Review of Probability

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Probability

- The world is a very uncertain place
- 30 years of Artificial Intelligence and Database research danced around this fact
- And then a few AI researchers decided to use some ideas from the eighteenth century

What we're going to do

- We will review the fundamentals of probability.
- It's really going to be worth it
- In this lecture, you'll see an example of probabilistic analytics in action: Bayes Classifiers

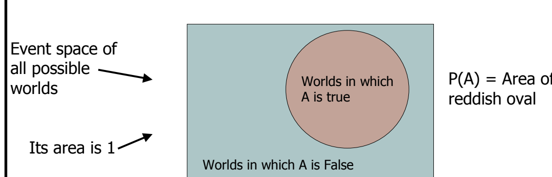
Discrete Random Variables

- A is a Boolean-valued random variable if A denotes an event, and there is some degree of uncertainty as to whether A occurs.
- Examples
- A = The US president in 2023 will be male
- A = You wake up tomorrow with a headache
- A = You have Ebola

Probabilities

- We write P(A) as "the fraction of possible worlds in which A is true"
- We could at this point spend 2 hours on the philosophy of this.
- But we won't.

Visualizing A



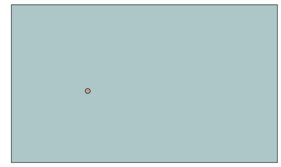
The Axioms of Probability

- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)

Where do these axioms come from? Were they "discovered"?
Answers coming up later.

Interpreting the axioms

- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)

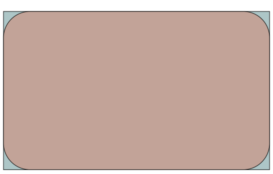


The area of A can't get any smaller than 0

And a zero area would mean no world could ever have A true

Interpreting the axioms

- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)

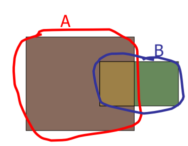


The area of A can't get any bigger than 1

And an area of 1 would mean all worlds will have A true

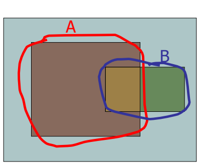
Interpreting the axioms

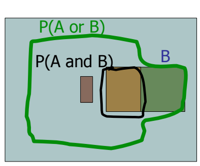
- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)



Interpreting the axioms

- 0 <= P(A) <= 1
- P(True) = 1
- P(False) = 0
- P(A or B) = P(A) + P(B) - P(A and B)





Simple addition and subtraction

These Axioms are Not to be Trifled With

- There have been attempts to do different methodologies for uncertainty
  - Fuzzy Logic
  - Three-valued logic
  - Dempster-Shafer
  - Non-monotonic reasoning
- But the axioms of probability are the only system with this property:

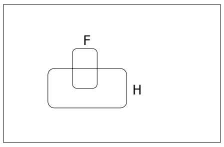
  If you gamble using them you can't be unfairly exploited by an opponent using some other system [di Finetti 1931]

Conditional Probability

- P(A|B) = Fraction of worlds in which B is true that also have A true

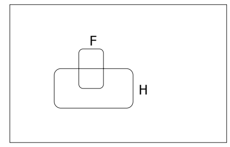
H = "Have a headache"
F = "Coming down with Flu"

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2



"Headaches are rare and flu is rarer, but if you're coming down with 'flu there's a 50-50 chance you'll have a headache."

Conditional Probability



H = "Have a headache"
F = "Coming down with Flu"

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

P(H|F) = Fraction of flu-inflicted worlds in which you have a headache

= #worlds with flu and headache

#worlds with flu

= Area of "H and F" region

Area of "F" region

= P(H ^ F)

P(F)

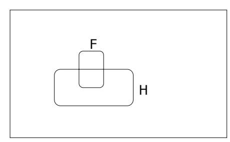
Definition of Conditional Probability

$$P(A|B) = \frac{P(A \wedge B)}{P(B)}$$

Corollary: The Chain Rule

$$P(A \wedge B) = P(A|B)\, P(B)$$

Probabilistic Inference



H = "Have a headache"
F = "Coming down with Flu"

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

One day you wake up with a headache. You think: "Drat! 50% of flus are associated with headaches so I must have a 50-50 chance of coming down with flu"

Is this reasoning good?

