Naïve Bayes

These slides were assembled by Eric Eaton, with grateful acknowledgement of the many others who made their course materials freely available online. Feel free to reuse or adapt these slides for your own academic purposes, provided that you include proper attribution. Please send comments and corrections to Eric.

Essential Probability Concepts

- Marginalization: $$P(B) = \sum_{v \in \mathrm{values}(A)} P(B \wedge A = v)$$

- Conditional Probability: $$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$

- Bayes' Rule: $$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)}$$

- 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...

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

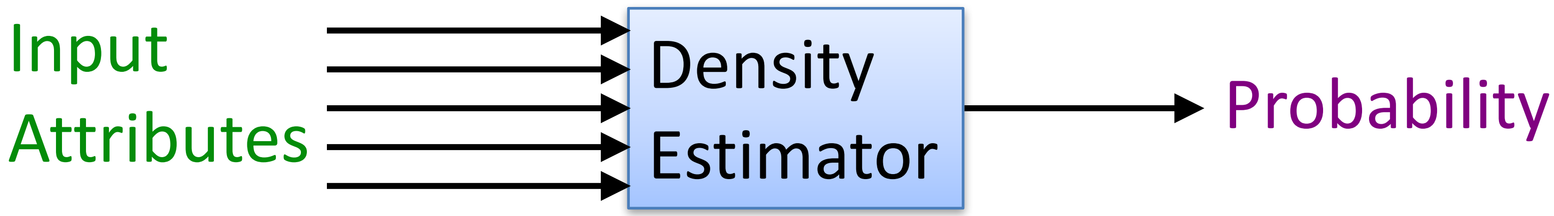
Example of Learning a Joint PD

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

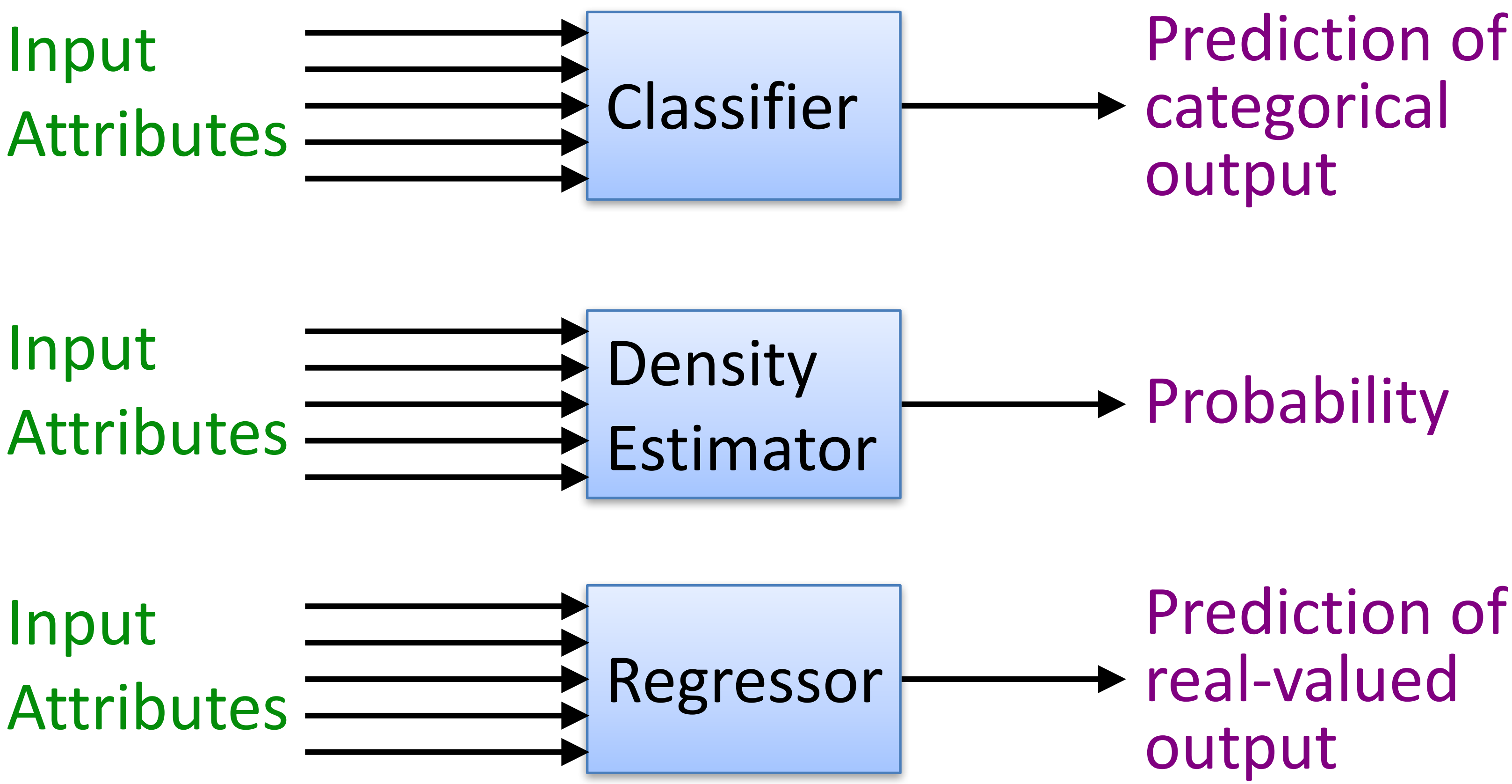
Density Estimation

- 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



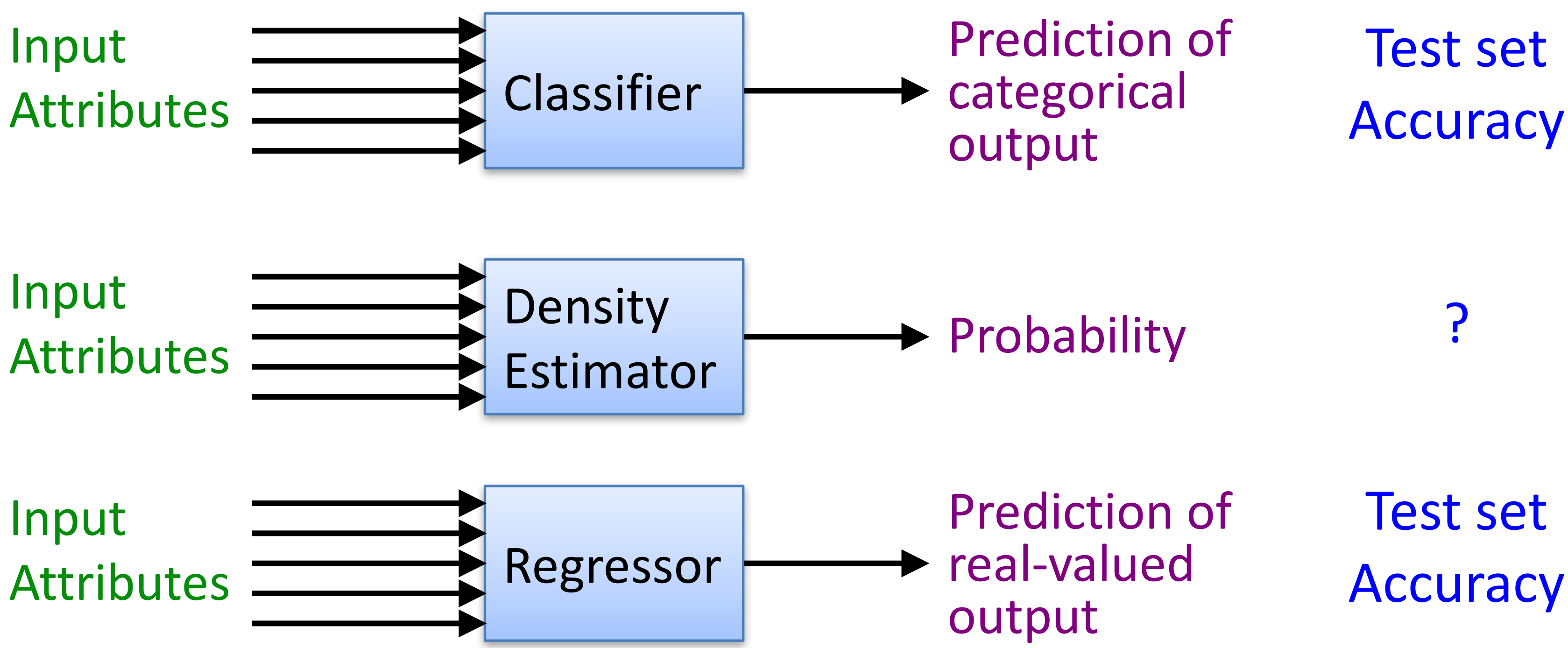
Density Estimation

Compare it against the two other major kinds of models:



Evaluating Density Estimation

Test-set criterion for
estimating performance
on future data



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.)

