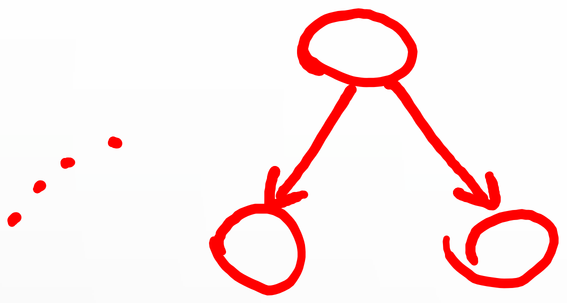
Artificial Intelligence
CE-417, Group 1
Computer Eng. Department
Sharif University of Technology

Spring 2024

By Mohammad Hossein Rohban, Ph.D.

Courtesy: Most slides are adopted from CSE-573 (Washington U.), original slides for the textbook, and CS-188 (UC. Berkeley).



Machine Learning → Data

Key Concepts

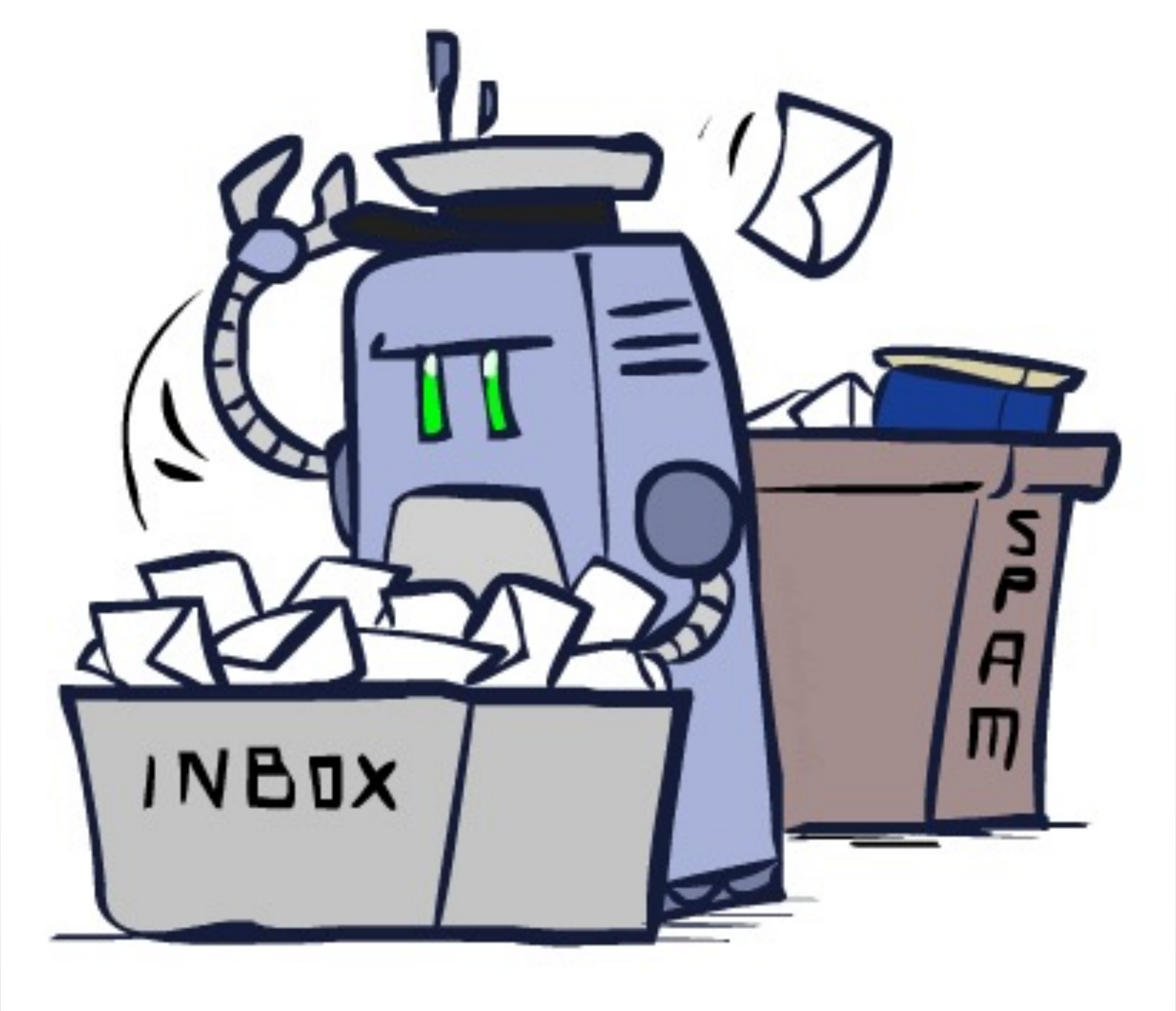
Generalization

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)

- Today: model-based classification with naive Bayes


INBOX
SPAM

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
  - …

Dear Sir.

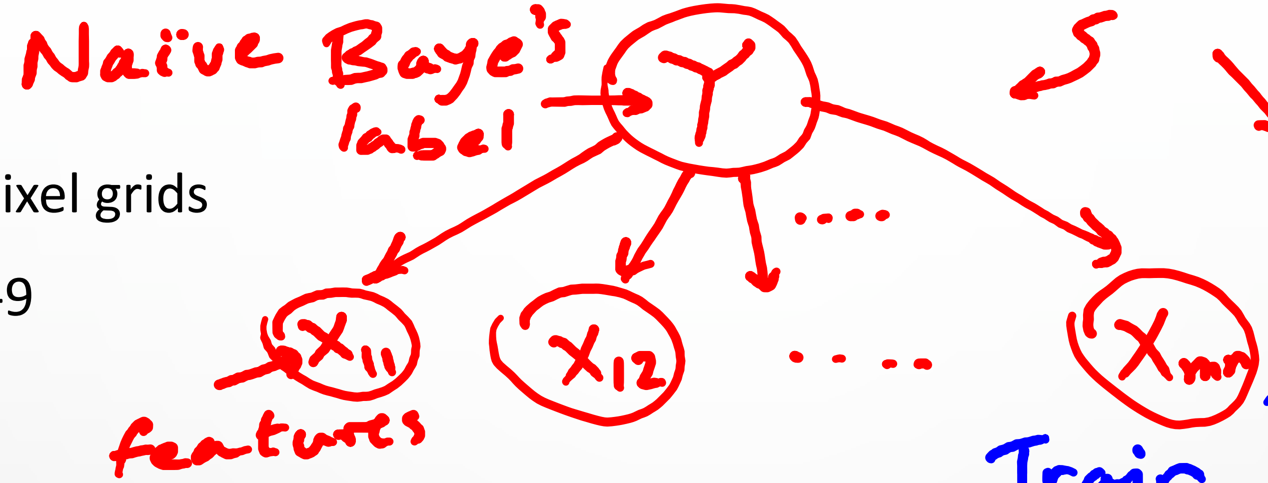
First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

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, Iknow 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



