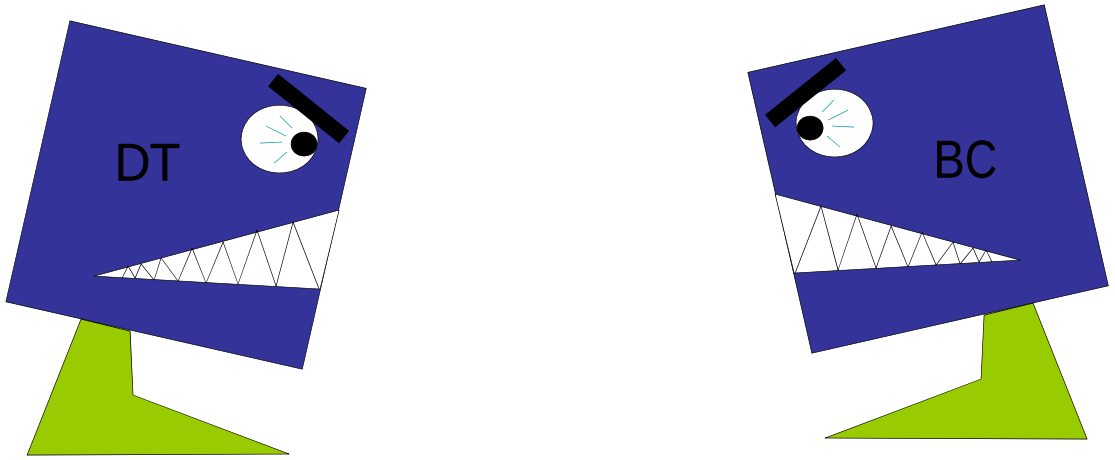
Bayes classifiers

Adapted from Professor Moore’s slides

Bayes Classifiers

- A formidable and sworn enemy of decision trees



How to build a Bayes Classifier

- Assume you want to predict output Y which has arity n_Y and values $v_1, v_2, \ldots v_{ny}$.
- Assume there are m input attributes called $X_1, X_2, \ldots X_m$
- Break dataset into n_Y smaller datasets called $DS_1, DS_2, \ldots DS_{ny}$.
- 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.

How to build a Bayes Classifier

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

How to build a Bayes Classifier

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

- 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 $P(X_1, X_2, \ldots X_m \mid Y=v_i)$ 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 $v_1, v_2, \ldots v_{ny}$.
- 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
- M_i estimates P(X_1, X_2, … X_m | $Y=v_i$)

This is a Maximum Likelihood classifier.

It can get silly if some Ys are very unlikely

- Idea: When a new set of input values ($X_1 = u_1$, $X_2 = u_2$, …. $X_m = u_m$) come along to be evaluated predict the value of Y that makes P(X_1, X_2, … 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 $v_1, v_2, \ldots v_{ny}$.
- Assume there are m input attributes called
- Break dataset into n_Y smaller datasets cal
- 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, … X_m | $Y=v_i$)

Much Better Idea

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

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

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:

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

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:

$$Y^{\text{predict}} = \operatorname{argmax}_v P(Y = v \mid X_1 = \cdots$$

$$= \operatorname{argmax}_v P(X_1 = u_1 \cdots X_m = u_m$$

We can use our favorite Density Estimator here.

Right now we have two options:

- Joint Density Estimator
- Naïve Density Estimator

Joint Density Bayes Classifier

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

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

$Y^{predict}$ = the class containing most 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

Example

X1	X2	X3	Y
0	0	1	0
0	1	0	0
1	1	0	0
0	0	1	1
1	1	1	1
0	0	1	1
1	1	0	1

Example: Cont

$$P(Y=0)=3/7 \qquad P(Y=1)=4/7$$

$$P(X_1=0, X_2=0, X_3=1/Y=0)=1/3$$

$$P(X_1=0, X_2=0, X_3=1/Y=1)=1/2$$

X1=0,X2=0, x3=1 sera asignado a la clase 1 . Notar tambien que en esta clase el record (0,0,1) aparece mas veces que en la clase 0.

