

## Naïve Bayes and Probability Theory

Benjamin Rosman

[Benjamin.Rosman1@wits.ac.za](mailto:Benjamin.Rosman1@wits.ac.za) / [benjros@gmail.com](mailto:benjros@gmail.com)

February 2018

# Probability

- Why do we care about probability?
- Not always confident in our statements
  - Features: how likely are our features?
  - Outcomes: how confident is our decision?
  - Predictions tend to not be certain



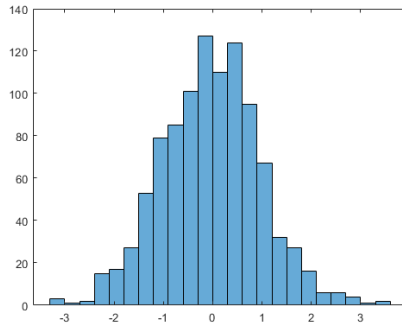
# Random Variables



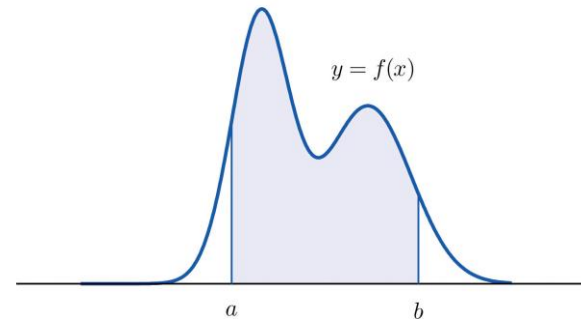
- Interested in how likely some event is
- A random variable (RV) takes on values from a collection of **mutually exclusive** and **collectively exhaustive** states (each corresponding to an event)
- E.g.
  - Outcome of game: win, lose, draw
  - Eye colour: brown, green, blue
  - Number of phones per house: 0, 1, 2, 3-5, 6+
  - Salary:  $0 \leq \textit{amount} \leq 1,000,000$
- We are interested in the probability of some event occurring

# Random Variables

- RVs: discrete or continuous



$P(a < X < b) = \text{area of shaded region}$



- Notation: capital letters denote RVs, lower case letters are the values. E.g.  $p(X = x)$ . This is the probability that RV  $X$  takes on the value  $x$ . Usually shortened to  $p(x)$ .
- Probabilities sum to 1, i.e. the chance of *anything* happening is 100%
  - Discrete RVs:  $\sum_x p(x) = 1$
  - Continuous RVs:  $\int p(x)dx = 1$ .  $p(x)$  is called the probability density function (pdf)
  - Often achieved by normalisation: divide by the total

# Joint Distributions

- Probabilities over multiple variables?
- Let  $X$  and  $Y$  be two RVs
- $X = \{\text{yes, no}\}$  indicating whether an email contains the word “millions”
- $Y = \{\text{ham, spam}\}$  indicating the type of email

	$Y = \text{ham}$	$Y = \text{spam}$
$X = \text{yes}$	0.01	0.25
$X = \text{no}$	0.49	0.25

- **Note:** sum = 1, as one of these four combinations must have happened
- Notation:  $p(X = \text{yes}, Y = \text{ham}) = 0.01$

“probability that an email is ham AND contains the word ‘millions’ is 0.01”

# Marginal Probabilities

- The **sum rule**:
  - $p(X) = \sum_y p(X, Y)$
- What does this say?
- E.g.
  - $P(\text{email doesn't contain "millions"}) = P(X=\text{no}) = ?$
  - $P(\text{email isn't spam}) = P(Y=\text{ham}) = ?$

# Conditional Probabilities

- We often want to know how probable something is **given that (conditioned on) something else has already happened**
- The conditional probability distribution (CPD) of  $X$  given  $Y = y$  (note: a specific value of  $y$ ) is:
  - $p(X = x | Y = y) = p(x|y) = \frac{p(x,y)}{p(y)}$
- Rewrite to give the **product rule**:
  - $p(X, Y) = p(X)p(Y|X) = p(Y)p(X|Y)$
- E.g.  $P(X=\text{yes} | Y=\text{ham}) = ?$
- **Note**  $\sum_x p(X = x | Y = y) = 1$ , for all  $y$

# Bayes' Rule

$$p(X, Y) = p(X)p(Y|X) = p(Y)p(X|Y)$$

- From the product rule

- $p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$

- From the sum rule, the denominator is
  - $p(X) = \sum_y p(X|Y)p(Y)$
- Most powerful rule in probabilistic ML!