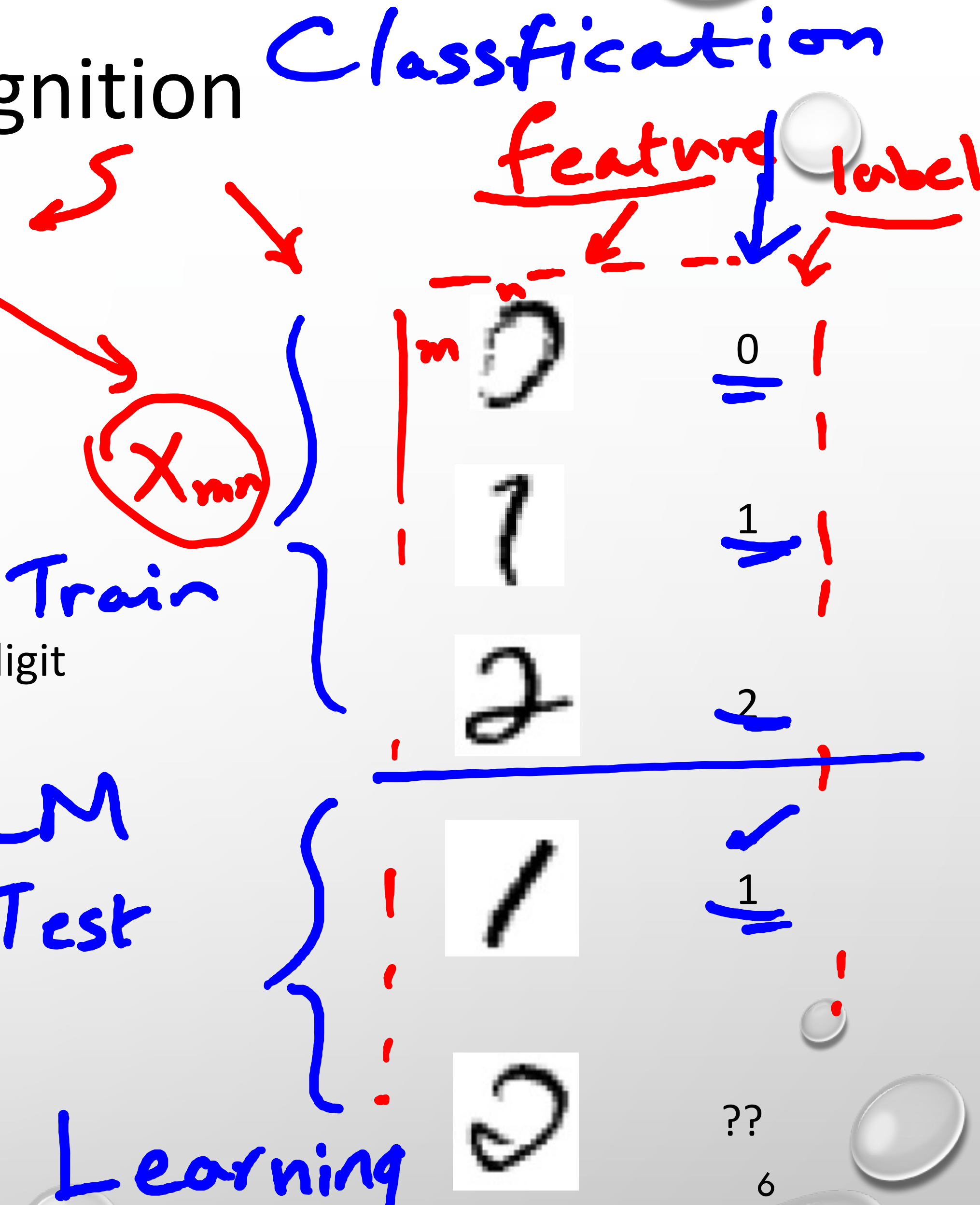
- Input: images / pixel grids
- Output: a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images

Clustering / Topic Modeling / LLM

- Features: the attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape patterns: NumComponents, AspectRatio, NumLoops
  - …

Unsupervised L. vs. Supervised Learning

Reinforcement L.



Other Classification Tasks

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

- Examples:
  - Spam detection (input: document, Classes: spam / ham)
  - OCR (input: images, classes: characters)
  - Medical diagnosis (input: symptoms, Classes: diseases)
  - Automatic essay grading (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

feature label

$X_1 \dots X_d$ Y

Model: joint distribution

Baye's Net

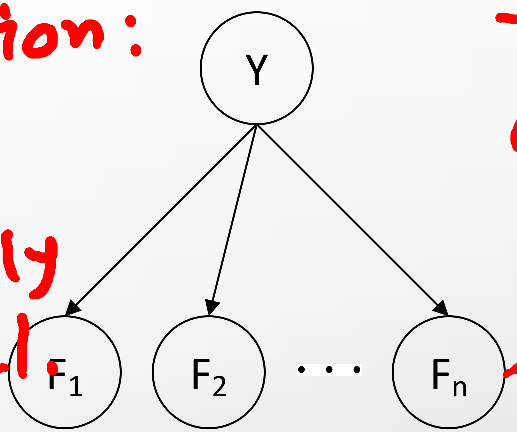
Model-Based Classification

- 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 digit recognition version:
  - One feature (variable) f_{ij} for each grid position <i,j>
  - 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.

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

  - Here: lots of features, each is binary valued

- Naïve Bayes model: $$P(Y|F_{0,0}\ldots F_{15,15}) \propto P(Y)\prod_{i,j} P(F_{i,j}|Y)$$

- What do we need to learn?

General Naïve Bayes

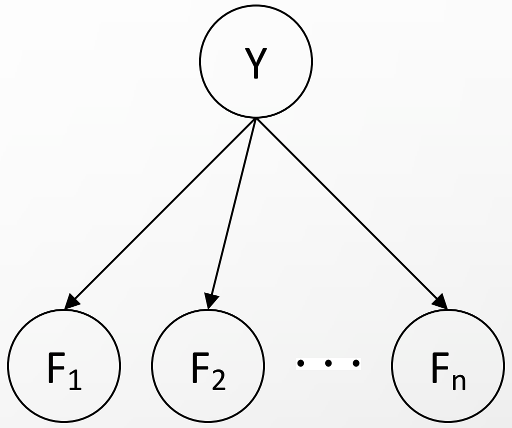
- A general naive Bayes model:

|Y| parameters

$$P(\mathsf{Y}, \mathsf{F}_1 \ldots \mathsf{F}_n) = \quad P(\mathsf{Y}) \prod_i P(\mathsf{F}_i|\mathsf{Y})$$

|Y| x $|F|^n$ values

n x |F| x |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

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

$$+ \quad \overline{\qquad\qquad\qquad} \quad P(f_1 \dots f_n)$$

$$\Downarrow$$

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

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

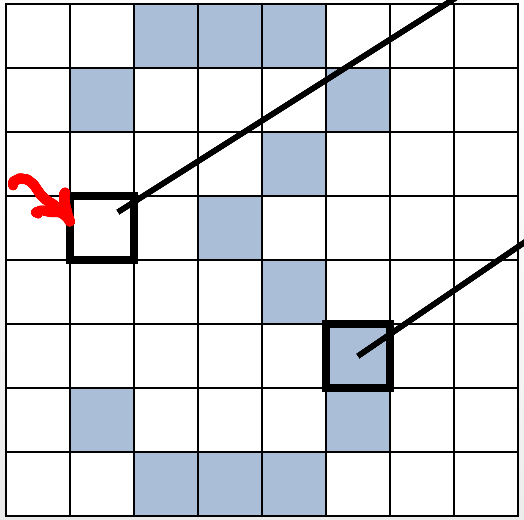
General Naïve Bayes

- 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 $P(Y|F_1 \ldots F_n)$
    - 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: $$P(Y, W_1 \ldots W_n) = P(Y) \prod_i P(W_i|Y)$$

