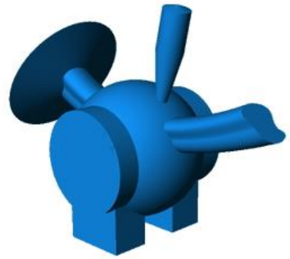
Naïve Bayes

Semester 1, 2023

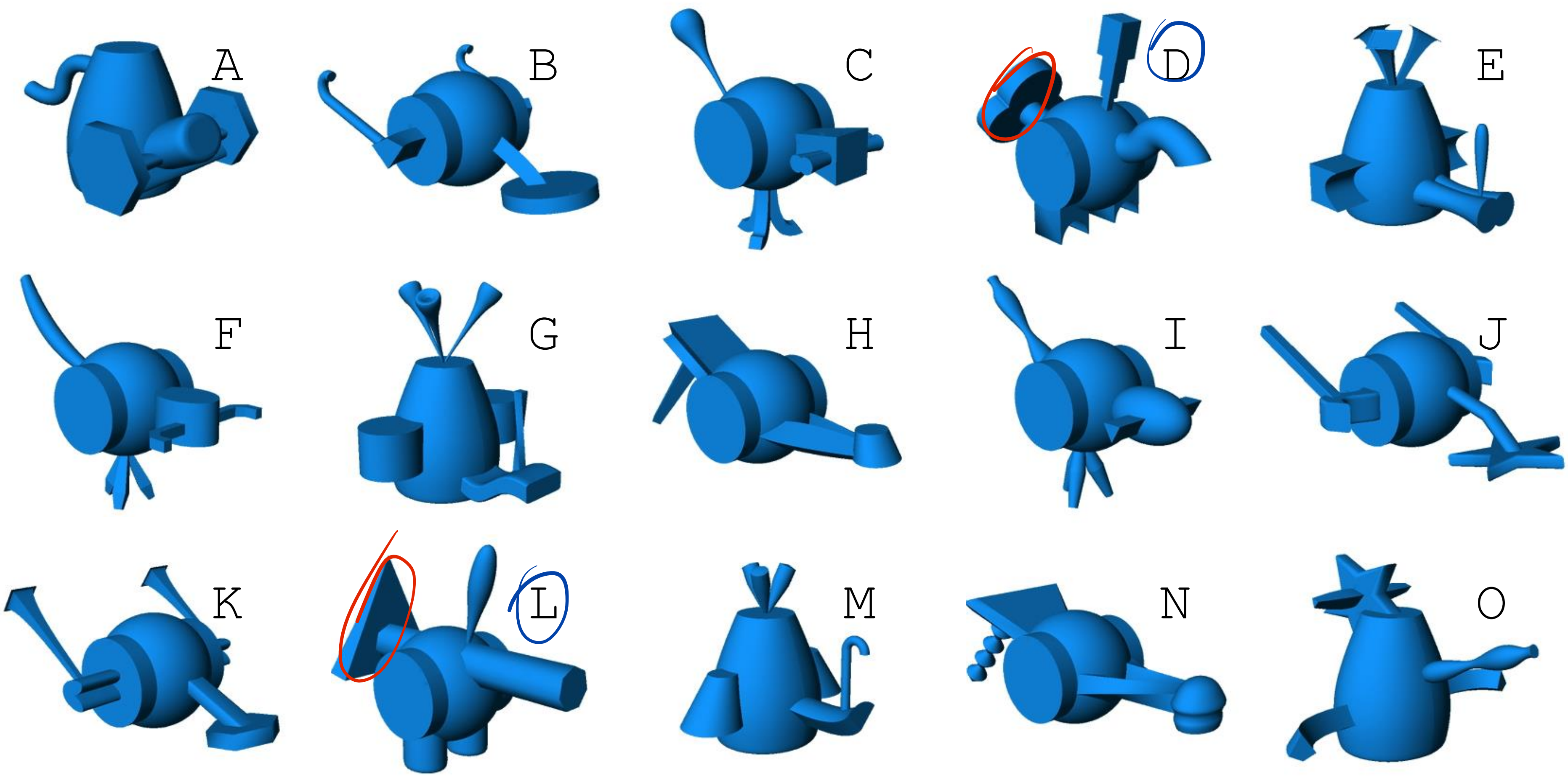
Kris Ehinger

Outline

- Probabilistic learning
- Bayes' Rule
- Naïve Bayes classifier
- Practical issues, assumptions, etc.