Rev. Thomas Bayes  
(1701 – 1761)



# Bayes' Rule



- Let  $X$  be an observation,  $Y$  a class label. We want to know  $p(Y|X)$ : what is the probability of this class, given this data?

**Posterior probability of label  $Y$  having seen data  $X$**

**Likelihood of seeing data  $X$  under label  $Y$**

**Prior probability of label  $Y$ : how likely to we expect  $Y$  to be**

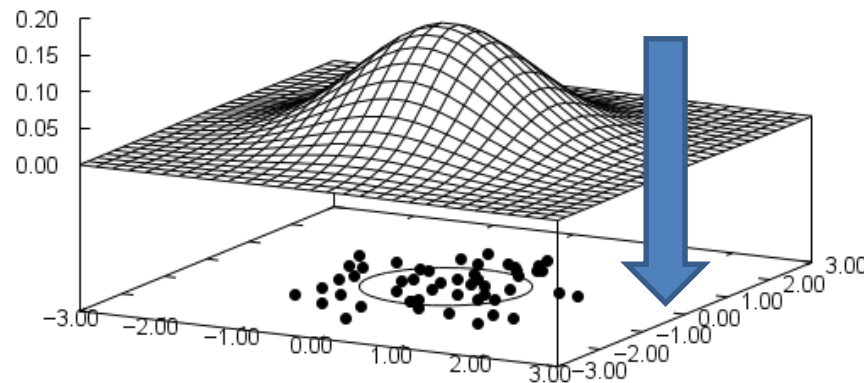
$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Think of this as:  
I thought the prob of  $y$  was  $p(y)$   
Then I saw  $x$   
Now I think it is  $p(y|x)$

Probability of data  $X$  under any  $Y$  (normalisation)

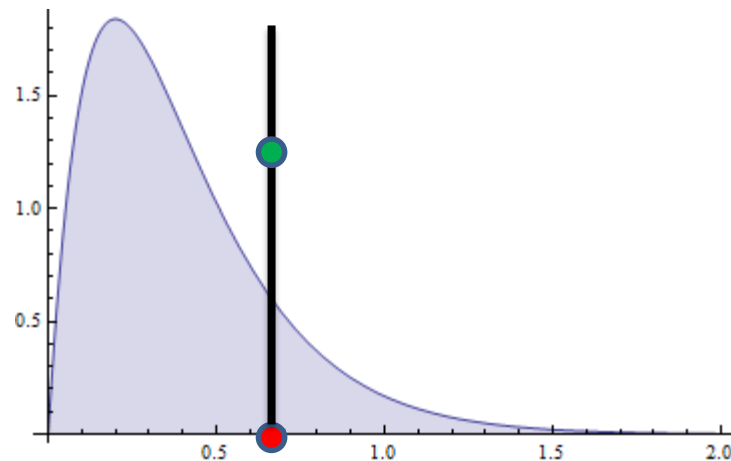
# What to do with distributions?

- Generate data  $x$  from distribution  $p$  (sampling)
  - $x \sim p$
  - Each data point generated independently and is identically distributed: iid
- Generative approach: use to create artificial data



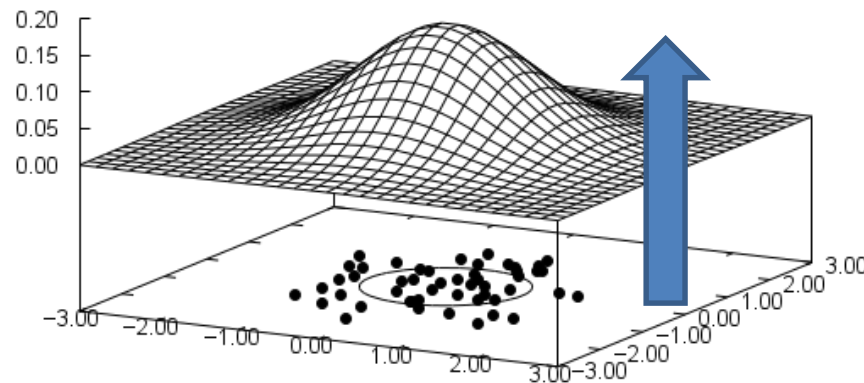
# Sampling

- Generate data  $x$  from distribution  $p$  (sampling)
  - How:
    - Pick a **random point  $x$  in domain of  $p$**
    - Accept if **random number  $< p(x)$**



