ECS171: Machine Learning

L11: Probabilistic methods II

Instructor: Prof. Maike Sonnewald
TAs: Pu Sun & Devashree Kataria

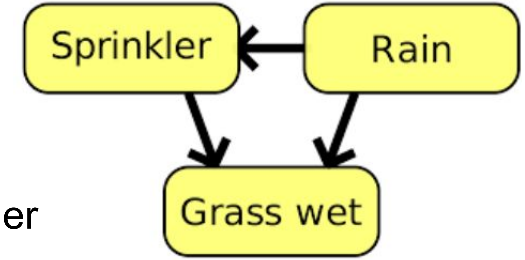

MOCO Amsterdam garden

Intended Learning Outcomes

- Understand the concept and calculate components of Bayes Theorem
    - Conditional, Marginal, Joint etc
    - Be able to apply the chain rule
- Describe how Directed Acyclic Graphs are used for a Bayes Network and how this differs from the Networks dealt with previously
    - Understand and apply the concept of probabilities in Networks
- Describe the distinction between Naive Bayes and a Bayesian network
    - Including concepts like conditional independence
- Calculate the Discriminant function in the context of a simple Naive Baye network

Probabilistic Graphical Models (PGM): Bayes Network

- Aka Belief Network and Decision Network
- In a nutshell: A network representing probabilistic dependencies between variables
    - The connections/dependencies can be described as a ‘graph’
    - Flexible models capable of capturing complex relationships
- Applications include:
    - Diagnostics, reasoning, causal modeling, decision making under uncertainty, anomaly detection, natural language processing
- We will cover a simple Bayesian Network is called “Naïve Bayes” with a simpler structure and much stronger assumptions about the independence of features.



https://goo.gl/images/Q3wuzF

Bayes theorem recap

If our data (input vector), and our outcome (e.g. labels) is A

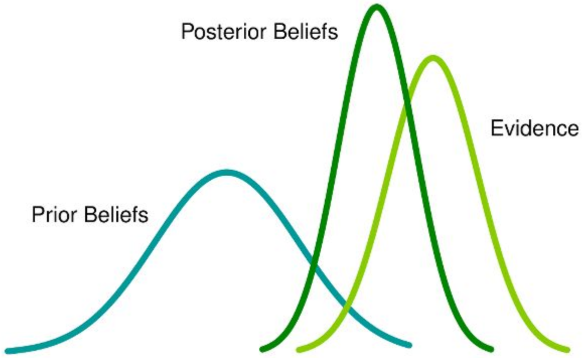
Conditional or posterior Probability: P(θ |data)
Probability of one (or more) event given the occurrence of another event, e.g. P(θ given data) or P(θ | data).

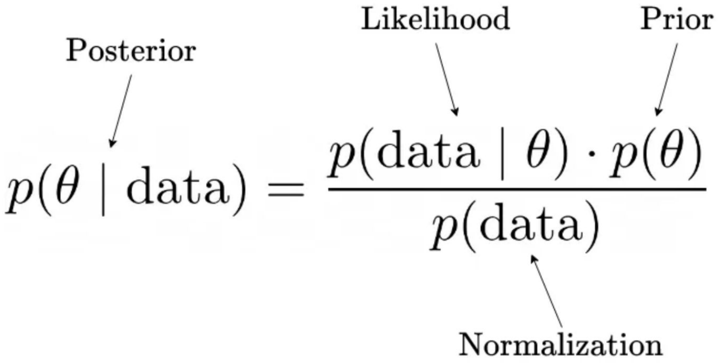
Likelihood: P(data|θ)

Marginal Probability or Prior: P(θ)
The probability of an event irrespective of the outcomes of other random variables

‘Evidence’: P(data)
What actually occurred

Joint Probability: P(θ, data) also written P(θ and data)
Probability of two (or more) simultaneous events

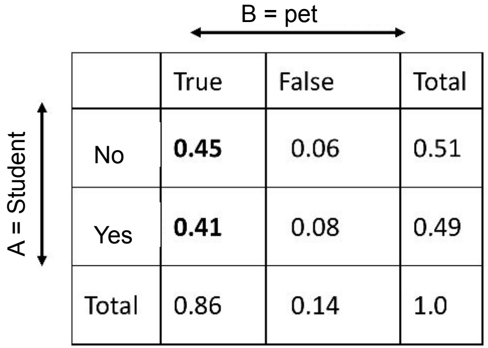




$$p(\theta \mid \text{data}) = \frac{p(\text{data} \mid \theta) \cdot p(\theta)}{p(\text{data})}$$