This is a “tufa”
Are there any more tufas
in the set below?



“Fribbles” dataset: P. Williams (1998)

Concept learning

- People can grasp a "class" concept from very limited data:
  - One (or few) examples
  - Noisy features
  - Diverse data set
  - Ambiguity about what features might be important
- How to get this generalisation performance from machines?

Why probabilistic models?

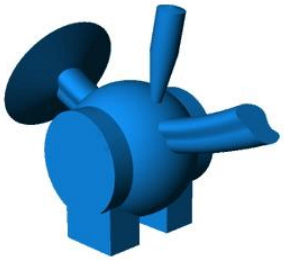
- Framework for modelling systems that are noisy/uncertain
- Rules that let you generalise from limited observations
- Based on laws of probability, so gives an optimal prediction given the available data
    - and the assumptions built into the model

Probabilistic learner

- Goal is classification, so we'll build a **supervised** model
- We need to build a probabilistic model of the training data, and then use that to predict the class of the test data
- In probability terms: given an instance T, which class c is most likely?
- $\hat{c} = \arg\max_{c \in C} P(c|T)$

Probabilistic learner

- How to do this?
- For each class c:
  - Find all examples of c in the training data
  - Count the number of times T has been observed
- Choose class ĉ with the greatest frequency of observed T



What if you've never seen this specific instance before?

Probabilistic learner

- Unfortunately, this requires a massive amount of data!
- We need to have seen every possible *combination* of attributes in the training set, ideally at least a few times per class, to have a good estimate of frequency
- For m attributes with k possible values and C classes, this mean $O(Ck^m)$ instances
  - 2 classes, 20 binary attributes: >2 million
  - 2 classes, 10 attributes with 10 values: 20 trillion

Naïve Bayes solution

- Assume different attributes are statistically independent, conditional on class
- Compute P(c|T) from P(T|c) using Bayes rule

$$P(c_j|T) = \frac{P(T|c_j)P(c_j)}{P(T)}$$ → all same for c_j

$$\max P(c_j|T) \rightarrow \max P(T|c_j)P(c_j)$$

Bayes’ Rule

Bayes' Rule example

- Consider this situation:

You go to a shop that you're pretty sure is open today. When you get close, however, you notice all the lights are off inside and the windows are dark.

- What do you conclude?

Bayesian mental model

- When a shop is open, it usually doesn't turn the lights off:

  P(off | open) = 0.01

- When a shop is closed, it usually does turn the lights off:

  P(off | closed) = 0.85

$P(\text{off} \cap \text{open})$ $P(\text{off} \cap \text{closed})$

$P(\text{off}) = P(\text{off} \mid \text{open})P(\text{open}) + P(\text{off} \mid \text{closed})P(\text{closed})$

- So what are the chances the shop is open given that the lights are off?

$P(\text{open} \mid \text{off})$

Bayes' Rule

- For hypothesis H and evidence x
  - P(H), the prior, is the initial degree of belief in H
  - P(H|x), the posterior, is the degree of belief in H, given x
  - P(x|H) is the likelihood of observing evidence x given that H is true

Likelihood of observing x when H is true

Posterior: Belief in H, given observation x

Prior: Belief in H before the observation

$$P(H|x) = \frac{P(H,x)}{P(x)} = \frac{P(x|H)P(H)}{P(x)}$$

Likelihood of observing x in general

Updating beliefs

- What are the chances the shop is open given that the lights are off?

$$P(H|x) = \frac{P(x|H)P(H)}{P(x)}$$

- Depends on prior P(open)…

0.01 0.95

$$\text{P(open | off)} = \frac{\text{P(off | open) P(open)}}{\text{P(off | open)P(open) + P(off | closed)P(closed)}}$$