# What to do with data?

- Estimate the distribution  $p$  from the data  $x$ 
  - Learning problem: learn  $p$  that fits  $x$
- How to measure goodness of fit?
  - Given a distribution, how likely is it to have generated the data?
  - i.e. what is the probability of this data set given the distribution?



# Likelihood

- So: want the probability of data  $D$  given distribution/model  $M$  = likelihood of  $M$  =  $p(D | M)$ 
  - $p(D|M) = \prod_{i=1}^N p(x_i|M)$ 
    - Product of probabilities of generating each data point independently
- Compute  $p(D | M)$  for different models  $M$
- Pick  $M$  which gives the **highest likelihood**
  - This is the **maximum likelihood estimate**
  - Optimisation problem!

# Example distribution: Gaussian

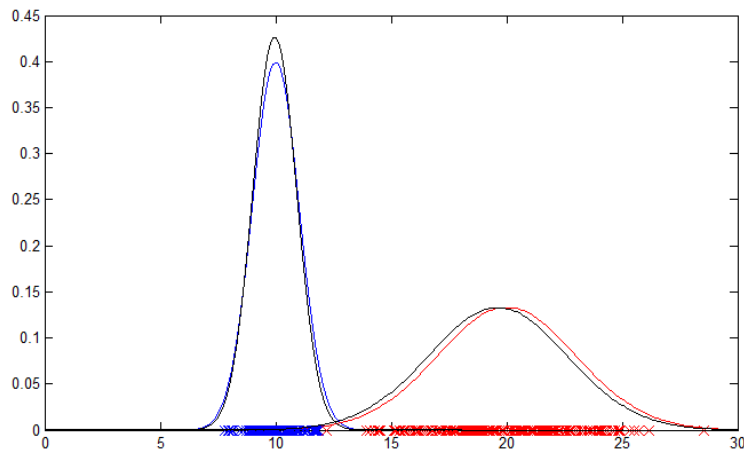
- Most common continuous distribution
- Often a very reasonable model
- Also called Normal distribution
- 1D Gaussian:

$$p(x|\mu, \sigma^2) = N(x; \mu, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

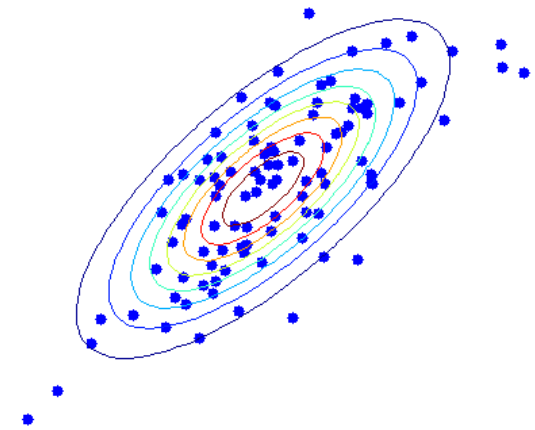
mean      variance

Normalisation factor:  
integrate to 1

Probability of  $x$  decreases  
as  $x$  moves further from  $\mu$ ,  
with speed governed by  $\sigma$



