Machine Learning – COMS3007

Naïve Bayes and Probability Theory

Benjamin Rosman

Based heavily on course notes by
Chris Williams and Victor Lavrenko,
Amos Storkey, and Clint van Alten

Sometimes we want to know the probability of an outcome

headache	*runny nose*	*fever*	*flu*
N	*Y*	*Y*	*N*
Y	*N*	*N*	*N*
N	*N*	*N*	*N*
Y	*Y*	*Y*	*Y*
Y	*Y*	*N*	*Y*
N	*N*	*Y*	*Y*

headache	*runny nose*	*fever*	*flu*
Y	*N*	*Y*	*?*

Why?

Probability

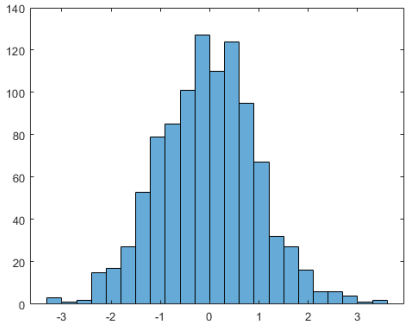
- Why do we care about probability?
- Not always confident in our statements
  - Features: how likely are our features?
  - Outcomes: how confident is our decision?
  - Predictions: tend not to be certain

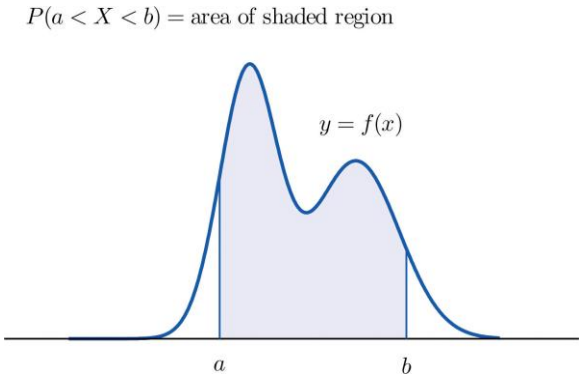
Random Variables

- 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: $0 \leq amount \leq 1{,}000{,}000$
- We are interested in the probability of some event occurring

Random Variables

- RVs: discrete or continuous





- 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 achieve by normalisation: divide by the total

Joint Distributions

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

	Y = ham	Y = spam
X = yes	0.01	0.25
X = no	0.49	0.25

- Note: sum = 1, as one of these four combinations must have happened
- Notation: p(X = yes, Y = 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(email doesn't contain "millions") = P(X=no) = ?
  - P(email isn't spam) = P(Y=ham) = ?

	Y = ham	Y = spam
X = yes	0.01	0.25
X = no	0.49	0.25

Conditional Probabilities

- We often what 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=yes|Y=ham) = ?
  "probability an email contains 'millions'
  IF I ALREADY KNOW IT IS HAM"
- Note $\sum_x p(X = x|Y = y) = 1$, for all y

	Y = ham	Y = spam
X = yes	0.01	0.25
X = no	0.49	0.25

Bayes' Rule

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

Rev. Thomas Bayes
(1701 – 1761)

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

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?

Likelihood of seeing data X under label Y

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

Posterior probability of label Y having seen data X

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

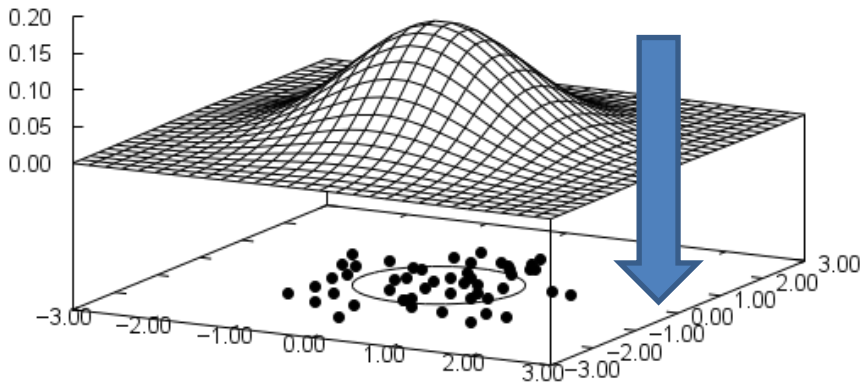
Think of this as:
I thought the prob of y was p(y)
Then I saw x
Now I think it is p(y|x)

Probability of data X under any Y (normalisation)

What to do with distributions?