0.01 0.95 0.85 0.05

P(open) = 0.95

$$\text{P(open | off)} = \frac{0.01 \times 0.95}{0.001 + 0.043} = 0.183$$

Bayes' Rule

- Bayes' Rule allows us to compute P(H|x) when P(x|H) can be estimated
  - In many situations, it's fairly easy to estimate P(x|H) from real world observations or a theoretical model
- Powerful tool to represent a model's (or person's) beliefs, and how those beliefs are updated through experience with the world

Naïve Bayes classifier

Bayes' Rule

$$P(C,X) = P(C|X)P(X) = P(X|C)P(C)$$

Posterior: Likelihood the class is C, given observation X

Likelihood of observing X in class C

$P(C \cap X)$

Prior: Likelihood of class C, in general

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Likelihood of observing X, in general

Naïve Bayes learner

- Task: classify an instance T into one of the possible classes $c_j \in C$

$$\hat{c} = \arg\max_{c_j \in C} P(c_j | T)$$

$$P(c_j \cap T) = P(c_j | T) P(T) = P(T | c_j) P(c_j)$$

$$= \arg\max_{c_j \in C} \frac{P(T | c_j) P(c_j)}{\cancel{P(T)}}$$

Same for all $c_j \in C$, so we can ignore it

Naïve Bayes learner

- Task: classify an instance T into one of the possible classes $c_j \in C$

$$\hat{c} = \arg\max_{c_j \in C} P(c_j|T)$$

$$= \arg\max_{c_j \in C} P(T|c_j)P(c_j)$$

- Each class generates instances
- Each class could have generated this instance, with some likelihood
- Which class is most likely to have generated this instance?

Naïve Bayes learner

attributes

- Task: classify an instance $T = < x_1, x_2, \ldots , x_n >$ into one of the possible classes $c_j \in C$

$$\hat{c} = \underset{c_j \in C}{\arg\max}\, P(c_j | x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\arg\max}\, \frac{P(x_1, x_2, \ldots, x_n | c_j) P(c_j)}{\cancel{P(x_1, x_2, \ldots, x_n)}}$$

Same for all $c_j \in C$, so we can ignore it

$$= \underset{c_j \in C}{\arg\max}\, P(x_1, x_2, \ldots, x_n | c_j) P(c_j)$$

?

hard to compute

What makes it "naïve"?

- Naïve Bayes assumes all attributes are independent, conditional on the class:

$$P(x_1, x_2, \ldots, x_n | c_j) \approx P(x_1 | c_j) P(x_2 | c_j) \ldots P(x_n | c_j)$$

AND $\cap$

$$= \prod_i P(x_i | c_j)$$

- Makes the problem tractable – easy to compute these probabilities
- But it's almost always untrue in real-world data
- But naïve Bayes (usually) works pretty well anyway

Complete naïve Bayes learner

- Task: classify an instance $T = < x_1, x_2, \ldots, x_n >$ into one of the possible classes $c_j \in C$

$$\hat{c} = \underset{c_j \in C}{\arg\max}\, P(c_j | x_1, x_2, \ldots, x_n) \quad // \frac{P(T|c_j)P(c_j)}{P(T)}$$

↓ bayes rule + ignore denominator

$$= \underset{c_j \in C}{\arg\max}\, P(x_1, x_2, \ldots, x_n | c_j) P(c_j)$$

Complete naïve Bayes learner

- Task: classify an instance T = < x_1, x_2, … , x_n > into one of the possible classes $c_j \in C$

$$\hat{c} = \underset{c_j \in C}{\arg\max}\, P(c_j | x_1, x_2, \dots, x_n)$$

$$\boxed{= \underset{c_j \in C}{\arg\max}\, P(c_j) \prod_i P(x_i | c_j)}$$

- Each class generates attributes, with some likelihood
- Which class is most likely to have generated these attributes?