Recap Conditional Probability: P(θ|data)

- Conditional probability is a measure of the likelihood of an event occurring, given that another event (or set of events) has already occurred: P(A|B) = P(A,B) / P(B)
- Example:
    - What is the probability of a randomly selected person is a student (A) given that they own a pet (B)?
    - P(A=Student|B=True) = P(A=Student,B=True)/ P(B=True) = .41/ (.45+.41) = .41/ .86 = .477 or 47.7%



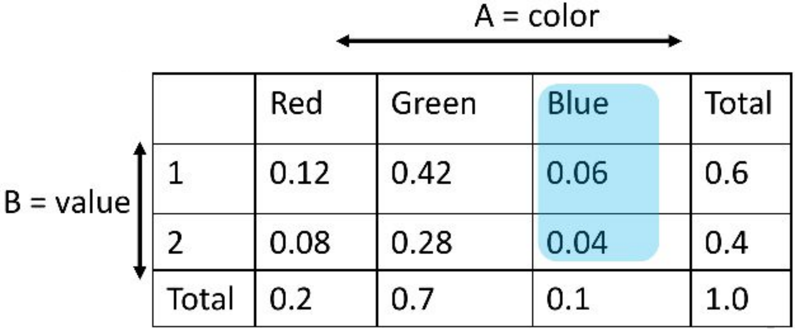
B = pet

A = Student

	True	False	Total
No	**0.45**	0.06	0.51
Yes	**0.41**	0.08	0.49
Total	0.86	0.14	1.0

Recap Marginal Probability : P(data)

- The probability of a single event or variable without considering the effects of any other events or variables
- The marginal probability of an event A can be calculated by summing the joint probabilities of A occurring with all possible outcomes of another event B
- Example:
    - What is the probability that a card drawn from a pile of cards is “blue”.
    - $P(A=blue) = \sum P(A, B) = P(A=Blue, B=1) + P(A=Blue, B=2) =.06+ .04 =.1$ or 10%



	Red	Green	Blue	Total
1	0.12	0.42	0.06	0.6
2	0.08	0.28	0.04	0.4
Total	0.2	0.7	0.1	1.0

Recap Joint Probability: P(θ, data) or P(θ and data)

- If A and B are two events, P(A,B) represents the probability of A and B occurring simultaneously
- Independent:
  - Occurrence of one event **does not** affect the probability of the other event: P(A)*P(B)
- Dependent:
  - Occurrence of one **does** affect the probability of the other event: P(A,B) = P(A|B) * P(B)
- Symmetric: P(A,B) = P(B,A) = P(B|A) * P(A)
- Example:
  - The joint probability that it rains (A) and the sky is cloudy (B), P(A,B)
  - If P(A=rain | B=cloudy) = 1/13 , P(B= cloudy) = ½ then, P(A,B) = 1/13 x 1/2 = 1/26

Recap Chain Rule in probability

For events $A_1, \ldots, A_n$ whose intersection has not probability zero, the chain rule states

$$\begin{aligned}
\mathbb{P}\left(A_1 \cap A_2 \cap \ldots \cap A_n\right) &= \mathbb{P}\left(A_n \mid A_1 \cap \ldots \cap A_{n-1}\right) \mathbb{P}\left(A_1 \cap \ldots \cap A_{n-1}\right) \\
&= \mathbb{P}\left(A_n \mid A_1 \cap \ldots \cap A_{n-1}\right) \mathbb{P}\left(A_{n-1} \mid A_1 \cap \ldots \cap A_{n-2}\right) \mathbb{P}\left(A_1 \cap \ldots \cap A_{n-2}\right) \\
&= \mathbb{P}\left(A_n \mid A_1 \cap \ldots \cap A_{n-1}\right) \mathbb{P}\left(A_{n-1} \mid A_1 \cap \ldots \cap A_{n-2}\right) \cdot \ldots \cdot \mathbb{P}(A_3 \mid A_1 \cap A_2) \mathbb{P}(A_2 \mid A_1) \mathbb{P}(A_1) \\
&= \mathbb{P}(A_1) \mathbb{P}(A_2 \mid A_1) \mathbb{P}(A_3 \mid A_1 \cap A_2) \cdot \ldots \cdot \mathbb{P}(A_n \mid A_1 \cap \cdots \cap A_{n-1}) \\
&= \prod_{k=1}^{n} \mathbb{P}(A_k \mid A_1 \cap \cdots \cap A_{k-1}) \\
&= \prod_{k=1}^{n} \mathbb{P}\left(A_k \,\middle|\, \bigcap_{j=1}^{k-1} A_j\right).
\end{aligned}$$

