

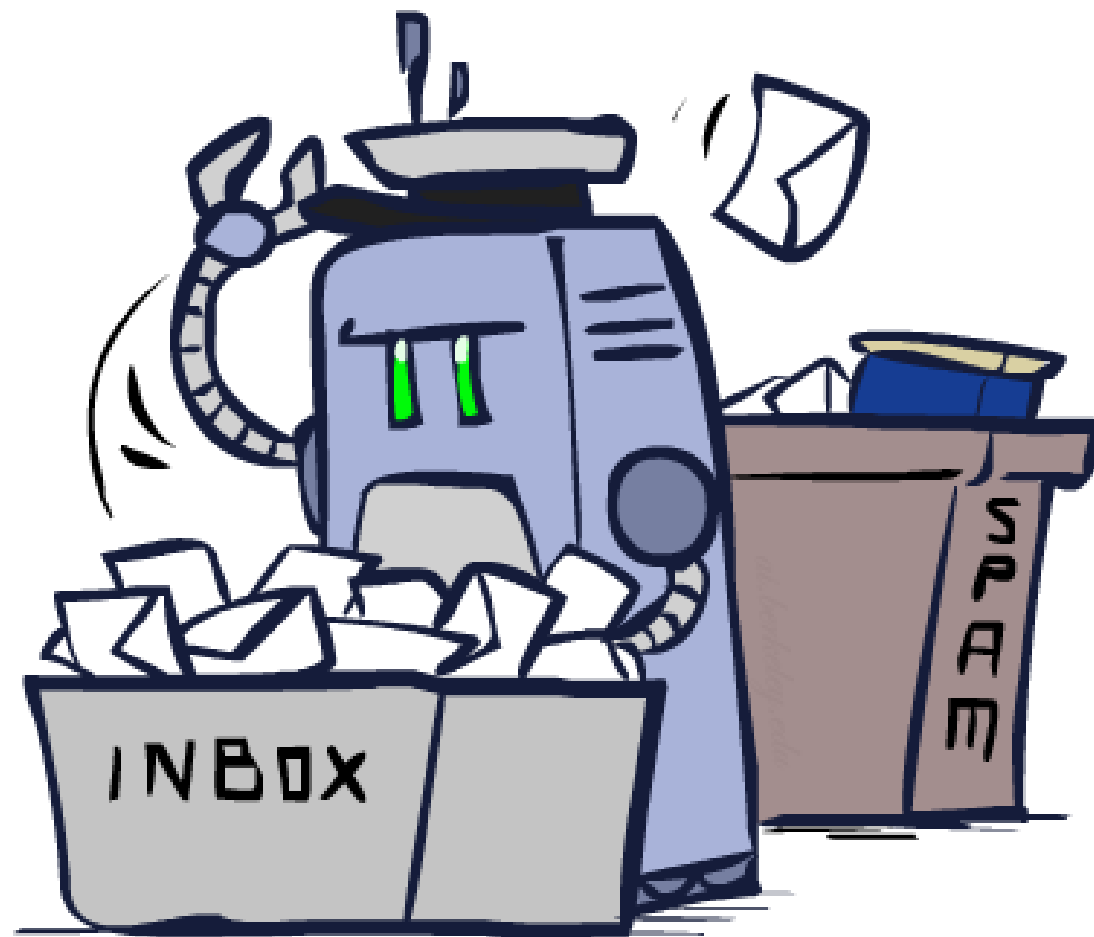
Naïve Bayes (朴素贝叶斯模型)



Machine Learning 机器学习

- 已经学习的: 如何利用模型求解最优决策
- Machine learning: 如何从数据或经验中获取一个模型
 - 学习模型参数 (e.g. probabilities)
 - 学习结构 (e.g. BN graphs)
 - 学习隐式的模式或概念 (e.g. clustering)
- 今天内容: 用于分类问题的朴素贝叶斯模型

Classification 分类任务



举例: Spam Filter 垃圾邮件过滤

- Input: an email
- Output: spam/ham
- 训练准备:
 - 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
- 提取特征: 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.

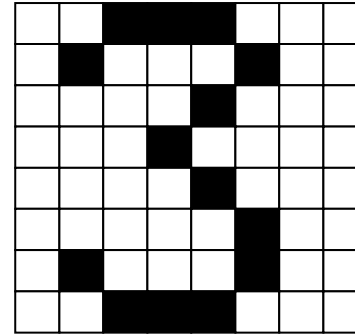
99 MILLION EMAIL ADDRESSES FOR ONLY \$99



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.

Digit Recognition 手写数字识别

- Input: images / pixel grids
- Output: a digit 0-9

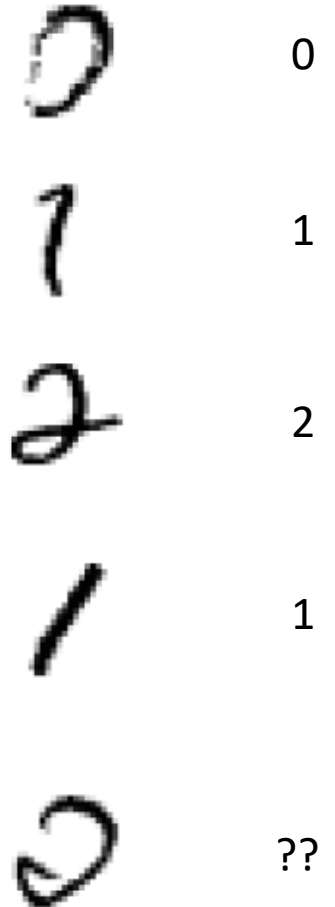


- 训练准备:

- 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

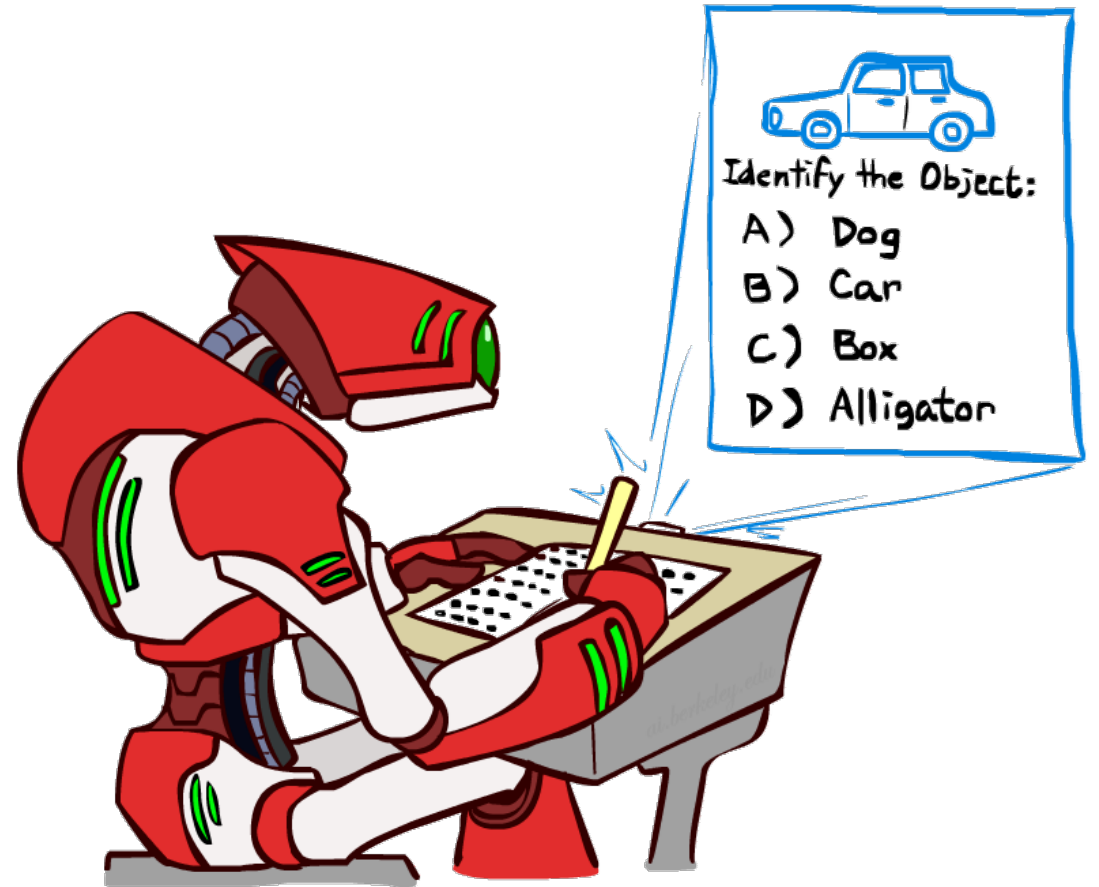
- 提取特征: The attributes used to make the digit decision