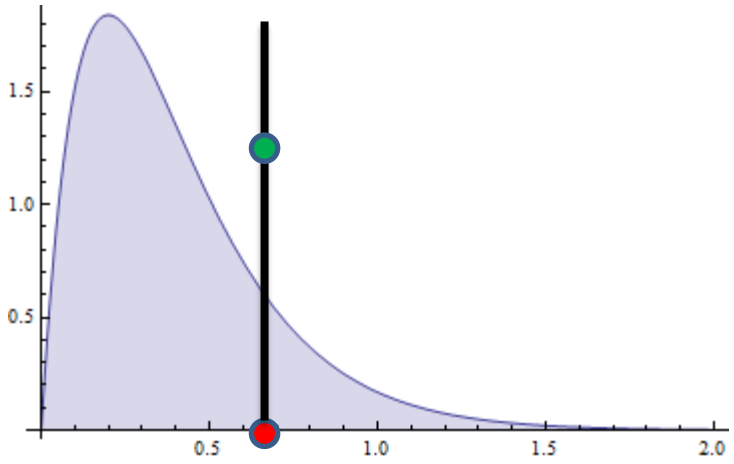
- Generate data x from distribution p (sampling)
  - $x \sim p$
  - Each data point generated **i**ndependently and is **i**dentically **d**istributed: iid

  - 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)



Sampling

https://thispersondoesnotexist.com/

SYSTEM PROMPT (HUMAN-WRITTEN)
In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

MODEL COMPLETION (MACHINE-WRITTEN)
The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

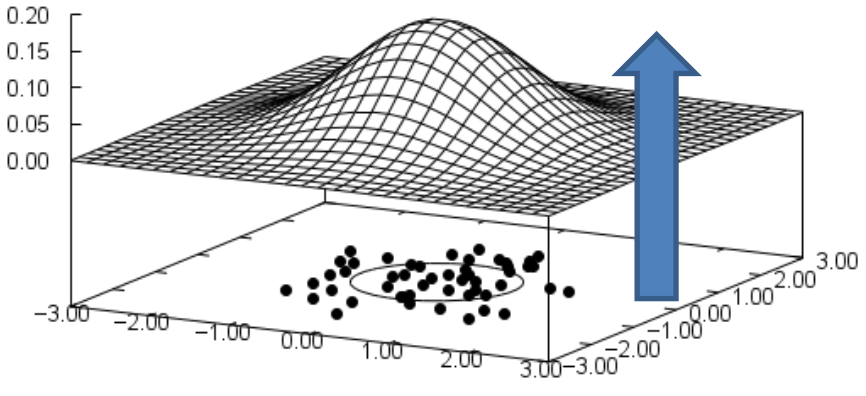
Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.
…

https://blog.openai.com/better-language-models/

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!

Bayesian Classification

- Recall: we're trying to learn a function y = f(x)
  - y is a class label
  - $x = (x_1, x_2, \ldots, x_n)$ 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:
    - $P(y|x) = \frac{P(x|y)P(y)}{\sum_{y'} P(x|y')P(y')}$ (Bayes' rule)

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. $x = [x_1\ x_2\ \ \ldots x_{100}]$
- Then $P(x|y) = P(x_1, x_2, \ldots, x_{100}|y)$
  - 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 $x = (x_1, x_2, \ldots, x_n)$:
  - $x_1, x_2, \ldots, x_n$ 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 $x = (x_1, x_2, \ldots, x_n)$ 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

E1	"Buy this online today"	Ham
E2	"Send us money today"	Spam
E3	"Send money today"	Spam
E4	"Buy online today"	Spam
E5	"Send us money"	Ham
E6	"Send this money"	Spam

- Priors?
  - P(spam) = 4/6
  - P(ham) = 2/6

"money" is in ¾ of spam emails

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(spam) = 4/6
- P(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)

