

CS 188: Artificial Intelligence Spring 2006

Lecture 9: Naïve Bayes 2/14/2006

Dan Klein – UC Berkeley
Many slides from either Stuart Russell or Andrew Moore

Today

- Bayes' rule
- Expectations and utilities
- Naïve Bayes models
 - Classification
 - Parameter estimation
 - Real world issues

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

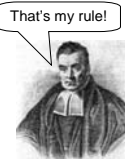
- Dividing, we get:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

- Why is this at all helpful?

- Lets us invert a conditional distribution
- Often the one conditional is tricky but the other simple
- Foundation of many systems we'll see later (e.g. ASR, MT)

- In the running for most important AI equation!



More Bayes' Rule

- Diagnostic probability from causal probability:

$$P(\text{Cause}|\text{Effect}) = \frac{P(\text{Effect}|\text{Cause})P(\text{Cause})}{P(\text{Effect})}$$

- Example:

- m is meningitis, s is stiff neck

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

- Note: posterior probability of meningitis still very small
- Does this mean you should ignore a stiff neck?

Expectations

- Real valued functions of random variables:

$$f : X \rightarrow R$$

- Expectation** of a function a random variable according to a distribution over the same variable

$$E_{P(X)}[f(X)] = \sum_x P(x)f(x)$$

- Example: Expected value of a fair die roll

$$\frac{1}{6} \times 1 + \frac{1}{6} \times 2 + \frac{1}{6} \times 3 + \frac{1}{6} \times 4 + \frac{1}{6} \times 5 + \frac{1}{6} \times 6 = 3.5$$



X	P	f
1	1/6	1
2	1/6	2
3	1/6	3
4	1/6	4
5	1/6	5
6	1/6	6

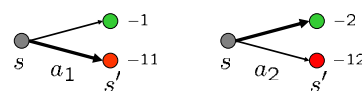
Utilities

- Preview of utility theory (much more later)

- Utilities:

- A *utility* or *reward* is a function from events to real numbers
- E.g. using a certain airport plan and getting there on time
- We often talk about actions having *expected* utilities in a given state

$$\text{utility}(a, s) = E_{P(s'|s,a)}[\text{reward}(s, a, s')]$$



- The rational action is the one which maximizes expected utility
- This depends on (1) the probability and (2) the magnitude of the outcomes

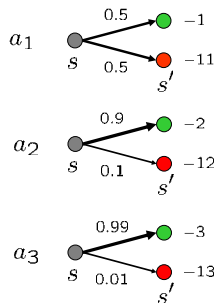
Example: Plane Plans

- How early to leave?

- Why might agents make different decisions?

- Different rewards
- Different evidence
- Different beliefs (different models)

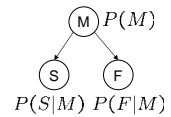
- We'll use the *principle of maximum expected utility* for classification, decision networks, reinforcement learning...



Combining Evidence

- What if there are multiple effects?

- E.g. diagnosis with two symptoms
- Meningitis, stiff neck, fever



$$P(m|s, f) \leftarrow \text{direct estimate}$$

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

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

$$+ \begin{cases} P(m, s, f) = P(s|m)P(f|m)P(m) \\ P(\bar{m}, s, f) = P(s|\bar{m})P(f|\bar{m})P(\bar{m}) \end{cases}$$

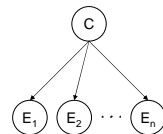
General Naïve Bayes

- This is an example of a *naïve Bayes* model:

$|C| \times |E|^n$
parameters

$$P(\text{Cause}, \text{Effect}_1 \dots \text{Effect}_n) = P(\text{Cause}) \prod_i P(\text{Effect}_i | \text{Cause})$$

$|C|$ parameters $n \times |E| \times |C|$ parameters



- Total number of parameters is *linear* in $n!$

Inference for Naïve Bayes

- Getting posteriors over causes

- Step 1: get joint probability of causes and evidence

$$P(C, e_1 \dots e_n) =$$

$$\begin{bmatrix} P(c_1, e_1 \dots e_n) \\ P(c_2, e_1 \dots e_n) \\ \vdots \\ P(c_k, e_1 \dots e_n) \end{bmatrix} \Rightarrow \begin{bmatrix} P(c_1) \prod_i P(e_i | c_1) \\ P(c_2) \prod_i P(e_i | c_2) \\ \vdots \\ P(c_k) \prod_i P(e_i | c_k) \end{bmatrix}$$

- Step 2: get probability of evidence

$$P(e_1 \dots e_n)$$

- Step 3: renormalize

$$P(C|e_1 \dots e_n)$$

General Naïve Bayes

- What do we need in order to use naïve Bayes?

- Some code to do the inference

- For fixed evidence, build $P(C, e)$
- Sum out C to get $P(e)$
- Divide to get $P(C|e)$

- Estimates of local *conditional probability tables* (CPTs)

- $P(C)$, the prior over causes
- $P(E|C)$ for each evidence variable
- These typically come from observed data
- These probabilities are collectively called the *parameters* of the model and denoted by θ

Parameter Estimation

- Estimating the distribution of a random variable X or $X|Y$?

- Empirically*: collect data

- For each value x , look at the *empirical rate* of that value:

$$\hat{P}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

$$\hat{P}(r) = 1/3$$

- This estimate maximizes the *likelihood of the data* (see homework)

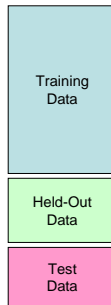
$$L(x, \theta) = \prod_i P_{\theta}(x_i)$$

- Elicitation*: ask a human!