- Pixels: (6,8)=ON
- Shape Patterns: NumComponents, AspectRatio, NumLoops
- ...

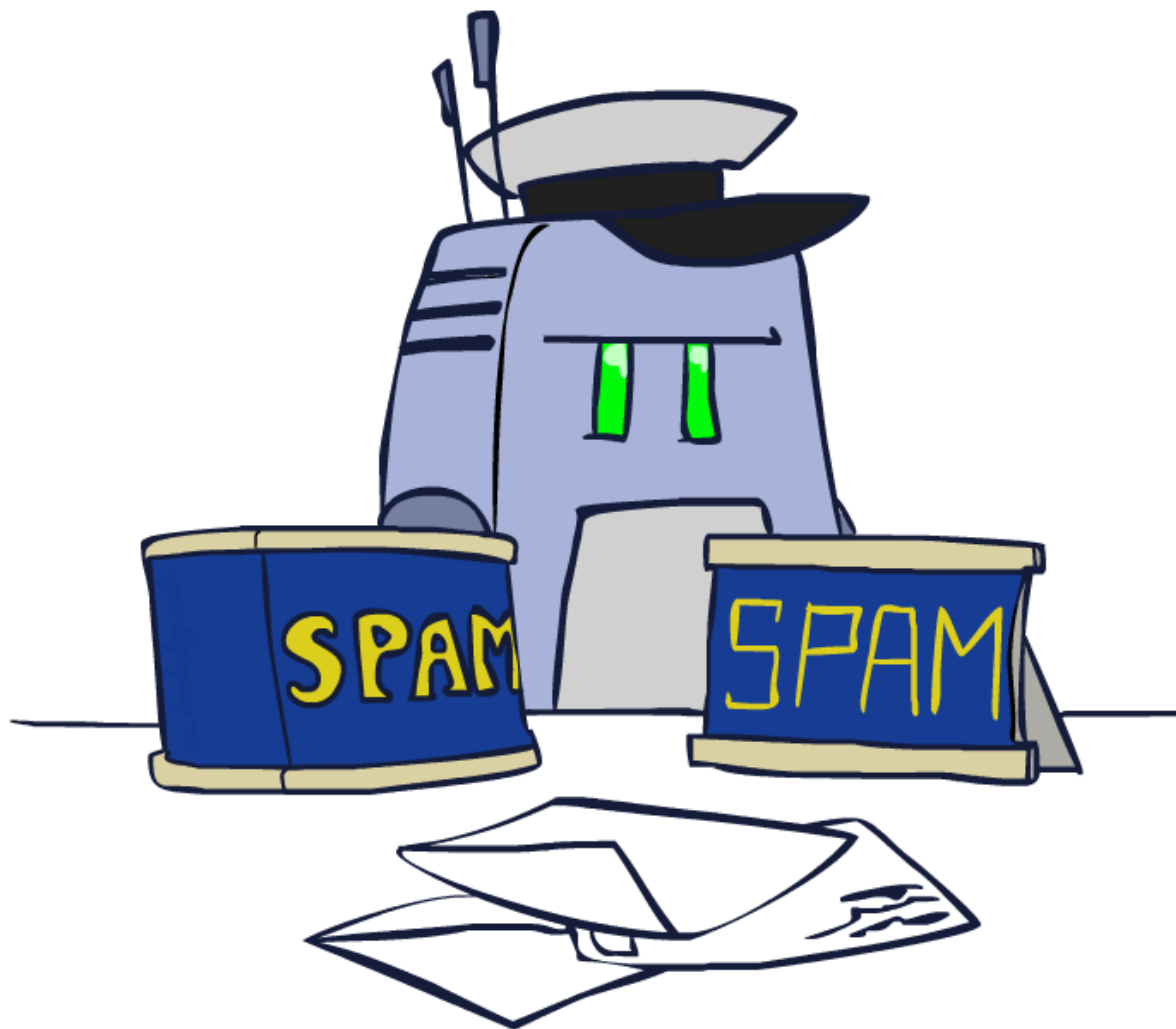


应用中的分类任务

- 分类任务: 给定输入 x , 预测标签 (类别) 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!