Bayes (Bayes'/Bayes's) Rule

P(A|B)P(B) = P(A^B) = P(B|A)P(A)
So

$$P(B|A) = \frac{P(A|B)\, P(B)}{P(A)}$$

This is Bayes Rule

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London,* **53:370-418**

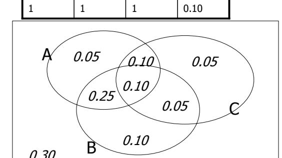
The Joint Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables then the table will have 2^M rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those numbers must sum to 1.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Using the Joint Distribution

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

One you have the JD you can ask for the probability of any logical expression involving your attribute

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor Male) = 0.4654

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor) = 0.7604

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

P(Male | Poor) = 0.4654 / 0.7604 = 0.612

Inference is a big deal

- I've got this evidence. What's the chance that this conclusion is true?
  - I've got a sore neck: how likely am I to have meningitis?
  - I see my lights are out and it's 9pm. What's the chance my spouse is already asleep?
- There's a thriving set of industries growing based around Bayesian Inference. Highlights are: Medicine, Pharma, Help Desk Support, Engine Fault Diagnosis

Where do Joint Distributions come from?

- Idea One: Expert Humans
- Idea Two: Simpler probabilistic facts and some algebra

Example: Suppose you knew

P(A) = 0.7 P(C|A^B) = 0.1
P(C|A^~B) = 0.8
P(B|A) = 0.2 P(C|~A^B) = 0.3
P(B|~A) = 0.1 P(C|~A^~B) = 0.1

Then you can automatically compute the JD using the chain rule

P(A=x ^ B=y ^ C=z) =
P(C=z|A=x^ B=y) P(B=y|A=x) P(A=x)

In another lecture: Bayes Nets, a systematic way to do this.

Where do Joint Distributions come from?

- Idea Three: Learn them from data!

Prepare to see one of the most impressive learning algorithms you'll come across in the entire course….

Learning a joint distribution

Build a JD table for your attributes in which the probabilities are unspecified

A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

The fill in each row with

$$\hat{P}(\text{row}) = \frac{\text{records matching row}}{\text{total number of records}}$$

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

Fraction of all records in which A and B are True but C is False

Example of Learning a Joint

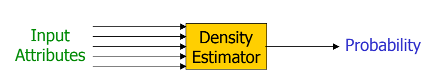
- This Joint was obtained by learning from three attributes in the UCI "Adult" Census Database [Kohavi 1995]

Where are we?

- We have recalled the fundamentals of probability
- We have become content with what JDs are and how to use them
- And we even know how to learn JDs from data.

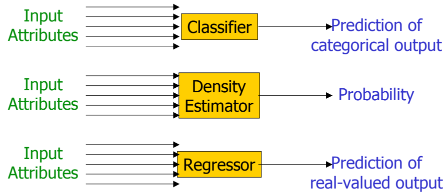
Density Estimation

- Our Joint Distribution learner is our first example of something called Density Estimation
- A Density Estimator learns a mapping from a set of attributes to a Probability



Density Estimation

- Compare it against the two other major kinds of models:



Using a density estimator

- Given a record $\mathbf{x}$, a density estimator M can tell you how likely the record is:

$$\hat{P}(\mathbf{x}|M)$$

- Given a dataset with R records, a density estimator can tell you how likely the dataset is:

(Under the assumption that all records were independently generated from the Density Estimator's JD)

$$\hat{P}(\text{dataset}|M) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

A small dataset: Miles Per Gallon

192 Training Set Records

mpg	modelyear	maker
good	75to78	asia
bad	70to74	america
bad	75to78	europe
bad	70to74	america
bad	70to74	america
bad	70to74	asia
bad	70to74	asia
bad	75to78	america
:	:	:
:	:	:
:	:	:
bad	70to74	america
good	79to83	america
bad	75to78	america
good	79to83	america
bad	75to78	america
good	79to83	america
good	79to83	america
bad	70to74	america
good	75to78	europe
bad	75to78	europe

From the UCI repository (thanks to Ross Quinlan)

A small dataset: Miles Per Gallon

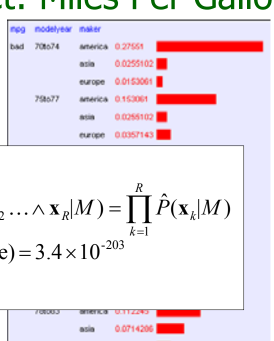
192 Training Set Records

mpg	modelyear	maker
good	75to78	asia
bad	70to74	america
bad	75to78	europe
bad	70to74	america
bad	70to74	america
bad	70to74	asia
bad	70to74	asia
bad	75to78	america
:	:	:
:	:	:
:	:	:
bad	70to74	america
good	79to83	america
bad	75to78	america
good	79to83	america
bad	75to78	america
good	79to83	america
good	79to83	america
bad	70to74	america
good	75to78	europe
bad	75to78	europe

