Probabilistic and Bayesian Analytics

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University

www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Copyright © 2001, Andrew W. Moore
Probabilistic Analytics: Slide 2

Probabilistic and Bayesian Analytics

Andrew W. Moore
Associate Professor
School of Computer Science
Carnegie Mellon University

www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Copyright © 2001, Andrew W. Moore
Aug 25th, 2001

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

Copyright © 2001, Andrew W. Moore
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

Event space of all possible worlds

Its area is 1

$P(A) = \text{Area of reddish oval}$

Worlds in which $A$ is true

Worlds in which $A$ is False
The Axioms of Probability

- $0 \leq P(A) \leq 1$
- $P(\text{True}) = 1$
- $P(\text{False}) = 0$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

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

Interpreting the axioms

- $0 \leq P(A) \leq 1$
- $P(\text{True}) = 1$
- $P(\text{False}) = 0$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

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

• \(0 \leq P(A) \leq 1\)
• \(P(\text{True}) = 1\)
• \(P(\text{False}) = 0\)
• \(P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)\)

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

- $0 \leq P(A) \leq 1$
- $P(\text{True}) = 1$
- $P(\text{False}) = 0$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

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]
Theorems from the Axioms

- \( 0 \leq P(A) \leq 1 \), \( P(\text{True}) = 1 \), \( P(\text{False}) = 0 \)
- \( P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B) \)

From these we can prove:
\[ P(\text{not } A) = P(\sim A) = 1 - P(A) \]

- How?

Side Note

- I am inflicting these proofs on you for two reasons:
  1. These kind of manipulations will need to be second nature to you if you use probabilistic analytics in depth
  2. Suffering is good for you
Another important theorem

- $0 \leq P(A) \leq 1$, $P(\text{True}) = 1$, $P(\text{False}) = 0$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

From these we can prove:

\[ P(A) = P(A \land B) + P(A \land \neg B) \]

- How?

Multivalued Random Variables

- Suppose $A$ can take on more than 2 values
- $A$ is a random variable with arity $k$ if it can take on exactly one value out of \( \{v_1, v_2, ..., v_k\} \)
- Thus...

\[
\begin{align*}
P(A = v_i \land A = v_j) &= 0 \text{ if } i \neq j \\
P(A = v_1 \lor A = v_2 \lor A = v_k) &= 1
\end{align*}
\]
An easy fact about Multivalued Random Variables:

- Using the axioms of probability...
  
  \[ 0 \leq P(A) \leq 1, \quad P(\text{True}) = 1, \quad P(\text{False}) = 0 \]
  
  \[ P(\text{A or B}) = P(\text{A}) + P(\text{B}) - P(\text{A and B}) \]
  
- And assuming that A obeys...
  
  \[ P(A = v_i \land A = v_j) = 0 \text{ if } i \neq j \]
  
  \[ P(A = v_1 \lor A = v_2 \lor A = v_k) = 1 \]
  
- It’s easy to prove that
  
  \[ P(A = v_1 \lor A = v_2 \lor A = v_i) = \sum_{j=1}^{i} P(A = v_j) \]

An easy fact about Multivalued Random Variables:

- Using the axioms of probability...
  
  \[ 0 \leq P(A) \leq 1, \quad P(\text{True}) = 1, \quad P(\text{False}) = 0 \]
  
  \[ P(\text{A or B}) = P(\text{A}) + P(\text{B}) - P(\text{A and B}) \]
  
- And assuming that A obeys...
  
  \[ P(A = v_i \land A = v_j) = 0 \text{ if } i \neq j \]
  
  \[ P(A = v_1 \lor A = v_2 \lor A = v_k) = 1 \]
  
- It’s easy to prove that
  
  \[ P(A = v_1 \lor A = v_2 \lor A = v_i) = \sum_{j=1}^{i} P(A = v_j) \]
  
- And thus we can prove
  
  \[ \sum_{j=1}^{k} P(A = v_j) = 1 \]
Another fact about Multivalued Random Variables:

- Using the axioms of probability...
  
  \[ 0 \leq P(A) \leq 1, \quad P(\text{True}) = 1, \quad P(\text{False}) = 0 \]
  
  \[ P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B) \]