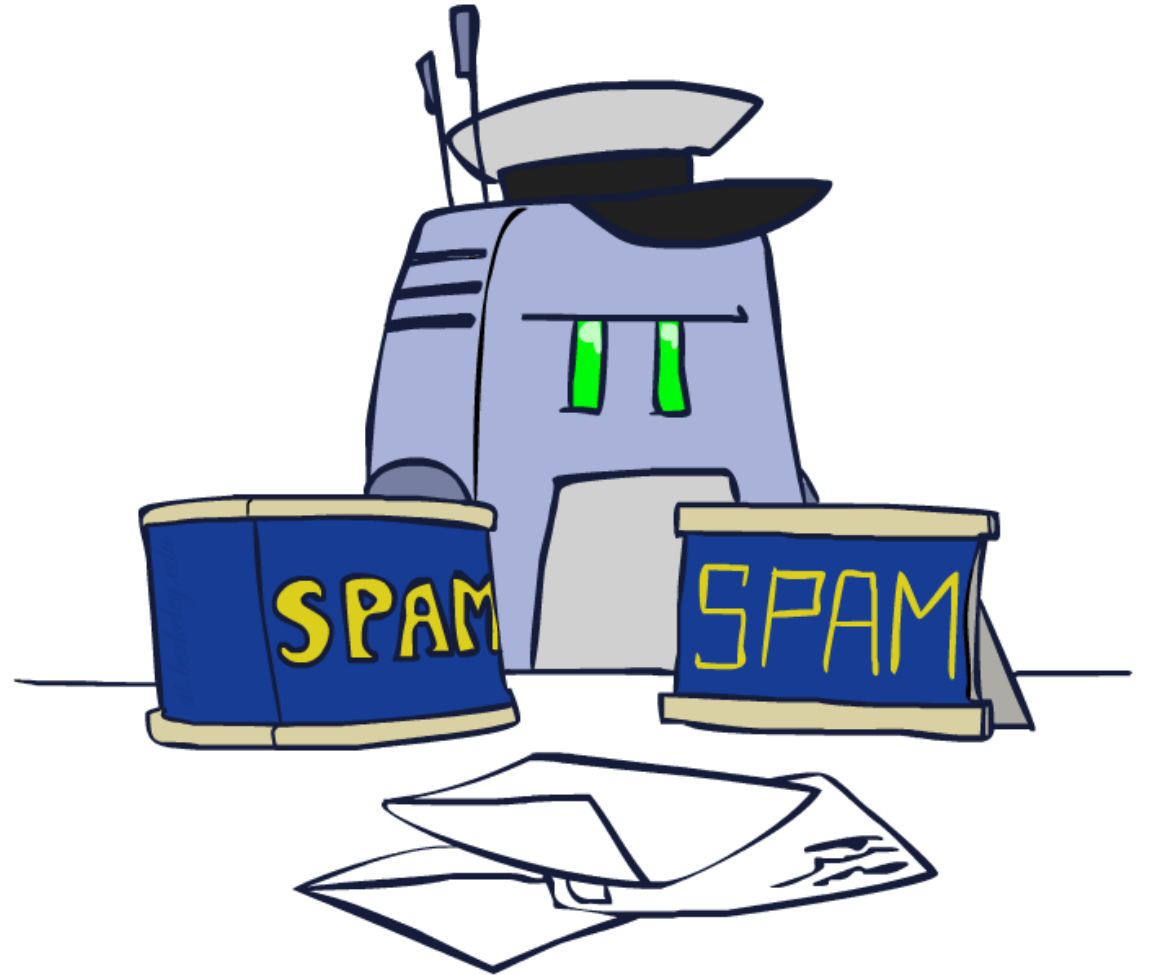


Model-Based Classification 基于模型的分类



基于模型的分类

- 基于模型的方法：
 - 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
- 问题是：
 - What structure should the BN have?
 - How should we learn its parameters?




朴素贝叶斯模型应用于手写数字分类

- Naïve Bayes: Assume all features are independent effects of the label

- Simple digit recognition version:

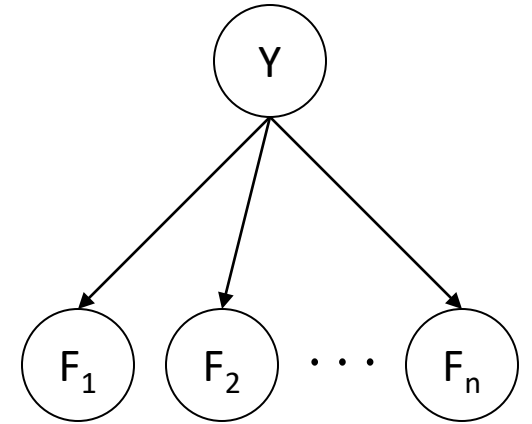
- One feature (variable) F_{ij} for each grid position $\langle i,j \rangle$
- Feature values are on / off, based on whether intensity is more or less than 0.5 in underlying image
- Each input maps to a feature vector, e.g.

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

- 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?



朴素贝叶斯模型

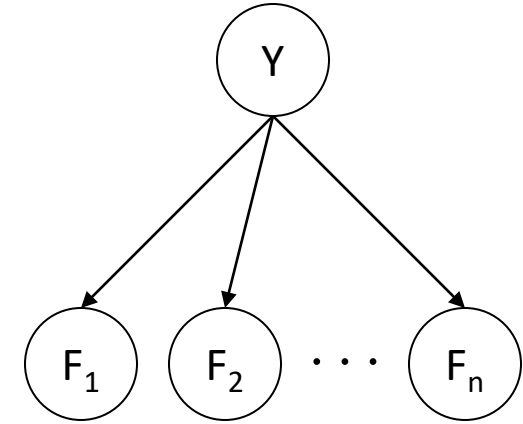
- A general Naive Bayes model:

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

|Y| parameters

|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

推理计算标签变量的后验分布

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

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

$$P(f_1 \dots f_n)$$

+ ↶

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

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

垃圾邮件过滤

- Naïve Bayes spam filter

- Data:

- Collection of emails, labeled spam or ham
- Note: someone has to hand label all this data!
- Split into training, held-out, test sets



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

- Classifiers

- Learn on the training set
- (Tune it on a held-out set)
- Test it on new emails



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.

朴素贝叶斯对文本数据的建模准备

■ 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

how many variables are there?
how many values?

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

■ Generative model: $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$

■ “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 distribution
 - Why make this assumption?
- Called “bag-of-words” because model is insensitive to word order or reordering

**in is lecture lecture next over person remember room
sitting the the the to to up wake when you**

举例: 垃圾邮件过滤

- 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 |
| ... | |

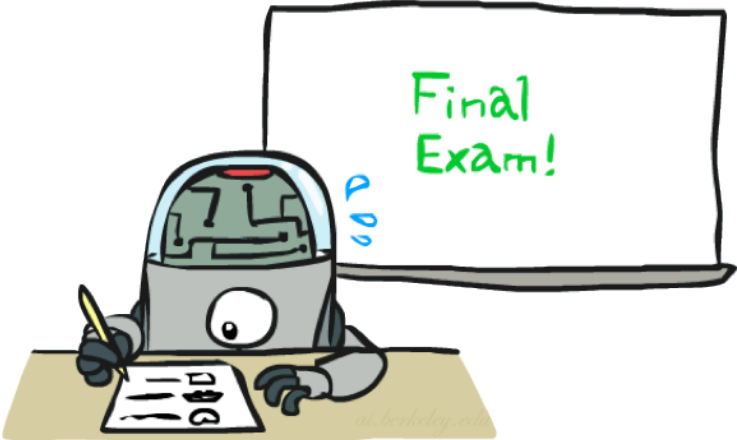
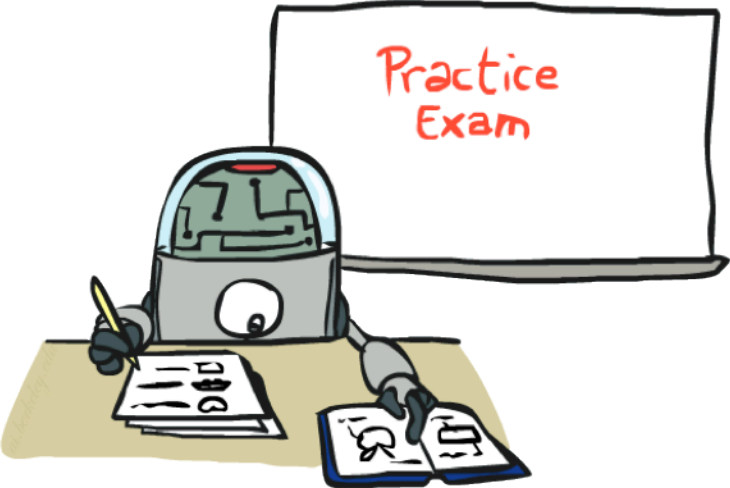
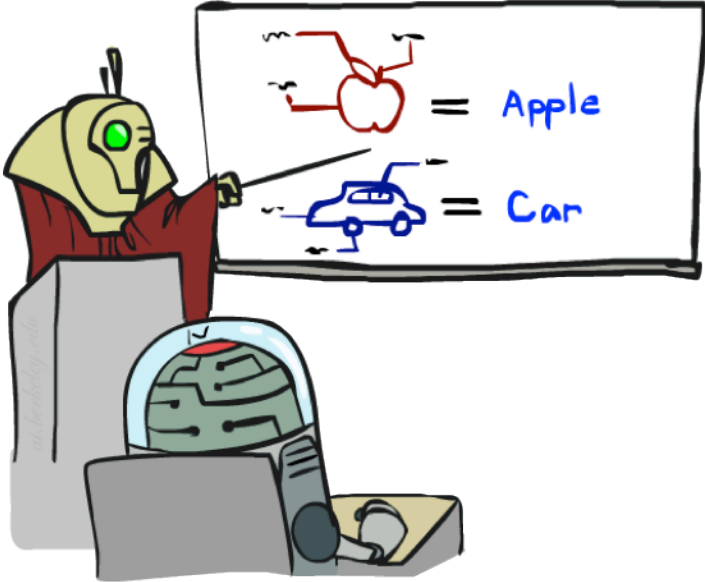
泛化的朴素贝叶斯模型