- Usually need domain experts, and sophisticated ways of eliciting probabilities (e.g. betting games)
- Trouble calibrating

Classification

- Data: labeled instances, e.g. emails marked spam/ham
 - Training set
 - Held out set
 - Test set
- Experimentation
 - Learn model parameters (probabilities) on training set
 - (Tune performance on held-out set)
 - Run a single test on the 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



A Spam Filter

- Running example: naive 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 a model on the training set
 - Tune it on the held-out set
 - Test it on new emails in the test set



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.

Baselines

- First task: 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
- For real research, usually use previous work as a (strong) baseline

Naïve Bayes for Text

- Naïve Bayes:
 - Predict unknown cause (spam vs. ham)
 - Independent evidence from observed variables (e.g. the words)

- 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
 - In a bag-of-words model
 - Each position is identically distributed
 - All share the same distributions
 - Why make this assumption?

*Minor detail: technically we're conditioning on the length of the document here

Example: Spam Filtering

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

$P(C)$	$P(W \text{spam})$	$P(W \text{ham})$
ham : 0.63 spam: 0.37	the : 0.0156 to : 0.0153 and : 0.0115 of : 0.0095 you : 0.0093 a : 0.0086 with: 0.0080 from: 0.0075 ...	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?

Example: Spam Filtering

- Raw probabilities don't affect the posteriors; relative probabilities (odds ratios) do:

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

- These parameters will overfit the training data!
 - 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?
 - 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

Estimation: Smoothing

- Problems with maximum likelihood estimates:
 - If I flip a coin once, and it's heads, what's the estimate for $P(\text{heads})$?
 - What if I flip it 50 times with 27 heads?
 - What if I flip 10M times with 8M heads?
- 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
 - Note: we also have priors over model assumptions!

Estimation: Smoothing

- Relative frequencies are the maximum likelihood estimates

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

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

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

Estimation: Laplace Smoothing

- Laplace's estimate:

- Pretend you saw every outcome once more than you actually did



$$\begin{aligned}P_{LAP}(x) &= \frac{c(x) + 1}{\sum_x [c(x) + 1]} & P_{ML}(X) &= \left\langle \frac{2}{3}, \frac{1}{3} \right\rangle \\ &= \frac{c(x) + 1}{N + |X|} & P_{LAP}(X) &= \left\langle \frac{3}{5}, \frac{2}{5} \right\rangle\end{aligned}$$

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

Estimation: Laplace Smoothing

- Laplace's estimate (extended):



- Pretend you saw every outcome k extra times
- What's Laplace smoothing with $k = 0$?
- k is the strength of the prior

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

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

- Laplace for conditionals:

- Smooth each condition independently:

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

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

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

- Get $P(X)$ from the data
- Make sure the estimate of $P(X|Y)$ isn't too different from $P(X)$

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

- What if α is 0? 1?

- For even better ways to estimate parameters, as well as details of the math see cs281a, cs294-5

Real NB: Smoothing

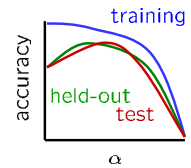
- For real classification problems, smoothing is critical
- New odds ratios:

$\frac{P(W \text{ham})}{P(W \text{spam})}$	$\frac{P(W \text{spam})}{P(W \text{ham})}$
helvetica : 11.4	verdana : 28.8
seems : 10.8	Credit : 28.4
group : 10.2	ORDER : 27.2
ago : 8.4	 : 26.9
areas : 8.3	money : 26.5
...	...

Do these make more sense?

Tuning on Held-Out Data

- Now we've got two kinds of unknowns
 - Parameters: the probabilities $P(Y|X)$, $P(Y)$
 - Hyper-parameters, like the amount of smoothing to do: k , α
- Where to learn?
 - Learn parameters from training data
 - Must tune hyper-parameters 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



Confidences from a Classifier

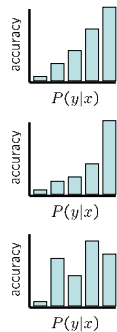
- The **confidence** of a probabilistic classifier:
 - Posterior over the top label

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

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

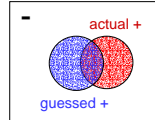
Calibration

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



Precision vs. Recall

- Let's say we want to classify web pages as homepages or not
 - In a test set of 1K pages, there are 3 homepages
 - Our classifier says they are all non-homepages
 - 99.7 accuracy!
 - Need new measures for rare positive events

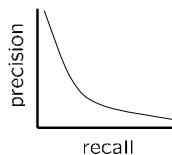


- Precision:** fraction of guessed positives which were actually positive
- Recall:** fraction of actual positives which were guessed as positive
- Say we guess 5 homepages, of which 2 were actually homepages
 - Precision: 2 correct / 5 guessed = 0.4
 - Recall: 2 correct / 3 true = 0.67
- Which is more important in customer support email automation?
- Which is more important in airport face recognition?

Precision vs. Recall

Precision/recall tradeoff

- Often, you can trade off precision and recall
- Only works well with weakly calibrated classifiers



To summarize the tradeoff:

- Break-even point: precision value when $p = r$
- F-measure: harmonic mean of p and r :

$$F_1 = \frac{2}{1/p + 1/r}$$

Summary

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption makes all effects independent given the cause
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- Classifier confidences are useful, when you can get them