A small dataset: Miles Per Gallon

192 Training Set

mpg	modelyear	maker
good	75to78	asia
bad	70to74	america
bad	75to78	europe
bad	70to74	america
bad	70to74	america
bad	70to74	asia
bad	70to74	asia



$$\hat{P}(\text{dataset}|M) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

$$= (\text{in this case}) = 3.4 \times 10^{-203}$$

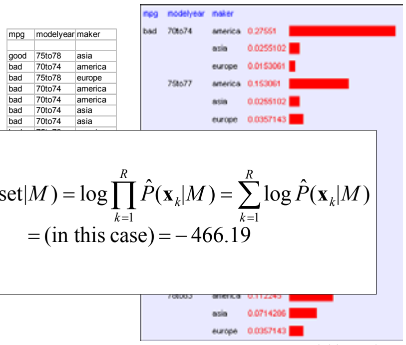
Log Probabilities

Since probabilities of datasets get so small we usually use log probabilities

$$\log \hat{P}(\text{dataset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

A small dataset: Miles Per Gallon

192 Training Set Records



$$\log \hat{P}(\text{dataset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$
$$= (\text{in this case}) = -466.19$$

Summary: The Good News

- We have a way to learn a Density Estimator from data. (Just count)
- Density estimators can do many good things…
  - Can sort the records by probability, and thus spot weird records (anomaly detection)
  - Can do inference: P(E1|E2)
    Automatic Doctor / Help Desk etc
  - Ingredient for Bayes Classifiers (see later)

Summary: The Bad News

- Density estimation by directly learning the joint is trivial, mindless and dangerous
- How much data do you need to accurately predict the probability of rare events? To fill in all possible situations?
- This is why probabilistic approaches were rejected earlier in AI. Interesting question: Why are probabilistic approaches popular now?

Using a test set

	Set Size	Log likelihood
Training Set	196	-466.1905
Test Set	196	-614.6157

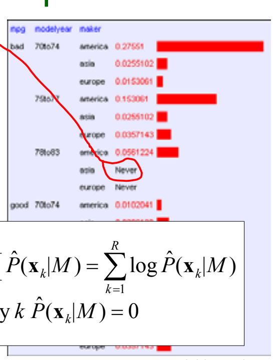
An independent test set with 196 cars has a worse log likelihood

(actually it's a billion quintillion quintillion quintillion quintillion times less likely)

….Density estimators can overfit (too many parameters, too little data). And the full joint density estimator is the overfittiest of them all!

The zero problem

If this ever happens, it means there are certain combinations that we learn are impossible



$$\log \hat{P}(\text{testset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$
$$= -\infty \text{ if for any } k \ \hat{P}(\mathbf{x}_k|M) = 0$$

Using a test set

	Set Size	Log likelihood
Training Set	196	-466.1905
Test Set	196	-614.6157

The only reason that our test set didn't score -infinity is that my code is hard-wired to always predict a probability of at least one in 10^{20}

We need Density Estimators that are less prone to overfitting

Naïve Density Estimation

The problem with the Joint Estimator is that it just mirrors the training data.

We need something which generalizes more usefully.

The naïve model generalizes strongly (and is usually wrong/approximate):

Assume that each attribute is distributed independently of any of the other attributes.

Independently Distributed Data

- Let *x[i]* denote the *i*th field of record *x*.
- The independently distributed assumption says that for any *i,v,* u_1 u_2*…* u_{i-1} u_{i+1}*…* u_M

$$P(x[i]=v \mid x[1]=u_1, x[2]=u_2, \ldots x[i-1]=u_{i-1}, x[i+1]=u_{i+1}, \ldots x[M]=u_M) = P(x[i]=v)$$

- Or in other words, *x[i]* is independent of *{x[1],x[2],..x[i-1], x[i+1],…x[M]}*
- This is often written as

$$x[i] \perp \{x[1], x[2], \ldots x[i-1], x[i+1], \ldots x[M]\}$$

A note about independence

- Assume A and B are Boolean Random Variables. Then

"A and B are independent"

if and only if

P(A|B) = P(A)

- "A and B are independent" is often notated as $A \perp B$

Independence Theorems

- Assume P(A|B) = P(A)
- Then P(A^B) =

= P(A) P(B)

- Assume P(A|B) = P(A)
- Then P(B|A) =

= P(B)

Independence Theorems

- Assume P(A|B) = P(A)
- Then P(~A|B) =

= P(~A)

- Assume P(A|B) = P(A)
- Then P(A|~B) =

= P(A)

Multivalued Independence

For multivalued Random Variables A and B,

$$A \perp B$$

if and only if

$$\forall u,v : P(A=u \mid B=v) = P(A=u)$$

from which you can then prove things like…

$$\forall u,v : P(A=u \wedge B=v) = P(A=u)P(B=v)$$

$$\forall u,v : P(B=v \mid A=v) = P(B=v)$$

Using the Naïve Distribution

- Once you have a Naïve Distribution you can easily compute any row of the joint distribution.
- Suppose *A, B, C* and *D* are independently distributed. What is *P(A^~B^C^~D)*?