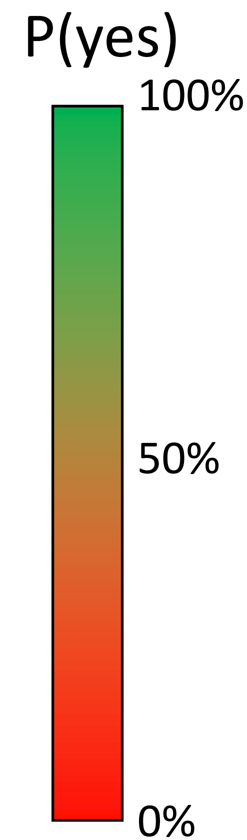
Bayesian prior

- The prior $P(c_j)$ can be estimated from the frequency of classes in the training set (maximum likelihood estimate)
- Naïve Bayes learns the priors from the training set and uses them in prediction
  - Good if the training set correctly reflects the real-world / test set distribution of classes (or is close)
  - But potentially a problem if you want to apply the classifier to a new situation with new priors

new class

Computing probability

Outlook	Temp	Humidity	Windy	Play?
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
sunny	cool	normal	false	**yes**
overcast	hot	high	true	no

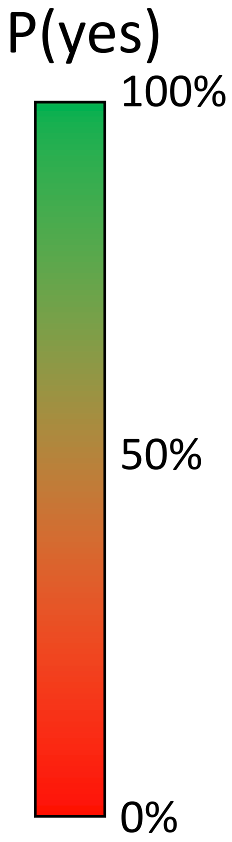


Today it's sunny, cool, normal, false. What's the class?

$P(c_j \mid x_1, x_2, x_3, x_4) \Rightarrow$ yes $P=1$

Computing probability

Outlook	Temp	Humidity	Windy	Play?
rainy	hot	normal	true	**yes**
rainy	hot	normal	true	no
rainy	hot	normal	true	**yes**
rainy	hot	normal	true	no
rainy	hot	normal	true	**yes**
rainy	hot	normal	true	no
sunny	cool	normal	false	**yes**
sunny	mild	high	false	no
overcast	cool	high	true	no



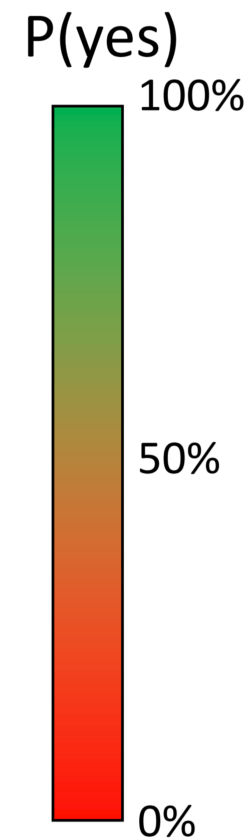
Today it's rainy, hot, normal, true. What's the class?

$P(c_j \mid \text{rainy, hot, normal, true})$, yes : $P=0.5$

no : $P=0.5$

Computing probability

Outlook	Temp	Humidity	Windy	Play?
overcast	mild	normal	true	**yes**
sunny	mild	normal	false	**yes**
overcast	hot	high	true	**yes**
sunny	cool	high	false	**yes**
rainy	cool	normal	true	no
overcast	hot	normal	true	no
sunny	hot	normal	false	no
sunny	mild	normal	true	no
overcast	cool	high	true	no



Today it's overcast, mild, high, false. What's the class?

$\max \; P(x_1|c_j)\, P(x_2|c_j)\, P(c_j)$

each of features are more common in yes

This is naive bayes, consider each feature independently

Naïve Bayes example

- Given a training data set, what probabilities do we need to estimate?