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

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

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

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

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

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

Example: Cont

$$P(Y=0)=3/7 \qquad P(Y=1)=4/7$$

$$P(X_1=0,X_2=0,X_3=1/Y=0)=P(X_1=0/Y=0)P(X_2=0/Y=0)P(X_3=1/Y=0)=$$

$$(2/3)(1/3)(1/3)=2/27$$

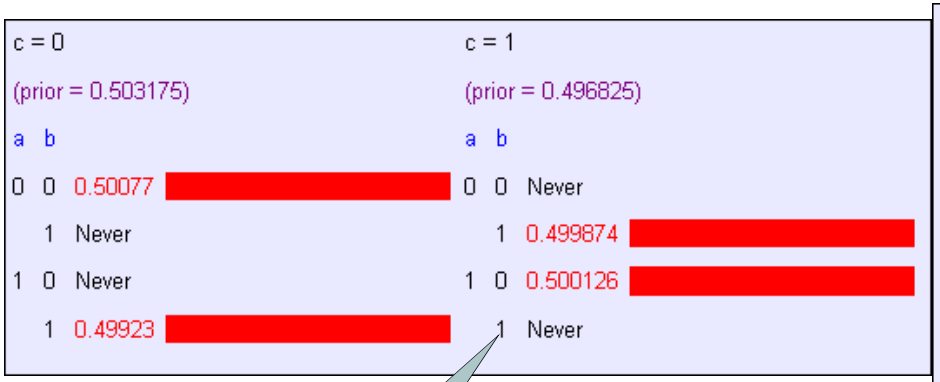
$$P(X_1=0,X_2=0,X_3=1/Y=1)=P(X_1=0/Y=1)P(X_2=0/Y=1)P(X_3=1/Y=1)=$$