- 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(spam | w) = 98.9

Training and Testing

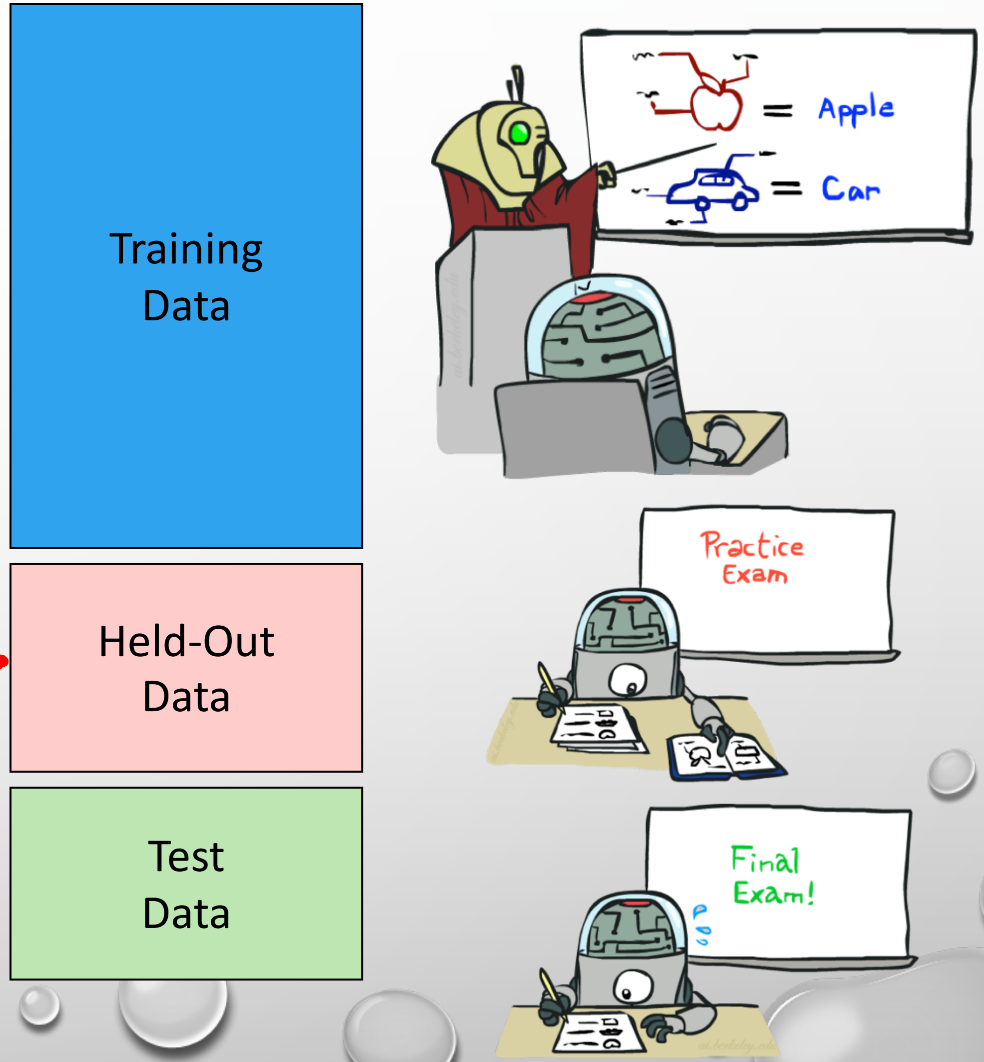
Hyperparameter vs. Parameter

Important Concepts

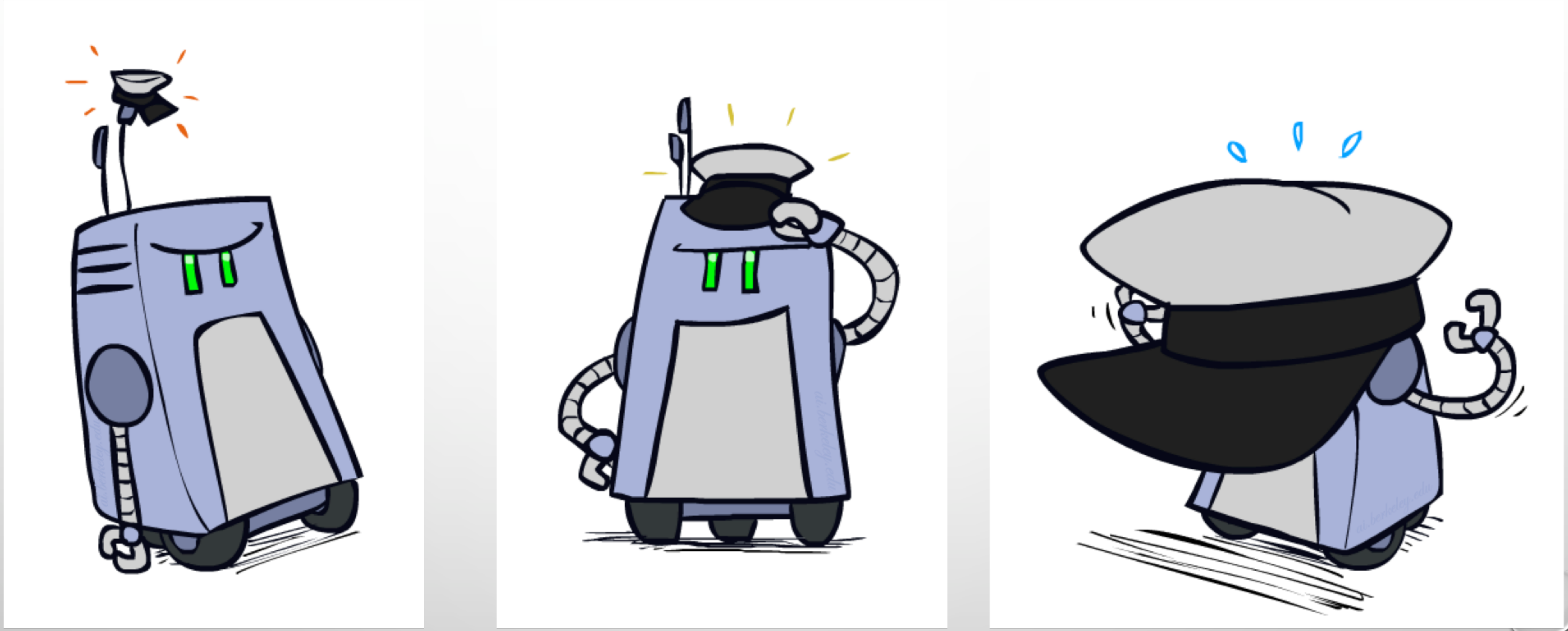
K → Laplace Smoothing

- 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

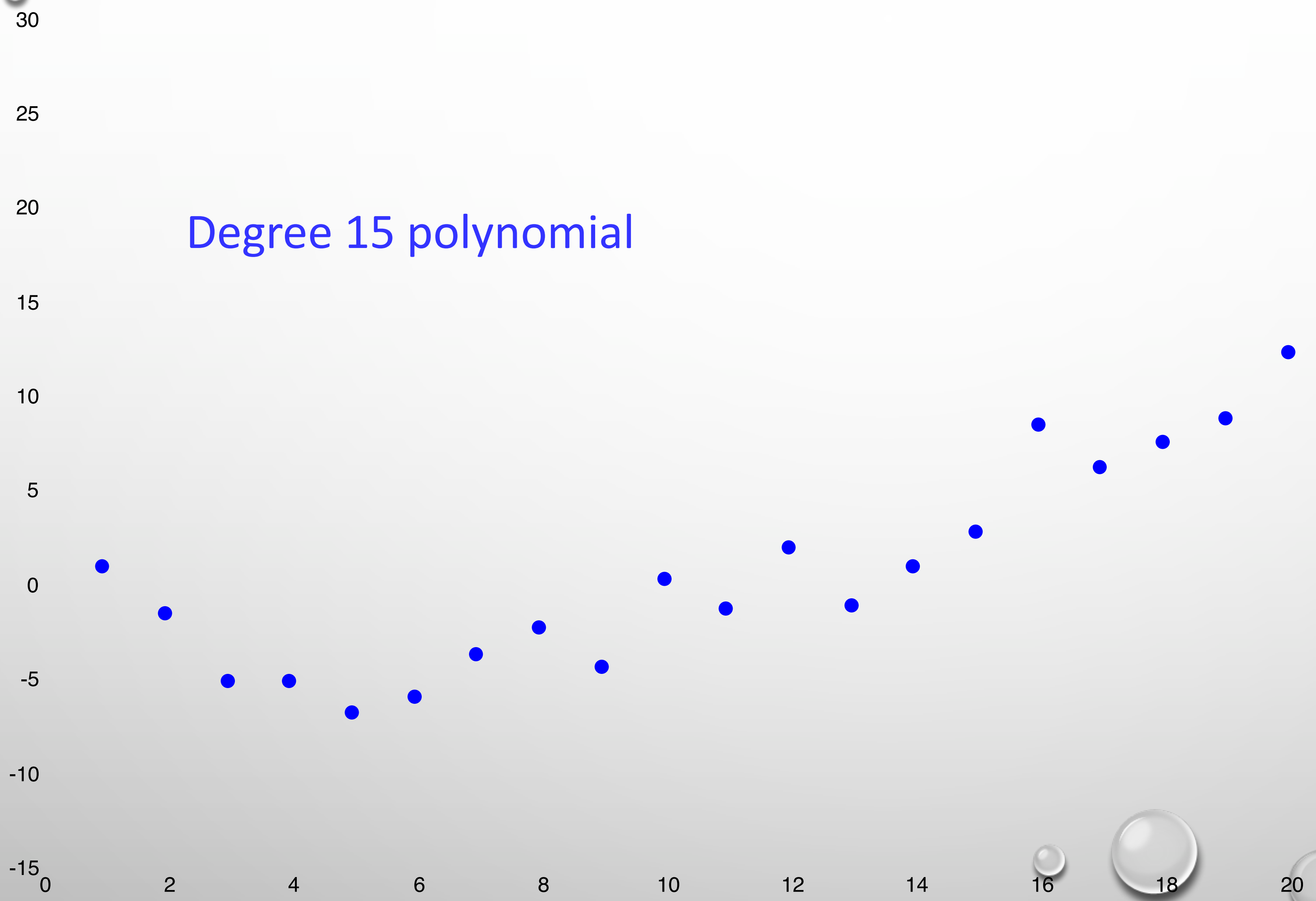
Validation Set

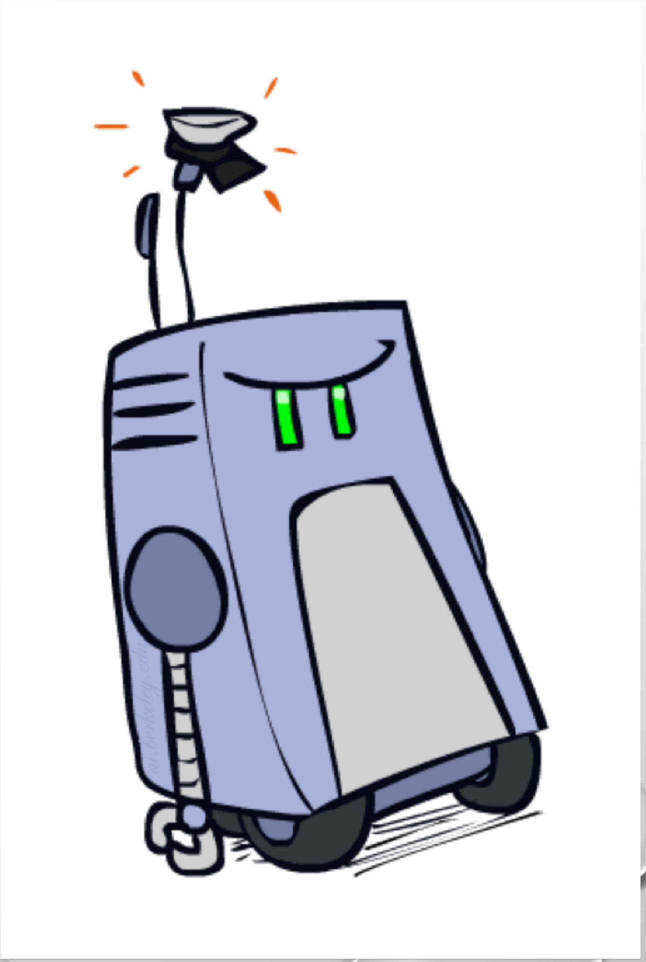


Generalization and Overfitting



Overfitting





Example: Overfitting

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

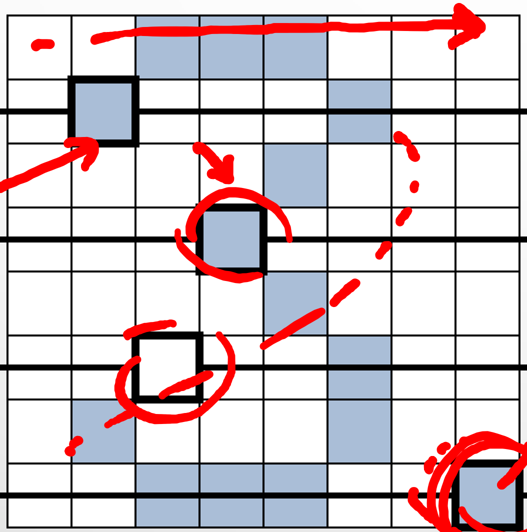