- And assuming that A obeys...

  \[ P(A = v_i \land A = v_j) = 0 \text{ if } i \neq j \]
  
  \[ P(A = v_1 \lor A = v_2 \lor A = v_k) = 1 \]

- It’s easy to prove that

  \[ P(B \land [A = v_1 \lor A = v_2 \lor A = v_j]) = \sum_{j=1}^{i} P(B \land A = v_j) \]

- And thus we can prove

  \[ P(B) = \sum_{j=1}^{k} P(B \land A = v_j) \]
Elementary Probability in Pictures

- $P(\sim A) + P(A) = 1$

Elementary Probability in Pictures

- $P(B) = P(B \land A) + P(B \land \sim A)$
Elementary Probability in Pictures

$$\sum_{j=1}^{k} P(A = v_j) = 1$$

Elementary Probability in Pictures

$$P(B) = \sum_{j=1}^{k} P(B \land A = v_j)$$
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.”
**Definition of Conditional Probability**

\[
P(A \cap B) = \frac{P(A|B)}{P(B)}
\]

**Corollary: The Chain Rule**

\[
P(A \cap B) = P(A|B) P(B)
\]

---

**Probabilistic Inference**

- **H** = “Have a headache”
- **F** = “Coming down with Flu”

\[
\begin{align*}
P(H) &= 1/10 \\
P(F) &= 1/40 \\
P(H|F) &= 1/2
\end{align*}
\]

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?
Probabilistic Inference

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

\[
P(H) = \frac{1}{10}
\]
\[
P(F) = \frac{1}{40}
\]
\[
P(H|F) = \frac{1}{2}
\]

\[
P(F \cap H) = \ldots
\]
\[
P(F|H) = \ldots
\]

Another way to understand the intuition

Thanks to Jahanzeb Sherwani for contributing this explanation:

Let's say we have \(P(F), P(H), \) and \(P(H|F)\), like in the example in class.

Areawise, \(P(F) = A + B\), \(P(H) = B + C\).

Also, \(P(H|F) = \frac{B}{A + B}\).

Thus, to get the opposite conditional probability, i.e., \(P(F|H)\), we need to figure out \(\frac{B}{B + C}\).

Since we know \(B / (A + B)\), we can get \(B / (B + C)\) by multiplying by \((A + B)\) and dividing by \((B + C)\). But since we already calculated, \(A + B = P(F)\), and \(B + C = P(H)\), so we are actually multiplying by \(P(F)\) and dividing by \(P(H)\). Which is Bayes Rule:

\[
P(F|H) = \frac{P(H|F) \cdot P(F)}{P(H)}
\]
What we just did...

\[
P(B|A) = \frac{P(A \land B)}{P(A)} = \frac{P(A|B)P(B)}{P(A)}
\]

This is Bayes Rule


Using Bayes Rule to Gamble

The “Win” envelope has a dollar and four beads in it

The “Lose” envelope has three beads and no money

Trivial question: someone draws an envelope at random and offers to sell it to you. How much should you pay?
Using Bayes Rule to Gamble

The "Win" envelope has a dollar and four beads in it.

The "Lose" envelope has three beads and no money.

Interesting question: before deciding, you are allowed to see one bead drawn from the envelope.
Suppose it’s black: How much should you pay?
Suppose it’s red: How much should you pay?

Calculation...

Copyright © 2001, Andrew W. Moore
More General Forms of Bayes Rule

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\sim A)P(\sim A)} \]

\[ P(A|B \land X) = \frac{P(B|A \land X)P(A \land X)}{P(B \land X)} \]

More General Forms of Bayes Rule

\[ P(A=v_i | B) = \frac{P(B|A=v_i)P(A=v_i)}{\sum_{k=1}^{n_A} P(B|A=v_k)P(A=v_k)} \]
Useful Easy-to-prove facts

\[ P(A \mid B) + P(\neg A \mid B) = 1 \]

\[ \sum_{k=1}^{n_A} P(A = v_k \mid B) = 1 \]

The Joint Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:
The Joint Distribution

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.