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

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

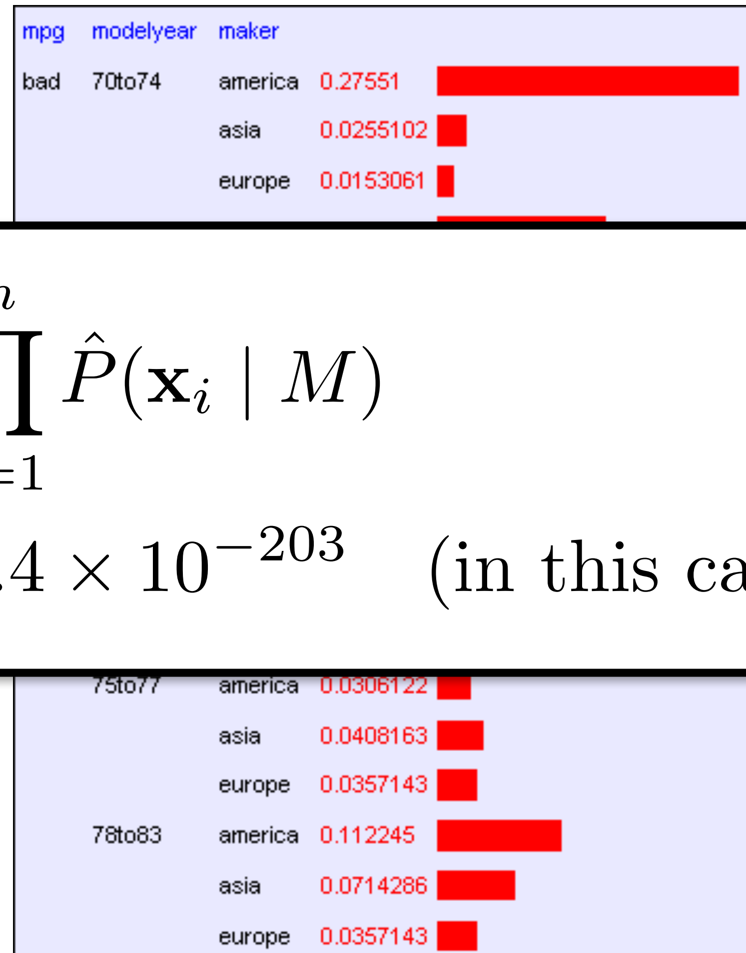
Slide by Andrew Moore

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

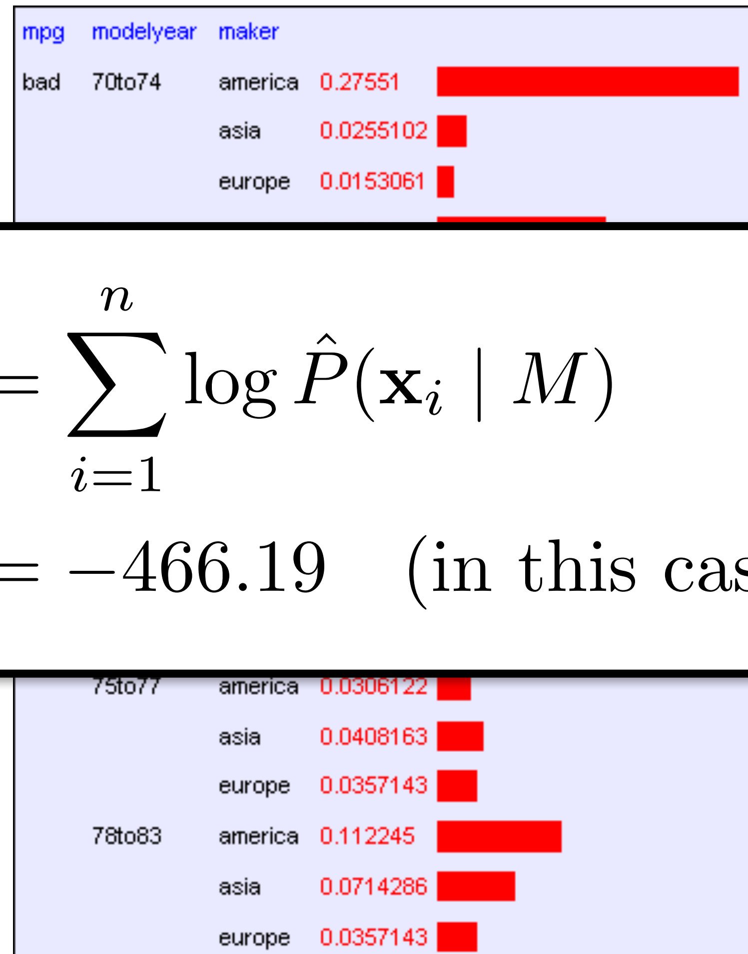
$$\log \hat{P}(\text{dataset} \mid M) = \log \prod_{i=1}^{n} \hat{P}(\mathbf{x}_i \mid M) = \sum_{i=1}^{n} \log \hat{P}(\mathbf{x}_i \mid M)$$

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

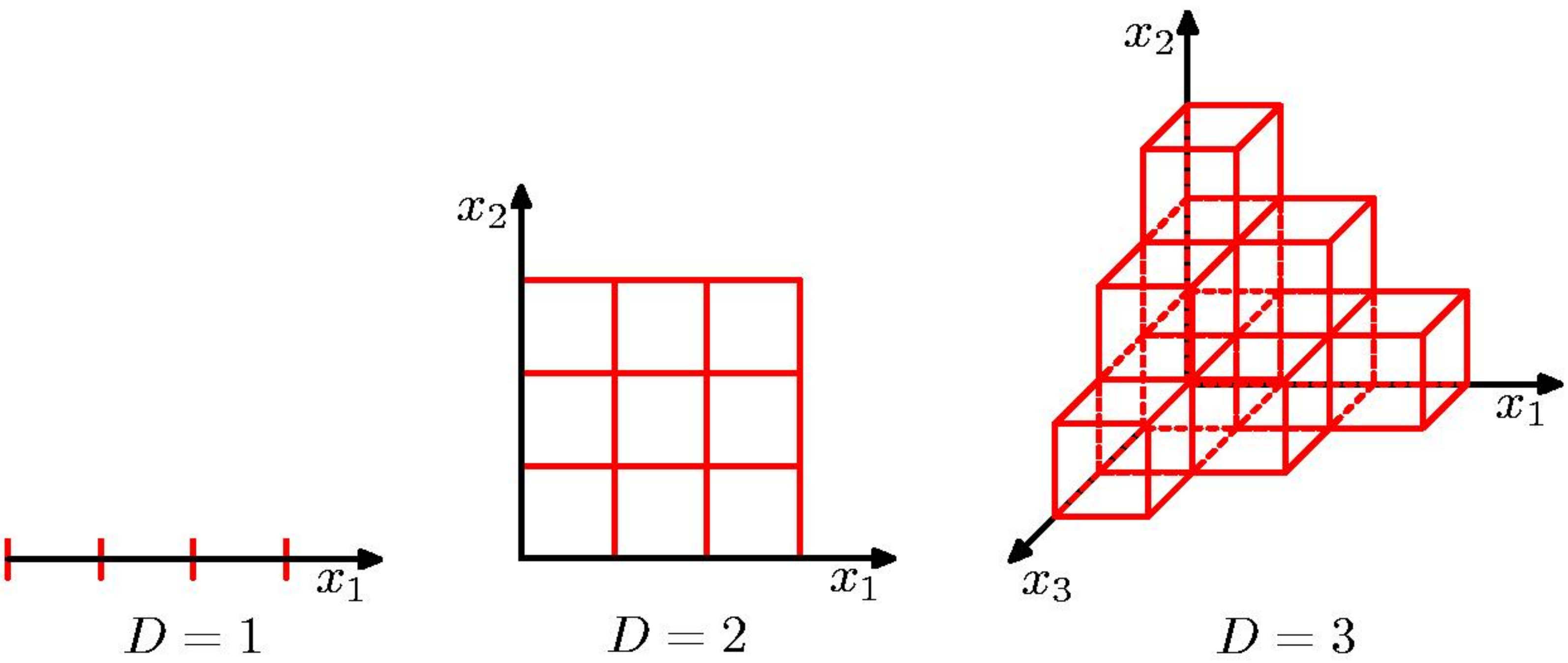
The Good News:

- 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 impractical, may result in adverse behavior

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

- 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!

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

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

$$= -\infty \quad \text{if for any } i,\ \hat{P}(\mathbf{x}_i \mid M) = 0$$

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

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

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

---

Easy to estimate from data

Impractical, but necessary

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

Unnecessary, as it turns out

- 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…

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*

P(play) = ?

P(Sky = sunny | play) = ?

P(Humid = high | play) = ?

...

P(¬play) = ?