# Multivariate Gaussian



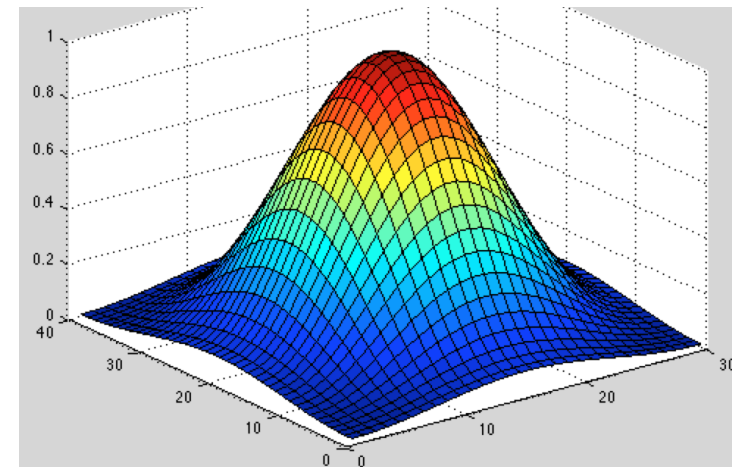
- Data  $\mathbf{x}$  is d-dimensional

- $$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\}$$

- $\boldsymbol{\mu}$  = d-dimensional mean vector
- $\Sigma$  = covariance matrix: symmetric and positive definite

- $\Sigma = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]$

- Parameters to learn:
  - $\Sigma = d(d+1)/2$ ,  $\boldsymbol{\mu} = d$ . Why?



# Max Likelihood Estimate of a Gaussian

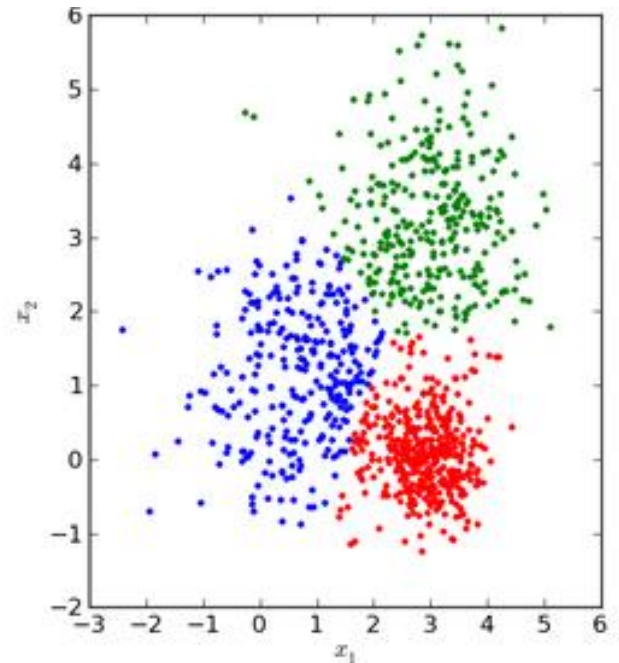
- Given data  $\{x_i, i = 1, 2, \dots, n\}$
- MLE of the data given a 1D Gaussian model, gives:
  - $\hat{\mu} = \frac{\sum_i x_i}{n}, \hat{\sigma}^2 = \frac{\sum_i (x_i - \mu)^2}{n}$
  - How?
    - Compute prob of data given model:  $P(D|M)$
    - We often compute the log probability instead
    - Differentiate and set = 0. Why?
- MLE of multivariate Gaussian:
  - $\mu = \frac{1}{n} \sum_i x_i$
  - $\Sigma = \frac{1}{n} \sum_i (x_i - \mu)(x_i - \mu)^T$

Note: we do direct optimisation here: we only need to compute the MLE



# Generating More Structured Data

- Assume we want 2D data from 3 classes = {A, B, C}
- Define 3 Gaussians in 2D, label them A, B, C
  - Define a  $\mu$  and  $\Sigma$  for each
- Define a prior for each (how likely is this class)
  - $P(A) + P(B) + P(C) = 1$
- To generate a data point:
  - Randomly pick a class  $y$  using the prior
  - Randomly sample a point  $\mathbf{x}$  from model  $y$
  - Add  $(\mathbf{x}, y)$  to dataset



# Bayesian Classification

- Recall: we're trying to learn a function  $y = f(x)$ 
  - $y$  is a class label
  - $x = (x_1, x_2, \dots, x_n)$  is a vector of attributes/features
- Probabilistic classification:
  - What is the most probable class given our data?
  - $y^* = \arg \max_y P(y|x)$
- Class probabilities:
  - $P(y|x) = \frac{P(x|y)P(y)}{\sum_{y'} P(x|y')P(y')}$

