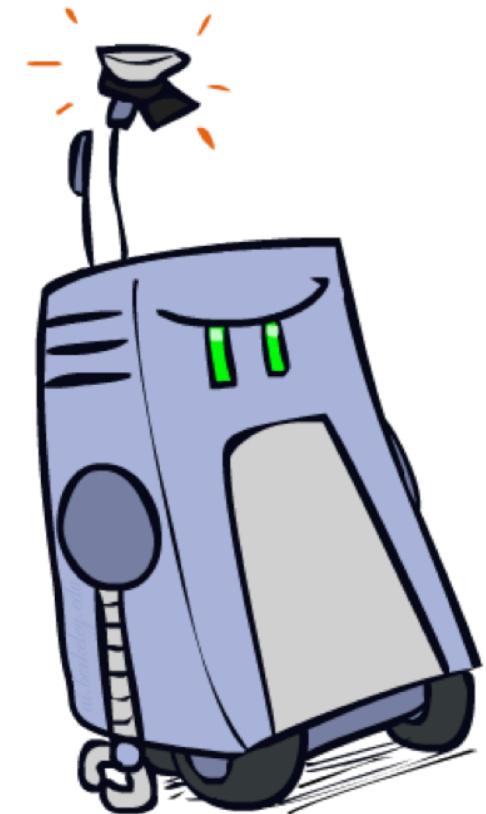
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

```
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 (e.g. document ID)
 - 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

Next Time: Parameter Estimation