P(Sky = sunny | ¬play) = ?

P(Humid = high | ¬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	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = ?

P(Sky = sunny | play) = ?

P(Humid = high | play) = ?

…

P(¬play) = ?

P(Sky = sunny | ¬play) = ?

P(Humid = high | ¬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	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = 3/4

P(Sky = sunny | play) = ?

P(Humid = high | play) = ?

...

P(¬play) = 1/4

P(Sky = sunny | ¬play) = ?

P(Humid = high | ¬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						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$ $\qquad$ $\mathrm{P}(\neg\mathrm{play}) = 1/4$

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

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

… $\qquad$ …

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*

P(play) = 3/4

P(¬play) = 1/4

P(Sky = sunny | play) = 1

P(Sky = sunny | ¬play) = ?

P(Humid = high | play) = ?

P(Humid = high | ¬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*

P(play) = 3/4 P(¬play) = 1/4

P(Sky = sunny | play) = 1 P(Sky = sunny | ¬play) = ?

P(Humid = high | play) = ? P(Humid = high | ¬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*

P(play) = 3/4

P(¬play) = 1/4

P(Sky = sunny | play) = 1

P(Sky = sunny | ¬play) = 0

P(Humid = high | play) = ?

P(Humid = high | ¬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?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

P(play) = 3/4 P(¬play) = 1/4

P(Sky = sunny | play) = 1 P(Sky = sunny | ¬play) = 0

P(Humid = high | play) = ? P(Humid = high | ¬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?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

P(play) = 3/4 P(¬play) = 1/4

P(Sky = sunny | play) = 1 P(Sky = sunny | ¬play) = 0

P(Humid = high | play) = 2/3 P(Humid = high | ¬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*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = 3/4 P(¬play) = 1/4

P(Sky = sunny | play) = 1 P(Sky = sunny | ¬play) = 0

P(Humid = high | play) = 2/3 P(Humid = high | ¬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*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = 3/4

P(¬play) = 1/4

P(Sky = sunny | play) = 1

P(Sky = sunny | ¬play) = 0

P(Humid = high | play) = 2/3

P(Humid = high | ¬play) = 1

…

…

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*

P(play) = 3/4

P(Sky = sunny | play) = 1

P(Humid = high | play) = 2/3

...

P(¬play) = 1/4

P(Sky = sunny | ¬play) = 0

P(Humid = high | ¬play) = 1

...

Laplace Smoothing

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

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

$$P(X_j = v \mid Y = y_k) = \frac{\mathrm{c}_v + 1}{\displaystyle\sum_{v' \in \mathrm{values}(X_j)} \mathrm{c}_{v'} + |\mathrm{values}(X_j)|}$$

  where
  - c_v is the count of training instances with a value of v for attribute j and class label y_k
  - $|\mathrm{values}(X_j)|$ 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*

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

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

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

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

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

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

…

…

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*

P(play) = 3/4

P(¬play) = 1/4

P(Sky = sunny | play) = 4/5

P(Sky = sunny | ¬play) = 1/3

P(Humid = high | play) = ?

P(Humid = high | ¬play) = ?

...

...

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*

P(play) = 3/4   P(¬play) = 1/4

P(Sky = sunny | play) = 4/5   P(Sky = sunny | ¬play) = 1/3

P(Humid = high | play) = 3/5   P(Humid = high | ¬play) = ?

...   ...

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:

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

P(play) = 3/4 P(¬play) = 1/4

P(Sky = sunny | play) = 4/5 P(Sky = sunny | ¬play) = 1/3

P(Humid = high | play) = 3/5 P(Humid = high | ¬play) = 2/3

... ...

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 this

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

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

  - 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

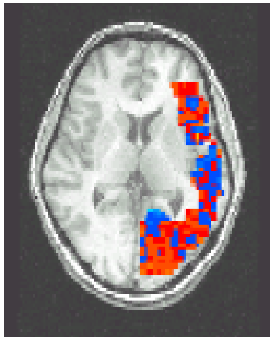
$$P(Y = y_k \mid X = \mathbf{x}) = \alpha \tilde{P}(Y = y_k \mid X = \mathbf{x})$$

Naïve Bayes Applications

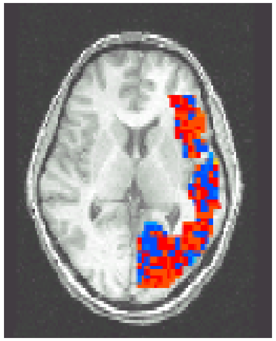
- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?

- Classifying mental states

Learning P(BrainActivity | WordCategory)