Headache	Sore	Temperature	Cough	Diagnosis
severe	mild	high	yes	**Flu**
no	severe	normal	yes	**Cold**
mild	mild	normal	yes	**Flu**
mild	no	normal	no	**Cold**
severe	severe	normal	yes	**Flu**

We need $P(c_j)$, $P(x_i|c_j)$ for every x_i, c_j

Naïve Bayes example

Headache	Sore	Temperature	Cough	Diagnosis
severe	mild	high	yes	**Flu**
no	severe	normal	yes	**Cold**
mild	mild	normal	yes	**Flu**
mild	no	normal	no	**Cold**
severe	severe	normal	yes	**Flu**

P(Flu) = 3/5
P(Headache = severe|Flu) = 2/3
P(Headache = mild|Flu) = 1/3
P(Headache = no|Flu) = 0/3

P(Cold) = 2/5
P(Headache = severe|Cold) = 0/2
P(Headache = mild|Cold) = 1/2
P(Headache = no|Cold) = 1/2

Naïve Bayes example

$P(Flu) = 3/5$
$P(Headache = severe|Flu) = 2/3$
$P(Headache = mild|Flu) = 1/3$
$P(Headache = no|Flu) = 0/3$
$P(Sore = severe|Flu) = 1/3$
$P(Sore = mild|Flu) = 2/3$
$P(Sore = no|Flu) = 0/3$
$P(Temp = high|Flu) = 1/3$
$P(Temp = normal|Flu) = 2/3$
$P(Cough = yes|Flu) = 3/3$
$P(Cough = no|Flu) = 0/3$

$P(Cold) = 2/5$
$P(Headache = severe|Cold) = 0/2$
$P(Headache = mild|Cold) = 1/2$
$P(Headache = no|Cold) = 1/2$
$P(Sore = severe|Cold) = 1/2$
$P(Sore = mild|Cold) = 0/2$
$P(Sore = no|Cold) = 1/2$
$P(Temp = high|Cold) = 0/2$
$P(Temp = normal|Cold) = 2/2$
$P(Cough = yes|Cold) = 1/2$
$P(Cough = no|Cold) = 1/2$

Naïve Bayes example

- A patient comes to the clinic with mild headache, severe soreness, normal temperature, and no cough. Are they more likely to have cold or flu?

$P(x_1|c_j)P(x_2|c_j)P(x_3|c_j)P(c_j) \approx P(c_j)P(x_1,x_2,x_3|c_j) =_{max} \frac{P(c_j \cap T)}{P(T)}$

$=_{max} P(c_j|T)$

Cold: $P(C)P(H=m|C)P(S=s|C)P(T=n|C)P(C=n|C)$

$$\frac{2}{5}\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)\left(\frac{2}{2}\right)\left(\frac{1}{2}\right)=0.05$$

not P(patient having cold)
just relative probability

Flu: $P(F)P(H=m|F)P(S=s|F)P(T=n|F)P(C=n|F)$

$$\frac{3}{5}\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)\left(\frac{0}{3}\right)=0$$

Naïve Bayes example

- A patient comes to the clinic with severe headache, mild soreness, high temperature, and no cough. Are they more likely to have cold or flu?

Cold: $P(C)P(H = s|C)P(S = m|C)P(T = h|C)P(C = n|C)$

$$\frac{2}{5}\left(\frac{0}{2}\right)\left(\frac{0}{2}\right)\left(\frac{0}{2}\right)\left(\frac{1}{2}\right) = \underline{0}$$

training dataset is too small
no all combination

Flu: $P(F)P(H = s|F)P(S = m|F)P(T = h|F)P(C = n|F)$

$$\frac{3}{5}\left(\frac{2}{3}\right)\left(\frac{2}{3}\right)\left(\frac{1}{3}\right)\left(\frac{0}{3}\right) = \underline{0}$$

$\therefore$ equal probability

Zero values

- Problem: if any $P(x_i|c_j) = 0$, the final value will be zero
  - This means we need to see every possible pairing of attributes to class, which means a lot of data; likely to miss some pairings in a real-world data set
  - The 0s are actually informative – the fact that we never observed a pairing means it's probably rare
- Solution: treat unobserved events as possible but unlikely (all probabilities > 0)