Example: Boolean variables $A$, $B$, $C$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
</tbody>
</table>
The Joint Distribution

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.

Example: Boolean variables A, B, C

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Using the Joint

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

\[ P(\text{Poor Male}) = 0.4654 \]

\[ P(E) = \sum \text{P(row)} \text{ matching } E \]

Using the Joint

\[ P(\text{Poor}) = 0.7604 \]

\[ P(E) = \sum \text{P(row)} \text{ matching } E \]
### Inference with the Joint

\[ P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum P(\text{row})}{\sum P(\text{row})} \]

<table>
<thead>
<tr>
<th>Gender</th>
<th>Hours_Worked</th>
<th>Wealth</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>v0:40.5-</td>
<td>poor</td>
<td>0.253122</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
<td>0.0245895</td>
</tr>
<tr>
<td></td>
<td>v1:40.5+</td>
<td>poor</td>
<td>0.0421760</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
<td>0.0116293</td>
</tr>
<tr>
<td>Male</td>
<td>v0:40.5-</td>
<td>poor</td>
<td>0.331313</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
<td>0.0971296</td>
</tr>
<tr>
<td></td>
<td>v1:40.5+</td>
<td>poor</td>
<td>0.134106</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
<td>0.105933</td>
</tr>
</tbody>
</table>

\[
P(\text{Male} \mid \text{Poor}) = \frac{0.4654}{0.7604} = 0.612
\]
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?
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

\[
\begin{align*}
P(A) &= 0.7 & P(C|A^B) &= 0.1 \\
P(C|A^~B) &= 0.8 \\
P(B|A) &= 0.2 & P(C|A^B) &= 0.3 \\
P(B|~A) &= 0.1 & P(C|~A^B) &= 0.1 \\
\end{align*}
\]

Then you can automatically compute the JD using the chain rule

\[
P(A=x \land B=y \land C=z) = P(C=z|A=x \land B=y) \ P(B=y|A=x) \ P(A=x)
\]

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>

The fill in each row with:

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.30</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.05</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.10</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.10</td>
</tr>
</tbody>
</table>

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]

<table>
<thead>
<tr>
<th>gender</th>
<th>hours_worked</th>
<th>wealth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>v0:40.5-</td>
<td>poor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
</tr>
<tr>
<td></td>
<td>v1:40.5+</td>
<td>poor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
</tr>
<tr>
<td>Male</td>
<td>v0:40.5-</td>
<td>poor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
</tr>
<tr>
<td></td>
<td>v1:40.5+</td>
<td>poor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rich</td>
</tr>
</tbody>
</table>

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:

- **Classifier**
  - Prediction of categorical output

- **Density Estimator**
  - Probability

- **Regressor**
  - Prediction of real-valued output
Evaluating Density Estimation

Test-set criterion for estimating performance on future data*

* See the Decision Tree or Cross Validation lecture for more detail

<table>
<thead>
<tr>
<th>Input Attributes</th>
<th>Classifier</th>
<th>Prediction of categorical output</th>
<th>Test set Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Attributes</td>
<td>Density Estimator</td>
<td>Probability</td>
<td>?</td>
</tr>
<tr>
<td>Input Attributes</td>
<td>Regressor</td>
<td>Prediction of real-valued output</td>
<td>Test set Accuracy</td>
</tr>
</tbody>
</table>

Evaluating 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)
  \[
  \hat{P}(\text{dataset}|M) = \hat{P}(\mathbf{x}_1 \land \mathbf{x}_2 \ldots \land \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)
  \]
A small dataset: Miles Per Gallon

<table>
<thead>
<tr>
<th>mpg</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>79/78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70/74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79/78</td>
<td>europe</td>
</tr>
</tbody>
</table>

From the UCI repository (thanks to Ross Quinlan)
**A small dataset: Miles Per Gallon**

<table>
<thead>
<tr>
<th>mpg</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>75to78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>75to78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>75to78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79to83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>75to78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>75to78</td>
<td>europe</td>
</tr>
</tbody>
</table>

\[
\hat{P} \text{(dataset} | M) = \prod_{k=1}^{R} \hat{P}(x_k | M) = 3.4 \times 10^{-203}
\]