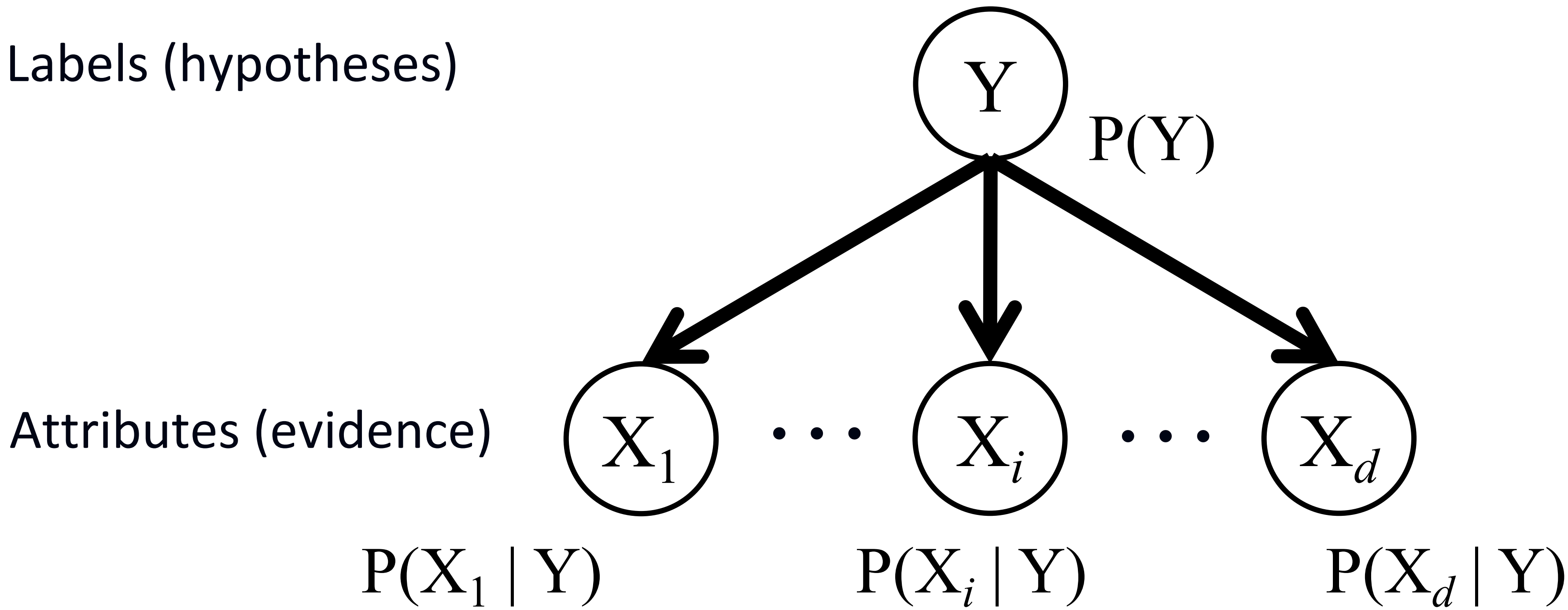
People Words



Animal Words

Pairwise Classification
Accuracy: 85%

The Naïve Bayes Graphical Model

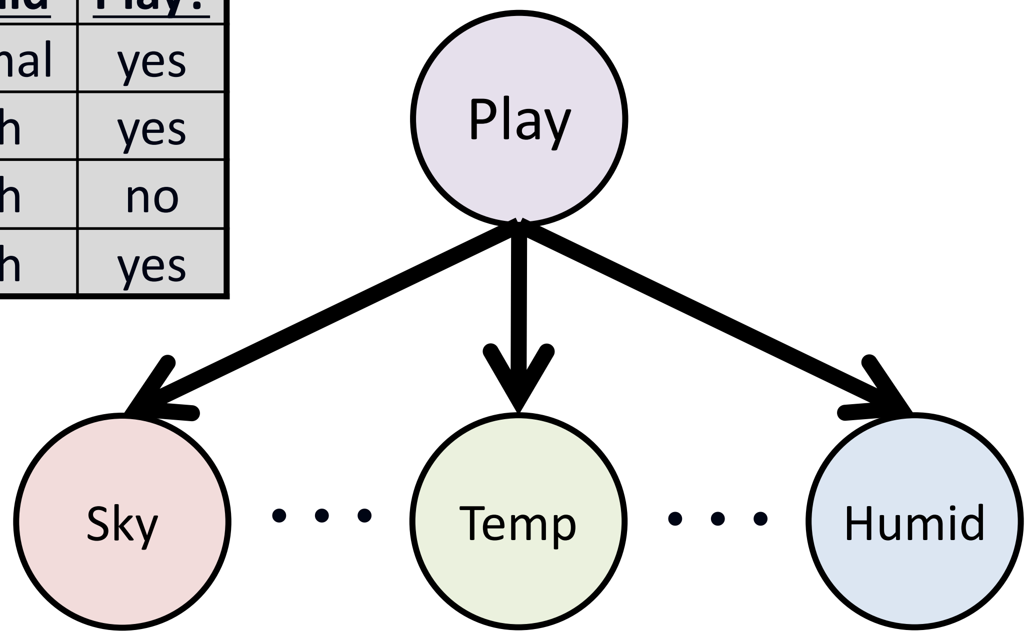


- Nodes denote random variables
- Edges denote dependency
- Each node has an associated conditional probability table (CPT), conditioned upon its parents

Example NB Graphical Model

Data:

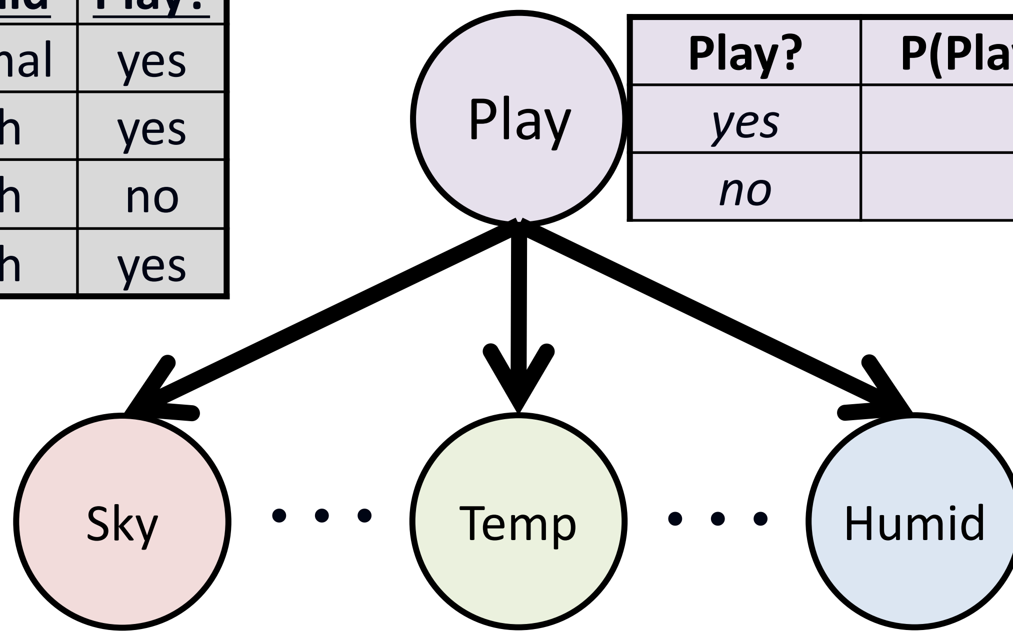
Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