# Understanding the Pieces

- $P(y|x) = \frac{P(x|y)P(y)}{P(x) = \sum_{y'} P(x|y')P(y')}$
- $P(y)$ : prior probability of each class
  - Some classes are more common than others
  - Pigeons are more common than eagles
- $P(x|y)$ : class conditional model
  - How likely is class  $y$  to generate observation  $x$ ?
- $P(x)$ : normalization term
  - Outliers may have low probability in every class

# Expensive Learning



- Consider  $P(x|y)$  for some class  $y$ 
  - We would need to learn this for all  $x$  and  $y$
- Assume  $n=100$ , i.e.  $x = [x_1 \ x_2 \ \dots \ x_{100}]$
- Then  $P(x|y) = P(x_1, x_2, \dots, x_{100}|y)$ 
  - This means that for each  $y$ , we need a 100-dim table to represent the joint probabilities!
  - The curse of dimensionality!
    - What are the odds we'll see every possibility during training??

# Conditional Independence

- We know:  $P(X,Y) = P(X) P(Y|X)$
- And so for a class  $C$ :  $P(X,Y|C) = P(X|C)P(Y|X,C)$
- What if we made an **assumption**:
  - $P(Y|X,C) = P(Y|C)$
- This means that knowing the value of  $X$  makes no difference to the value of  $Y$  **as long as we know class  $C$**
- We say that  $X$  and  $Y$  are conditionally independent given  $C$

# Conditional Independence

- In general, for  $x = (x_1, x_2, \dots, x_n)$ :
  - $x_1, x_2, \dots, x_n$  are conditionally independent given  $c$  iff
    - $P(x|c) = \prod_{i=1}^n P(x_i|c)$
- So considering the learning problem, instead of learning an  $n$ -dimensional table, we now instead need to learn  $n$  1-d tables 😊

# Conditional Independence Example

- Imagine Alice (A) and Bob (B) are late for work!
- These events are most likely not independent
- But: they may be independent if we know that there was a train strike (T).
- $P(A, B) \neq P(A)P(B)$
- $P(A, B|T) = P(A|T)P(B|T)$
- T explains all the dependence between A and B.



# Naïve Bayes Classifier

- The Naïve Bayes (NB) model is given by the **conditional independence assumption** applied to Bayes' rule:

$$P(c|x) = \frac{\prod_{i=1}^n P(x_i|c)P(c)}{P(x)}$$

for attribute vector  $x = (x_1, x_2, \dots, x_n)$  and class  $c$

- We need to learn  $P(x_i|c)$  and  $P(c)$  from the training data. The form of  $P(x_i|c)$  is given.
- We then want to find the best  $c$  for new data  $x$ , which is  $c^* = \arg \max_c P(c|x)$ 
  - This is the **maximum a posteriori** (or MAP) solution

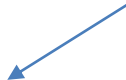


# Discrete Example: Learning

- Identify spam vs ham
- Training emails:

“Bag of words” representation:

Treat each email as a long vector, with 0/1 indicating absence/presence of a word



<b>E1</b>	“Buy this online today”	Ham
<b>E2</b>	“Send us money today”	Spam
<b>E3</b>	“Send money today”	Spam
<b>E4</b>	“Buy online today”	Spam
<b>E5</b>	“Send us money”	Ham
<b>E6</b>	“Send this money”	Spam

- Priors?
  - $P(\text{spam}) = 4/6$
  - $P(\text{ham}) = 2/6$

Class conditional models:

Word	Spam	Ham
Buy	1/4	1/2
This	1/4	1/2
Online	1/4	1/2
Send	3/4	1/2
Us	1/4	1/2
Money	3/4	1/2
Today	3/4	1/2

# Discrete Example: Inference

- $P(\text{spam}) = 4/6$
- $P(\text{ham}) = 2/6$

$$P(c|x) = \frac{\prod_{i=1}^n P(x_i|c) P(c)}{P(x)}$$

- New email:
  - E7 = “Buy money online today”
  - Encode as (1,0,1,0,0,1,1)

- $P(E7|\text{spam})$ 
  - $= \frac{1}{4} \left(1 - \frac{1}{4}\right) \frac{1}{4} \left(1 - \frac{3}{4}\right) \left(1 - \frac{1}{4}\right) \frac{3}{4} \frac{3}{4} = 0.0049$
- $P(E7|\text{ham})$ 
  - $= \frac{1}{2} \left(1 - \frac{1}{2}\right) \frac{1}{2} \left(1 - \frac{1}{2}\right) \left(1 - \frac{1}{2}\right) \frac{1}{2} \frac{1}{2} = 0.0078$