**Log Probabilities**

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

\[
\log \hat{P} \text{(dataset} | M) = \sum_{k=1}^{R} \log \hat{P}(x_k | M)
\]
A small dataset: Miles Per Gallon

192 Training Set Records

<table>
<thead>
<tr>
<th>mpg</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>75-78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>75-78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>75-78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79-83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>75-78</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79-83</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>79-83</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>70-74</td>
<td>america</td>
</tr>
<tr>
<td>good</td>
<td>75-78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>75-78</td>
<td>europe</td>
</tr>
</tbody>
</table>

\[
\log \hat{P}(\text{dataset}|M) = \log \prod_{k=1}^{R} \hat{P}(x_k|M) = \sum_{k=1}^{R} \log \hat{P}(x_k|M)
= (\text{in this case}) = -466.19
\]

Summary: The Good News

- We have a way to learn a Density Estimator from data.
- Density estimators can do many good things...
  - Can sort the records by probability, and thus spot weird records (anomaly detection)
  - Can do inference: \( P(E_1|E_2) \)
    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

Using a test set

<table>
<thead>
<tr>
<th>Set</th>
<th>Size</th>
<th>Log likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>196</td>
<td>-466.1905</td>
</tr>
<tr>
<td>Test</td>
<td>196</td>
<td>-614.6157</td>
</tr>
</tbody>
</table>

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. And the full joint density estimator is the overfittiest of them all!
Overfitting Density Estimators

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

\[
\log \hat{P}(\text{testset}|M) = \log \prod_{k=1}^{R} \hat{P}(x_k|M) = \sum_{k=1}^{R} \log \hat{P}(x_k|M)
\]

\[= -\infty \text{ if for any } k \hat{P}(x_k|M) = 0\]

Using a test set

<table>
<thead>
<tr>
<th>Set Size</th>
<th>Log likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>196</td>
</tr>
<tr>
<td>Test Set</td>
<td>196</td>
</tr>
</tbody>
</table>

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:
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 $i, v$, $u_1 u_2 \ldots u_{i-1} u_{i+1} \ldots u_M$
  $$P(x[i] = v | x[1] = u_1, x[2] = u_2, \ldots x[i-1] = u_{i-1}, x[i+1] = u_{i+1}, \ldots x[M] = u_M) = P(x[i] = v)$$
- Or in other words, $x[i]$ is independent of \{x[1],x[2],\ldots,x[i-1], x[i+1],\ldots,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 $P(A^B) =$

$= P(A) \cdot P(B)$

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

$= P(B)$
Independence Theorems

- Assume $P(A|B) = P(A)$
- Then $P(\neg A|B) = P(\neg A)$
- Assume $P(A|B) = P(A)$
- Then $P(A|\neg B) = P(A)$

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...

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

$$\forall u, v : P(B = v \mid A = v) = P(B = v)$$
Back to Naïve Density Estimation

- Let \( x[i] \) denote the \( i \)th field of record \( x \):
- Naïve DE assumes \( x[i] \) is independent of \( \{x[1], x[2], \ldots, x[i-1], x[i+1], \ldots, x[M]\} \)
- Example:
  - Suppose that each record is generated by randomly shaking a green dice and a red dice
    - Dataset 1: \( A = \) red value, \( B = \) green value
    - Dataset 2: \( A = \) red value, \( B = \) sum of values
    - Dataset 3: \( A = \) sum of values, \( B = \) difference of values
  - Which of these datasets violates the naïve assumption?

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 \( P(A \sim B \sim C \sim D) \)?
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 $P(A \wedge \neg B \wedge C \wedge \neg D)$?

$$
P(A \wedge \neg B \wedge C \wedge \neg D) = P(A | \neg B \wedge C \wedge \neg D) \cdot P(\neg B \wedge C \wedge \neg D)
$$

$$
= P(A) \cdot P(\neg B \wedge C \wedge \neg D)
$$

$$
= P(A) \cdot P(\neg B | C \wedge \neg D) \cdot P(C \wedge \neg D)
$$

$$
= P(A) \cdot P(\neg B) \cdot P(C | \neg D) \cdot P(\neg D)
$$

$$
= P(A) \cdot P(\neg B) \cdot P(C) \cdot P(\neg D)
$$

Naïve Distribution General Case

- Suppose $x[1], x[2], \ldots x[M]$ 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