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



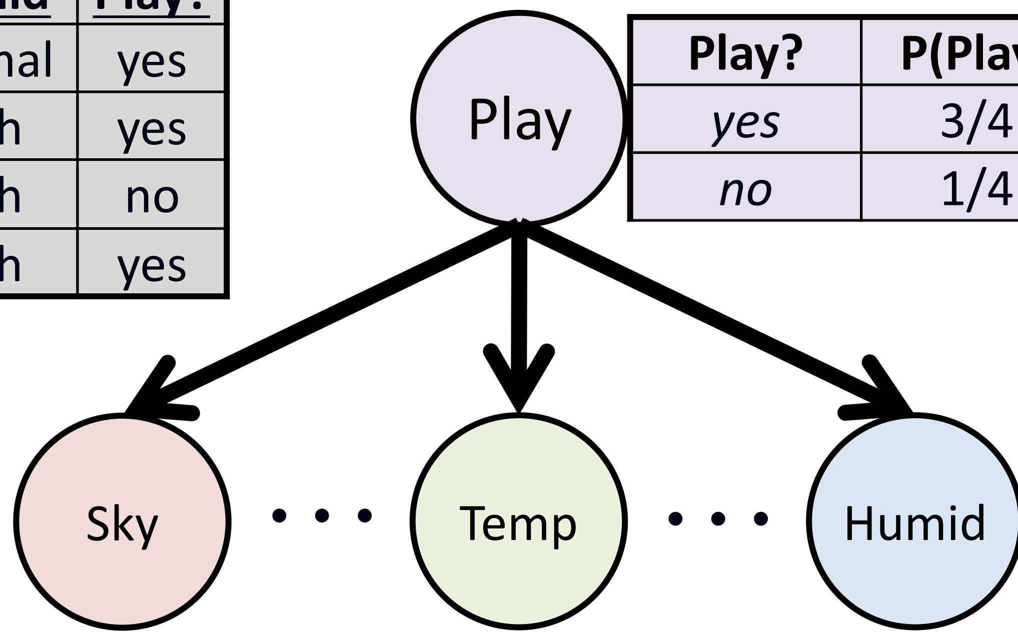
Play?	P(Play)
yes	
no	

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

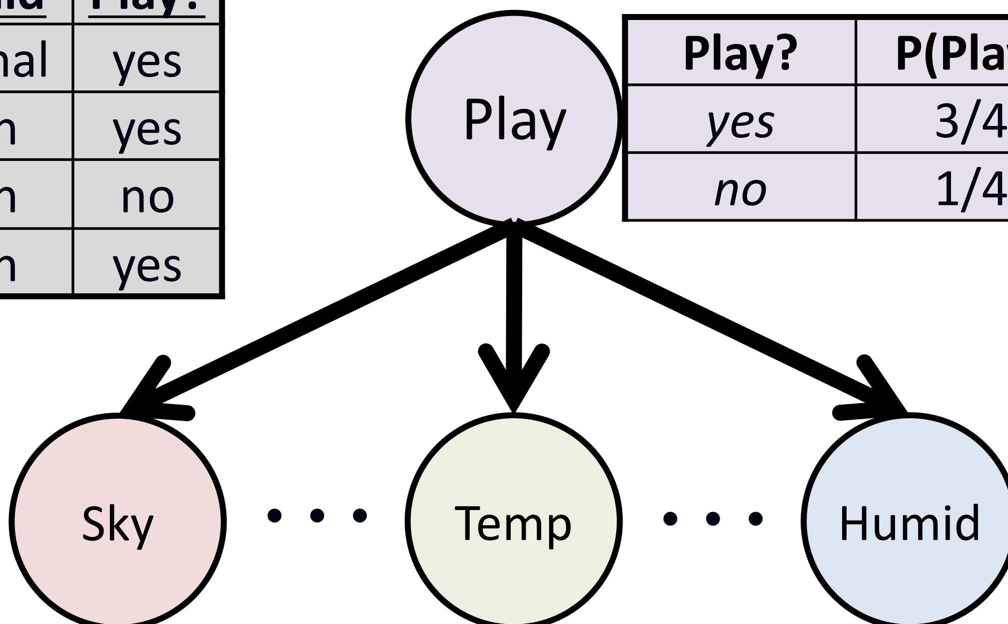
Play?	P(Play)
yes	3/4
no	1/4



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



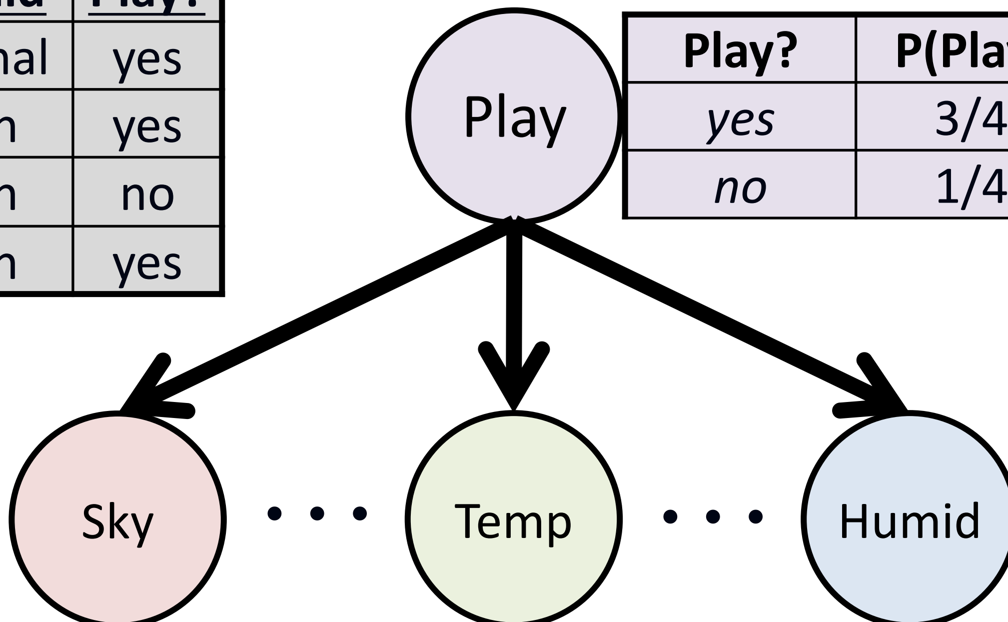