$P(C = 2) = 0.1$

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

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

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

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



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

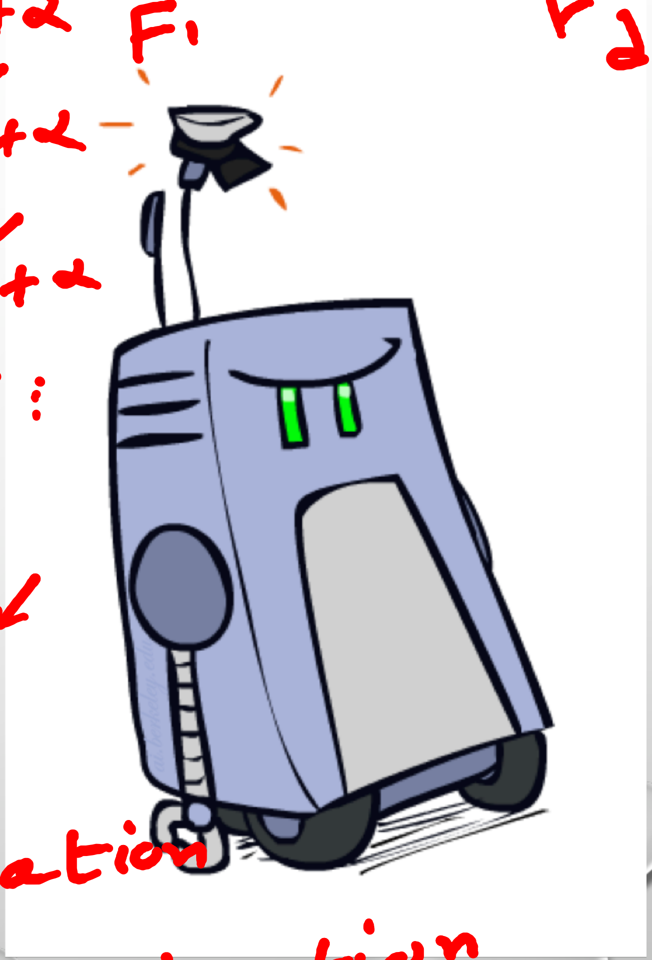
$P(C = 3) = 0.1$

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

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

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

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



2 wins!!

