CSE 4950/6950
Naïve Bayes

Instructor: AKM Kamrul Islam
aislam5@cs.gsu.edu
Dept. of Computer Science
Georgia State University

(Materials are highly adapted from different online sources)

Georgia State University

Bayesian Learning

Boolean random variables: cavity might be true or false

• Discrete random variables: weather might be sunny, rainy, cloudy, snow

–P(Weather=sunny) –P(Weather=rainy)

–P(Weather=cloudy) –P(Weather=snow)

• Continuous random variables: the temperature has continuous values

Sample space: the set of possible outcomes of an experiment.

• Event: a subset of the sample space.

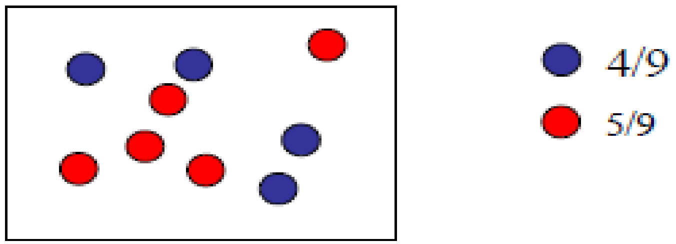
• If *S* is a finite sample space of equally likely outcomes, and *E* is an event, that is, a subset of *S*, then the *probability* of *E* is

$$p(E) = \frac{|E|}{|S|}$$

Probability

- Before the evidence is obtained; prior probability
  - *P(a)* the prior probability that the proposition is true
  - P(cavity)=0.1

- After the evidence is obtained; posterior probability
  - P(a|b)
  - The probability of a given that all we know is b
  - *P(cavity|toothache)=0.8*



Axioms of Probability

- All probabilities are between 0 and 1. For any proposition *a*, $0 \le P(a) \le 1$
- The probability of disjunction is given by

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

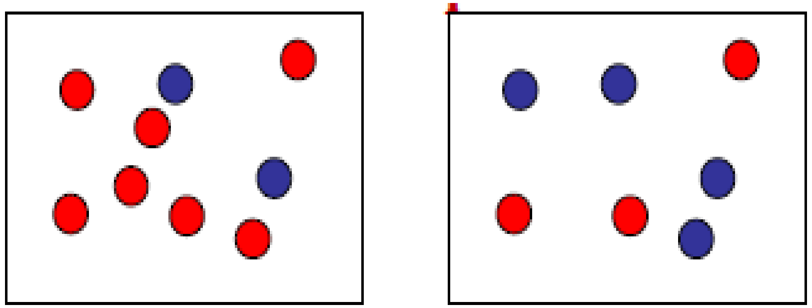
Product Rule

$$P(a \wedge b) = P(a \mid b) P(b)$$

$$P(a \wedge b) = P(b \mid a) P(a)$$

Bayes Rule

$$P(b \mid a) = \frac{P(a \mid b)P(b)}{P(a)}$$



1. Choose one of the two boxes at random.
2. Select one of the balls in this box at random

If a red ball is selected, what is the probability that this ball is from the first box?

$$p(F \mid E) = \frac{p(F \cap E)}{p(E)}$$

E : a red ball is selected
$\bar{E}$: a blue ball is selected
F : a ball is selected from the first box
$\bar{F}$: a ball is selected from the second box

$p(F \mid E) = p(F \cap E)/p(E) = 49/76 \approx 0.645$
$p(E \mid F) = p(E \cap F)/p(F) = 7/9$, $p(F) = p(\bar{F}) = 1/2$
$\rightarrow p(E \cap F) = 7/18$
$p(E \mid \bar{F}) = p(E \cap \bar{F})/p(\bar{F}) = 3/7 \rightarrow p(E \cap \bar{F}) = 3/14$
$E = (E \cap F) \cup (E \cap \bar{F}) \rightarrow p(E) = p(E \cap F) + p(E \cap \bar{F}) = 38/63$

Bayes Theorem

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- *P(h)* = prior probability of hypothesis h
- *P(D)* = prior probability of training data *D*
- *P(h|D)* = probability of *h* given *D*
- *P(D|h)* = probability of *D* given *h*

Choosing Hypotheses

- Generally want the most probable hypothesis given the training data
- **Maximum a posteriori** hypothesis h_{MAP}:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(h|D) \\ &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\ &= \arg\max_{h \in H} P(D|h)P(h) \end{aligned}$$

Georgia State University

Maximum Likelihood (ML)

- If assume $P(h_i) = P(h_j)$ for all h_i and h_j, then can further simplify, and choose the
- **Maximum likelihood**(ML) hypothesis

$$h_{ML} = \arg\max_{h_i \in H} P(D|h_i)$$

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result (+) in only 98% of the cases in which the disease is actually present, and a correct negative result (-) in only 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

$$P(cancer) = 0.008 \qquad P(\neg cancer) = 0.992$$

$$P(+|cancer) = 0.98 \qquad P(-|cancer) = 0.02$$

$$P(+|\neg cancer) = 0.03 \qquad P(-|\neg cancer) = 0.97$$

$$P(+|cancer) \cdot P(cancer) = 0.98 \cdot 0.008 = 0.0078$$

$$P(+|\neg cancer) \cdot P(\neg cancer) = 0.03 \cdot 0.992 = 0.0298$$

$$h_{MAP} = \neg cancer$$

Normalization

$$\frac{0.0078}{0.0078+0.0298}=0.20745 \quad \frac{0.0298}{0.0078+0.0298}=0.79255$$

The result of Bayesian inference depends strongly on the prior probabilities, which must be available in order to apply the method

Brute-Force Bayes Concept Learning

- For each hypothesis h in H, calculate the posterior probability

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- Output the hypothesis h_{MAP} with the highest posterior probability

$$h_{MAP} =_{h\in H} P(h|D)$$

Given no prior knowledge that one hypothesis is more likely than another, what values should we specify for *P(h)*?

- What choice shall we make for *P(D|h)* ?
- The algorithm may require significant computation, it applies Bayes theorem to each hypothesis in H to calculate P(h|D)

Essential Probability Concepts

- Marginalization: $P(B) = \sum_{v \in \text{values}(A)} P(B \wedge A = v)$
- Conditional Probability: $P(A \mid B) = \dfrac{P(A \wedge B)}{P(B)}$
- Bayes' Rule: $P(A \mid B) = \dfrac{P(B \mid A) \times P(A)}{P(B)}$
- Independence:

$$
\begin{aligned}
A \perp\!\!\!\perp B \quad &\leftrightarrow \quad P(A \wedge B) = P(A) \times P(B) \\
&\leftrightarrow \quad P(A \mid B) = P(A) \\
A \perp\!\!\!\perp B \mid C \quad &\leftrightarrow \quad P(A \wedge B \mid C) = P(A \mid C) \times P(B \mid C)
\end{aligned}
$$

Density Estimation

Recall the Joint Distribution...

	alarm		¬alarm	
	earthquake	¬earthquake	earthquake	¬earthquake
burglary	0.01	0.08	0.001	0.009
¬burglary	0.01	0.09	0.01	0.79

How Can We Obtain a Joint Distribution?

Option 1: Elicit it from an expert human

Option 2: Build it up from simpler probabilistic facts

- e.g, if we knew

$$P(a) = 0.7 \qquad P(b|a) = 0.2 \qquad P(b|\neg a) = 0.1$$

then, we could compute $P(a \wedge b)$

Option 3: Learn it from data...