Play?	P(Play)
yes	3/4
no	1/4

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



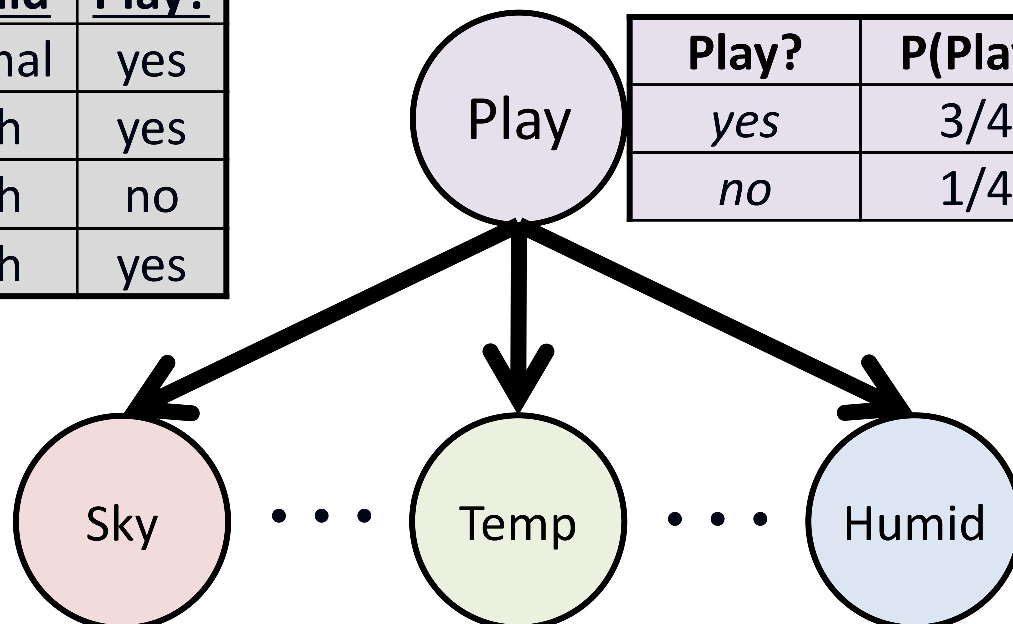
Play?	P(Play)
yes	3/4
no	1/4

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

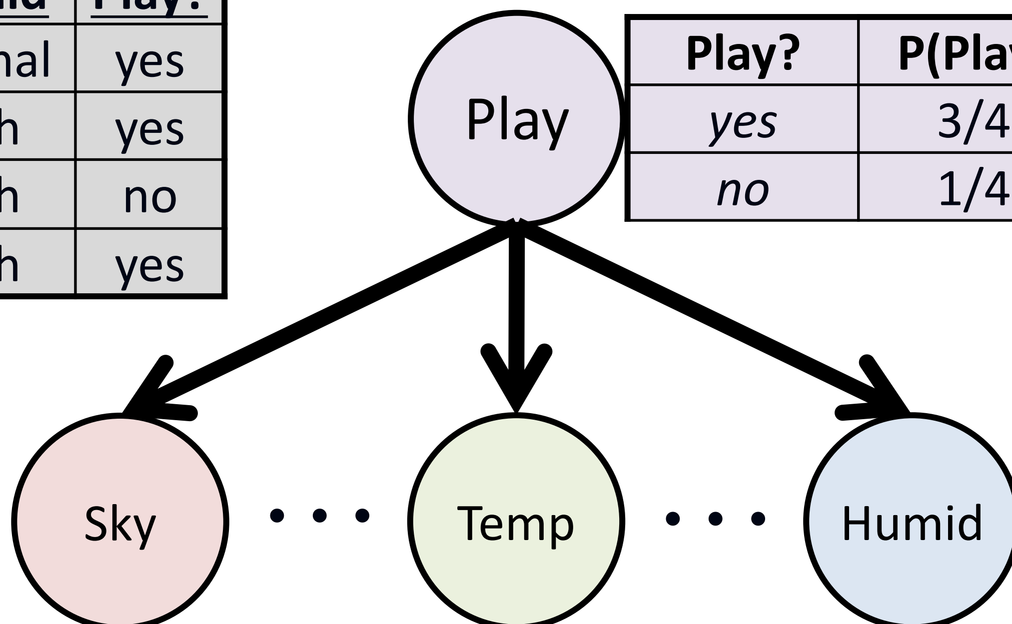
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