- $P(E7|spam)$
  - $= \frac{1}{4}\left(1-\frac{1}{4}\right)\frac{1}{4}\left(1-\frac{3}{4}\right)\left(1-\frac{1}{4}\right)\frac{3}{4}\frac{3}{4} = 0.0049$
- $P(E7|ham)$
  - $= \frac{1}{2}\left(1-\frac{1}{2}\right)\frac{1}{2}\left(1-\frac{1}{2}\right)\left(1-\frac{1}{2}\right)\frac{1}{2}\frac{1}{2} = 0.0078$
- So: $P(spam|E7)$
  - $= \frac{0.0049\times\frac{4}{6}}{0.0049\times\frac{4}{6}+0.0078\times\frac{2}{6}} = 0.5586$

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(ham|E7) = 0.4414
= 1 – 0.5586

Problems with Naïve Bayes

- Assumes word independence
  - Every word contributes independently to P(spam|email)
  - Fool NB by adding many ham-like words to spam emails
  - Need more complex models to work around this

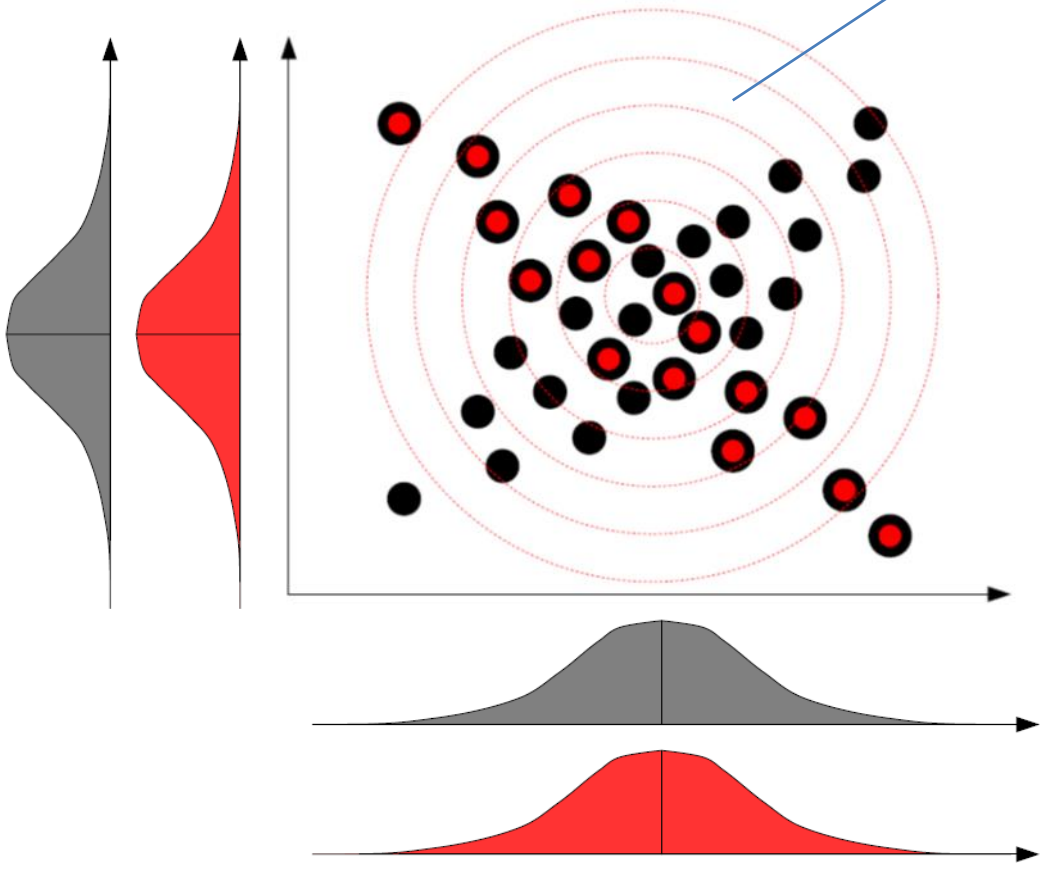
- Zero-frequency problem
  - Some words may never appear in the training data!
  - Never allow zero probabilities
    - Laplace smoothing: add small positive number to each count
  - Very common: Zipf's law

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)

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

- Confusion matrix:

This means that 3 examples that were actually cats were incorrectly misclassified as dogs

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

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

$$Accuracy = \frac{diagonals}{total} = \frac{5+3+11}{5+2+3+3+2+1+11} = \frac{19}{27} = 0.704$$

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
- Naïve Bayes
  - Conditional independence
  - The naïve Bayes classifier
  - Discrete examples
  - Smoothing