1. (**Naive Bayes, revision of COMS10003**) Suppose a naive Bayesian spam filter uses a vocabulary consisting of the words 'Viagra', 'CONFIDENTIAL', 'COMS21202' and 'Gaussian', and has estimated the class-conditional likelihoods of these words occurring in spam and non-spam emails as in Table 1.

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word \mid spam)$	$P(word \mid \neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email \mid spam)}{P(email \mid \neg spam)}$.

More on decision rules

The following decision rules are equivalent:

- if **p(lightness|sea bass) ≥ p(lightness|salmon)** then sea bass else salmon (*maximum likelihood, ML*)
- if $\frac{p(\text{lightness} \mid \text{sea bass})}{p(\text{lightness} \mid \text{salmon})} \geq 1$ **then** sea bass **else** salmon (*likelihood ratio*)
- $\arg\max_{\omega \in \{\text{bass}, \text{salmon}\}} p(\text{lightness} \mid \omega)$ (works for more than two different classes)

With non-uniform prior probabilities (class probabilities) we should use

- if **p(lightness|sea bass)P(sea bass) ≥ p(lightness|salmon)P(salmon)** then sea bass else salmon (*maximum a posteriori or MAP*)
- **if** $\frac{p(\text{lightness} \mid \text{sea bass})}{p(\text{lightness} \mid \text{salmon})} \geq \frac{P(\text{salmon})}{P(\text{sea bass})}$ **then** sea bass **else** salmon
- $\arg\max_{\omega \in \{\text{bass}, \text{salmon}\}} p(\text{lightness} \mid \omega) P(\omega)$

The Naive-Bayes classifier

"Naively" assumes independent features within each class:

$$P(\mathbf{x} \mid \omega) = P(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \mid \omega)$$

- **unconditional independence:** knowledge about one feature does not tell us anything about the others
- **class-conditional independence:** within each class, knowledge about one feature does not tell us anything about the others

$$\approx P(x_1 \mid \omega) P(x_2 \mid \omega) \ldots P(x_d \mid \omega) = \prod_{i=1}^{d} P(x_i \mid \omega)$$

Now the MAP (*Maximum A Posteriori*) decision rule becomes

$$\arg\max_{\omega} P(\omega \mid \mathbf{x}) = \arg\max_{\omega} P(\mathbf{x} \mid \omega) P(\omega) \approx \arg\max_{\omega} \left(\prod_{i=0}^{d} P(x_i \mid \omega) \right) P(\omega)$$

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email|spam)}{P(email|\neg spam)}$.

(a) We use $P(word|spam)$ for words that occur in a spam email, and $P(\neg word|spam) = 1 - P(word|spam)$ for words that don't occur in a spam email (similarly for non-spam). We make the naive-Bayesian assumption that the occurrence or absence of words is independent within each class, and so we can decompose the likelihood ratio as follows:

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email|spam)}{P(email|\neg spam)}$.

(a) We use $P(word|spam)$ for words that occur in a spam email, and $P(\neg word|spam) = 1 - P(word|spam)$ for words that don't occur in a spam email (similarly for non-spam). We make the naive-Bayesian assumption that the occurrence or absence of words is independent within each class, and so we can decompose the likelihood ratio as follows:

$$\frac{P(A|spam)}{P(A|\neg spam)}$$

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email|spam)}{P(email|\neg spam)}$.

(a) We use $P(word|spam)$ for words that occur in a spam email, and $P(\neg word|spam) = 1 - P(word|spam)$ for words that don't occur in a spam email (similarly for non-spam). We make the naive-Bayesian assumption that the occurrence or absence of words is independent within each class, and so we can decompose the likelihood ratio as follows:

$$\frac{P(A|spam)}{P(A|\neg spam)} = \frac{P(Viagra|spam)P(\neg CONFIDENTIAL|spam)P(\neg COMS21202|spam)P(\neg Gaussian|spam)}{P(Viagra|\neg spam)P(\neg CONFIDENTIAL|\neg spam)P(\neg COMS21202|\neg spam)P(\neg Gaussian|\neg spam)}$$

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email|spam)}{P(email|\neg spam)}$.

(a) We use $P(word|spam)$ for words that occur in a spam email, and $P(\neg word|spam) = 1 - P(word|spam)$ for words that don't occur in a spam email (similarly for non-spam). We make the naive-Bayesian assumption that the occurrence or absence of words is independent within each class, and so we can decompose the likelihood ratio as follows:

$$\begin{aligned}
\frac{P(A|spam)}{P(A|\neg spam)} &= \frac{P(Viagra|spam)P(\neg CONFIDENTIAL|spam)P(\neg COMS21202|spam)P(\neg Gaussian|spam)}{P(Viagra|\neg spam)P(\neg CONFIDENTIAL|\neg spam)P(\neg COMS21202|\neg spam)P(\neg Gaussian|\neg spam)} \\
&= \frac{0.20}{0.01} \times \frac{1-0.30}{1-0.05} \times \frac{1-0.02}{1-0.20} \times \frac{1-0.05}{1-0.10} \\
&= 20 \times 14/19 \times 49/40 \times 19/18 = 19.06
\end{aligned}$$

What is the most *likely* class for classifying e-mail *A*? Why?