Probabilistic smoothing I

- Simple option: whenever you encounter $P(x_i | c_j) = 0$, just replace 0 with a small positive constant ε
- ε should be very small, much less than (1/N) (N = number of instances), given the class Flu: N=3, Cold: N=2
- Since ε is tiny, assume all probabilities still sum to 1 (no need to scale other values)
- In practice, tends to reduce to comparisons to the cardinality of ε (meaning, whichever class has fewest εs wins)

underestimate the rare event

Probabilistic smoothing example

- A patient comes to the clinic with severe headache, mild soreness, high temperature, and no cough. Are they more likely to have cold or flu?

Cold: $P(C)P(H=s|C)P(S=m|C)P(T=h|C)P(C=n|C)$

$$\frac{2}{5}(\varepsilon)(\varepsilon)(\varepsilon)\left(\frac{1}{2}\right)=\frac{\varepsilon^3}{5}$$

Flu: $P(F)P(H=s|F)P(S=m|F)P(T=h|F)P(C=n|F)$

$$\frac{3}{5}\left(\frac{2}{3}\right)\left(\frac{2}{3}\right)\left(\frac{1}{3}\right)(\varepsilon)=\frac{4\varepsilon}{45}$$

Probabilistic smoothing II

- Slightly more complicated option: increase *all* counts by 1 (**Laplace smoothing**)
  - Unseen events get a count of 1
  - Events seen once become 2, twice 3, etc.
  - A more general version: increase all counts by α, which is a value between 0 and 1
  - Formula for an attribute X with d values:

|t| =

Unsmoothed:

$$P_i = \frac{x_i}{N}$$

Smoothed:

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

→ level

headache: d=3

Probabilistic smoothing example

Headache	Sore	Temperature	Cough	Diagnosis
severe	mild	high	yes	**Flu**
no	severe	normal	yes	**Cold**
mild	mild	normal	yes	**Flu**
mild	no	normal	no	**Cold**
severe	severe	normal	yes	**Flu**

$\alpha d = 1 \times 3 = 3$

P(Headache = severe|Flu) = ($\overset{\alpha}{1}$+2) / ($\overset{\alpha d}{3}$+3) = 3/6

P(Headache = mild|Flu) = ($\overset{\alpha}{1}$+1) / (3+3) = 2/6

P(Headache = no|Flu) = ($\overset{\alpha}{1}$+0) / (3+3) = 1/6

P(Headache = severe|Cold) = (1+0) / ($\overset{\alpha d}{3}$+$\overset{N}{2}$) = 1/5

P(Headache = mild|Cold) = (1+1) / (3+2) = 2/5

P(Headache = no|Cold) = (1+1) / (3+2) = 2/5

Probabilistic smoothing II

Laplace smoothing: N is large denominator is large. probability don't change much

- Most common method of Laplace smoothing is **add-one smoothing** (α=1, all counts increase by 1)
- Probabilities are changed drastically when there are few instances, but the changes are smaller with more instances (mimics confidence)

  N is small

- Add-one smoothing is known to overestimate the likelihood of rare events
  - But ε or lower values of α can underestimate
  - Hard to choose the right α in practice