point estimation vs. interval estimation

Example: Overfitting

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

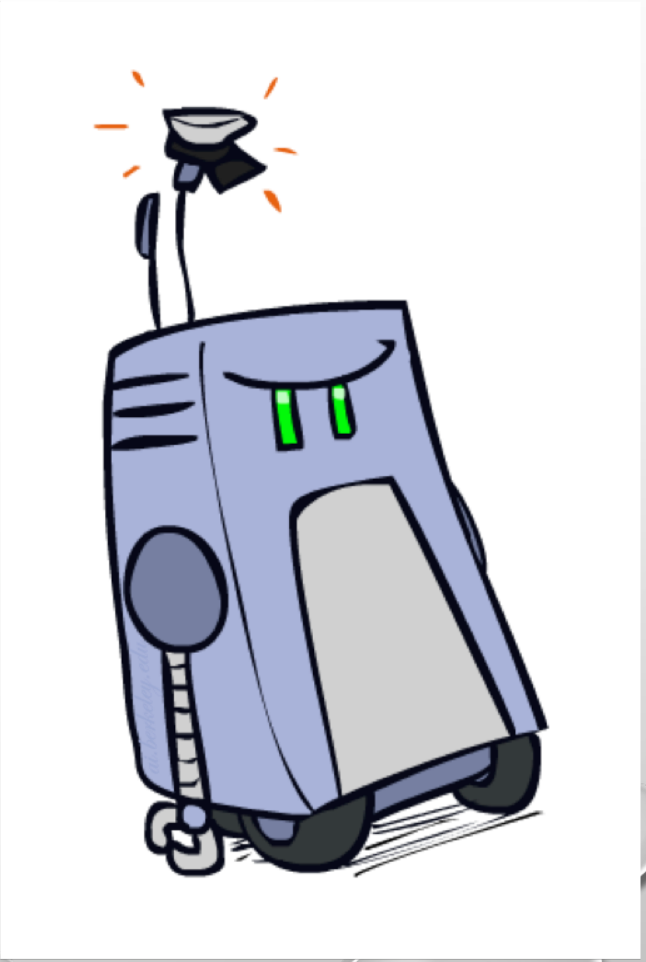
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

```
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?



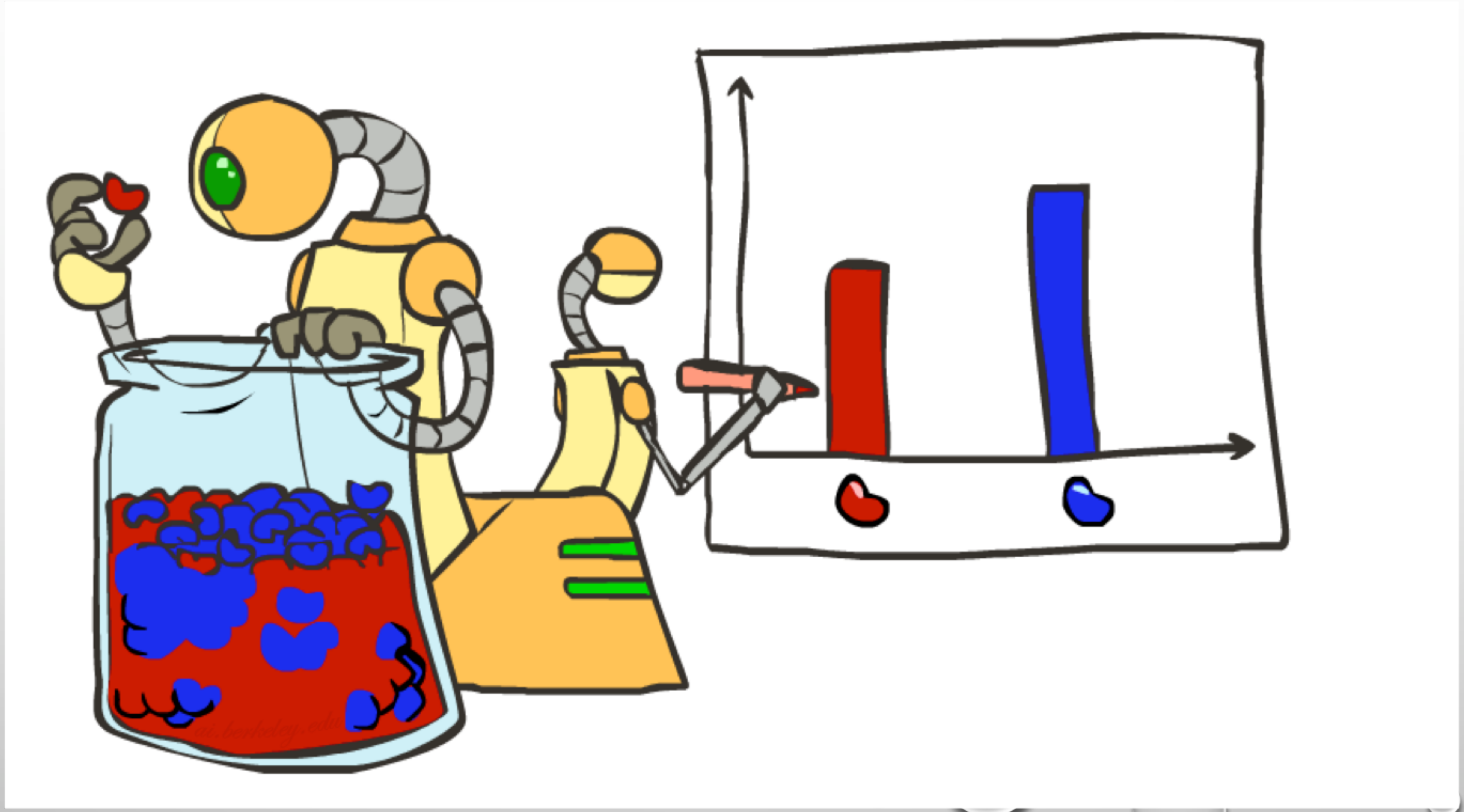
Generalization and Overfitting

- 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**

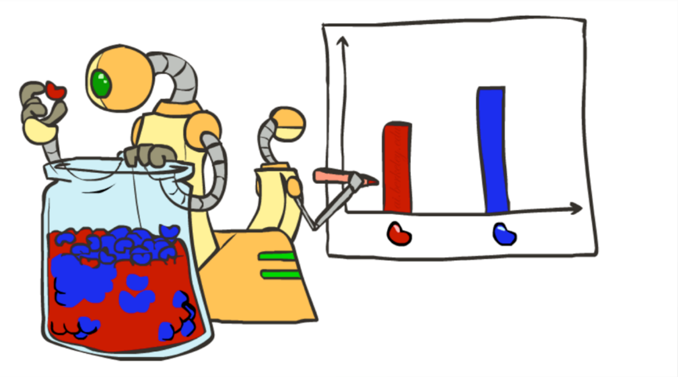
- As an extreme case, imagine using the entire email as the only feature
  - Would get the training data perfect (if deterministic labeling)
  - Wouldn't *generalize* at all
  - Just making the bag-of-words assumption gives us some generalization, but isn't enough

- To generalize better: we need to smooth or regularize the estimates

Parameter Estimation



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:

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

r r b

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

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

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

Maximum Likelihood