For $n = 4$, i.e. four events, the chain rule reads

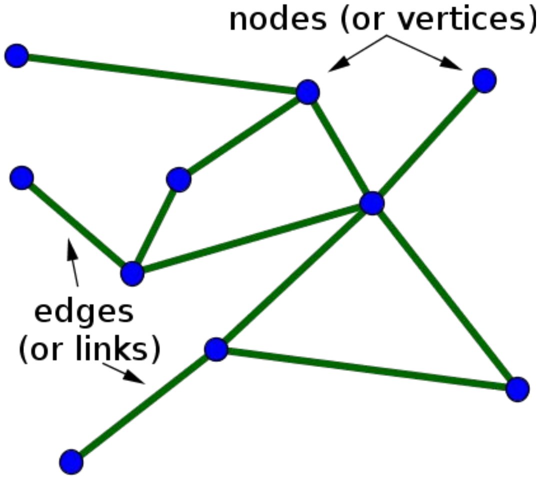
$$\begin{aligned}
\mathbb{P}(A_1 \cap A_2 \cap A_3 \cap A_4) &= \mathbb{P}(A_4 \mid A_3 \cap A_2 \cap A_1) \mathbb{P}(A_3 \cap A_2 \cap A_1) \\
&= \mathbb{P}(A_4 \mid A_3 \cap A_2 \cap A_1) \mathbb{P}(A_3 \mid A_2 \cap A_1) \mathbb{P}(A_2 \cap A_1) \\
&= \mathbb{P}(A_4 \mid A_3 \cap A_2 \cap A_1) \mathbb{P}(A_3 \mid A_2 \cap A_1) \mathbb{P}(A_2 \mid A_1) \mathbb{P}(A_1)
\end{aligned}$$

Bayesian Networks: Probabilistic **Graphical** Models

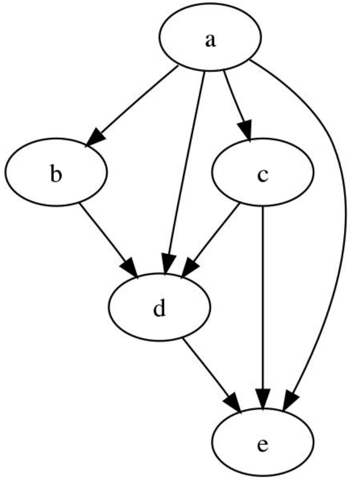
- A ‘graph’ in the context of Networks is a mathematical structure used to model pairwise relations between objects
    - Graphs consist of vertices and edges, where edges (aka links) connect vertices (aka nodes)

- The directed edges in the graph indicates the flow of information and the dependencies between variables

- For a Bayes Network, we use a ‘Directed Acyclic Graph’ to represents the probabilistic dependencies among a set of variables



Bayesian Networks as Directed Acyclic Graphs



- A directed graph with no cycles
- A Directed Acyclic Graph G can be thought of as a compact representation of a joint probability distribution over n variables X_1, X_2, X_3,..., X_n :

$$P(X_1 \cap X_2, \cap X_3 \cap ... \cap X_n) = P(X_1 \mid X_2 \cap X_3 \cap ... \cap X_1)P(X_2 \mid X_3 \cap ... \cap X_n) \cdot ... \cdot P(X_{n-1} \mid X_n)P(x_n)$$

They are a generalization of random processes that depend on each other:

- Example 1: rainy weather pattern: Dark clouds increase the probability of raining later the same day
- Example 2: The probability of detecting a malware is influenced by the values of internal CPU events in a microprocessor