Using the Naïve Distribution

- Once you have a Naïve Distribution you can easily compute any row of the joint distribution.
- Suppose A, B, C and D are independently distributed. What is P(A^~B^C^~D)?

= P(A|~B^C^~D) P(~B^C^~D)
= P(A) P(~B^C^~D)
= P(A) P(~B|C^~D) P(C^~D)
= P(A) P(~B) P(C^~D)
= P(A) P(~B) P(C|~D) P(~D)
= P(A) P(~B) P(C) P(~D)

Naïve Distribution General Case

- Suppose *x[1], x[2], … x[M]* are independently distributed.

$$P(x[1]=u_1, x[2]=u_2, \ldots x[M]=u_M) = \prod_{k=1}^{M} P(x[k]=u_k)$$

- So if we have a Naïve Distribution we can construct any row of the implied Joint Distribution on demand.
- So we can do any inference
- But how do we learn a Naïve Density Estimator?

Learning a Naïve Density Estimator

$$\hat{P}(x[i]=u) = \frac{\#\text{records in which } x[i]=u}{\text{total number of records}}$$

Another trivial learning algorithm!

Contrast

Joint DE	Naïve DE
Can model anything	Can accurately model only very boring distributions, but often good approximation
No problem to model "C is a noisy copy of A"	Outside Naïve's scope
Given 100 records and more than 6 Boolean attributes will screw up badly	Given 100 records and 10,000 multivalued attributes will be fine

Reminder: The Good News

- We have two ways to learn a Density Estimator from data.
- *In other lectures we'll see vastly more impressive Density Estimators (Mixture Models, Bayesian Networks, Density Trees, Kernel Densities and many more)
- Density estimators can do many good things…
  - Anomaly detection
  - Can do inference: P(E1|E2) Automatic Doctor / Help Desk etc
  - Ingredient for Bayes Classifiers

Bayes Classifiers

How to build a Probabilistic Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \dots v_{ny}$.
- Assume there are m input attributes called $X_1, X_2, \dots X_m$
- Sort dataset into n_Y smaller datasets called $DS_1, DS_2, \dots DS_{ny}$, with DS_i = Records in which $Y=v_i$
- For each DS_i, learn Density Estimator M_i to model the input distribution among the $Y=v_i$ records.

How to build a Probabilistic Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \dots v_{ny}$.
- Assume there are m input attributes called $X_1, X_2, \dots X_m$
- Sort dataset into n_Y smaller datasets called $DS_1, DS_2, \dots DS_{ny}$, with DS_i = Records in which $Y=v_i$
- For each DS_i, learn Density Estimator M_i to model the input distribution among the $Y=v_i$ records.
- M_i estimates P($X_1, X_2, \dots X_m$ / $Y=v_i$)

How to build a Probabilistic Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \dots v_{ny}$.
- Assume there are m input attributes called $X_1, X_2, \dots X_m$
- Break dataset into n_Y smaller datasets called $DS_1, DS_2, \dots DS_{ny}$.
- Define DS_i = Records in which $Y=v_i$
- For each DS_i, learn Density Estimator M_i to model the input distribution among the $Y=v_i$ records.
- M_i estimates P($X_1, X_2, \dots X_m$ / $Y=v_i$)
- Idea: When a new set of input values ($X_1 = u_1, X_2 = u_2, \dots. X_m = u_m$) come along to be evaluated predict the value of Y that makes P($X_1, X_2, \dots X_m$ / $Y=v_i$) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

Is this a good idea?

How to build a Probabilistic Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \dots v_{ny}$.
- Assume there are m input attribu
- Break dataset into n_Y smaller dat
- Define DS_i = Records in which Y
- For each DS_i, learn Density Esti
  distribution among the $Y=v_i$ reco

This is a Maximum Likelihood classifier.

