Naive Bayes

Supervised Naive Bayes Prediction

Marcus Birkenkrahe

March 9, 2023

Naive Bayes

```
- Lecture notes in Markdown file (4_naive_bayes.md)
- Source: Lantz (2019), chapter 4, pp. 89-123
- DataCamp assignment: "Supervised learning with R", ch. 2
```

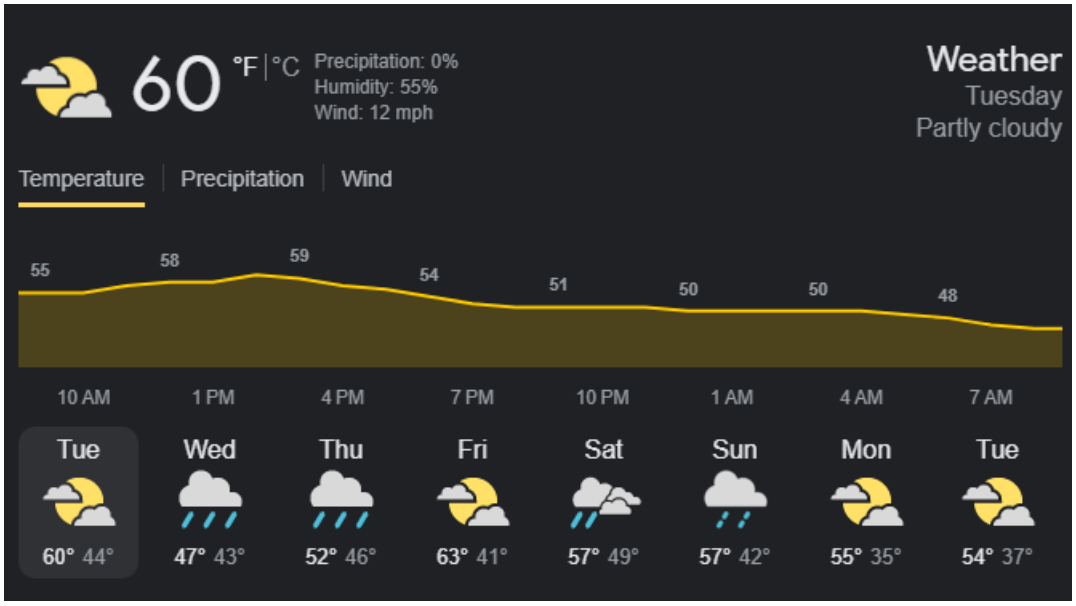
What you will learn

- Classification using Naive Bayes

- Bayes' theorem and naive assumptions
- Text classification use case
- R packages for text mining & visualization
- Application: SMS junk message filter

A little bit of math in here, but nothing more than basic arithmetic. The main complexity is to understand the relationship between features, class, and probability measures, which are our "similarity" measure for this type of classifier.

Probabilistic methods




```
- Probabilistic methods describe uncertainty
- They use data on past events to extrapolate future events
- Such predictions are subject to many assumptions
```

- The chance of rain for example describes the proportion of prior days with similar atmospheric conditions in which in rained.

- A 70 percent chance of rain implies that in 7 out of 10 past cases with similar conditions, it rained somewhere in the area.
- The Naive Bayes algorithm uses probabilities in a similar way to a weather forecast.

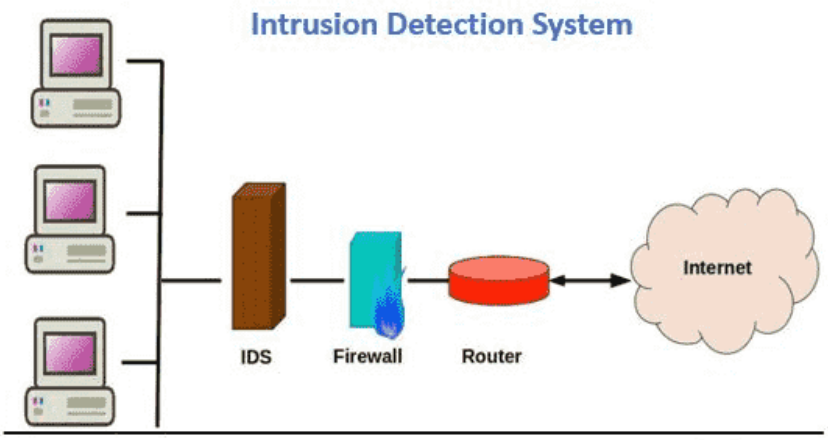
Probability

```
- A probability P is a number between 0 and 1 (0% to 100%)
- P captures the chance that an event will occur based on evidence
- P = 0 indicates that the event will definitely not occur
- P = 1 indicates that the event will occur with absolute certainty
```

Bayesian methods

```
- Training data are used to calculate outcome probability
- Evidence is provided by labeled feature values
- Classifier uses calculated probabilities to estimate class
```

Applications



- Text classification, e.g. spam filter
- Anomaly detection in computer networks
- Diagnosing medical conditions

- Best for problems where information from numerous attributes should be considered simultaneously to estimate overall probability of an outcome.
- E.g. spam filter: various words found in an example/message instance
- Unlike other ML methods, Bayesian methods use all available evidence to make predictions.
- Even if a large number of features have minor effects, their combined impact in a Bayesian model could have a major impact.