Bayesian Networks as Directed Acyclic Graphs

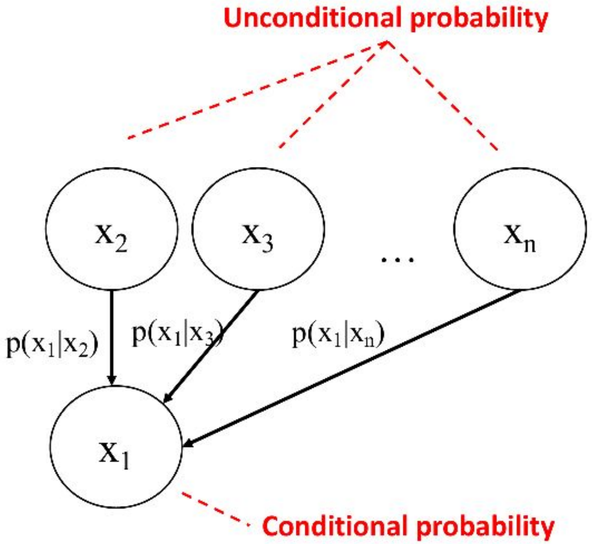
- Vertices : Variables
- Edges: A conditional probability
    - An edge from y to x represents P(x|y)
- For vertex X_1 the conditional probability is:

$$P(X_1 \mid X_2, X_3, \ldots, X_n)$$

- Recall:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | Parents(x_i))$$

- The joint distribution contains the information we need to compute the probability of interest using Bayesian Networks

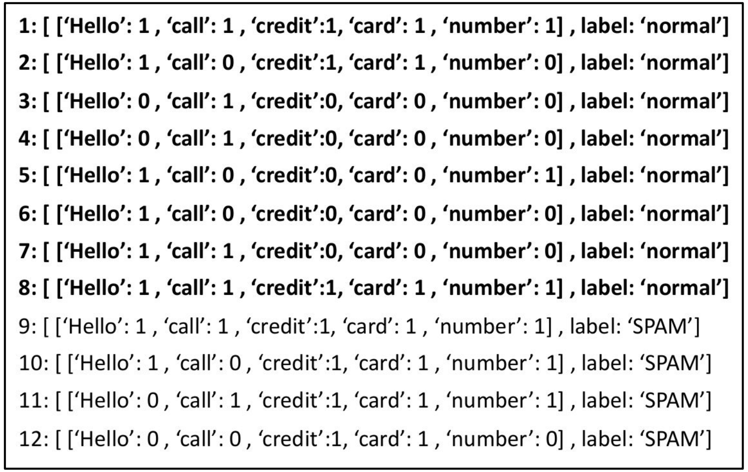


The advantages of Bayesian Networks

- Offers a more nuanced model of a system, particularly important with imperfect data
- Interoperable and visual structure
- Answer probabilistic queries and compute Inference
    - Example: “What is the probability of an email being SPAM if it has the words “provide your credit card information”?

Probabilities: Spam filter example

- Given training data (right), determine the conditional probability of seeing ‘Hello’ in a ‘normal’ message
- The probabilities of discrete individual words (not a continuous property) and can be called ‘Likelihood’



1: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
2: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘normal’]
3: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
4: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
5: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 1] , label: ‘normal’]
6: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
7: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
8: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
9: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
10: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
11: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
12: [[‘Hello’: 0 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘SPAM’]

$$P(\text{Hello} \mid \text{normal}) = \frac{\text{Number of times ‘Hello' is seen in a ‘normal' message}}{\text{Total number of words in the ‘normal' messages}} = \frac{8}{20} = 0.3$$

Spam filter example: ‘normal’

1: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
2: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘normal’]
3: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
4: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
5: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 1] , label: ‘normal’]
6: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
7: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
8: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
9: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
10: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
11: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
12: [[‘Hello’: 0 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘SPAM’]

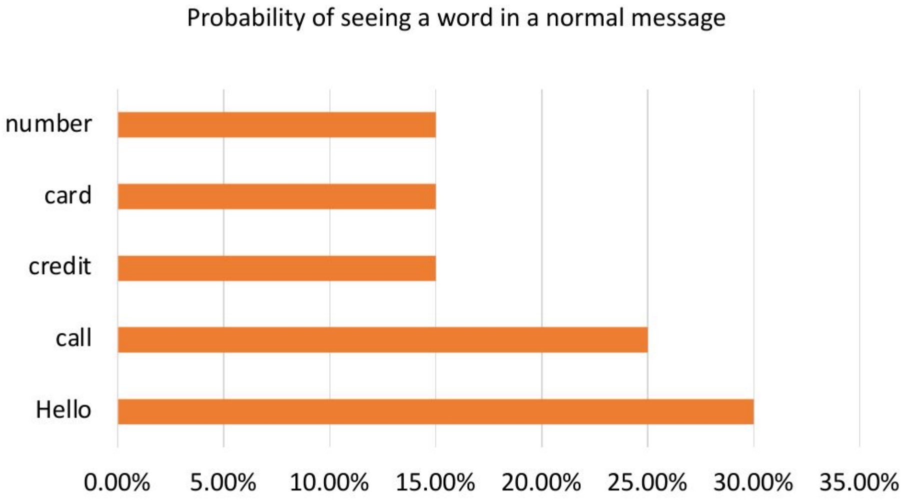
$$P(\text{credit} \mid \text{normal}) = \frac{3}{20} = 0.15$$

$$P(\text{Hello} \mid \text{normal}) = \frac{8}{20} = 0.3$$

$$P(\text{call} \mid \text{normal}) = \frac{5}{20} = 0.25$$

$$P(\text{card} \mid \text{normal}) = \frac{3}{20} = 0.15$$

$$P(\text{number} \mid \text{normal}) = \frac{3}{20} = 0.15$$


Probability of seeing a word in a normal message

Spam filter example: ‘SPAM’

1: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
2: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘normal’]
3: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
4: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
5: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 1] , label: ‘normal’]
6: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
7: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
8: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
9: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
10: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
11: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
12: [[‘Hello’: 0 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘SPAM’]

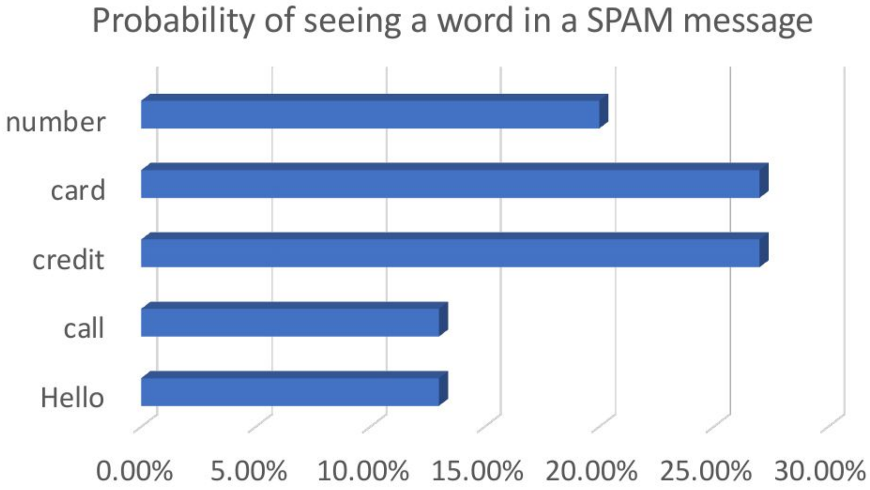
$$P(\text{credit} \mid \text{SPAM}) = \frac{4}{15} = 0.15$$

$$P(\text{Hello} \mid \text{SPAM}) = \frac{2}{15} = 0.3$$

$$P(\text{call} \mid \text{SPAM}) = \frac{2}{15} = 0.25$$

$$P(\text{card} \mid \text{SPAM}) = \frac{4}{15} = 0.15$$

$$P(\text{number} \mid \text{SPAM}) = \frac{3}{15} = 0.15$$



Spam filter example: Score proportional to probability

- Prior probability P(normal) is the initial guess about the probability that any message is ‘normal’:

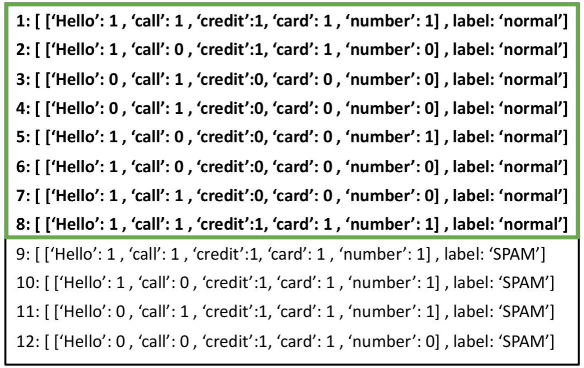
$$P(\text{normal}) = \frac{8}{12} = 0.67$$

- The probability score of a message that contains ‘credit’ and ‘card’ being normal:

$$P(\text{normal}) \times P(\text{credit} \mid \text{normal}) \times P(\text{card} \mid \text{normal}) = 0.67 \times 0.15 \times 0.15 = 0.015$$

- The score is proportional to the probability that a message is normal given that it has the words ‘credit’ and ‘card’ in it:

$$0.015 \propto P(\text{normal} \mid \text{credit},\ \text{card})$$



1: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
2: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘normal’]
3: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
4: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
5: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 1] , label: ‘normal’]
6: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
7: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
8: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
9: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
10: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
11: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
12: [[‘Hello’: 0 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘SPAM’]

Spam filter example: Score proportional to probability

- Prior probability P(SPAM) is the initial guess about the probability that any message is ‘SPAM’:

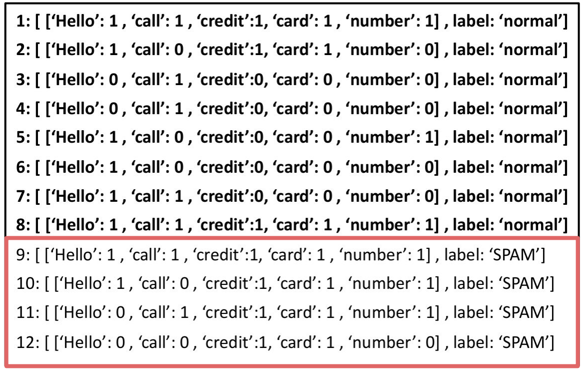
$$P(\text{SPAM}) = \frac{4}{12} = 0.33$$

- The probability score of a message that contains ‘credit’ and ‘card’ being SPAM:

$$P(\text{SPAM}) \times P(\text{credit} \mid \text{SPAM}) \times P(\text{card} \mid \text{SPAM}) = 0.33 \times 0.27 \times 0.27 = 0.024$$

- The score is proportional to the probability that a message is SMAP given that it has the words ‘credit’ and ‘card’ in it:

$$0.024 \propto P(\text{SPAM} \mid \text{credit},\ \text{card})$$



1: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
2: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘normal’]
3: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
4: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
5: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 1] , label: ‘normal’]
6: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
7: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:0, ‘card’: 0 , ‘number’: 0] , label: ‘normal’]
8: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘normal’]
9: [[‘Hello’: 1 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
10: [[‘Hello’: 1 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
11: [[‘Hello’: 0 , ‘call’: 1 , ‘credit’:1, ‘card’: 1 , ‘number’: 1] , label: ‘SPAM’]
12: [[‘Hello’: 0 , ‘call’: 0 , ‘credit’:1, ‘card’: 1 , ‘number’: 0] , label: ‘SPAM’]

Spam filter example

P(credit | normal) = $\frac{4}{20} = 0.2$

P(Hello | normal) = $\frac{4}{20} = 0.2$

P(call | normal) = $\frac{5}{20} = 0.25$

P(card | normal) = $\frac{4}{20} = 0.2$

P(number | normal) = $\frac{3}{20} = 0.15$

P(credit | SPAM) = $\frac{4}{15} = 0.27$

P(Hello | SPAM) = $\frac{2}{15} = 0.13$

P(call | SPAM) = $\frac{2}{15} = 0.13$

P(card | SPAM) = $\frac{4}{15} = 0.27$

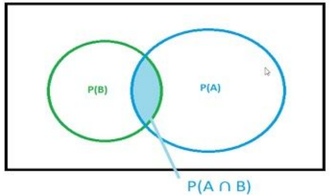
P(number | SPAM) = $\frac{3}{15} = 0.2$

$$0.015 \propto P(normal|credit\ card)$$

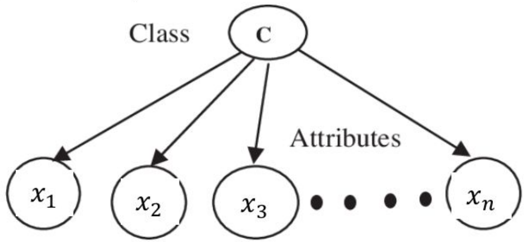
$$\boxed{0.024 \propto P(SPAM|credit\ card)}$$