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(a) Determine the most likely class of each of these emails by calculating the likelihood ratios $\frac{P(email|spam)}{P(email|\neg spam)}$.

(a) We use $P(word|spam)$ for words that occur in a spam email, and $P(\neg word|spam) = 1 - P(word|spam)$ for words that don't occur in a spam email (similarly for non-spam). We make the naive-Bayesian assumption that the occurrence or absence of words is independent within each class, and so we can decompose the likelihood ratio as follows:

$$\begin{aligned}
\frac{P(A|spam)}{P(A|\neg spam)} &= \frac{P(Viagra|spam)P(\neg CONFIDENTIAL|spam)P(\neg COMS21202|spam)P(\neg Gaussian|spam)}{P(Viagra|\neg spam)P(\neg CONFIDENTIAL|\neg spam)P(\neg COMS21202|\neg spam)P(\neg Gaussian|\neg spam)} \\
&= \frac{0.20}{0.01} \times \frac{1-0.30}{1-0.05} \times \frac{1-0.02}{1-0.20} \times \frac{1-0.05}{1-0.10} \\
&= 20 \times 14/19 \times 49/40 \times 19/18 = 19.06
\end{aligned}$$

Now you will…

- Use the same **Likelihood Ratio** rule to classify e-mails *B* and *C*

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(b) Now assume that typically 10% of your emails are spam. Using MAP estimation, investigate how this affects your predictions.

(b) We now need to take the prior probability of spam into account, and consider the posterior odds

$$\frac{P(spam|email)}{P(\neg spam|email)} = \frac{P(email|spam)P(spam)}{P(email|\neg spam)P(\neg spam)}$$

For e-mail *A:*

$$\frac{P(spam\,|A)}{P(\neg spam\,|A)} = \mathbf{19.06}\,\frac{0.1}{0.9} = 2.11$$

Table 1: Class-conditional likelihoods for words in the vocabulary.

word	$P(word\|spam)$	$P(word\|\neg spam)$
Viagra	0.20	0.01
CONFIDENTIAL	0.30	0.05
COMS21202	0.02	0.20
Gaussian	0.05	0.10

Consider three test emails as follows: A contains the word 'Viagra' but none of the others; B contains the word 'CONFIDENTIAL' but none of the others; C contains the words 'COMS21202' and 'Gaussian' but none of the others.

(b) Now assume that typically 10% of your emails are spam. Using MAP estimation, investigate how this affects your predictions.

(b) We now need to take the prior probability of spam into account, and consider the posterior odds

$$\frac{P(spam|email)}{P(\neg spam|email)} = \frac{P(email|spam)P(spam)}{P(email|\neg spam)P(\neg spam)}$$

For e-mail *A*:

$$\frac{P(spam\,|A)}{P(\neg spam\,|A)} = \mathbf{19.06}\,\frac{0.1}{0.9} = 2.11$$

Higher *posterior* ratio than 1, still classified as spam!

4. (**Decision trees**) Suppose we have a training set of 32 spam emails and 32 non-spam emails, and the numbers of emails containing particular words are as in Table 2. We want to build a decision tree using these words as boolean features: if the word occurs in an email the feature is true, else it is false. Which feature results in the best split, as measured by information gain?

Table 2: Numbers of spam and non-spam emails containing particular words.

word	spam	non-spam
Viagra	15	1
CONFIDENTIAL	28	4
COMS21202	1	15
Gaussian	4	12

4. **(Decision trees)** Suppose we have a training set of 32 spam emails and 32 non-spam emails, and the numbers of emails containing particular words are as in Table 2. We want to build a decision tree using these words as boolean features: if the word occurs in an email the feature is true, else it is false. Which feature results in the best split, as measured by information gain?

Table 2: Numbers of spam and non-spam emails containing particular words.

word	spam	non-spam
Viagra	15	1
CONFIDENTIAL	28	4
COMS21202	1	15
Gaussian	4	12

Answer:
Information gain is calculated in this case as the entropy of the training set minus the weighted average entropy after splitting on the feature. The training set has entropy $-(32/64)\log_2(32/64) - (32/64)\log_2(32/64) = -\log_2(1/2) = 1$ *(i.e. a uniform distribution: no calculation necessary!)*

Information gain:

$$Imp(Parent) - \sum_{i=1}^{k} \frac{n_i}{N} Imp(Child_i)$$

Entropy (a specific measure for impurity in a node):

$$\sum_{j=1}^{c} -p_j \log_2 p_j$$

4. (**Decision trees**) Suppose we have a training set of 32 spam emails and 32 non-spam emails, and the numbers of emails containing particular words are as in Table 2. We want to build a decision tree using these words as boolean features: if the word occurs in an email the feature is true, else it is false. Which feature results in the best split, as measured by information gain?