- So:  $P(\text{spam}|E7)$ 
  - $= \frac{0.0049 \times \frac{4}{6}}{0.0049 \times \frac{4}{6} + 0.0078 \times \frac{2}{6}} = 0.5586$

Class conditional models:

Word	Spam	Ham
Buy	1/4	1/2
This	1/4	1/2
Online	1/4	1/2
Send	3/4	1/2
Us	1/4	1/2
Money	3/4	1/2
Today	3/4	1/2

Note:  $P(\text{ham}|E7) = 0.4414$   
 $= 1 - 0.5586$

# Problems with Naïve Bayes

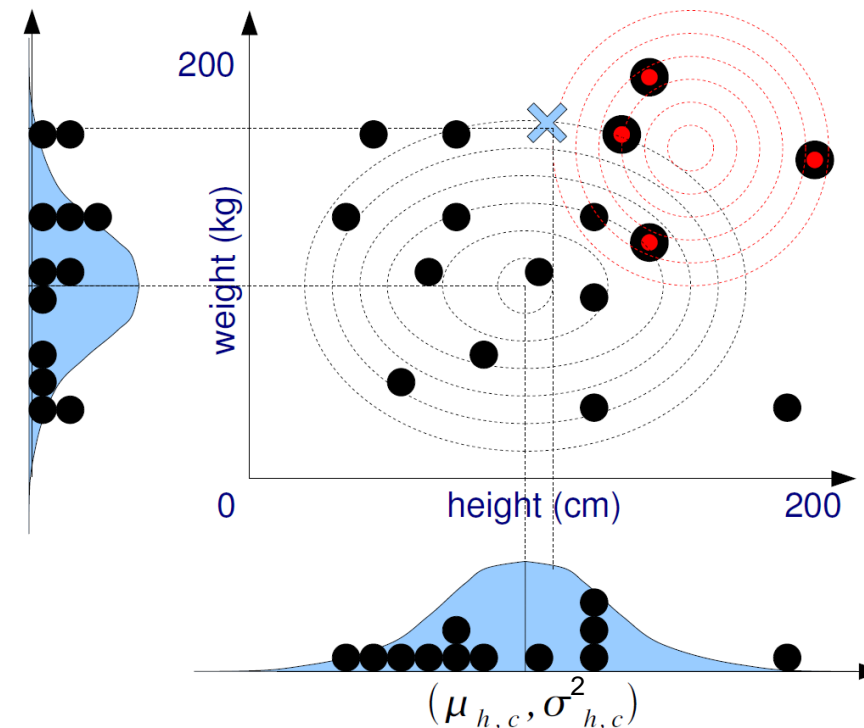
- **Assumes word independence**
  - Every word contributes independently to  $P(\text{spam} | \text{email})$
  - Fool NB by adding many ham-like words to spam emails
  - Need more complex models to work around this
- **Zero-frequency problem**
  - Some words may never appear in the training data!
  - Never allow zero probabilities
    - Laplace smoothing: **add small positive number to each count**
  - Very common: Zipf's law

# Smoothing

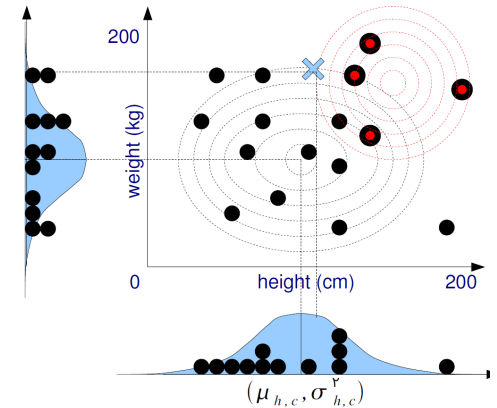
- What if the query email contained the word “robot”, but we had **never seen that in any spam email** (but in some ham ones)?
- $P(email|spam)$  would contain a factor = 0
  - And so  $P(email|spam) = 0$
- Why is this bad?
  - We may just have not seen a training spam email like this.  
**Instead we're implying it is impossible!**
- Laplacian smoothing:
  - Without:  $P(robot|spam) = \frac{\#\{X_{robot}=1, Y=spam\}}{\#\{Y=spam\}}$
  - With:  $P(robot|spam) = \frac{\#\{X_{robot}=1, Y=spam\} + k}{\#\{Y=spam\} + n_{robot}k}$
  - $n_{robot}$  = number of values  $X_{robot}$  can take ( $\{0,1\} = 2$ )
  - $k = 1$  (usually)