Bayes vs Naïve Bayes network

- Naive Bayes is a simplified form of a Bayesian network
    - A single root node represents the class label C
    - Feature nodes are directly connected with directe edges from the class to the feature nodes
- Key simplification:
    - All features are conditionally independent given th class label:
        - An observation is irrelevant or redundant when evaluating the hypothesis (no overlap)
    - Greatly reduces the complexity of the model
    - Allows for efficient computation of probabilities

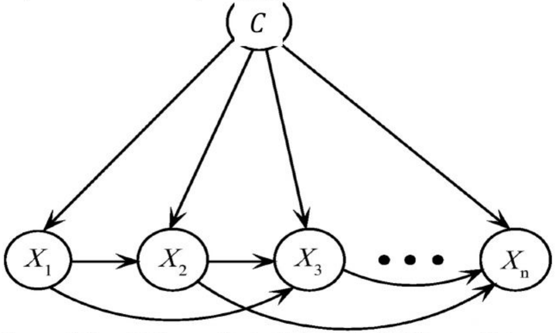


Naïve Bayes



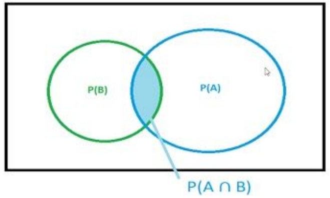
$P(C \mid X_1,\dots, X_n) \propto P(X_1 \mid C) * P(X_2 \mid C)*\dots * P(X_n \mid C)* P(C)$

Bayesian Network



$P(C \mid X_1,\dots, X_n) = P(X_1,\dots, X_n \mid C) * P(C) / P(X_1,\dots, X_n)$

Bayes vs Naïve Bayes network



- We have three binary features X_1, X_2, X_3, and a binary class label C
- Bayesian Network *without* conditional independence between the feature:
    - Estimate 8 possible combinations of feature values for each class
    - How many possible combinations in total for both class labels?

$$P(C \mid X_1, X_2, X_3) = \frac{P(X_1, X_2, X_3 \mid C)P(C)}{P(X_1, X_2, X_3)}$$

- Using the Naïve Bayes approach *with* conditional independence, we can reduce the computational complexity by relaxing the dependencies
    - How many estimations?

$$P(C \mid X_1, X_2, X_3) \propto P(X_1 \mid C)P(X_2 \mid C)P(X_3 \mid C)P(C)$$

Example: Building a classifier using a Discriminant Function

- One way to build a classifier is to calculate all posterior probabilities for the data points, given a certain class, and assign it to the class with the highest probability
  - Problem: multiplying small probabilities can lead to loss of precision as they can become extremely small

$$G(x) = log\frac{P(x|c1).P(c1)}{P(x|c2).p(c2)} \geq 0 \quad \rightarrow then\ assign\ to\ c1$$

$$i.e., P(x|c1).P(c1) > P(x|c2).P(c2)$$

- Simpler way: Discriminant function (X: attribute and C_1, C_2: class labels)
  - Simpler as it doesn’t need the calculation of evidence
  - Less subject to underflow issues
  - However, the complexity of computation increases with multiple attributes

Likelihood Term

$$G(X) = log\frac{P(x_1, \ldots, x_n|c1).P(c1)}{P(x_1, \ldots, x_n|c2).p(c2)} \geq 0$$

$$\rightarrow then\ assign\ to\ c1$$

Motivation for Naïve Bayes network

- The likelihood term in Bayes Theorem accounts for the probability of samples represented by features , given a certain class
- With several features and the dependencies between the variables, the computational cost will be high.
- A lot of features means we have to calculate the joint probability of all the features even in discriminant function.
- So, what is the solution?
    - Naïve Bayes Classifier