$$P_{\theta}(x) = \theta^{x}(1-\theta)^{1-x} \qquad x \in \{0,1\}$$

$$\rightarrow X_1 \dots X_n \overset{iid}{\sim} P_\theta(x)$$

Likelihood

$$\max_\theta \mathbb{P}(X_1 \dots X_n \mid \theta) = \prod_{i=1}^{n} P(X_i \mid \theta)$$

$$\max_\theta \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i}$$

$$\equiv \max_\theta \sum x_i \log\theta + (1-x_i)\log(1-\theta)$$

$$\sum \frac{x_i}{\theta} + (1-x_i)\left(-\frac{1}{1-\theta}\right) = 0 \rightarrow \theta = \frac{1}{n}\sum_{i=1}^{n} x_i$$

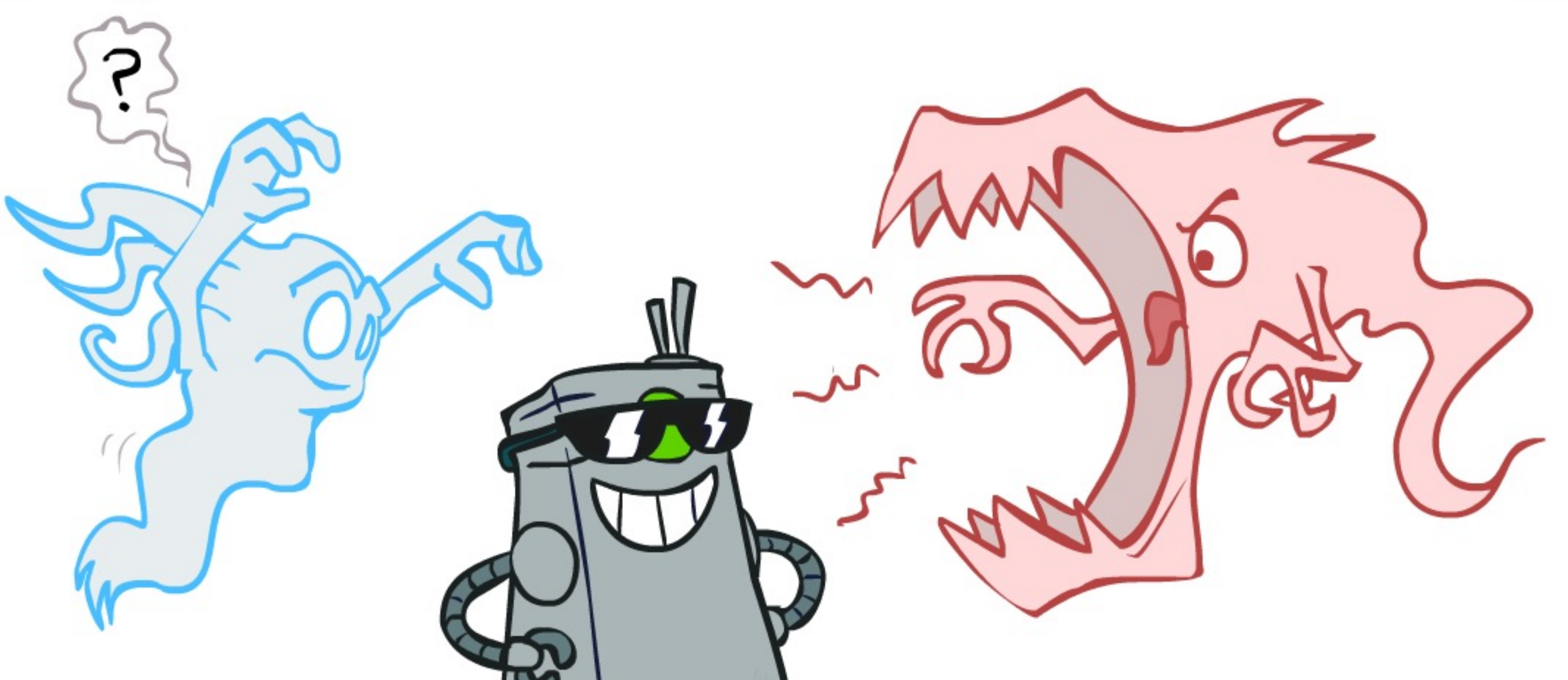
$$\theta_1^{\alpha_1}\theta_2^{\alpha_2}\cdots\theta_K^{\alpha_K}$$

Dirichlet

$\rightarrow \mathbb{P}(\theta) \rightsquigarrow$ prior prob.

$$P(\theta \mid X_1 \dots X_n) = \frac{P(X_1 \dots X_n \mid \theta)\,\mathbb{P}(\theta)}{\mathbb{P}(X_1 \dots X_n)}$$

Smoothing



Maximum Likelihood?

- Relative frequencies are the maximum likelihood estimates

$$\theta_{ML} = \arg\max_{\theta} P(\mathbf{X}|\theta)$$

$$= \arg\max_{\theta} \prod_i P_\theta(X_i)$$

→

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

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