Based on slide by Andrew Moore

Learning a Joint Distribution

Step 1:
Build a JD table for your attributes in which the probabilities are unspecified

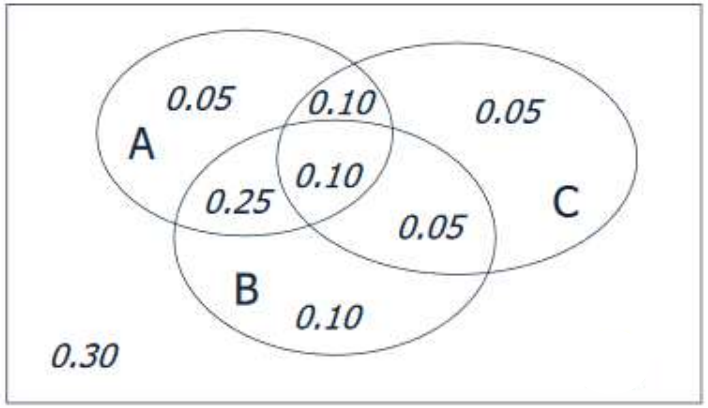
A	B	C	Prob
0	0	0	?
0	0	1	?
0	1	0	?
0	1	1	?
1	0	0	?
1	0	1	?
1	1	0	?
1	1	1	?

Step 2:
Then, 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



Slide © Andrew Moore

Recipe for making a joint distribution of dvariables:1.Make a truth table listing all combinations of values of your variables (if there are d Boolean variables then the table will have 2^D 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 of Learning a Joint PD

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

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

Inferring Probabilities from the Joint

	alarm		¬alarm	
	earthquake	¬earthquake	earthquake	¬earthquake
burglary	0.01	0.08	0.001	0.009
¬burglary	0.01	0.09	0.01	0.79

$$P(alarm) = \sum_{b,e} P(alarm \wedge \text{Burglary} = b \wedge \text{Earthquake} = e)$$

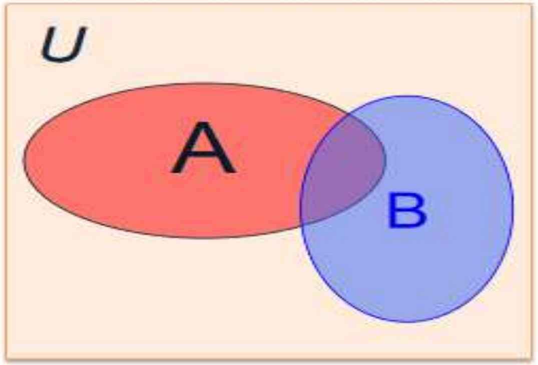
$$= 0.01 + 0.08 + 0.01 + 0.09 = 0.19$$

$$P(burglary) = \sum_{a,e} P(\text{Alarm} = a \wedge burglary \wedge \text{Earthquake} = e)$$

$$= 0.01 + 0.08 + 0.001 + 0.009 = 0.1$$

Conditional Probability

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



What if we already know that B is true?

That knowledge changes the probability of A

- Because we know we're in a world where B is true

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

$$P(A \wedge B) = P(A \mid B) \times P(B)$$

Example: Conditional Probabilities

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

$$P(A \wedge B) = P(A \mid B) \times P(B)$$

P(Alarm, Burglary) =

	alarm	¬alarm
burglary	0.09	0.01
¬burglary	0.1	0.8

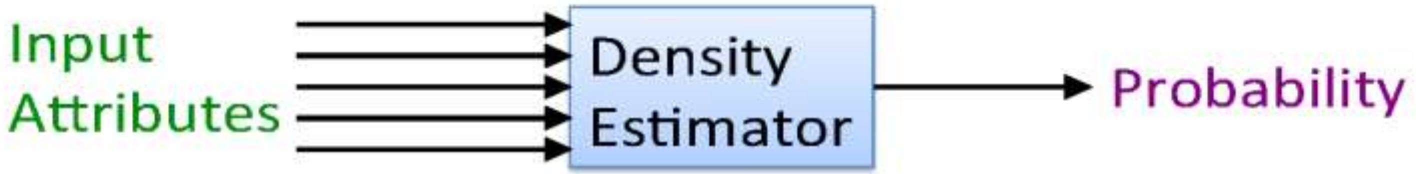
P(burglary | alarm) = P(burglary ∧ alarm) / P(alarm)
= 0.09 / 0.19 = 0.47

P(alarm | burglary) = P(burglary ∧ alarm) / P(burglary)
= 0.09 / 0.1 = 0.9

P(burglary ∧ alarm) = P(burglary | alarm) P(alarm)
= 0.47 * 0.19 = 0.09

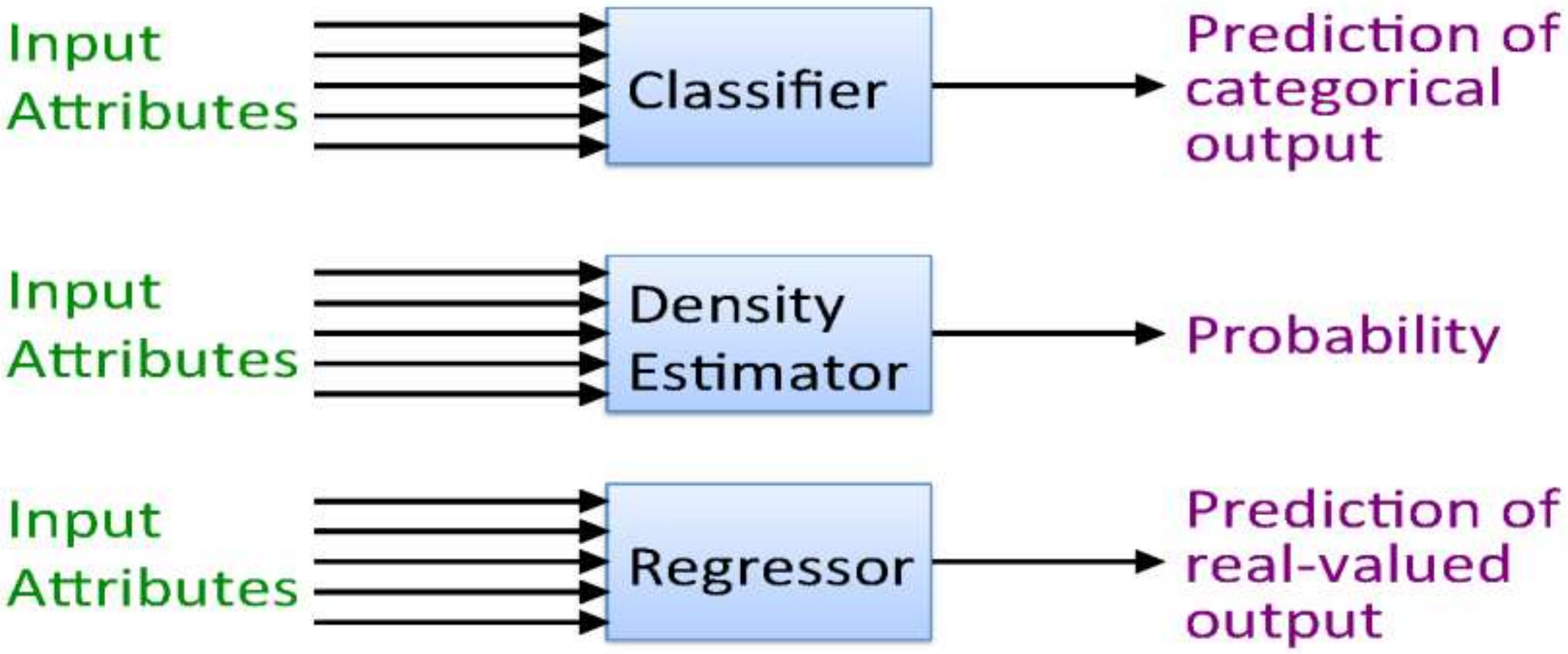
Density Estimation

- Our joint distribution learner is an example of something called **Density Estimation**
- A Density Estimator learns a mapping from a set of attributes to a probability