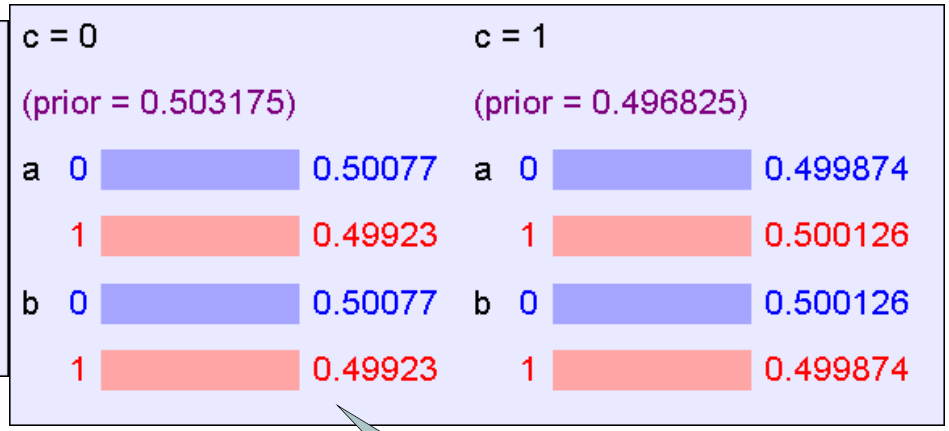
$$(2/4)(2/4)(3/4)=3/16$$

X1=0,X2=0, x3=1 sera asignado a la clase 1

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



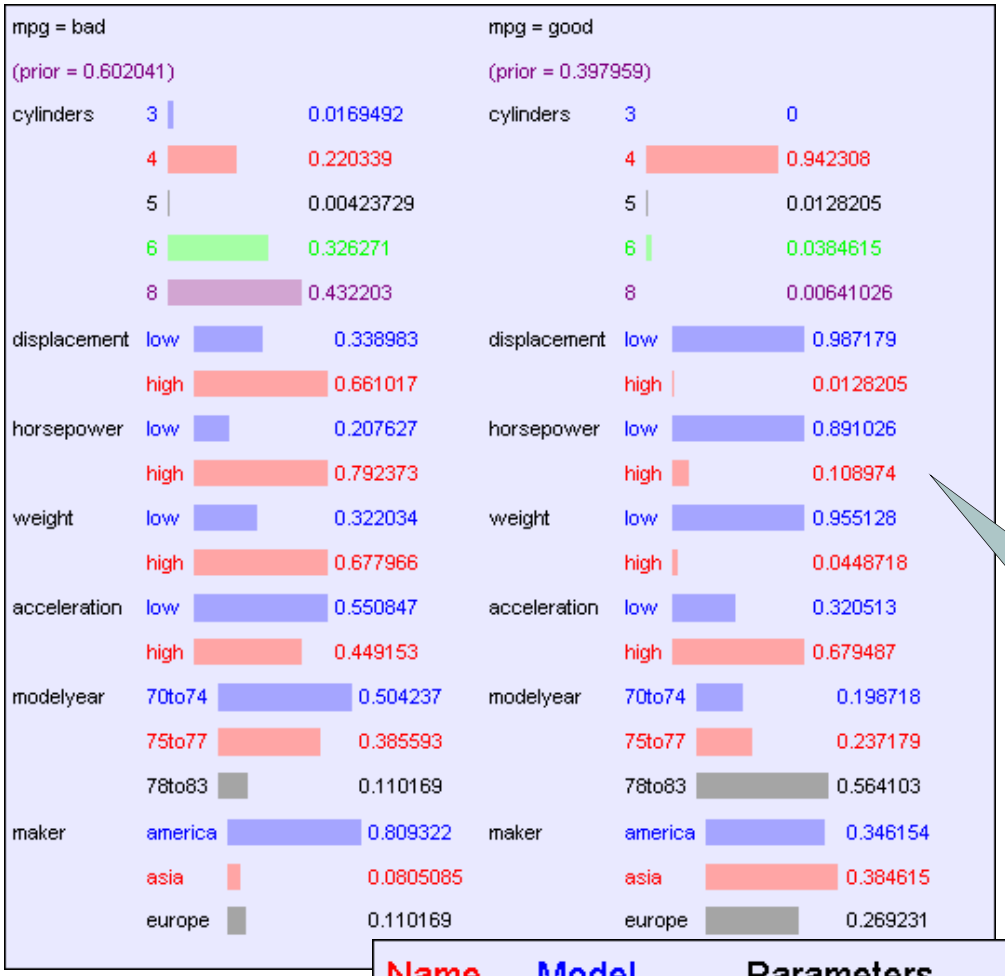


The Classifier learned by “Joint BC”

The Classifier learned by “Naive BC”

	del	Parameters	FracRight		
	yesclass	density=joint submodel=gauss gausstype=general	1	+/-	0
Model2	bayesclass	density=naive submodel=gauss gausstype=general	0.500125	+/-	0.00529626

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

Naïve Bayes classifier

Naïve Bayes classifier puede ser aplicado cuando hay predictoras continuas, pero hay que aplicar previamente un metodo de discretizacion tal como: Usando intervalos de igual ancho, usando intervalos con igual frecuencia, ChiMerge,1R, Discretizacion usando el metodo de la entropia con distintos criterios de parada, Todos ellos estan disponible en la libreria dprep(ver disc.mentr, disc.ew, disc.ef, etc) .

La libreria e1071 de R contiene una funcion **naiveBayes** que calcula el clasificador naïve Bayes. Si la variable es continua asume que sigue una distribucion Gaussiana.

Naive Bayes para Bupa

Sin discretizar

```
> a=naiveBayes(V7~.,data=bupa)
> pred=predict(a,bupa[,-7],type="raw")
> pred1=max.col(pred)
> table(pred1,bupa[,7])

pred1   1   2
    1 112 119
    2  33  81
> error=152/345
[1] 0.4405797
```

Discretizando con el metodo de la entropia

```
> dbupa=disc.mentr(bupa,1:7)
> b=naiveBayes(V7~.,data=dbupa)
> pred=predict(b,dbupa[,-7])
> table(pred,dbupa[,7])

pred   1   2
   1  79  61
   2  66 139
> error1=127/345
[1] 0.3681159
```

Naïve Bayes for Diabetes

Sin Descritizar

```
> a=naiveBayes(V9~.,data=diabetes)
> pred=predict(a,diabetes[,-9],type="raw")
> pred1=max.col(pred)
> table(pred1,diabetes[,9])

pred1   1   2
    1 421 104
    2  79 164
> error=(79+104)/768
[1] 0.2382813
```

Discretizando

```
> ddiabetes=disc.mentr(diabetes,1:9)
> b=naiveBayes(V9~.,data=ddiabetes)
> pred=predict(b,ddiabetes[,-9])
> table(pred,ddiabetes[,9])

pred   1   2
   1 418  84
   2  82 184
> 166/768
[1] 0.2161458
```

Naïve Bayes using ChiMerge discretization

```
> chibupa=chiMerge(bupa,1:6)
> b=naiveBayes(V7~.,data=chibupa)
> pred=predict(b,chibupa[,-7])
> table(pred,chibupa[,7])
pred   1   2
   1 117  21
   2  28 179
> error=49/345
[1] 0.1420290
> chidiab=chiMerge(diabetes,1:8)
> b=naiveBayes(V9~.,data=chidiab)
> pred=predict(b,chidiab[,-9])
> table(pred,chidiab[,9])
pred   1   2
   1 457  33
   2  43 235
> error=76/768
[1] 0.09895833
```