

CS 188: Artificial Intelligence

Spring 2010

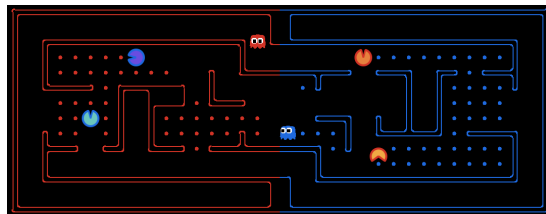
Lecture 22: Naïve Bayes

4/13/2010

Pieter Abbeel – UC Berkeley
Adapted from Dan Klein

Announcements

- Project 4 due Thursday
- Contest up since last night.
 - Nightly tournaments starting 11pm.



Machine Learning

- Up until now: how to reason in a model and how to make optimal decisions
- Machine learning: how to acquire a model on the basis of data / experience
 - Learning parameters (e.g. probabilities)
 - Learning structure (e.g. BN graphs)
 - Learning hidden concepts (e.g. clustering)

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.

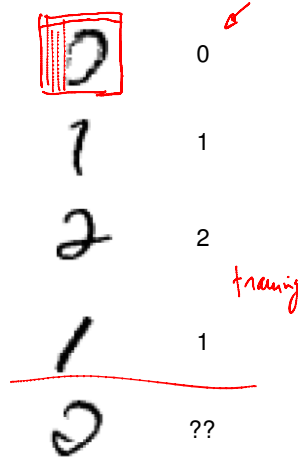
99 MILLION EMAIL ADDRESSES FOR ONLY \$99




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
 - ...

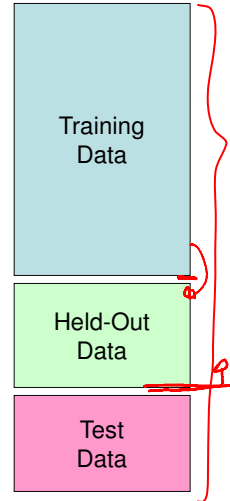


Other Classification Tasks

- 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! 

Important Concepts

- Data: labeled instances, e.g. emails marked spam/ham
 - Training set $\{(x^{(i)}, y^{(i)})\}_{i=1}^m$
 - 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



Bayes Nets for Classification

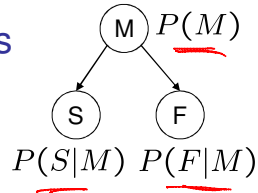
- One method of classification:
 - Use a probabilistic model! ←
 - Features are observed random variables F_i
 - Y is the query variable
 - Use probabilistic inference to compute most likely Y

→ $y = \operatorname{argmax}_y P(y|f_1 \dots f_n)$

- You already know how to do this inference

Simple Classification

- Simple example: two binary features



$P(m|s, f)$ ← direct estimate

→ $P(m|s, f) = \frac{P(s, f|m)P(m)}{P(s, f)}$ ← Bayes estimate (no assumptions)

$P(m|s, f) = \frac{P(s|m)P(f|m)P(m)}{P(s, f)}$ ← Conditional independence

+ $\left\{ \begin{array}{l} P(+m, s, f) = P(s|+m)P(f|+m)P(+m) \\ P(-m, s, f) = P(s|-m)P(f|-m)P(-m) \end{array} \right.$

General Naïve Bayes

- A general *naive Bayes* model:

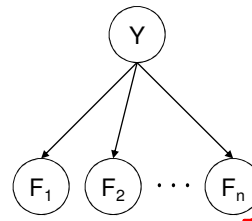
Labels → $|Y| \times |F|^n$ parameters
domains for F_i

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

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

$|Y|$ 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

y₁ vs y₂: $\log P(y_1) + \sum_i \log P(f_i|y_1) \stackrel{?}{>} \log P(y_2) + \sum_i \log P(f_i|y_2)$

Inference for Naïve Bayes

- Goal: compute posterior over causes *$\log \frac{P(y_1)}{P(y_2)} + \sum_i \log \frac{P(f_i|y_1)}{P(f_i|y_2)} \stackrel{?}{>} 0$*
 - Step 1: get joint probability of causes and evidence *$\log P(y_1) + \sum_i \log P(f_i|y_1)$*

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

- Step 2: get probability of evidence $P(f_1 \dots f_n)$
- Step 3: renormalize

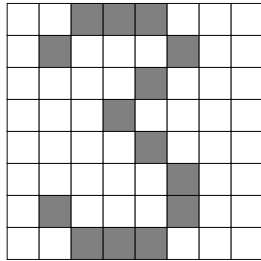
\downarrow
 $P(Y|f_1 \dots f_n)$

General Naïve Bayes

- What do we need in order to use naïve Bayes?
 - Inference (you know this part)
 - Start with a bunch of conditionals, P(Y) and the P(F_i|Y) tables
 - Use standard inference to compute P(Y|F₁...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 $\theta = \{ P(Y=y_1), \dots, P(Y=y_k), P(F_1=f_1|Y=y_1), \dots \}$
 - Up until now, we assumed these appeared by magic, but...
 - ...they typically come from training data: we'll look at this now

A Digit Recognizer

- Input: pixel grids



- Output: a digit 0-9



Naïve Bayes for Digits

- Simple version:
 - 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.