- 朴素贝叶斯的基本要求?
 - 推理判别方法 (比如上页幻灯片的例子)
 - 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
 - 局部条件概率表的估计
 - $P(Y)$, 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

模型参数估计(简单版)

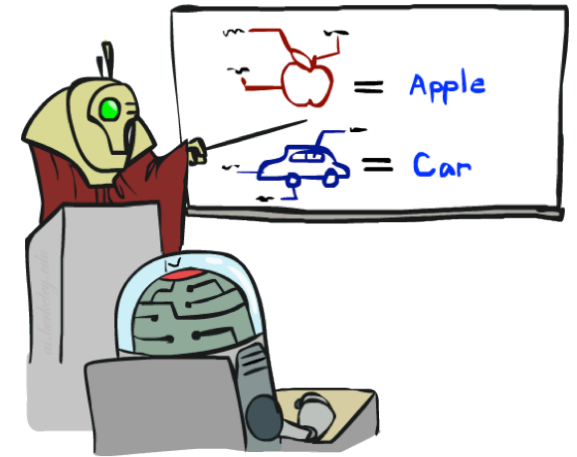
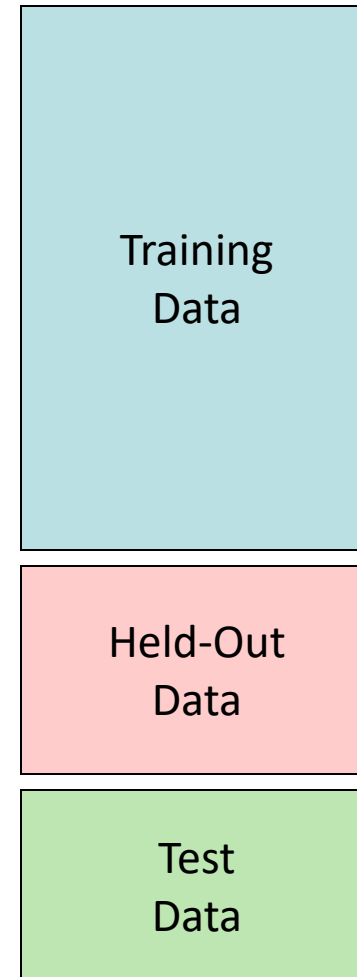
- 如何估计局部条件概率分布表?
 - Naïve version: just count!
 - $P(Y)$ – what ratio of the data is the digit 1?
 - $P(F_i|Y)$ – what ratio of the digit 1 has this pixel on?
- Need to be careful though ... let's see what goes wrong..

模型的训练与测试

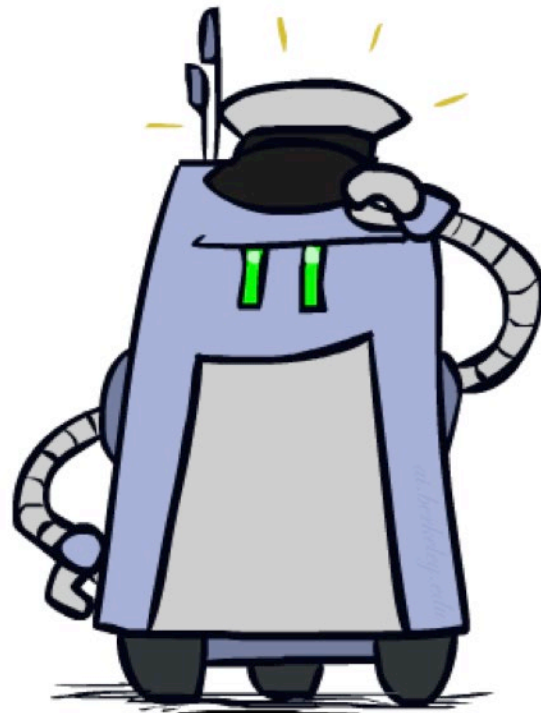
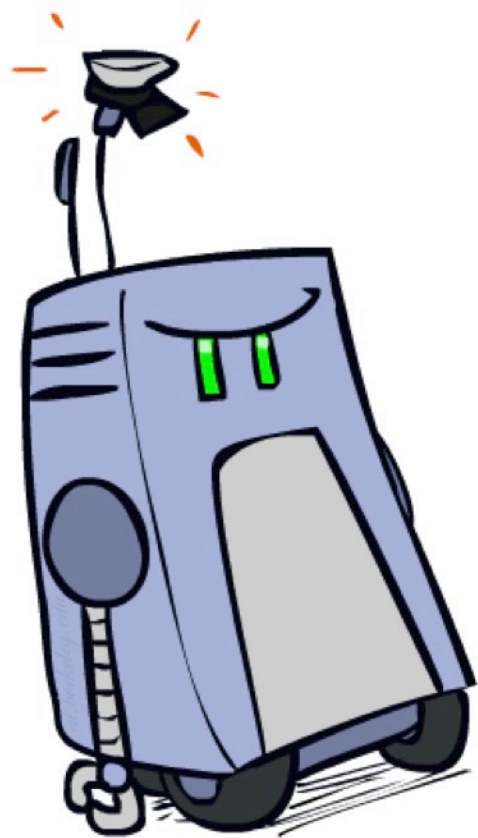