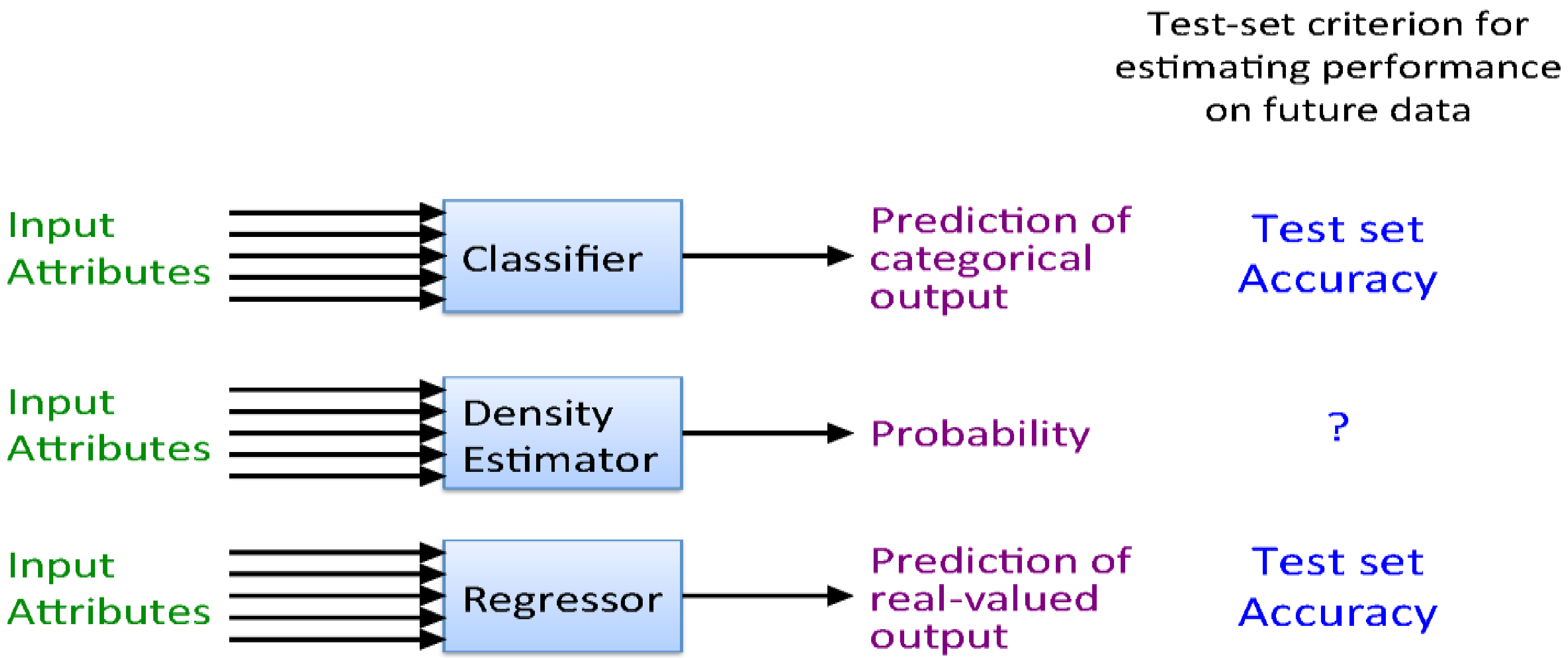
Density Estimation

Compare it against the two other major kinds of models:



Slide © Andrew Moore

Evaluating Density Estimation



Slide © Andrew Moore

Evaluating a Density Estimator

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

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

- The density estimator can also tell you how likely the dataset is:
  - Under the assumption that all records were independently generated from the Density Estimator's JD (that is, i.i.d.)

$$\hat{P}(\underbrace{\mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \ldots \wedge \mathbf{x}_n}_{\text{dataset}} \mid M) = \prod_{i=1}^{n} \hat{P}(\mathbf{x}_i \mid M)$$

Slide by Andrew Moore

Example Small Dataset: Miles Per Gallon

From the UCI repository (thanks to Ross Quinlan)

- 192 records in the 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
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

mpg	modelyear	maker	
bad	70to74	america	0.27551
		asia	0.0255102
		europe	0.0153061
	75to77	america	0.153061
		asia	0.0255102
		europe	0.0357143
	78to83	america	0.0561224
		asia	Never
		europe	Never
good	70to74	america	0.0102041
		asia	0.0306122
		europe	0.0459184
	75to77	america	0.0306122
		asia	0.0408163
		europe	0.0357143
	78to83	america	0.112245
		asia	0.0714286
		europe	0.0357143

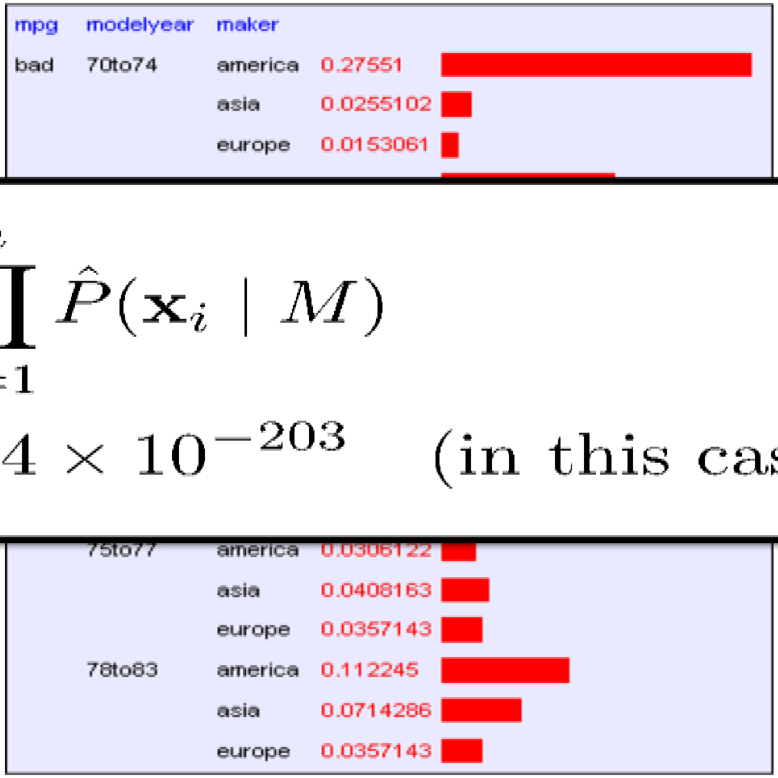
Slide by Andrew Moore

Example Small Dataset: Miles Per Gallon

From the UCI repository (thanks to Ross Quinlan)

- 192 records in the training set

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



$$
\hat{P}(\text{dataset} \mid M) = \prod_{i=1}^{n} \hat{P}(\mathbf{x}_i \mid M)
$$

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

Slide by Andrew Moore

Log Probabilities

- For decent sized data sets, this product will underflow

$$\hat{P}(\text{dataset} \mid M) = \boxed{\prod_{i=1}^{n}} \hat{P}(\mathbf{x}_i \mid M)$$

- Therefore, since probabilities of datasets get so small, we usually use log probabilities

$$\log \hat{P}(\text{dataset} \mid M) = \log \prod_{i=1}^{n} \hat{P}(\mathbf{x}_i \mid M) = \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i \mid M)$$