<table>
<thead>
<tr>
<th>Joint DE</th>
<th>Naïve DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can model anything</td>
<td>Can model only very boring distributions</td>
</tr>
<tr>
<td>No problem to model “C is a noisy copy of A”</td>
<td>Outside Naïve’s scope</td>
</tr>
<tr>
<td>Given 100 records and more than 6 Boolean attributes will screw up badly</td>
<td>Given 100 records and 10,000 multivalued attributes will be fine</td>
</tr>
</tbody>
</table>
Empirical Results: “Hopeless”

The “hopeless” dataset consists of 40,000 records and 21 Boolean attributes called a, b, c, ..., u. Each attribute in each record is generated 50-50 randomly as 0 or 1.

Despite the vast amount of data, “Joint” overfits hopelessly and does much worse.

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Parameters</th>
<th>LogLike</th>
<th>Average test set log probability during 10 folds of k-fold cross-validation*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>joint</td>
<td>submodel=gauss</td>
<td>-272625</td>
<td>+/- 301.109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gausstype=general</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model2</td>
<td>naive</td>
<td>submodel=gauss</td>
<td>-68225.6</td>
<td>+/- 0.554747</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gausstype=general</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Empirical Results: “Logical”

The “logical” dataset consists of 40,000 records and 4 Boolean attributes called a, b, c, d where a, b, c are generated 50-50 randomly as 0 or 1. \( D = A^{\sim}C \), except that in 10% of records it is flipped.

The DE learned by “Joint”

The DE learned by “Naive”
Empirical Results: “Logical”

The “logical” dataset consists of 40,000 records and 4 Boolean attributes called a, b, c, d where a, b, c are generated 50-50 randomly as 0 or 1. D = A^~C, except that in 10% of records it is flipped.

The DE learned by “Joint”

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Parameters</th>
<th>LogLike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>joint</td>
<td>submodel=gauss gausstype=general</td>
<td>-9613.79 +/- 26.6781</td>
</tr>
<tr>
<td>Model2</td>
<td>naive</td>
<td>submodel=gauss gausstype=general</td>
<td>-10763.4 +/- 11.0538</td>
</tr>
</tbody>
</table>

Empirical Results: “MPG”

The “MPG” dataset consists of 392 records and 8 attributes.

The DE learned by “Joint”

A tiny part of the DE learned by “Naive”
Empirical Results: “MPG”

The “MPG” dataset consists of 392 records and 8 attributes.

- **Model 1 (Joint)**: submodel=gauss, gausstype=general, LogLike=-472.486, AIC=77.2184
- **Model 2 (Naive)**: submodel=gauss, gausstype=general, LogLike=-257.212, AIC=302246

Empirical Results: “Weight vs. MPG”

Suppose we train only from the “Weight” and “MPG” attributes.

- **Weight**
  - Bad: 0.193878
  - High: 0.408163
- **MPG**
  - Bad: 0.602041
  - Good: 0.397959
  - Weight: 0.57398
  - High: 0.42602
Empirical Results: “Weight vs. MPG”

Suppose we train only from the “Weight” and “MPG” attributes.

```
<table>
<thead>
<tr>
<th>mpg</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>bad</td>
<td>0.193878</td>
</tr>
<tr>
<td>good</td>
<td>0.408163</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>mpg</th>
<th>bad</th>
<th>0.602041</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>0.397959</td>
<td></td>
</tr>
</tbody>
</table>
```

### Empirical Results: “Weight vs. MPG”

“Weight vs. MPG”: The best that Naïve can do

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
<th>LogLike</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>joint</td>
<td>-44.3562 + 2.27547</td>
</tr>
<tr>
<td>submodel=gauss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gausstype=general</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model2</td>
<td>naive</td>
<td>-53.2231 + 0.610411</td>
</tr>
<tr>
<td>submodel=gauss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gausstype=general</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The DE learned by “Naïve”.