Probabilistic smoothing III

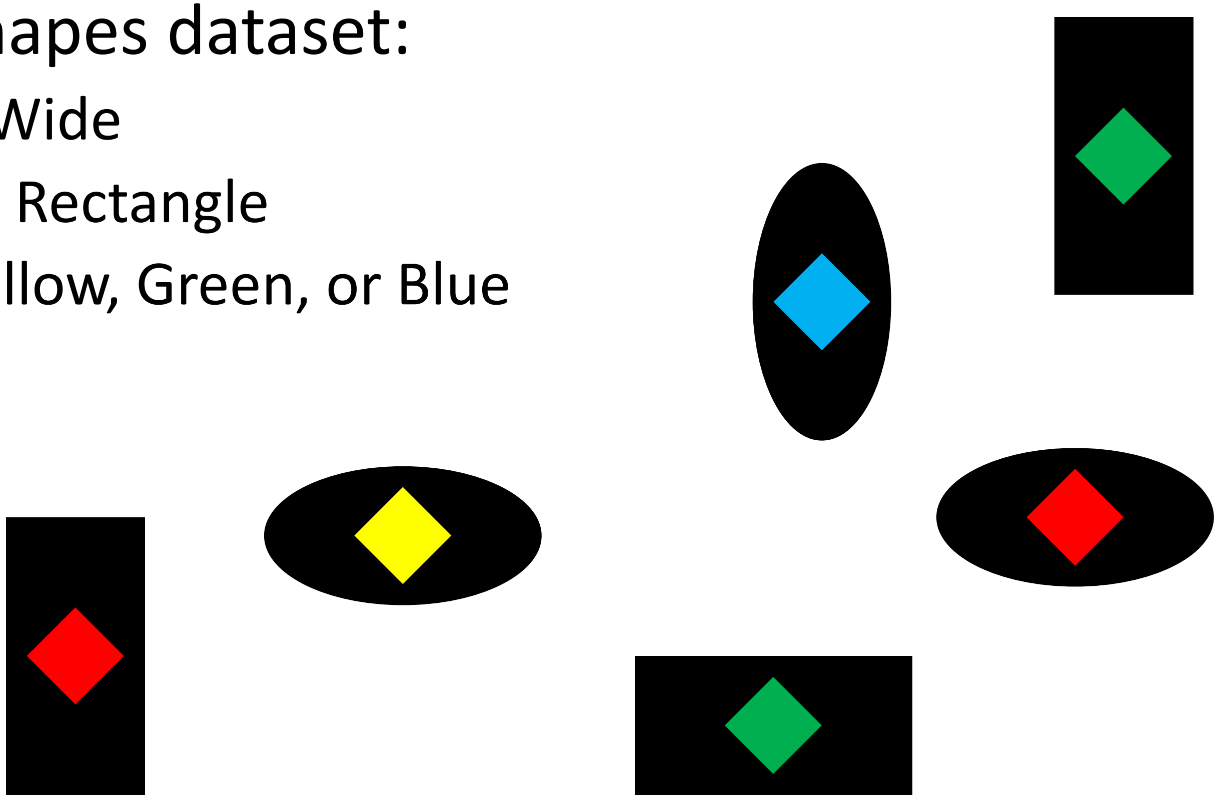
- Even more options:
  - Add-k smoothing: like Laplace smoothing, but adds a value k > 1 overestimate likelihood of rare events
  - Good-Turing estimation: uses the observed counts of different events to estimate how likely you are to see a never-before-seen event use observed data to predict unobserved data
  - Regression

Missing values

- What if an instance is missing some attribute?
- Missing values at test can simply be ignored – compute the likelihood of each class from the non-missing values
- Missing values in training can also be ignored – don't include them in the attribute-class counts, and the probabilities will be based on the non-missing values