Basic idea

Event	Trial
Heads result	Coin flip
Rainy weather	A single day
Message is spam	Incoming email message
Candidate becomes president	Presidential election
Win the lottery	Lottery ticket

The estimated likelihood of an **event** or potential outcome is based on the evidence from multiple **trials** or opportunities for the event to occur.

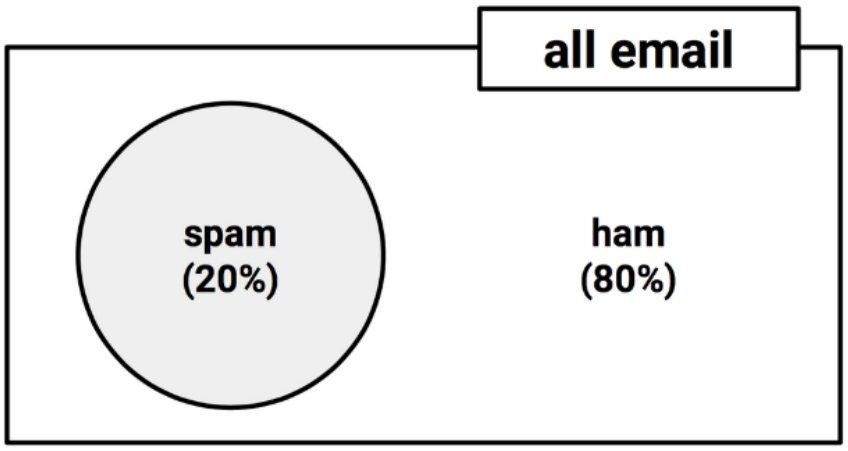
The more trials the better for the accuracy of the estimate - by way of the **law of large numbers**: if you repeat an experiment independently a large number of times and average the result, your result is close to the expected value (the arithmetic mean):

- Large number of coin flips - P(head) = P(tail) = 50%

- Large number of observed days - weather averages
- Large number of email messages - certain spam prediction
- Large number of elections - certain presidential prediction
- Large number of lottery tickets - certain win

But: real events are never mathematically independent.

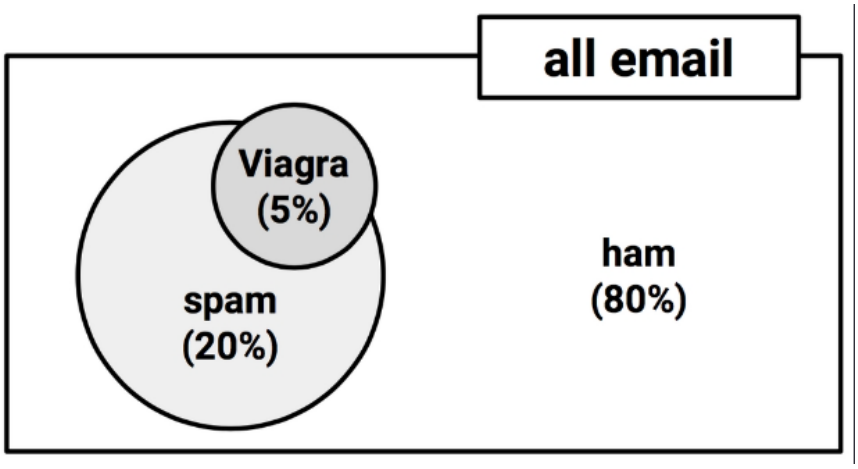
Spam vs. Ham



In email trials, spam and ham are mutually exclusive and exhaustive events.

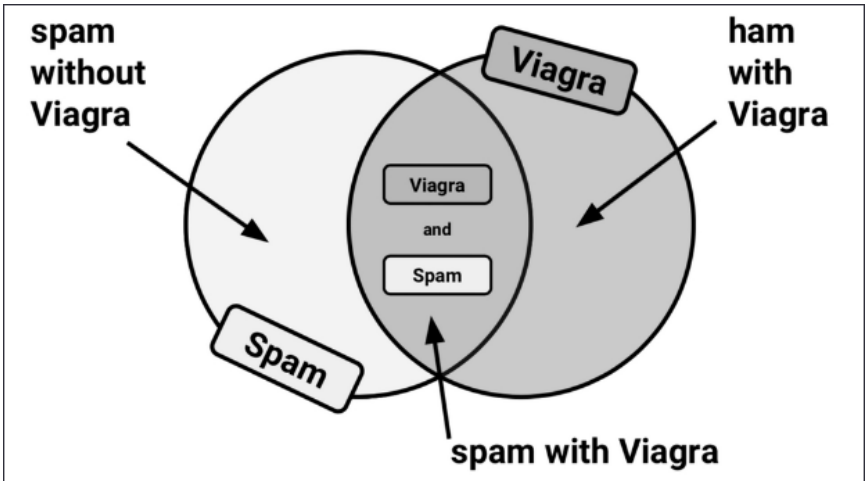
- P(event) = no. of occurrences / no. of trials
- Rain on 3/10 days w/similar conditions: 30% prob today
- Adding all P => 100% of the data or $\sum$ P(event)=1 because a trial always results in an outcome.
- P(spam) + P(ham) = 1 implies that spam/ham **mutually exclusive and exhaustive**.
- An alternative way of saying this uses a table of records: if you record many, many instances, say 1000, you have 200 lines marked as 'spam' and 800 lines marked as 'ham'.