# Continuous Example

- Distinguish children from adults based on size
  - Classes: {a, c}
  - Attributes: height (cm), weight (kg)
  - Training data:
    - $\{h_i, w_i, y_i\}$
  - 4 adults, 12 children



# Continuous Example



- Class priors?

- $P(a) = \frac{4}{4+12} = 0.25, P(c) = 0.75$

- Model for adults (assume independence)?

- $height \sim N(\mu_{h,a}, \sigma_{h,a}^2)$

- $\mu_{h,a} = \frac{1}{4} \sum_{i:y_i=a} h_i$

- $\sigma_{h,a}^2 = \frac{1}{4} \sum_{i:y_i=a} (h_i - \mu_{h,a})^2$

- $weight \sim N(\mu_{w,a}, \sigma_{w,a}^2)$

- Similarly for children

- $height \sim N(\mu_{h,c}, \sigma_{h,c}^2), weight \sim N(\mu_{w,c}, \sigma_{w,c}^2)$

# Example

- $P(a) = 0.25, P(c) = 0.75$

- $p(h_x|c) = \frac{1}{\sqrt{2\pi\sigma_{h,c}^2}} \exp\left\{-\frac{1}{2}\left(\frac{(h_x - \mu_{h,c})^2}{\sigma_{h,c}^2}\right)\right\}$

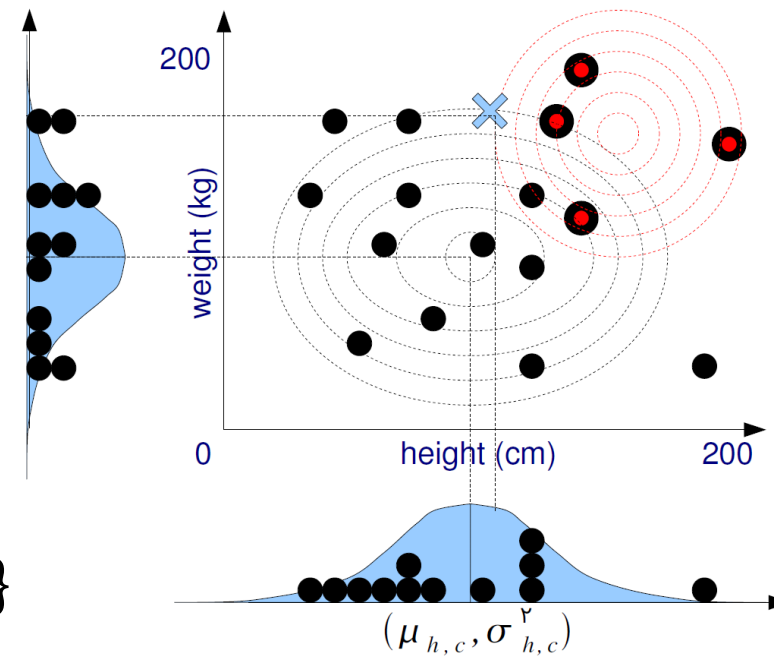
- $p(w_x|c) = \frac{1}{\sqrt{2\pi\sigma_{w,c}^2}} \exp\left\{-\frac{1}{2}\left(\frac{(w_x - \mu_{w,c})^2}{\sigma_{w,c}^2}\right)\right\}$