机器学习中的一一些重要概念

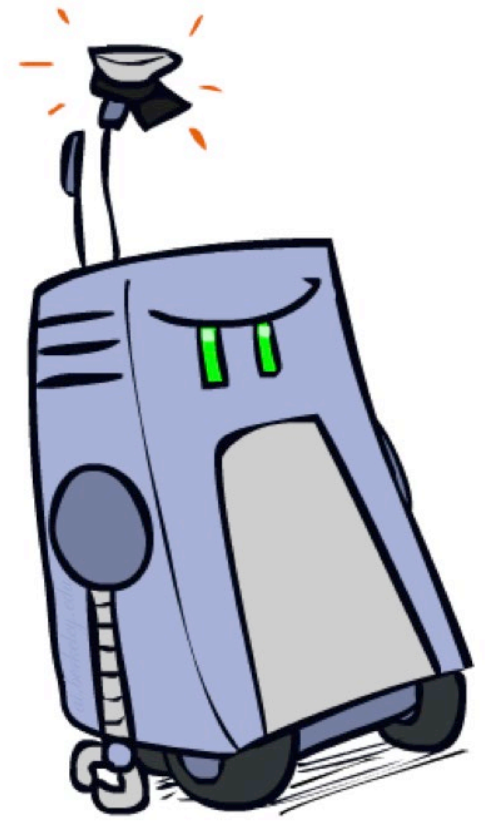
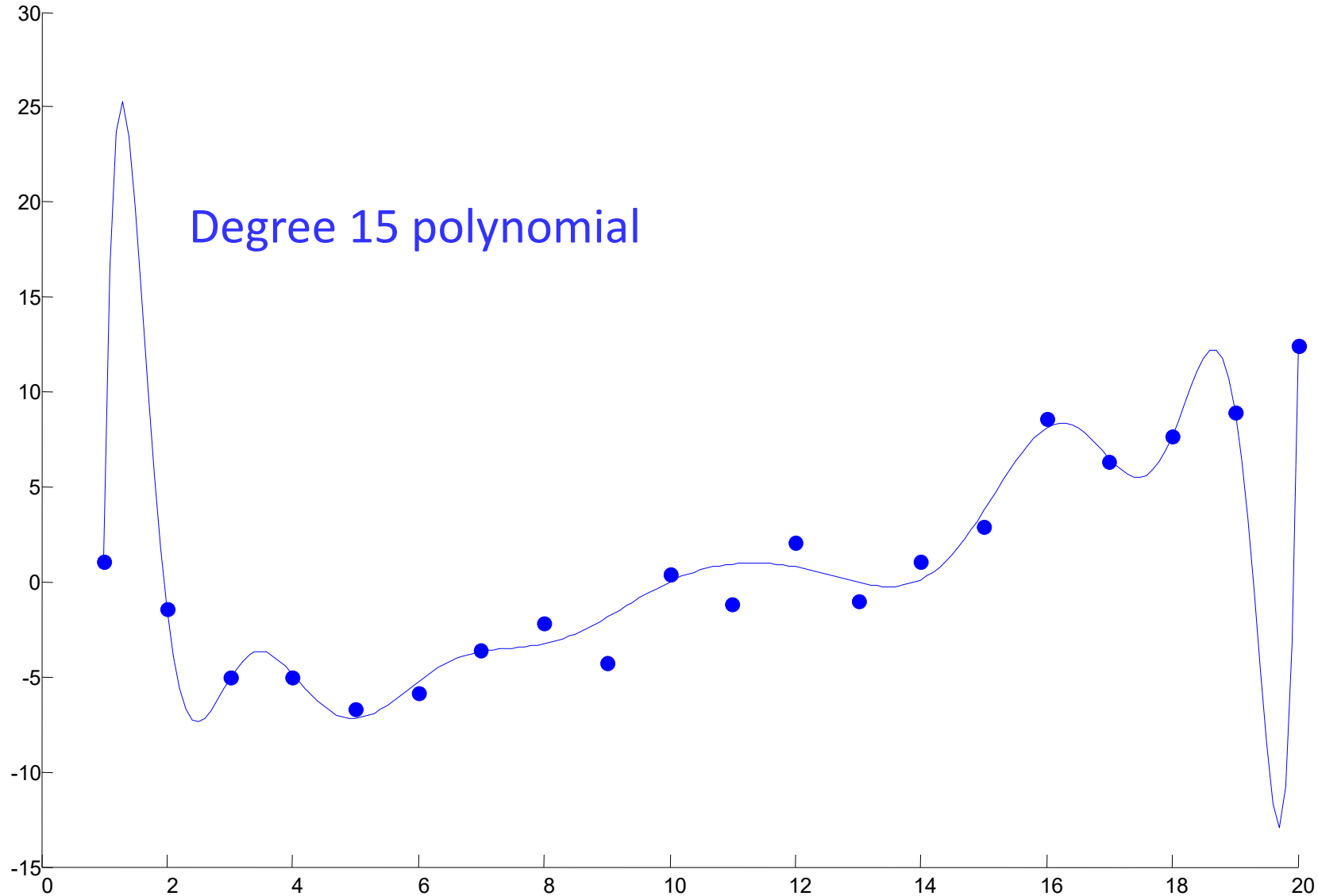
- Data: labeled instances, e.g. emails marked spam/ham
 - Training set
 - Held out set
 - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
 - Learn parameters (e.g. model probabilities) on training set
 - (Tune hyperparameters on held-out set)
 - Compute accuracy 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
 - Underfitting: fits the training set poorly



拟合不充分与过拟合



Overfitting 过拟合



过拟合举例

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

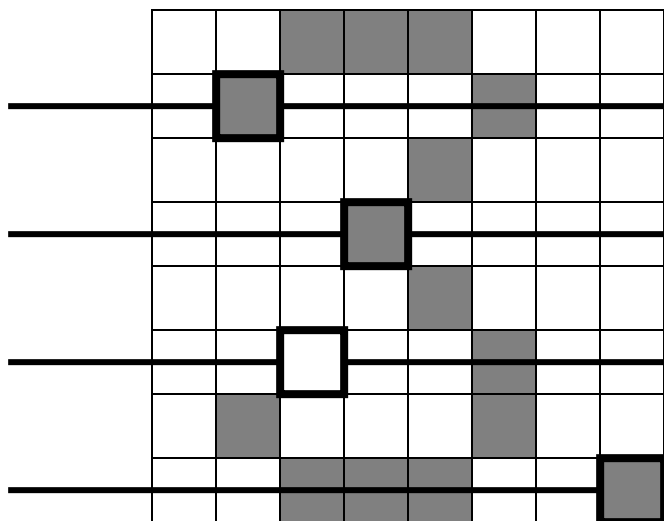
$P(C = 2) = 0.1$

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

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

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

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



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

$P(C = 3) = 0.1$

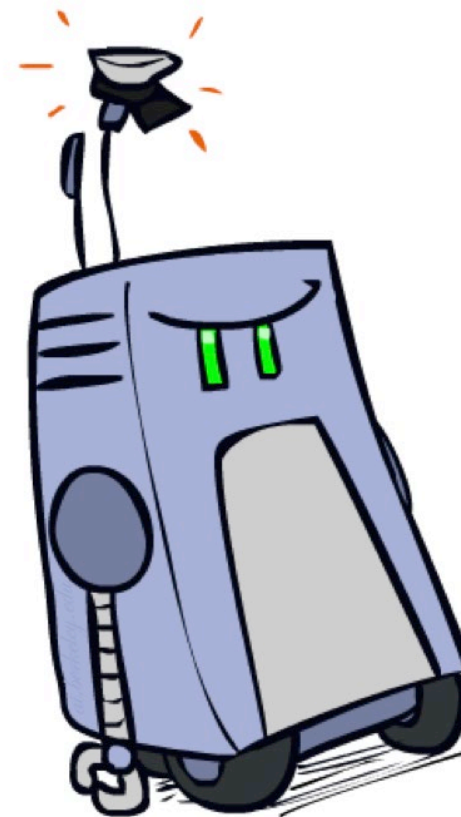
$P(\text{on}|C = 3) = 0.8$

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

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

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

2 wins!!



过拟合举例

- relative probabilities (odds ratios):

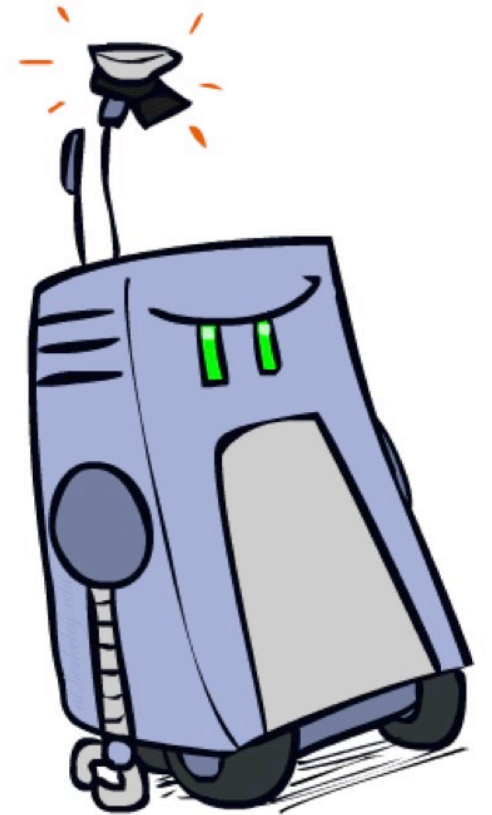
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