It can get silly if some Ys are very unlikely

- M_i estimates P($X_1, X_2, \dots X_m$ / $Y=v_i$)
- Idea: When a new set of input values ($X_1 = u_1, X_2 = u_2, \dots. X_m = u_m$) come along to be evaluated predict the value of Y that makes P($X_1, X_2, \dots X_m$ / $Y=v_i$) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

Is this a good idea?

How to build a Bayes Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \dots v_{ny}$.
- Assume there are m input attributes calle
- Break dataset into n_Y smaller datasets call
- Define DS_i = Records in which $Y=v_i$
- For each DS_i, learn Density Estimator M_i distribution among the $Y=v_i$ records.
- M_i estimates P($X_1, X_2, \dots X_m$ / $Y=v_i$)

Much Better Idea

- Idea: When a new set of input value ($X_1 = u_1, X_2 = u_2, \dots. X_m = u_m$) come along to be evaluate predict the value of Y that makes P($Y=v_i$ / $X_1, X_2, \dots X_m$) most likely

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

Is this a good idea?

Terminology

- MLE (Maximum Likelihood Estimator):

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

- MAP (Maximum A-Posteriori Estimator):

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

Getting what we need

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

Getting a posterior probability

$$P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

$$= \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v)}{P(X_1 = u_1 \cdots X_m = u_m)}$$

$$= \frac{P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v)}{\sum_{j=1}^{n_Y} P(X_1 = u_1 \cdots X_m = u_m \mid Y = v_j)P(Y = v_j)}$$

Bayes Classifiers in a nutshell

1. Learn the distribution over inputs for each value Y.
2. This gives P(X_1, X_2, … X_m / $Y=v_i$).
3. Estimate P($Y=v_i$). as fraction of records with $Y=v_i$.
4. For a new prediction:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$
$$= \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v)$$

Bayes Classifiers in a nutshell

1. Learn the distribution over inputs for each value Y.
2. This gives P(X_1, X_2, … X_m / $Y=v_i$).
3. Estimate P($Y=v_i$). as fraction of records w… $Y=v_i$.
4. For a new prediction:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v \mid X_1 \cdots$$
$$= \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \cdots$$

We can use our favorite Density Estimator here.

Right now we have two options:

- •Joint Density Estimator
- •Naïve Density Estimator

Joint Density Bayes Classifier

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v)$$

In the case of the joint Bayes Classifier this degenerates to a very simple rule:

$Y^{predict}$ = the most common value of Y among records in which $X_1 = u_1$, $X_2 = u_2$, …. $X_m = u_m$.

Note that if no records have the exact set of inputs $X_1 = u_1$, $X_2 = u_2$, …. $X_m = u_m$, then P(X_1, X_2, … X_m / $Y=v_i$) = 0 for all values of Y.

In that case we just have to guess Y's value

Naïve Bayes Classifier

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v) P(Y = v)$$

In the case of the naive Bayes Classifier this can be simplified:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v) \prod_{j=1}^{n_Y} P(X_j = u_j \mid Y = v)$$

Naïve Bayes Classifier

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(X_1 = u_1 \cdots X_m = u_m \mid Y = v) P(Y = v)$$

In the case of the naive Bayes Classifier this can be simplified:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}}\, P(Y = v) \prod_{j=1}^{n_Y} P(X_j = u_j \mid Y = v)$$

Technical Hint:
If you have 10,000 input attributes that product will underflow in floating point math. You should use logs:

$$Y^{\text{predict}} = \underset{v}{\operatorname{argmax}} \left(\log P(Y = v) + \sum_{j=1}^{n_Y} \log P(X_j = u_j \mid Y = v) \right)$$

More Facts About Bayes Classifiers

- Many other density estimators can be slotted in*.
- Density estimation can be performed with real-valued inputs*
- Bayes Classifiers can be built with real-valued inputs*
- Rather Technical Complaint: Bayes Classifiers don't try to be maximally discriminative---they merely try to honestly model what's going on*
- Zero probabilities are painful for Joint and Naïve. A hack (justifiable with the magic words "Dirichlet Prior") can help*.
- Naïve Bayes is wonderfully cheap. And survives 10,000 attributes cheerfully!

*See future Andrew Lectures

What you should know

- Probability
  - Fundamentals of Probability and Bayes Rule
  - What's a Joint Distribution
  - How to do inference (i.e. P(E1|E2)) once you have a JD
- Density Estimation
  - What is DE and what is it good for
  - How to learn a Joint DE
  - How to learn a naïve DE

What you should know

- Bayes Classifiers
  - How to build one
  - How to predict with a BC
  - Contrast between naïve and joint BCs