- Same for  $p(h_x|a)$  and  $p(w_x|a)$

- $p(x|c) = p(h_x|c)p(w_x|c)$

- $p(x|a) = p(h_x|a)p(w_x|a)$

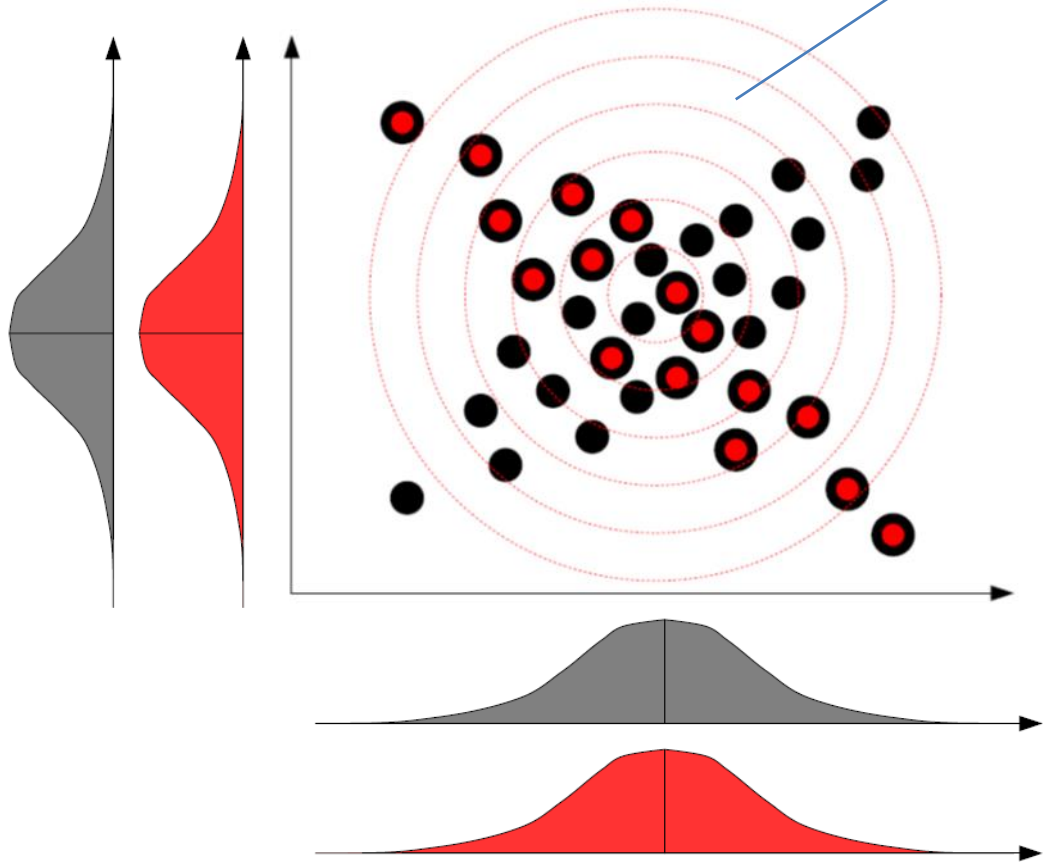
- $p(c|x) = \frac{p(x|c)p(c)}{p(x|c)p(c) + p(x|a)p(a)}$



$$P(c|x) = \frac{\prod_{i=1}^n P(x_i|c) P(c)}{P(x)}$$

# Being too Naïve

Easy to classify based  
on joint distribution  
 $P(x_1, x_2|c)$



Impossible to classify  
based on marginal  
distributions  $P(x_1|c)$  or  
 $P(x_2|c)$ :

Independence assumptions  
do not hold!



# Testing

- How do we report accuracy of a classifier?
- Confusion matrix:

		Actual class		
		Cat	Dog	Rabbit
Predicted class	Cat	5	2	0
	Dog	3	3	2
	Rabbit	0	1	11

We want these diagonal elements as high as possible, and the others as low as possible

This is evaluated using the test data, NOT the training data!

# Verdict



- Independence assumption is very naïve:
  - Usually doesn't hold, but still useful
  - Need more sophisticated Bayesian methods (not in this course)
- But:
  - Easy to program. Simple to understand.
  - Fast to train and use.
  - Probabilistic: can deal with uncertainty

# Recap

- Probability
  - RVs, joint/conditional/marginal distributions
  - Sum, product, Bayes' rules
  - Generating data
  - Estimating distributions
  - Likelihood & max likelihood
  - The Gaussian distribution
- Naïve Bayes
  - Conditional independence
  - The naïve Bayes classifier
  - Discrete and continuous examples
  - Smoothing