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

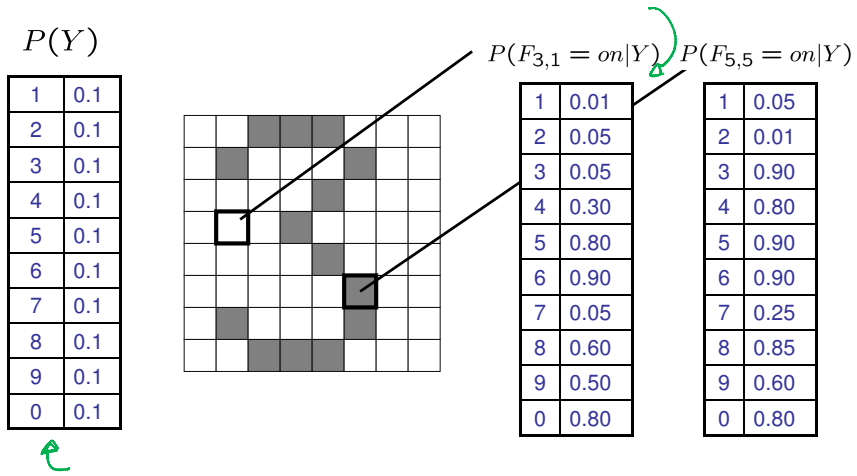
15x15

- 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?

Examples: CPTs



Parameter Estimation

- Estimating distribution of random variables like X or $X | Y$
- Empirically*: use training data
 - For each outcome x , look at the *empirical rate* of that value:

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

(r) (g) (g)

$$P_{ML}(r) = 1/3$$

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

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

- Elicitation*: ask a human!
 - Usually need domain experts, and sophisticated ways of eliciting probabilities (e.g. betting games)
 - Trouble calibrating

Handwritten notes:

$$\frac{\partial}{\partial \theta} = \dots$$

$$\max_{\theta} \log \theta (1-\theta) (1-\theta)$$

$$\max_{\theta} \log \theta + 2 \log (1-\theta)$$

$$\frac{\partial}{\partial \theta} = \frac{1}{\theta} + 2 \frac{-1}{1-\theta} = 0$$

$$1-\theta = 2\theta = 0$$

A Spam Filter

- 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

- Classifiers

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



Dear Sir.

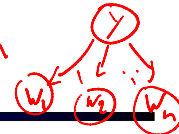
First, I must solicit your confidence in this transaction, this is by virtue of its nature as being utterly confidential and top secret. ...

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.

Naïve Bayes for Text



- Bag-of-Words Naïve Bayes:

- Predict unknown class label (spam vs. ham)
- Assume evidence features (e.g. the words) are independent
- Warning: subtly different assumptions than before!

- 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?

Example: Spam Filtering G

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

$P(C)$	$P(W spam)$	$P(W ham)$
ham : 0.66	the : 0.0156	the : 0.0210
spam: 0.33	to : 0.0153	to : 0.0133
	and : 0.0115	of : 0.0119
	of : 0.0095	2002: 0.0110
	you : 0.0093	with: 0.0108
	a : 0.0086	from: 0.0107
	with: 0.0080	and : 0.0105
	from: 0.0075	a : 0.0100

- Where do these tables come from? ?

Spam Example

Q: $P(Y=spam) + \log P(w|spam) + \dots$

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

$P(spam | w) = 98.9$

Example: Overfitting

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

$P(C = 2) = 0.1$

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

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

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

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

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

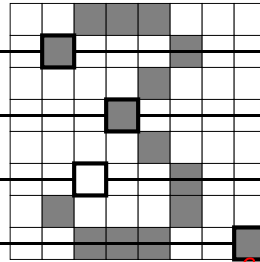
$P(C = 3) = 0.1$

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

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

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

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



2 wins!!

eg $\frac{P(y_1)}{P(y_2)}$ vs $\frac{P(f_1=y_1)}{P(f_1=y_2)}$

Example: Overfitting

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

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

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

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

```
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**

no inductive prior knowledge

ε

$$|\hat{\theta} - \theta^*| < \epsilon \text{ w.p. } 1 - \delta \quad n \geq \frac{1}{2\epsilon^2} \log \frac{1}{\delta}$$

Estimation: Smoothing

- **Problems with maximum likelihood estimates:**
 - If I flip a coin once, and it's heads, what's the estimate for P(heads)? *ML estimate: P(heads) = 1.0*
 - What if I flip 10 times with 8 heads? *ML estimate P(heads) = 0.8*
 - What if I flip 10M times with 8M heads? *ML estimate P(heads) = 0.8*
- **Basic idea:**
 - We have some prior expectation about parameters (here, the probability of heads)
 - Given little evidence, we should skew towards our prior
 - Given a lot of evidence, we should listen to the data

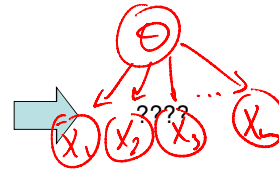
Estimation: Smoothing

- Relative frequencies are the maximum likelihood estimates

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

- In Bayesian statistics, we think of the parameters as just another random variable, with its own distribution

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



Estimation: Laplace Smoothing

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

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



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

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

- Can derive this as a MAP estimate with *Dirichlet priors* (see cs281a)

$$P(\theta) \propto \theta^{n_H} (1-\theta)^{n_T}$$

Estimation: Laplace Smoothing

- Laplace's estimate (extended):



- Pretend you saw every outcome k extra times

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

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

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

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

- Laplace for conditionals:

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

- Smooth each condition independently:

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