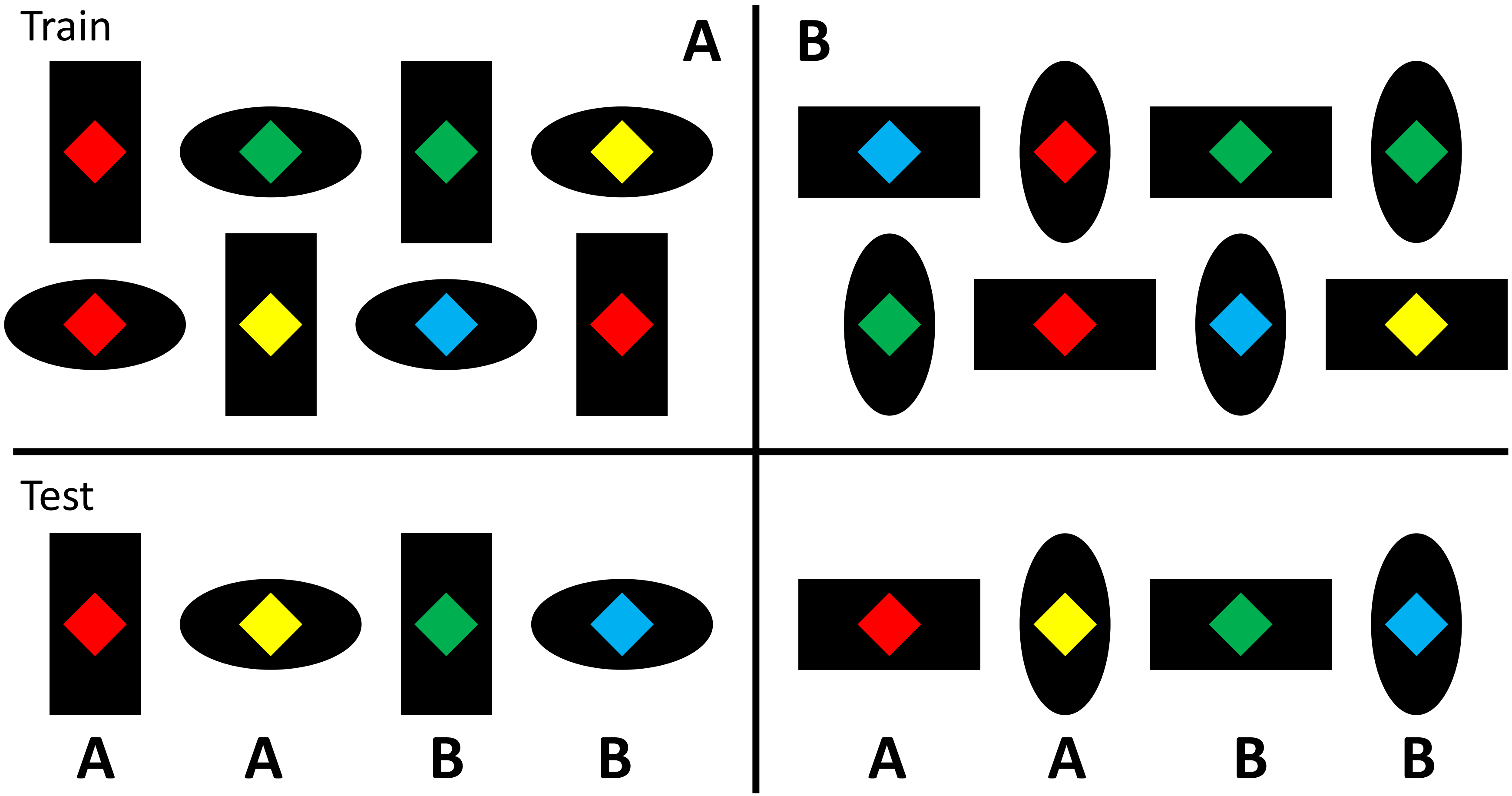
Naïve Bayes example

- Simple shapes dataset:
  - Tall or Wide
  - Oval or Rectangle
  - Red, Yellow, Green, or Blue



Tall, Rectangle, Red   Wide, Rectangle, Green

Naïve Bayes example



Naïve Bayes summary

- Why does it (usually) work, despite the wrong assumption of conditional independence?
  - We don't need a perfect estimate of $P(c|T)$ for every class – we just need to know which class is most likely
  - Ignoring the fact that some attributes are correlated tends to make all the class probabilities higher, but doesn't typically change their rank
  - Naïve Bayes is also robust to small errors in estimating $P(x_i|c_j)$ – these can change class probabilities but typically don't change class rank

Naïve Bayes

- Strengths:
  - Simple to build, fast
  - Computations scale well to high-dimensional datasets (1000s of attributes)
  - Explainable – generally easy to understand why the model makes the decision it does
- Weaknesses:
  - Inaccurate when there are many missing $P(x_i|c_j)$ values
  - Conditional independence assumption becomes problematic for complex systems