Naïve Bayes Classifier

- Assumption: unlike Bayes Theorem, the assumption is that the input features are independent variables (remove the dependency)
- With the above assumption we have for the likelihood term:

$$P(x_1, \dots, x_n | c1) = P(x_1 | c1) \dots P(x_n | c1) = \prod_{i=1}^{n} P(x_i | c1)$$

- Now, the discriminant function is:

$$G(x_1, \dots, x_n) = log \frac{\prod_{i=1}^{n} P(x_i | c1)\ .\ P(c1)}{\prod_{i=1}^{n} P(x_i | c2)\ .p(c2)} \geq 0 \ \rightarrow then\ assign\ to\ c1$$

Naïve Bayes Classifier: Flu test example

- We have a dataset of patients with attributes:

  Feature: Test: {positive, Negative}
  Class label: Flu: {True, False}

- Determine if the flu test generates accurate results for diagnosing flu
- Problem: calculate the probability of a patient having flu given a positive test using Bayes theorem:

$$P(Flu = T|Test = Positive) = \frac{P(Test=Positive|Flu=T) * P(Flu=T)}{P(Test=Positive)} = ?$$

Naïve Bayes Classifier: Flu test example

Test (feature)	Has flu (c1)	Healthy (c2)
positive	0.85	0.05
Negative	0.15	0.9998

Assume:

Likelihood term: P(Test=positive|Flu=T) = 0.85

Prior: P(Flu=T) = 0.0002

Using joint probability distribution formula, the Evidence term is:

P(Test=positive) = P(Test=positive|Flu=T) * P(Flu=T) + P(Test=positive|Flu=F) * P(Flu=F)

= 0.85*0.0002 + 0.05*0.9998 = 0.00017+ 0.14997 = 0.05

P(Flu=F) = 1-P(Flu=T) = 1- 0.0002 = 0.9998

P(Test=positive|Flu=F) = 0.05

$$P(Flu = T|Test = Positive) = \frac{P(Test=Positive|Flu=T)*P(Flu=T)}{P(Test=Positive)}$$

$$= \frac{0.85*0.0002}{0.05} = 0.0034 \sim 0.34\%$$

Conclusion: Very bad flu test

Naïve Bayes Classifier: Flu test example

Test (feature)	Has flu (c1)	Healthy (c2)
positive	0.85	0.05
Negative	0.15	0.9998

- Now build a classifier using the Discriminant function G(X) as: $\boldsymbol{P(Flu = T|Test = Positive)}$
- We can write G(X):

$$G(X) = log \frac{P(Test = positive|has\ flu).P(has\ flu)}{P(Test = positive|healthy).p(healthy)} = log \frac{0.85 * 0.0002}{0.05*0.9998} = -2.46$$

- If G(x) < 0 the patient with a *positive* test is less likely to have flu. Classified as healthy.
- Similarly:

$$\boldsymbol{P(Flu = T|Test = Negative)} =$$

$$G(X) = log \frac{P(Test = negative|has\ flu).P(has\ flu)}{P(Test = negative|healthy).p(healthy)} = log \frac{0.15 * 0.0002}{0.95*0.9998} = -4.5$$

- If G(x) < 0 the patient with a *negative* test is less likely to have flu. Classified as healthy.
- This test is not very accurate

Naïve Bayes Classifier: Adding more features

- In general, Discriminant function for multiple features:

$$G(X) = log\frac{P(x_1, \dots, x_n|c1).P(c1)}{P(x_1, \dots, x_n|c2).p(c2)} \geq 0 \quad \rightarrow then\ assign\ to\ c1$$

- However, with Naïve Bayes and assuming variable independence:

$$G(x_1, \dots, x_n) = log\frac{\prod_{i=1}^{n} P(x_i|c1)\ .P(c1)}{\prod_{i=1}^{n} P(x_i|c2)\ .p(c2)} \geq 0 \quad \rightarrow then\ assign\ to\ c1$$

- In the example we have accordingly:

$$G(x_1, x_2) = log\frac{P(x_1|c1).P(x_2|c1).P(c1)}{P(x_1|c2).P(x_2|c2).p(c2)}$$