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

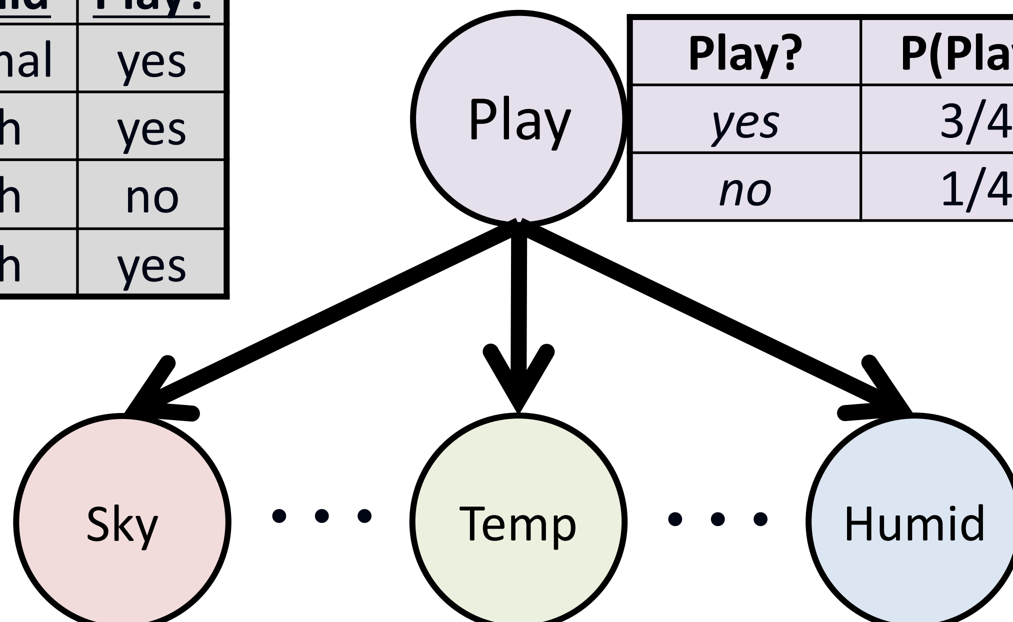
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

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:

<u>**Sky**</u>	<u>**Temp**</u>	<u>**Humid**</u>	<u>**Play?**</u>
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

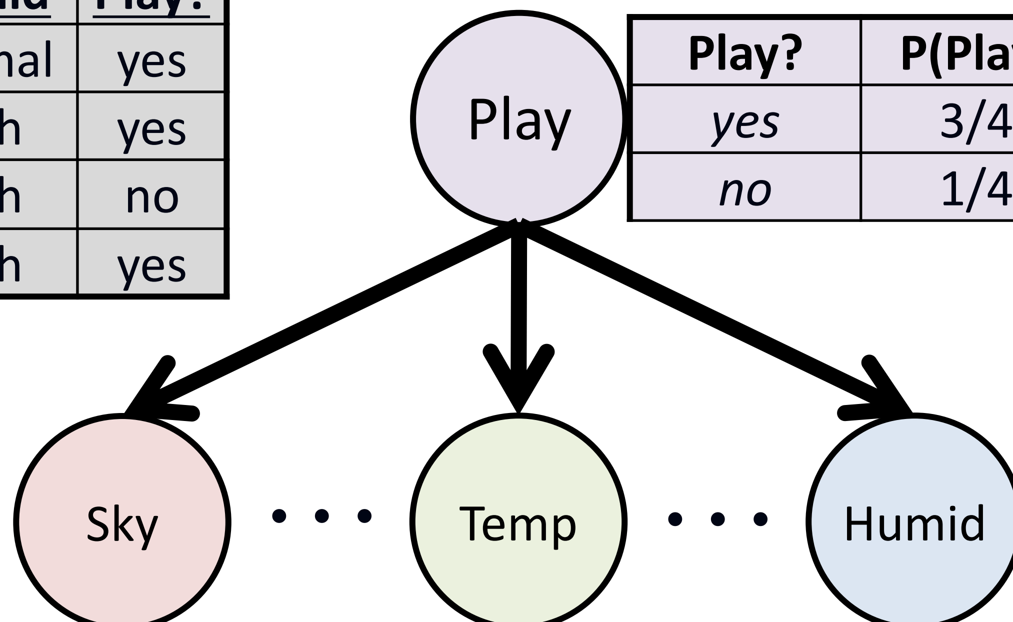
Temp	**Play?**	**P(Temp | Play)**
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

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



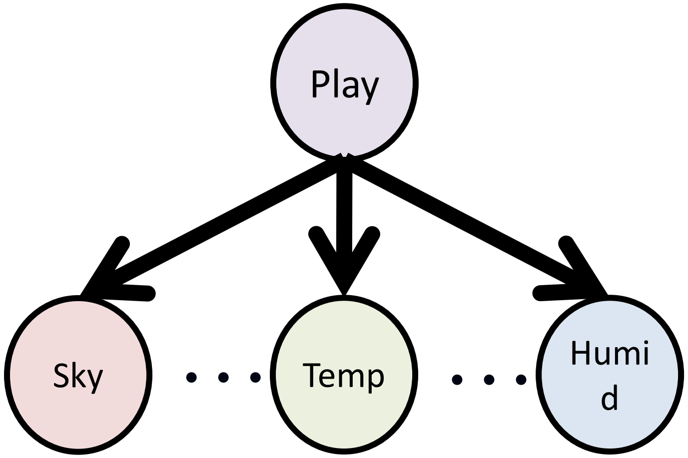
Play?	P(Play)
yes	3/4
no	1/4

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 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

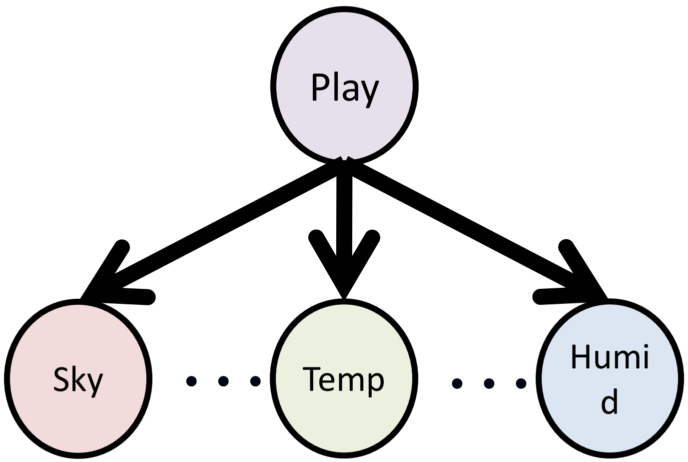
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

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:

$\mathbf{x}$ = (rainy, warm, normal)

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

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

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

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

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