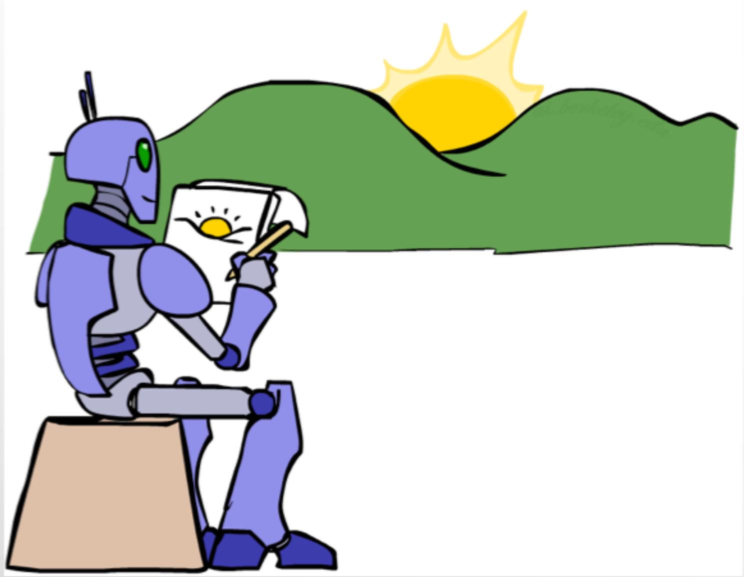
$$\theta_{MAP} = \arg\max_{\theta} P(\theta|\mathbf{X})$$

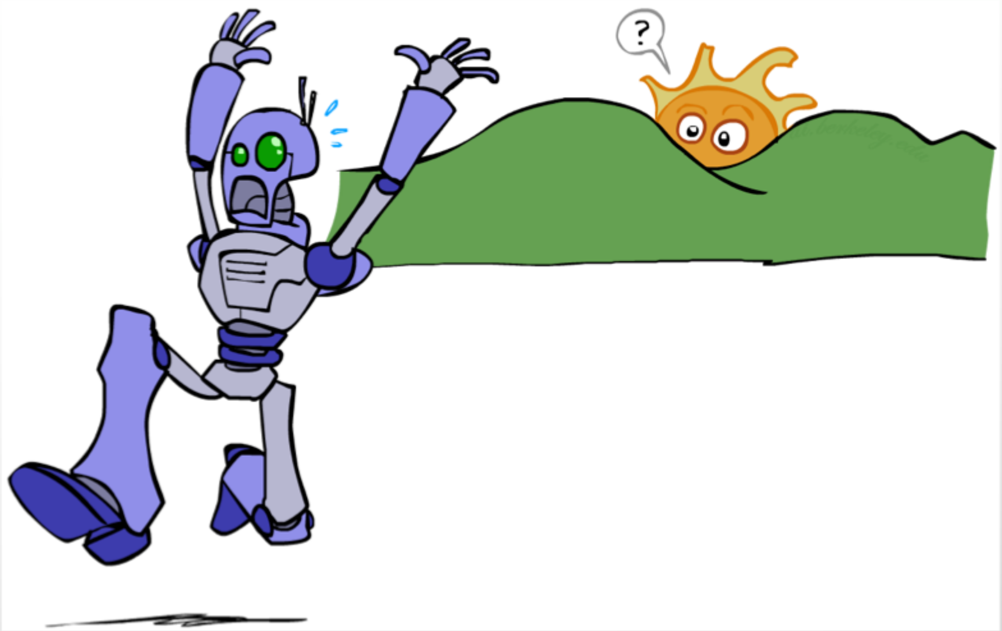
$$= \arg\max_{\theta} P(\mathbf{X}|\theta)P(\theta)/P(\mathbf{X})$$

→ ????

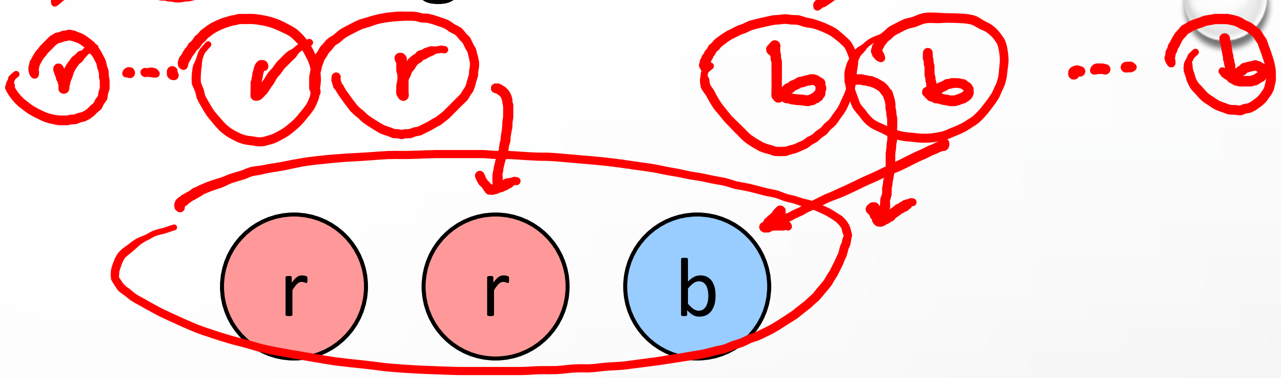
$$= \arg\max_{\theta} P(\mathbf{X}|\theta)P(\theta)$$

Unseen Events





Laplace Smoothing



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

$$P_{LAP}(x) = \frac{c(x)+1}{\sum_x [c(x)+1]}$$

$$= \frac{c(x)+1}{N+|X|}$$

$$P_{ML}(X) =$$

$$P_{LAP}(X) =$$

- Can derive this estimate with *Dirichlet priors (See Probabilistic Graphical Models course)*

Laplace Smoothing

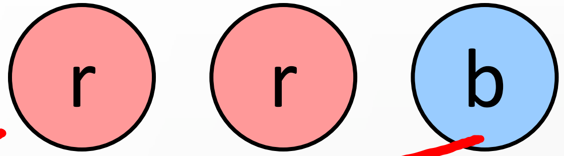
- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

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

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

- Laplace for conditionals:
  - Smooth each condition independently:

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



Data

$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

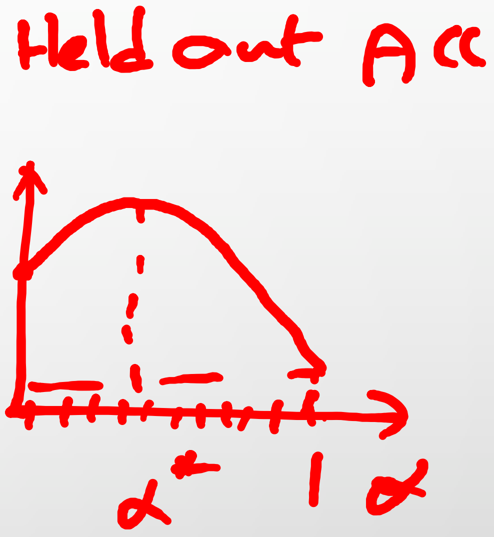
prior knowledge

Estimation: Linear Interpolation