Table 2: Numbers of spam and non-spam emails containing particular words.

word	spam	non-spam
Viagra	15	1
CONFIDENTIAL	28	4
COMS21202	1	15
Gaussian	4	12

Answer:

Information gain is calculated in this case as the entropy of the training set minus the weighted average entropy after splitting on the feature. The training set has entropy $-(32/64)\log_2(32/64) - (32/64)\log_2(32/64) = -\log_2(1/2) = 1$ *(i.e. a uniform distribution: no calculation necessary!)*

- *16 emails in the training set contain the word 'Viagra', 15 of which are spam and 1 of which is non-spam. The entropy of those emails is* $-(15/16)\log_2(15/16) - (1/16)\log_2(1/16) = 0.337$.
  The remaining 48 emails in the training set do not contain the word 'Viagra', 17 of which are spam and 31 of which are non-spam. The entropy of those emails is $-(17/48)\log_2(17/48) - (31/48)\log_2(31/48) = 0.938$.

Analysing the quality of the split based on the first attribute

4. (**Decision trees**) Suppose we have a training set of 32 spam emails and 32 non-spam emails, and the numbers of emails containing particular words are as in Table 2. We want to build a decision tree using these words as boolean features: if the word occurs in an email the feature is true, else it is false. Which feature results in the best split, as measured by information gain?

Table 2: Numbers of spam and non-spam emails containing particular words.

word	spam	non-spam
Viagra	15	1
CONFIDENTIAL	28	4
COMS21202	1	15
Gaussian	4	12

Answer:

Information gain is calculated in this case as the entropy of the training set minus the weighted average entropy after splitting on the feature. The training set has entropy $-(32/64)\log_2(32/64)-(32/64)\log_2(32/64) = -\log_2(1/2) = 1$ *(i.e. a uniform distribution: no calculation necessary!)*

- *16 emails in the training set contain the word 'Viagra', 15 of which are spam and 1 of which is non-spam. The entropy of those emails is* $-(15/16)\log_2(15/16)-(1/16)\log_2(1/16) = 0.337$.
  The remaining 48 emails in the training set do not contain the word 'Viagra', 17 of which are spam and 31 of which are non-spam. The entropy of those emails is $-(17/48)\log_2(17/48)-(31/48)\log_2(31/48) = 0.938$.
  The weighted average of these two entropies is $(16/64)0.337+(48/64)0.938 = 0.788$. *The decrease in entropy before and after splitting is thus* $1-0.788 = 0.212$.

$$Imp(Parent) - \sum_{i=1}^{k} \frac{n_i}{N} Imp(Child_i)$$

4. **(Decision trees)** Suppose we have a training set of 32 spam emails and 32 non-spam emails, and the numbers of emails containing particular words are as in Table 2. We want to build a decision tree using these words as boolean features: if the word occurs in an email the feature is true, else it is false. Which feature results in the best split, as measured by information gain?

Table 2: Numbers of spam and non-spam emails containing particular words.

word	spam	non-spam
Viagra	15	1
CONFIDENTIAL	28	4
COMS21202	1	15
Gaussian	4	12

Now you will…

- Calculate the Information gain for the other three attributes and **determine the best split**

6. (**Nearest-neighbour classification**) Assume $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and $\mathbf{z} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$ are three training instances labelled $+$, $+$ and $-$, respectively. Derive the k-nearest neighbour prediction for the test points $\mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{q} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, using Euclidean distance, for $k = 1$ and $k = 3$.

6. (**Nearest-neighbour classification**) Assume $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and $\mathbf{z} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$ are three training instances labelled $+$, $+$ and $-$, respectively. Derive the k-nearest neighbour prediction for the test points $\mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{q} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, using Euclidean distance, for $k = 1$ and $k = 3$.

- *($k = 1$) We have $L_2(\mathbf{p}, \mathbf{x}) = 1$, $L_2(\mathbf{p}, \mathbf{y}) = \sqrt{5}$, and $L_2(\mathbf{p}, \mathbf{z}) = \sqrt{10}$. So $\mathbf{x}$ is the nearest neighbour of $\mathbf{p}$, and we predict $+$.*

6. (**Nearest-neighbour classification**) Assume $\mathbf{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $\mathbf{y} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ and $\mathbf{z} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$ are three training instances labelled $+$, $+$ and $-$, respectively. Derive the k-nearest neighbour prediction for the test points $\mathbf{p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\mathbf{q} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$, using Euclidean distance, for $k = 1$ and $k = 3$.

- *($k = 1$) We have $L_2(\mathbf{p}, \mathbf{x}) = 1$, $L_2(\mathbf{p}, \mathbf{y}) = \sqrt{5}$, and $L_2(\mathbf{p}, \mathbf{z}) = \sqrt{10}$. So $\mathbf{x}$ is the nearest neighbour of $\mathbf{p}$, and we predict $+$.*

Now you will …

- Calculate the distance between **q** and **x, y, z** for *k=1*
- Derive the 1-NN prediction for **q**
- Derive the 3-NN prediction for **p** and **q**