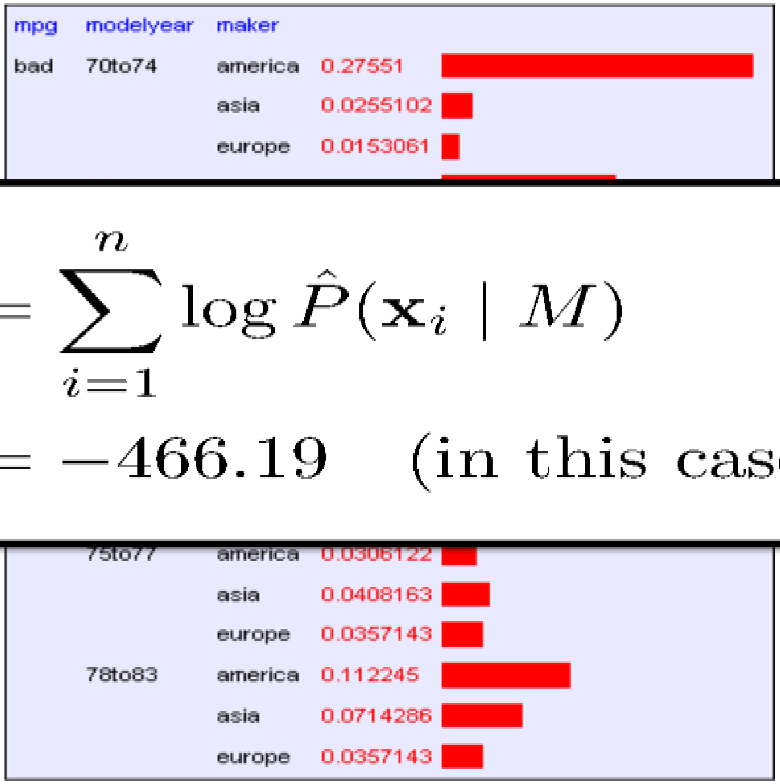
Based on slide by Andrew Moore

Example Small Dataset: Miles Per Gallon

From the UCI repository (thanks to Ross Quinlan)

- 192 records in the training set

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



$$\log \hat{P}(\text{dataset} \mid M) = \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i \mid M)$$

$$= -466.19 \quad \text{(in this case)}$$

Slide by Andrew Moore

Pros/Cons of the Joint Density Estimator

The Good News:

- We can 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
  - Ingredient for Bayes Classifiers (coming very soon...)

The Bad News:

- Density estimation by directly learning the joint is trivial, mindless, and dangerous

Slide by Andrew Moore

The Joint Density Estimator on 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 much 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!

Slide by Andrew Moore

Overfitting Density Estimators

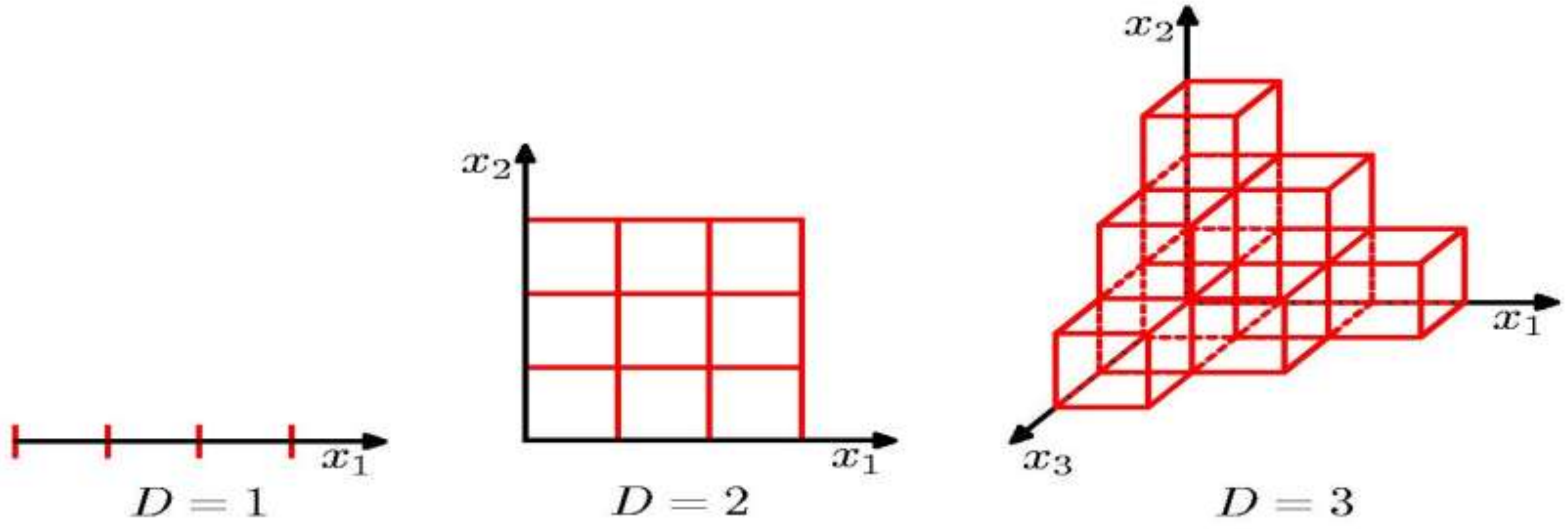
If this ever happens, the joint PDE learns there are certain combinations that are impossible

mpg	modelyear	maker	
bad	70to74	america	0.27551
		asia	0.0255102
		europe	0.0153061
	75to77	america	0.153061
		asia	0.0255102
		europe	0.0357143
	78to83	america	0.0561224
		asia	Never
		europe	Never
good	70to74	america	0.0102041
		asia	0.0306122

$$
\begin{aligned}
\log \hat{P}(\text{dataset} \mid M) &= \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i \mid M) \\
&= -\infty \quad \text{if for any } i,\ \hat{P}(\mathbf{x}_i \mid M) = 0
\end{aligned}
$$

Slide by Andrew Moore

Curse of Dimensionality



Slide by Christopher Bishop

The Joint Density Estimator on a Test Set

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

- The only reason that the test set didn't score -∞ is that the code was hard-wired to always predict a probability of at least $1/10^{20}$

We need Density Estimators that are less prone to overfitting...

Slide by Andrew Moore

The Naïve Bayes Classifier

Bayes' Rule

- Recall Baye's Rule:

$$P(\text{hypothesis} \mid \text{evidence}) = \frac{P(\text{evidence} \mid \text{hypothesis}) \times P(\text{hypothesis})}{P(\text{evidence})}$$

- Equivalently, we can write:

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k)P(X = \mathbf{x}_i \mid Y = y_k)}{P(X = \mathbf{x}_i)}$$

where X is a random variable representing the evidence and Y is a random variable for the label

- This is actually short for:

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k)P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d} \mid Y = y_k)}{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d})}$$

where X_j denotes the random variable for the j^{th} feature

Naïve Bayes Classifier

Idea: Use the training data to estimate

$P(X \mid Y)$ and $P(Y)$.

Then, use Bayes rule to infer $P(Y|X_{\text{new}})$ for new data

---

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{\overbrace{\boxed{P(Y = y_k)}}^{\text{Easy to estimate from data}}\overbrace{\boxed{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d} \mid Y = y_k)}}^{\text{Impractical, but necessary}}}{\underbrace{\boxed{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d})}}_{\text{Unnecessary, as it turns out}}}$$

- Recall that estimating the joint probability distribution $P(X_1, X_2, \ldots, X_d \mid Y)$ is not practical

Naïve Bayes Classifier

Problem: estimating the joint PD or CPD isn't practical

– Severely overfits, as we saw before

However, if we make the assumption that the attributes are independent given the class label, estimation is easy!

$$P(X_1, X_2, \ldots, X_d \mid Y) = \prod_{j=1}^{d} P(X_j \mid Y)$$

- In other words, we assume all attributes are *conditionally independent* given Y
- Often this assumption is violated in practice, but more on that later…

Naive Bayesian Classifier

- The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors.
- A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets.
- Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Algorithm

- Bayes theorem provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$.
- Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- P(c) is the prior probability of *class*.
- P(x|c) is the likelihood which is the probability of *predictor* given *class*.
- P(x) is the prior probability of *predictor*.

In ZeroR model there is no predictor, in OneR model we try to find the single best predictor, naive Bayesian includes all predictors using Bayes' rule and the independence assumptions between predictors.