- In practice, Laplace often performs poorly for P(X|Y):
  - When |X| is very large
  - When |Y| is very large

- Another option: linear interpolation
  - Also get the empirical P(X) from the data
  - Make sure the estimate of P(X|Y) isn't too different from the empirical P(X)

$$P_{LIN}(x|y) = \alpha \hat{P}(x|y) + (1.0 - \alpha)\hat{P}(x)$$



  - What if α is 0? 1?
  - See *Stochastic Processes* course for more interesting options of making the estimation.

Real NB: Smoothing

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

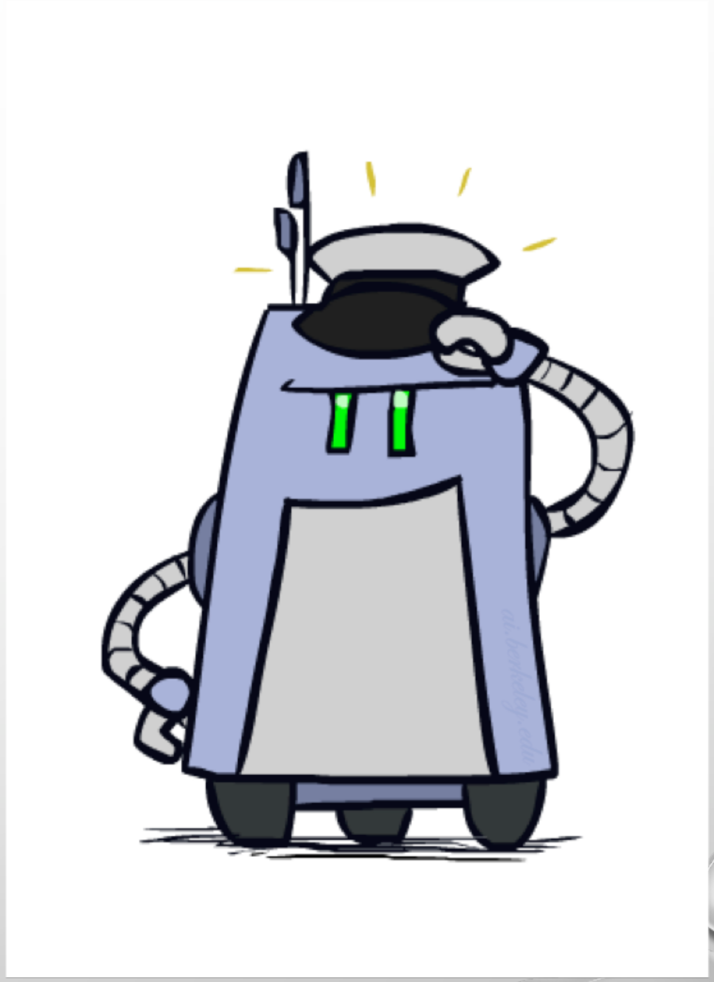
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

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

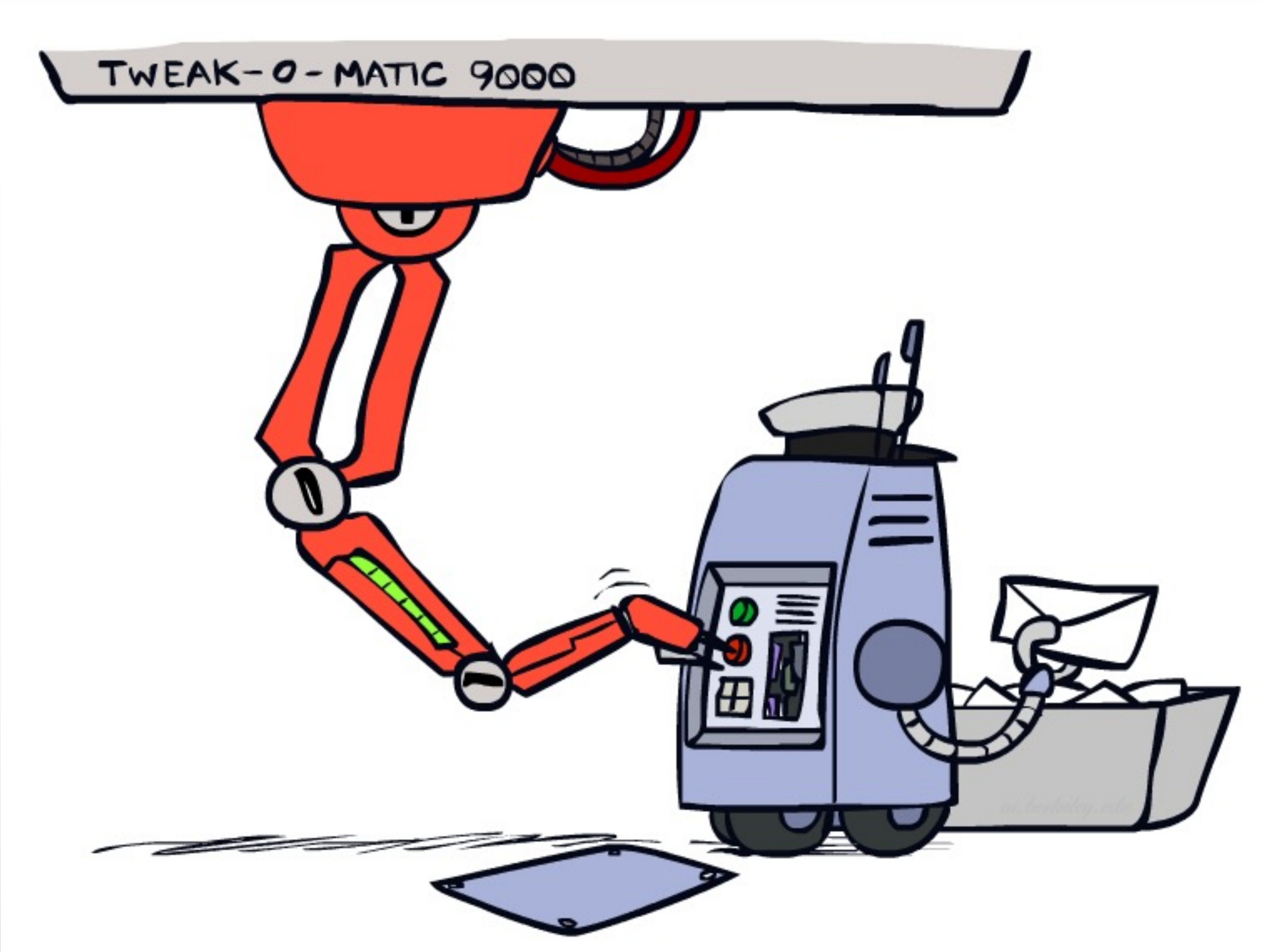
$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

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

Do these make more sense?



Tuning



Tuning on Held-Out Data

- Now we've got two kinds of unknowns
  - Parameters: the probabilities P(X|Y), P(Y)
  - Hyperparameters: e.g. The amount / type of smoothing to do, k, α

- What should we learn where?
  - 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