The light color shades denote predicted densities. The dark shades are real data.

“Naïve”
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: \( P(E_1|E_2) \)  Automatic Doctor / Help Desk etc
  - Ingredient for Bayes Classifiers

Bayes Classifiers

- A formidable and sworn enemy of decision trees

![Bayes Classifiers Diagram](image)
How to build a Bayes Classifier

• Assume you want to predict output $Y$ which has arity $n_Y$ and values $v_1, v_2, \ldots, v_{n_Y}$.
• Assume there are $m$ input attributes called $X_1, X_2, \ldots, X_m$.
• Break dataset into $n_Y$ smaller datasets called $DS_1, DS_2, \ldots, DS_{n_Y}$.
• 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 $P(X_1, X_2, \ldots, X_m \mid Y = v_i)$
How to build a Bayes Classifier

- Assume you want to predict output \( Y \) which has arity \( n_Y \) and values \( v_1, v_2, \ldots, v_{n_Y} \).
- Assume there are \( m \) input attributes called \( X_1, X_2, \ldots, X_m \).
- Break dataset into \( n_Y \) smaller datasets called \( D_{S_1}, D_{S_2}, \ldots, D_{S_{n_Y}} \).
- Define \( D_{S_i} = \) Records in which \( Y = v_i \).
- For each \( D_{S_i} \), learn Density Estimator \( M_i \) to model the input distribution among the \( Y = v_i \) records.
- \( M_i \) estimates \( P(X_1, X_2, \ldots, X_m | Y = v_i) \).

Idea: When a new set of input values \( (X_1 = u_1, X_2 = u_2, \ldots, X_m = u_m) \) come along to be evaluated predict the value of \( Y \) that makes \( P(X_1, X_2, \ldots, X_m | Y = v_i) \) most likely

\[
Y_{\text{predict}} = \arg\max_v P(X_1 = u_1, \ldots, X_m = u_m | Y = v)
\]

Is this a good idea?

This is a Maximum Likelihood classifier.
It can get silly if some \( Y \)s are very unlikely.
How to build a Bayes Classifier

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

$$Y_{\text{predict}} = \arg \max_v P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

Is this a good idea?

Much Better Idea

Terminology

- MLE (Maximum Likelihood Estimator):

$$Y_{\text{predict}} = \arg \max_v P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)$$

- MAP (Maximum A-Posteriori Estimator):

$$Y_{\text{predict}} = \arg \max_v P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$
Getting what we need

\[ Y_{\text{predict}} = \arg \max_v P(Y = v \mid X_1 = u_1 \cdots X_m = u_m) \]

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 $P(X_1, X_2, \ldots, X_m \mid Y=v)$. 
3. Estimate $P(Y=v)$ as fraction of records with $Y=v$. 
4. For a new prediction:

$$Y^{\text{predict}} = \arg\max_v P(Y = v \mid X_1 = u_1 \cdots X_m = u_m)$$

$$= \arg\max_v P(X_1 = u_1 \cdots X_m = u_m \mid Y = v) P(Y = v)$$
Joint Density Bayes Classifier

\[ Y^{\text{predict}} = \arg\max_{Y} P(X_1 = u_1, \ldots X_m = u_m \mid Y = \nu) P(Y = \nu) \]

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

\[ Y^{\text{predict}} = \text{the most common value of } Y \text{ among records in which } X_1 = u_1, X_2 = u_2, \ldots X_m = u_m. \]

Note that if no records have the exact set of inputs \( X_1 = u_1, X_2 = u_2, \ldots X_m = u_m \) then \( P(X_1, X_2, \ldots X_m \mid Y = \nu_i) = 0 \) for all values of \( Y \).

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

Joint BC Results: “Logical”

The “logical” dataset consists of 40,000 records and 4 Boolean attributes called \( a, b, c, d \) where \( a, b, c \) are generated 50-50 randomly as 0 or 1. \( D = A \land \neg C, \) except that in 10% of records it is flipped.