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = ?$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = ?$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

...

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

...

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\text{play}) = 3/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 1$

$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\text{play}) = 1/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = 0$

$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P}(\text{play}) = 3/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 1$

$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = 2/3$

$\ldots$

$\mathrm{P}(\neg\text{play}) = 1/4$

$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = 0$

$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$$\mathrm{P}(\text{play}) = 3/4$$
$$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 1$$
$$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = 2/3$$
$$\ldots$$

$$\mathrm{P}(\neg\text{play}) = 1/4$$
$$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = 0$$
$$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = ?$$
$$\ldots$$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = 1$

$\ldots$

GeorgiaStateUniversity

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	**Temp**	**Humid**	**Wind**	**Water**	**Forecast**	**Play?**
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = 2/3$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 0$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = 1$

$\ldots$

Example

We use the same simple Weather dataset here.

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

- The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target.
- Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

$$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(No) = 5/14 = 0.36$$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

The likelihood tables for all four predictors.

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

In this example we have 4 inputs (predictors). The final posterior probabilities can be standardized between 0 and 1

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

The zero-frequency problem

Add 1 to the count for every attribute value-class combination (*Laplace estimator*) when an attribute value (*Outlook=Overcast*) doesn’t occur with every class value (*Play Golf=no*).

Numerical Predictors

- Numerical variables need to be transformed to their categorical counterparts (binning) before constructing their frequency tables.
- The other option we have is using the distribution of the numerical variable to have a good guess of the frequency. For example, one common practice is to assume normal distributions for numerical variables.
- The probability density function for the normal distribution is defined by two parameters (mean and standard deviation).

		Humidity	Mean	StDev
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

$$P(\text{humidity} = 74 \mid \text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74 \mid \text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$ Mean

$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5}$$ Standard deviation

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$ Normal distribution

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

- Fix by using Laplace smoothing:
  - Adds 1 to each count

$$P(X_j = v \mid Y = y_k) = \frac{c_v + 1}{\sum_{v' \in \text{values}(X_j)} c_{v'} + |\text{values}(X_j)|}$$

where

- c_v is the count of training instances with a value of v for attribute j and class label y_k
- $|\text{values}(X_j)|$ is the number of values X_j can take on

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 1/3$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = 3/5$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 1/3$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$$\mathrm{P}(\text{play}) = 3/4$$

$$\mathrm{P}(\text{Sky} = \text{sunny} \mid \text{play}) = 4/5$$

$$\mathrm{P}(\text{Humid} = \text{high} \mid \text{play}) = 3/5$$

$$\ldots$$

$$\mathrm{P}(\neg\text{play}) = 1/4$$

$$\mathrm{P}(\text{Sky} = \text{sunny} \mid \neg\text{play}) = 1/3$$

$$\mathrm{P}(\text{Humid} = \text{high} \mid \neg\text{play}) = 2/3$$

$$\ldots$$

Using the Naïve Bayes Classifier

- Now, we have

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_{i,j} \mid Y = y_k)}{\boxed{P(X = \mathbf{x}_i)}}$$

This is constant for a given instance, and so irrelevant to our prediction

  – In practice, we use log-probabilities to prevent underflow

- To classify a new point $\mathbf{x}$,

$$h(\mathbf{x}) = \arg\max_{y_k} \; P(Y = y_k) \prod_{j=1}^{d} P(X_j = \underbrace{x_j}_{j^{\text{th}} \text{ attribute value of } \mathbf{x}} \mid Y = y_k)$$

$$= \arg\max_{y_k} \; \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

The Naïve Bayes Classifier Algorithm

- For each class label y_k
  - Estimate $\mathrm{P}(\mathrm{Y} = y_k)$ from the data
  - For each value $x_{i,j}$ of each attribute X_i
    - Estimate $\mathrm{P}(\mathrm{X}_i = x_{i,j} \mid \mathrm{Y} = y_k)$

- Classify a new point via:

$$h(\mathbf{x}) = \arg\max_{y_k} \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite it

Computing Probabilities (Not Just Predicting Labels)

- NB classifier gives predictions, not probabilities, because we ignore $\mathrm{P}(\mathrm{X})$ (the denominator in Bayes rule)
- Can produce probabilities by:
  - For each possible class label y_k, compute

$$\underbrace{\tilde{P}(Y = y_k \mid X = \mathbf{x})}_{\substack{\text{This is the numerator of Bayes rule, and is} \\ \text{therefore off the true probability by a factor} \\ \text{of } \alpha \text{ that makes probabilities sum to 1}}} = P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_j \mid Y = y_k)$$

  - α is given by $\alpha = \dfrac{1}{\sum_{k=1}^{\#classes} \tilde{P}(Y = y_k \mid X = \mathbf{x})}$
  - Class probability is given by

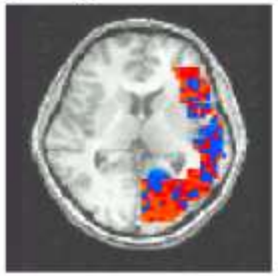
$$P(Y = y_k \mid X = \mathbf{x}) = \alpha \tilde{P}(Y = y_k \mid X = \mathbf{x})$$

Naïve Bayes Applications

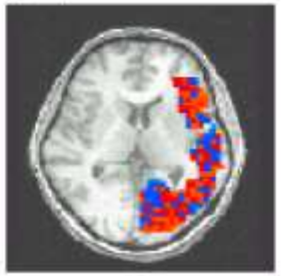
- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?