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

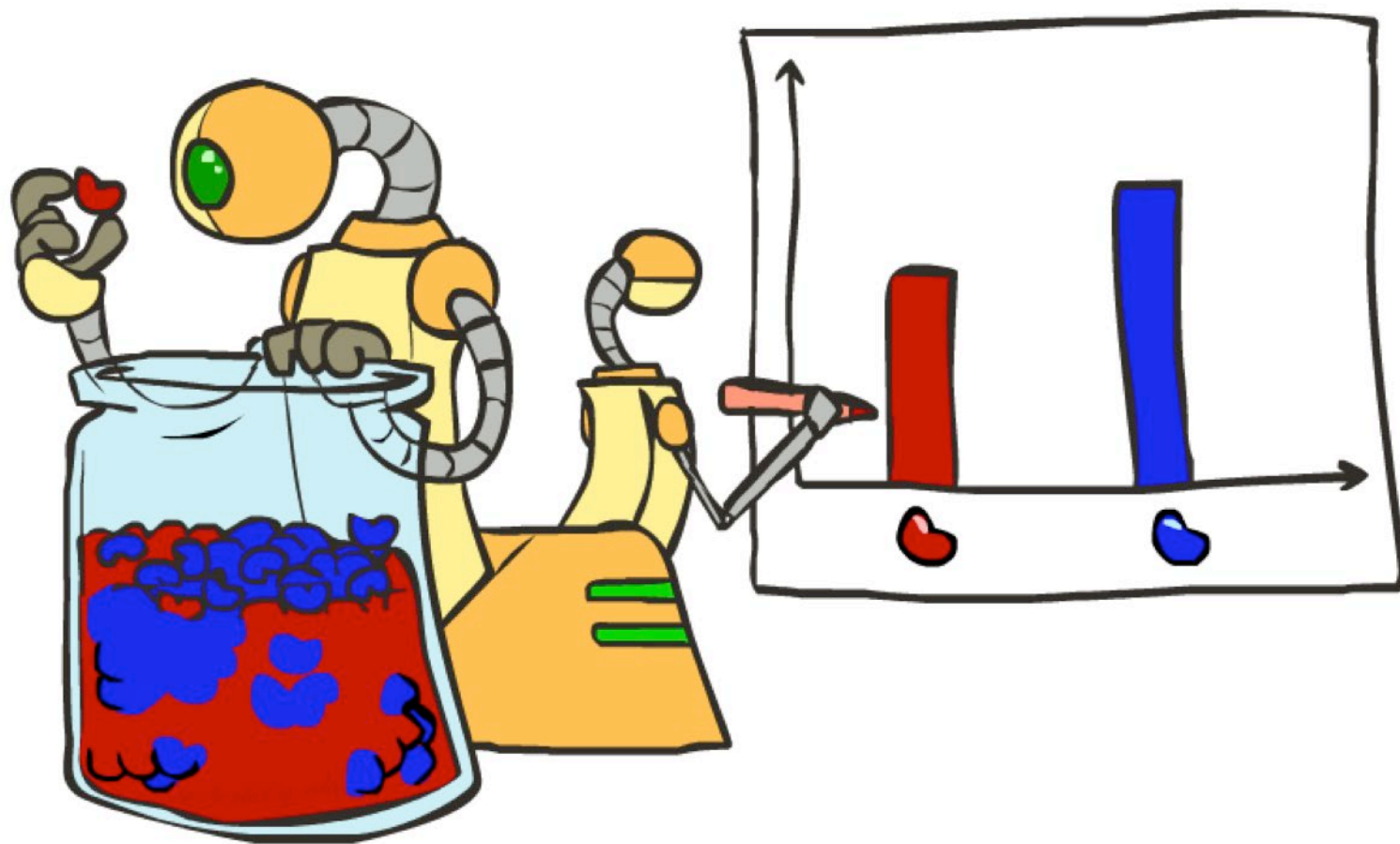
What went wrong here?



泛化性与过拟合

- 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 (防止过拟合)

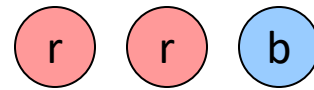
模型参数估算



模型参数估算

- 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:

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



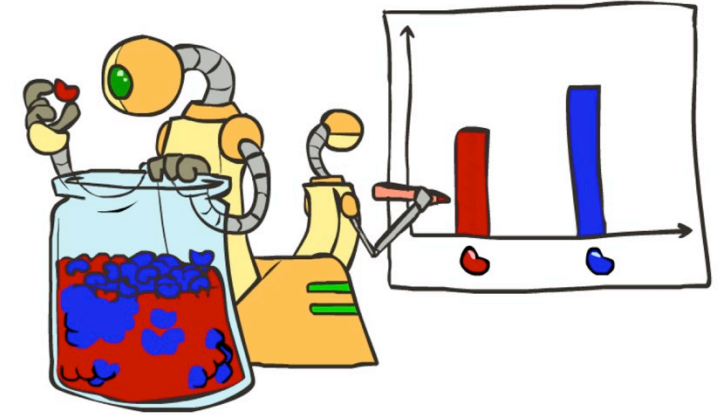
$$P_{\text{ML}}(r) = 2/3$$

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

$$L(x, \theta) = \prod_i P_{\theta}(x_i) = \theta \cdot \theta \cdot (1 - \theta)$$

$$P_{\theta}(x = \text{red}) = \theta$$

$$P_{\theta}(x = \text{blue}) = 1 - \theta$$



Your First Consulting Job

- A billionaire tech entrepreneur asks you a question:
 - **He says:** I have a thumbtack, if I flip it, what's the probability it will fall with the nail up?
 - **You say:** Please flip it a few times:



- **You say:** The probability is:
 - $P(H) = 3/5$
- **He says: Why???**
- **You say:** Because...

Your First Consulting Job

- $P(\text{Heads}) = \theta$, $P(\text{Tails}) = 1 - \theta$



- Flips are *i.i.d.*: $D = \{x_i | i=1 \dots n\}$, $P(D | \theta) = \prod_i P(x_i | \theta)$
 - Independent events
 - Identically distributed according to unknown distribution
- Sequence D of α_H Heads and α_T Tails

$$P(D | \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}$$

最大似然法估计

- **Data:** Observed set D of α_H Heads and α_T Tails
- **Hypothesis space:** Binomial distributions
- **Learning:** finding θ is an optimization problem

- What's the objective function?

$$P(\mathcal{D} | \theta) = \theta^{\alpha_H} (1 - \theta)^{\alpha_T}$$

- **MLE:** Choose θ to maximize probability of D

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} P(\mathcal{D} | \theta) \\ &= \arg \max_{\theta} \ln P(\mathcal{D} | \theta)\end{aligned}$$

最大似然法估计

$$\begin{aligned}\hat{\theta} &= \arg \max_{\theta} \ln P(\mathcal{D} | \theta) \\ &= \arg \max_{\theta} \ln \theta^{\alpha_H} (1 - \theta)^{\alpha_T}\end{aligned}$$

- Set derivative to zero, and solve!

$$\frac{d}{d\theta} \ln P(\mathcal{D} | \theta) = \frac{d}{d\theta} [\ln \theta^{\alpha_H} (1 - \theta)^{\alpha_T}]$$