The Classifier learned by “Joint BC”
Joint BC Results: “All Irrelevant”
The “all irrelevant” dataset consists of 40,000 records and 15 Boolean attributes called a, b, c, d..o where a, b, c are generated 50-50 randomly as 0 or 1. $v$ (output) = 1 with probability 0.75, 0 with prob 0.25

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Parameters</th>
<th>FracRight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>bayesclass</td>
<td>density=joint</td>
<td>0.70425</td>
</tr>
<tr>
<td></td>
<td></td>
<td>submodel=gauss</td>
<td>+/− 0.0083537</td>
</tr>
<tr>
<td></td>
<td></td>
<td>gausstype=general</td>
<td></td>
</tr>
</tbody>
</table>

Naïve Bayes Classifier

\[ Y_{\text{predict}} = \arg\max_v P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v) \]

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

\[ Y_{\text{predict}} = \arg\max_v P(Y = v) \prod_{j=1}^{n_v} P(X_j = u_j \mid Y = v) \]
Naïve Bayes Classifier

\[ Y^{\text{predict}} = \arg\max_v P(X_1 = u_1 \cdots X_m = u_m \mid Y = v)P(Y = v) \]

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

\[ Y^{\text{predict}} = \arg\max_v P(Y = v) \prod_{j=1}^{n_y} P(X_j = u_j \mid Y = v) \]

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

\[ Y^{\text{predict}} = \arg\max_v \left( \log P(Y = v) + \sum_{j=1}^{n_y} \log P(X_j = u_j \mid Y = v) \right) \]

BC Results: “XOR”

The “XOR” dataset consists of 40,000 records and 2 Boolean inputs called a and b, generated 50-50 randomly as 0 or 1. c (output) = a XOR b

The Classifier learned by “Joint BC”

Model bayesclass Parameters FracRight
\[
\begin{align*}
\text{Model} &\quad \text{bayesclass} \\
\text{density} &\quad \text{joint} \\
\text{submodel} &\quad \text{gauss} \\
\text{gaustype} &\quad \text{general} \\
\frac{1}{0} &\quad +/- \quad 0 \\
\end{align*}
\]

The Classifier learned by “Naive BC”

Model2 bayesclass Parameters
\[
\begin{align*}
\text{Model} &\quad \text{bayesclass} \\
\text{density} &\quad \text{naive} \\
\text{submodel} &\quad \text{gauss} \\
\text{gaustype} &\quad \text{general} \\
\text{0.500125} &\quad +/- \quad 0.00529626 \\
\end{align*}
\]
Naive BC Results: “Logical”

The “logical” dataset consists of 40,000 records and 4 Boolean attributes called a, b, c, d where a, b, c are generated 50-50 randomly as 0 or 1. D = A^~C, except that in 10% of records it is flipped.

This result surprised Andrew until he had thought about it a little.

The data shown in the figure is merely a subsample of the full dataset. The light color shades denote predicted classes. The dark shades are real data.
Naïve BC Results: "All Irrelevant"

The "all irrelevant" dataset consists of 40,000 records and 15 Boolean attributes called a,b,c,d..o where a,b,c are generated 50-50 randomly as 0 or 1. v (output) = 1 with probability 0.75, 0 with prob 0.25

The Classifier learned by "Naive BC"

Name     Model     Parameters                   FracRight
Model1   bayesclass  density=joint submodel=gauss gausstype=general 0.70425 +/- 0.00583537

Model2   bayesclass  density=naive submodel=gauss gausstype=general 0.75056 +/- 0.00281976

BC Results: "MPG": 392 records

The Classifier learned by "Naive BC"

Name     Model     Parameters                   FracRight
Model1   bayesclass  density=joint submodel=gauss gausstype=general 0.885256 +/- 0.0247796

Model2   bayesclass  density=naive submodel=gauss gausstype=general 0.852372 +/- 0.0400495
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. $P(E_1|E_2)$) 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
Interesting Questions

• Suppose you were evaluating NaiveBC, JointBC, and Decision Trees
  • Invent a problem where only NaiveBC would do well
  • Invent a problem where only Dtree would do well
  • Invent a problem where only JointBC would do well
  • Invent a problem where only NaiveBC would do poorly
  • Invent a problem where only Dtree would do poorly
  • Invent a problem where only JointBC would do poorly