- Classifying mental states

Learning $P(\mathrm{BrainActivity} \mid \mathrm{WordCategory})$



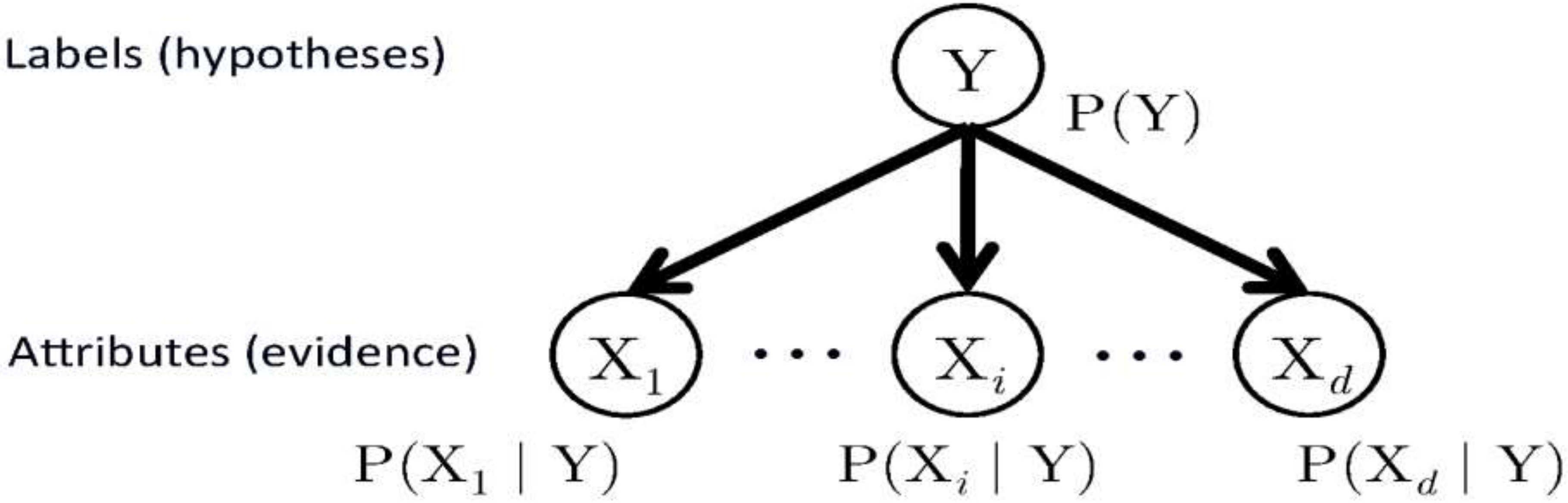
People Words



Animal Words

Pairwise Classification
Accuracy: 85%

The Naïve Bayes Graphical Model

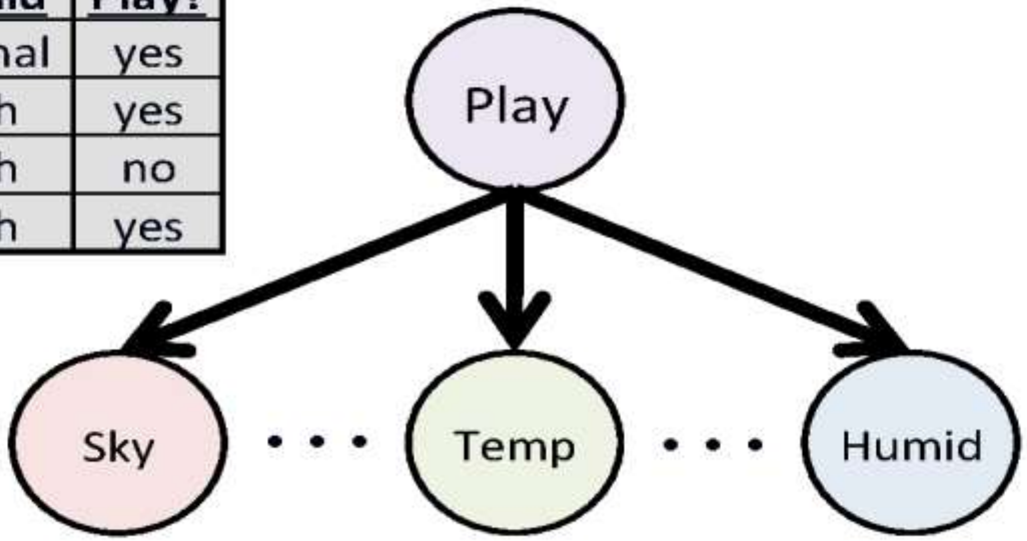


- Nodes denote random variables
- Edges denote dependency
- Each node has an associated conditional probability table (CPT), conditioned upon its parents

Example NB Graphical Model

Data:

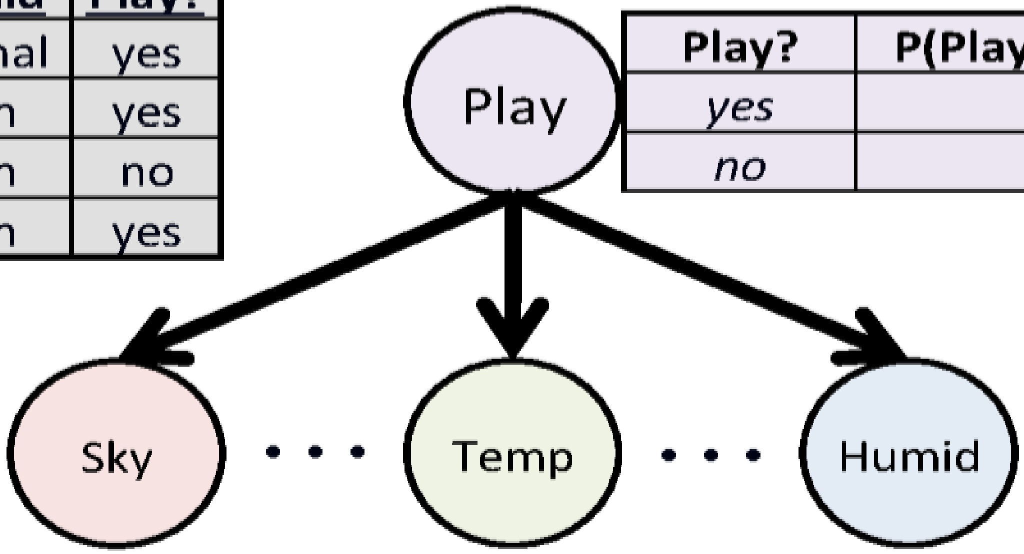
Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes

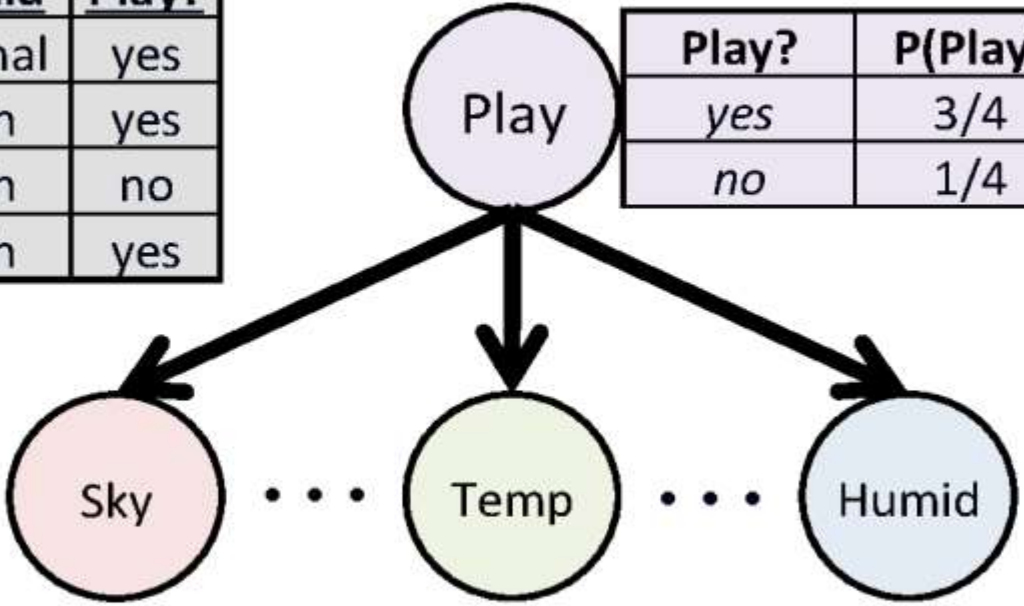


Play?	P(Play)
yes	
no	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes

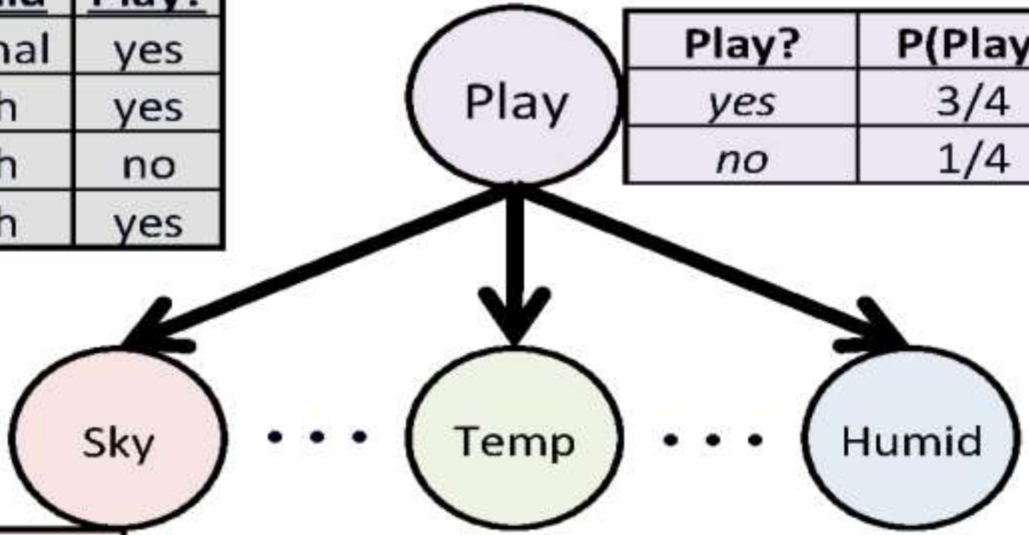


Play?	P(Play)
yes	3/4
no	1/4

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



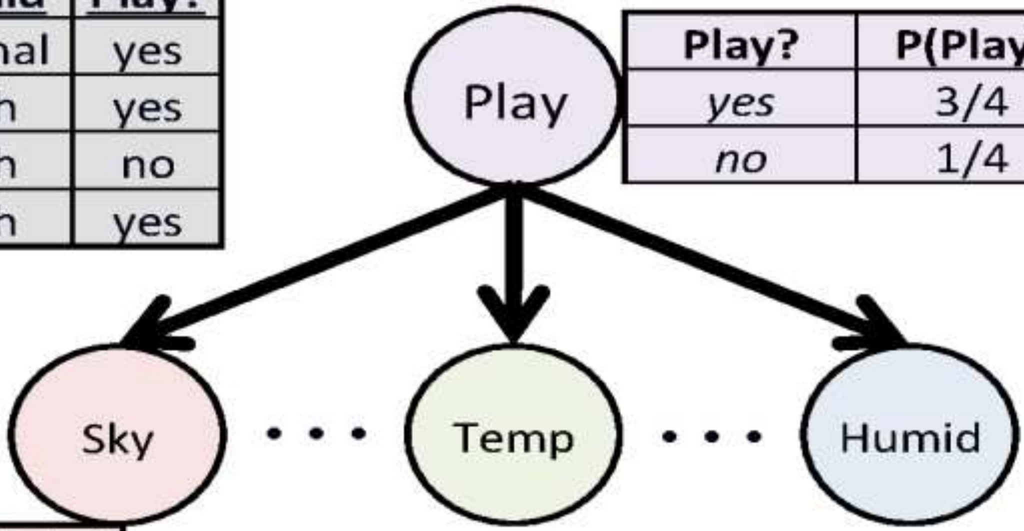
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	
rainy	*yes*	
sunny	*no*	
rainy	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



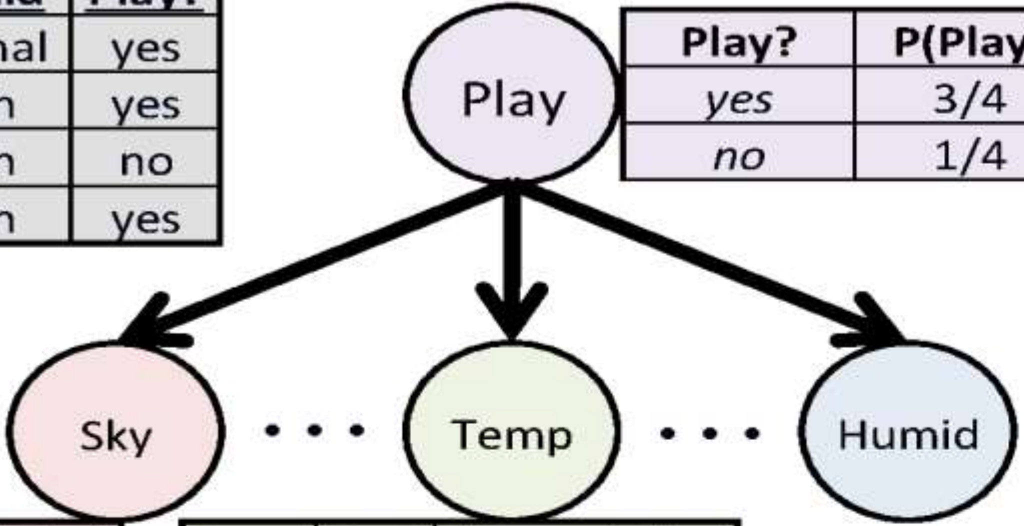
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

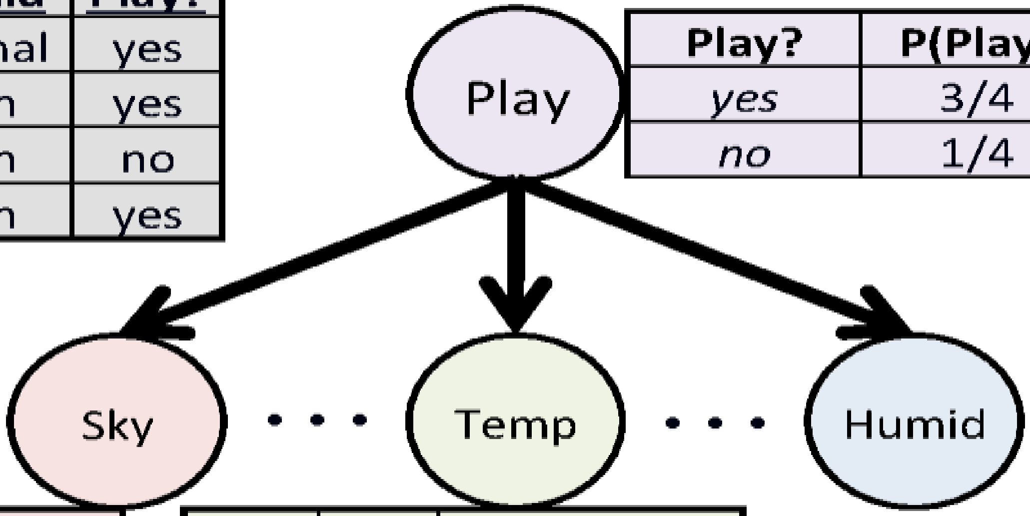
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	
cold	*yes*	
warm	*no*	
cold	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

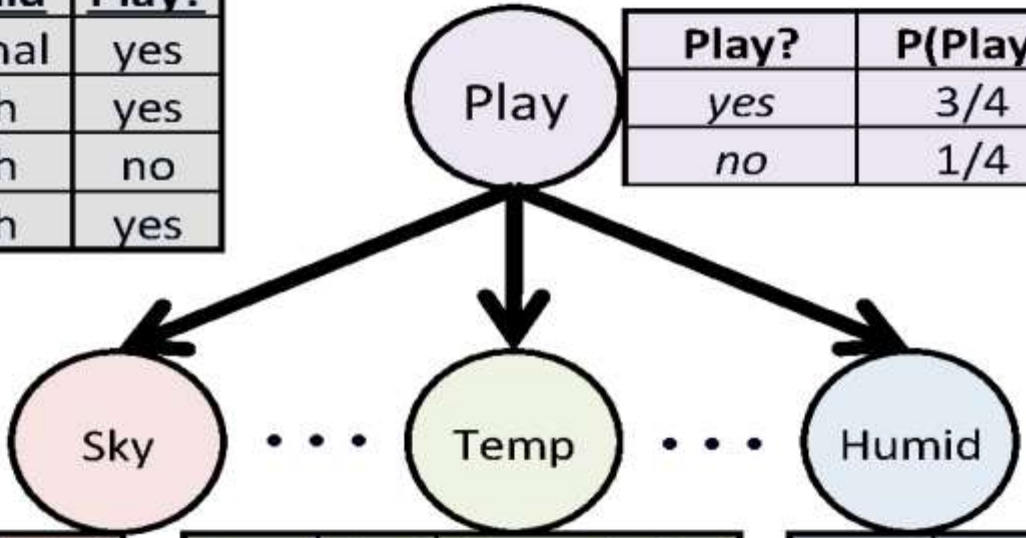
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