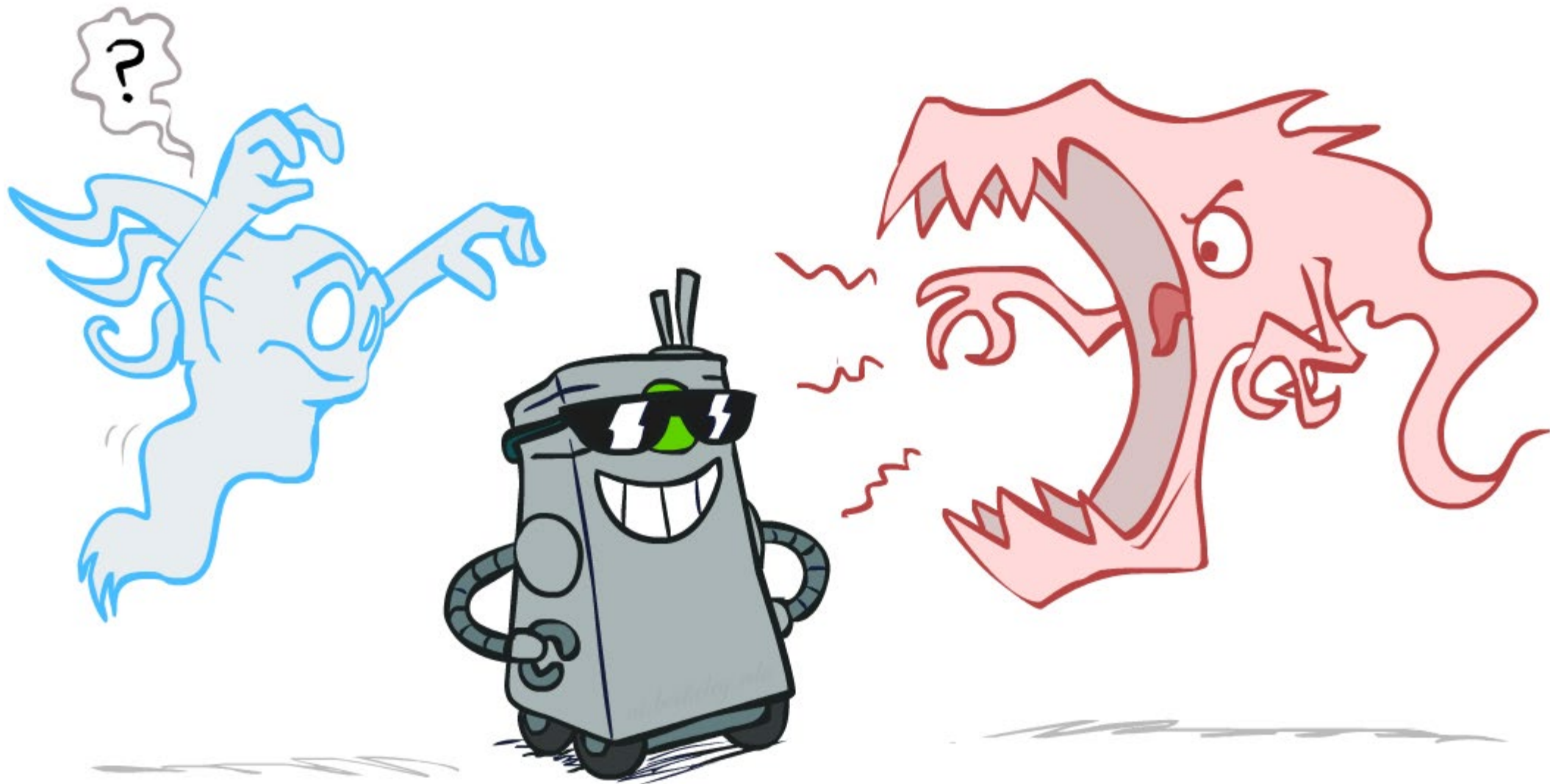
$$= \frac{d}{d\theta} [\alpha_H \ln \theta + \alpha_T \ln(1 - \theta)]$$

$$= \alpha_H \frac{d}{d\theta} \ln \theta + \alpha_T \frac{d}{d\theta} \ln(1 - \theta)$$

$$= \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1 - \theta} = 0$$

$$\hat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}$$

平滑化估值



最大似然法?

- Relative frequencies are the maximum likelihood estimates

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

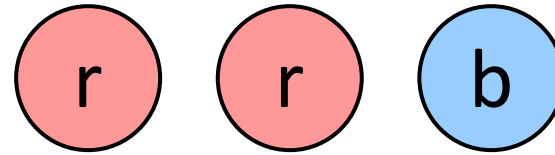
- Another option is to consider the most likely parameter value given the data

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

Laplace Smoothing 平滑化处理

- 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*

平滑化处理

- Laplace's estimate (extended):

- Pretend you saw every outcome k extra times

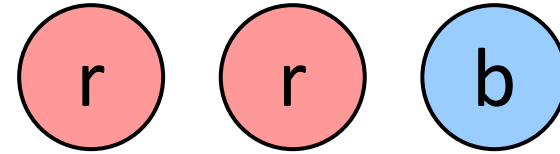
$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

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

- Laplace for conditionals:

- Smooth each condition independently:

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



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

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

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

线性插值估计*

- In practice, Laplace can perform poorly for $P(X|Y)$:

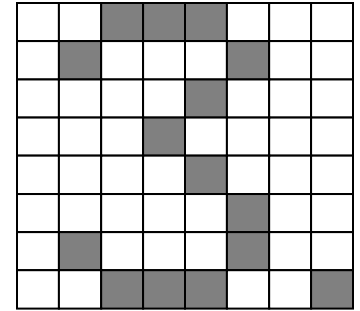
- When $|X|$ is very large
- When $|Y|$ is very large

- 另一个选择: 线性插值法

- 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?
- 还有一些其他参数估计的方法



现实应用: 平滑化估计参数

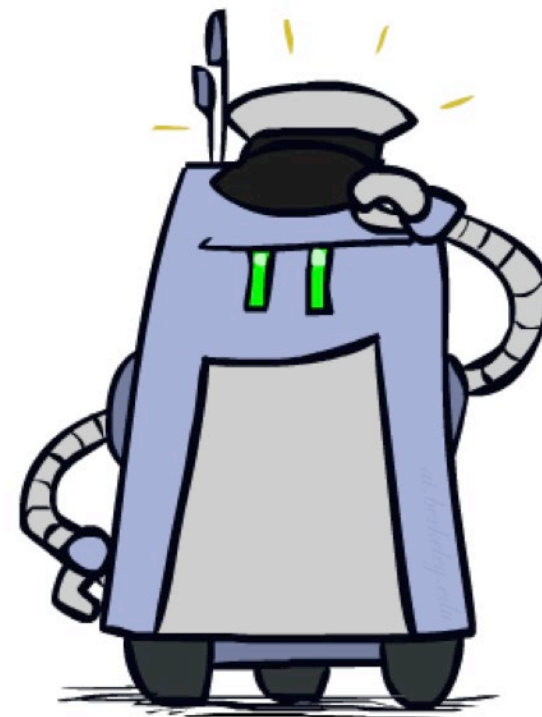
- For real classification problems, smoothing is critical
- New odds ratios:

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

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

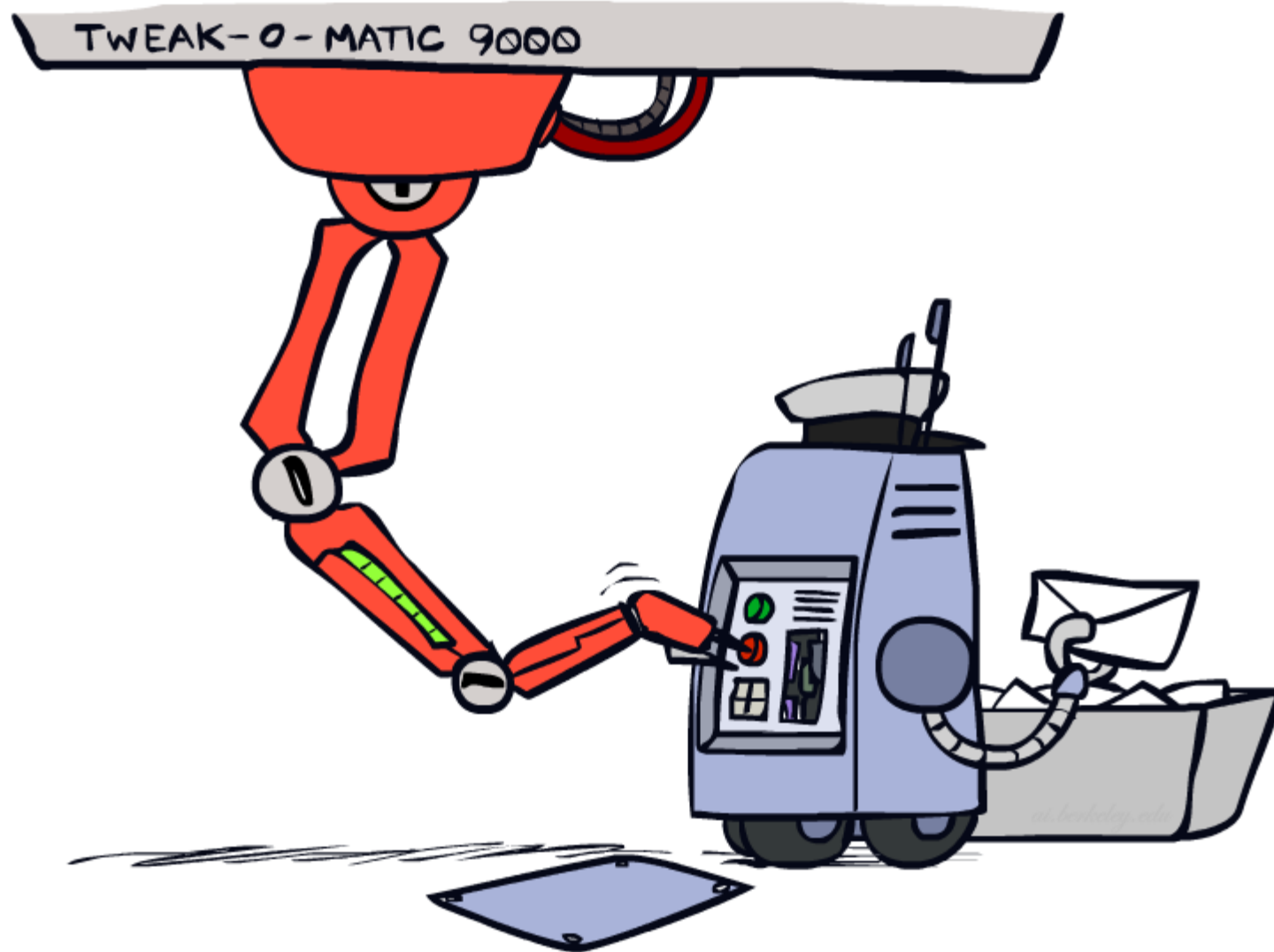
$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

| | | |
|---------|---|------|
| verdana | : | 28.8 |
| Credit | : | 28.4 |
| ORDER | : | 27.2 |
| | : | 26.9 |
| money | : | 26.5 |
| ... | | |



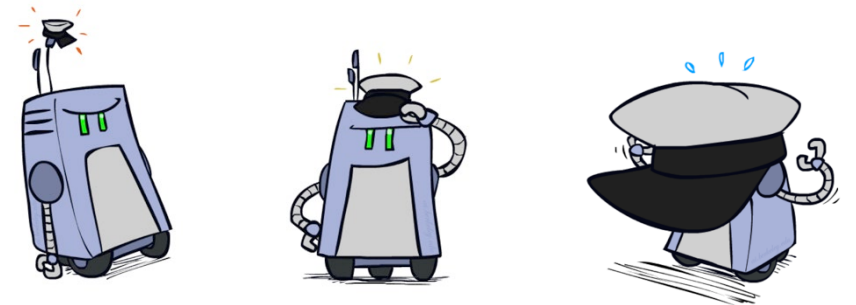
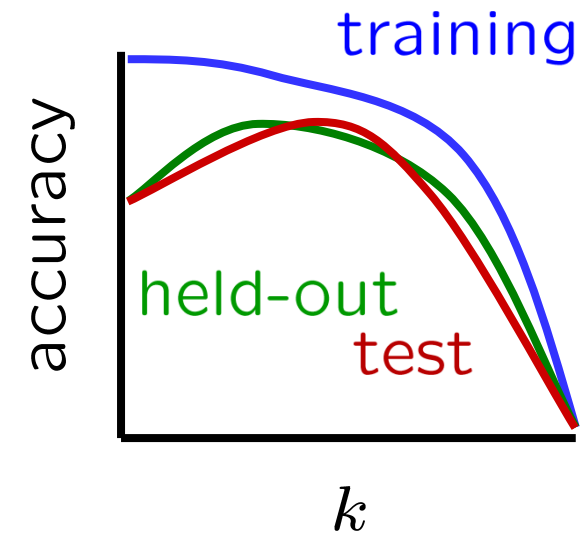
Do these make more sense?

调参数



在验证集上进行调节参数

- 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



基准表现作为对比

- First step: get a **baseline**
 - Baselines are very simple “straw man” procedures
 - Help determine how hard the task is
 - Help know what a “good” accuracy is
- Weak baseline: most frequent label classifier
 - Gives all test instances whatever label was most common in the training set
 - E.g. for spam filtering, might label everything as ham
 - Accuracy might be very high if the problem is skewed
 - E.g. calling everything “ham” gets 66%, so a classifier that gets 70% isn’t very good...
- For real research, usually use previous work as a (strong) baseline

分类器的置信度

- The **confidence** of a probabilistic classifier:

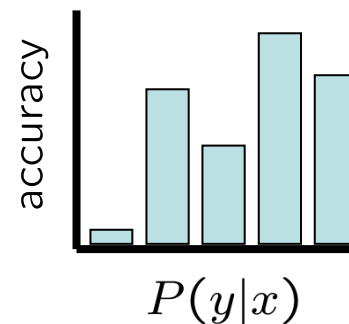
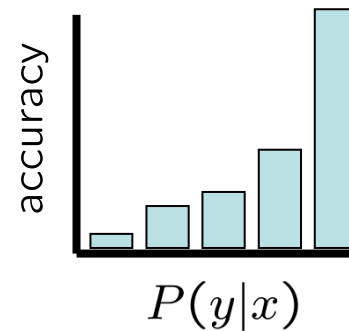
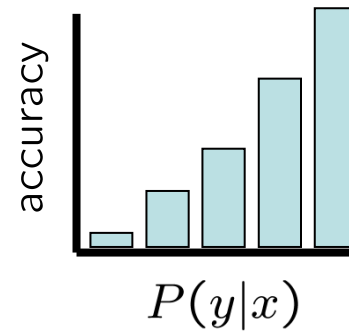
- Posterior over the top label

$$\text{confidence}(x) = \max_y P(y|x)$$

- Represents how sure the classifier is of the classification
- Any probabilistic model will have confidences
- No guarantee confidence is correct

- Calibration (校对)

- Weak calibration: higher confidences mean higher accuracy
- Strong calibration: confidence predicts accuracy rate
- What's the value of calibration?



总结

- 贝叶斯法则使得我们可以利用类似因果关系的条件概率，对未知变量进行诊断式查询（推理计算）
- 朴素贝叶斯模型假设所有特征变量之间都是条件于分类标签相互独立的
- 我们可以构建分类器，使用训练数据计算朴素贝叶斯模型的参数
- 对模型参数的平滑化估计在现实应用中很重要
- 分类器的置信度评价也很有用，如果可以计算的话