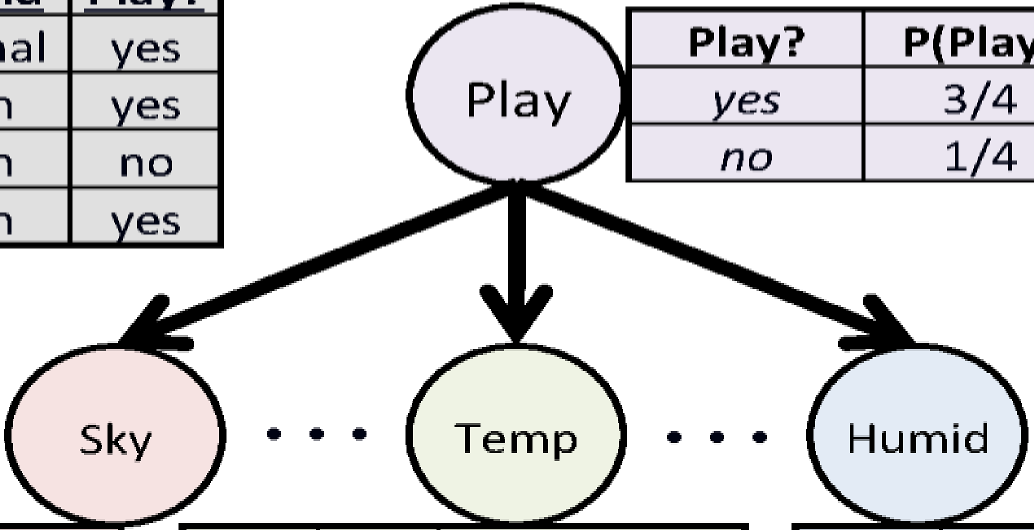
Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	
norm	*yes*	
high	*no*	
norm	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



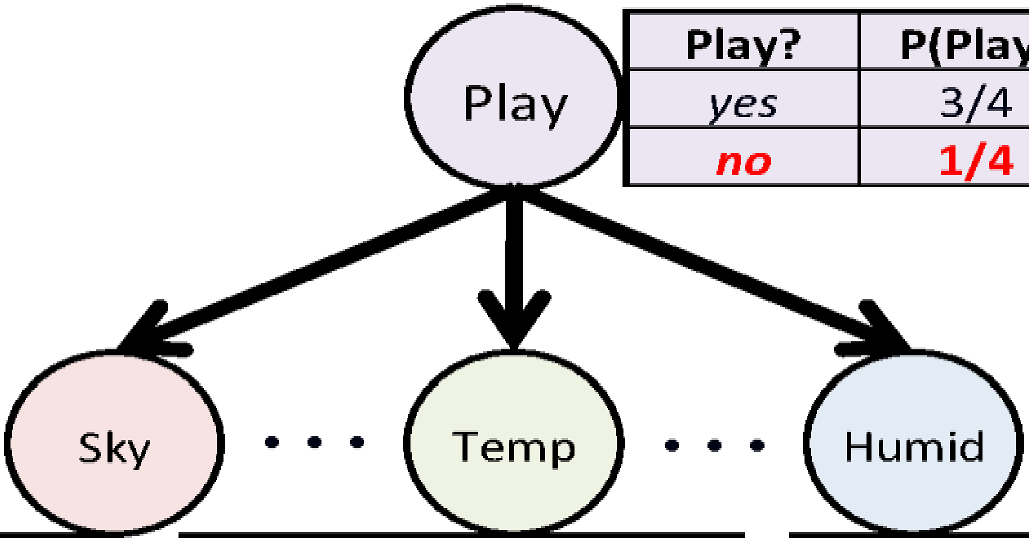
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

Example NB Graphical Model



Play?	P(Play)
yes	3/4
no	**1/4**

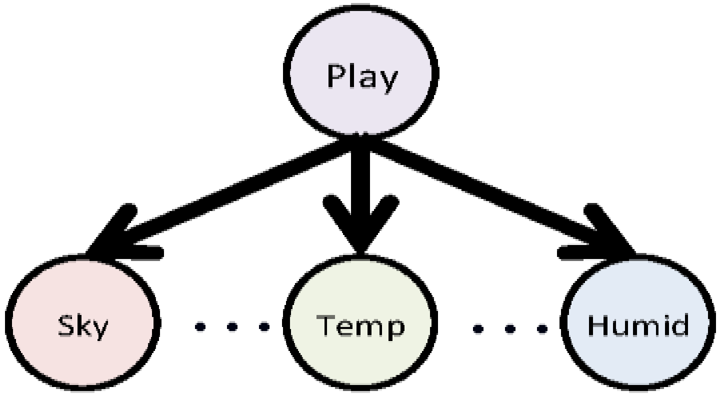
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	***yes***	**1/5**
sunny	*no*	1/3
rainy	***no***	**2/3**

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	***yes***	**1/5**
warm	*no*	1/3
cold	***no***	**2/3**

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	***yes***	**2/5**
high	*no*	2/3
norm	***no***	**1/3**

- Some **redundancies** in CPTs that can be eliminated

Example Using NB for Classification



Play?	P(Play)
yes	3/4
no	1/4

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

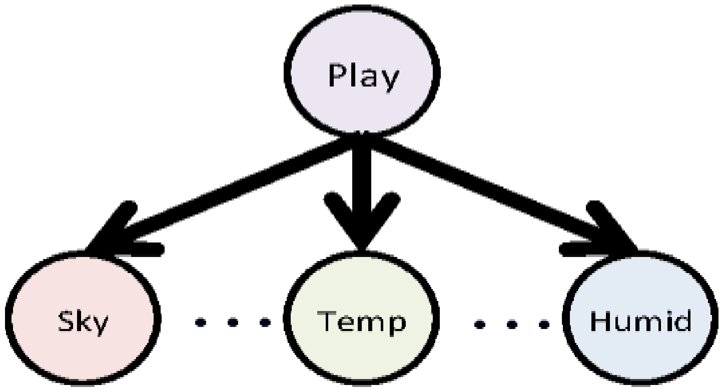
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

$$h(\mathbf{x}) = \arg\max_{y_k} \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

Goal: Predict label for $\mathbf{x}$ = (rainy, warm, normal)

Example Using NB for Classification



Play?	P(Play)
yes	3/4
no	1/4

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

Predict label for:

x = (rainy, warm, normal)

$$P(\text{play} \mid \mathbf{x}) \propto \log P(\text{play}) + \log P(\text{rainy} \mid \text{play}) + \log P(\text{warm} \mid \text{play}) + \log P(\text{normal} \mid \text{play})$$

$$\propto \log 3/4 + \log 1/5 + \log 4/5 + \log 2/5 = \boxed{-1.319 \quad \text{predict PLAY}}$$

$$P(\neg\text{play} \mid \mathbf{x}) \propto \log P(\neg\text{play}) + \log P(\text{rainy} \mid \neg\text{play}) + \log P(\text{warm} \mid \neg\text{play}) + \log P(\text{normal} \mid \neg\text{play})$$

$$\propto \log 1/4 + \log 2/3 + \log 1/3 + \log 1/3 = -1.732$$

Naive Bayes model

Naive Bayes model under the scikit-learn library:

- **Gaussian:** It is used in classification and it assumes that features follow a normal distribution.
- **Multinomial:** It is used for discrete counts. For example, let's say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of "word occurring in the document", we have "count how often word occurs in the document", you can think of it as "number of times outcome number x_i is observed over the n trials".
- **Bernoulli:** The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with 'bag of words' model where the 1s & 0s are "word occurs in the document" and "word does not occur in the document" respectively.

Naïve Bayes Summary

Advantages:

- Fast to train (single scan through data)
- Fast to classify
- Not sensitive to irrelevant features
- Handles real and discrete data
- Handles streaming data well

Disadvantages:

- Assumes independence of features

Online Links

https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/

https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html

https://www.kaggle.com/lovedeepsaini/fraud-detection-with-naive-bayes-classifier

http://jonathansoma.com/lede/foundations/classes/classification/naive-bayes/


Thank You!