Joint probability



'Viagra' is a non-mutually exclusive event. Its overlap with 'spam' is large than its overlap with 'ham'.

Venn diagrams



Calculating P(spam ∩ Viagra) depends on the joint probabilities of the two events, on their **dependency**.

- The Venn diagram illustrates instances that are only spam, only Viagra but not spam and spam with Viagra messages.

- Named after 19th century mathematician John Venn
- If the circles aren't touching, the joint prob is 0 and the events are said to be **independent**. They can still occur simultaneously.
- $A \cap B = 0$: Knowing something about the outcome of A reveals nothing about the outcome of B. Hard to illustrate in the real world, but:
- The outcome of a coin flip is unlikely to depend on the weather being sunny or rainy on any given day.
- **Dependent events are the basis of predictive modeling.**
- The appearance of clouds is predictive of rain, the appearance of the word 'Viagra' is predictive of spam.

Bayes' theorem

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

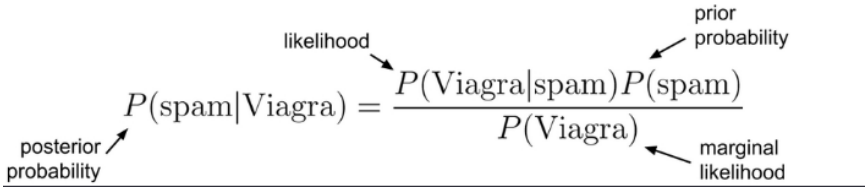
- For independent events, $P(A \cap B) = P(A) * P(B)$
- P(Viagra AND spam) = (5/100) * (20/100) = 0.01
- P(A|B) is the probability of A given B occurred
- P(A|B) is the probability of A conditional on B

$$\boxed{P(A \mid B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B \mid A)P(A)}{P(B)}}$$

- Recall: we're trying to predict the chance that a message that contains the word 'Viagra' (B) is spam (A).

- The formula states that the best estimate of P(A|B) is the proportion of trials in which A occurred with B, P(A ∩ B), out of all trials in which B occurred (all 'Viagra' messages).
- Extreme cases: if B is very rare, P(B) is small and the correction to P(A) is negligible (independence)
- If A and B occur together very often, P(A|B) will be high regardless of P(B).
- If Viagra and spam were independent, P(A ∩ B) = 0.05 * 0.20 = 0.01

Bayesian spam filter



To calculate the components, construct a frequency table that records how often 'Viagra' appeared in 'spam' and 'ham' messages.

	Viagra		
Frequency	**Yes**	**No**	**Total**
spam	4	16	20
ham	1	79	80
Total	5	95	100

- Without knowing anything about an incoming messages, our best estimate would be P(spam) - the **prior probability** (20%)
- The chance of having any 'Viagra' in a spam message is the **marginal likelihood** - having any 'Viagra' at all is the **marginal likelihood**
- What we're after is a computation of the **posterior probability** (i.e. **after** applying the condition 'Viagra').

Likelihood table

	Viagra		
Likelihood	**Yes**	**No**	**Total**
spam	4 / 20	16 / 20	20
ham	1 / 80	79 / 80	80
Total	5 / 100	95 / 100	100

The rows of the likelihood table contain the conditional probabilities for "Viagra" (yes/no) given that an email was spam or ham:

```
P(Viagra = Yes | spam) = 4/20 = 0.20
P(spam & Viagra) = P(Viagra|spam) * P(spam) = (4/20)*(20/100) = 0.04
P(spam|Viagra) = (4/20) * (20/100) / (5/100) = 0.80
```

- The computed chance of getting spam AND Viagra is FOUR times as large as the chance when independence was assumed (P(Viagra)*P(spam)=0.01)
- The posterior probability that a message containing Viagra is spam is 80% - any message containing this term should be filtered.
- This is how commercial spam filters work: they consider a much larger number of words simultaneously when computing frequency and likelihood tables.
- The Naive Bayes algorithm accounts for these additional difficulties. It also relies on careful text pre-processing of the message data.

References

- Lantz (2019). Machine Learning with R (3e). Packt.
- Majka M (2019). naivebayes: High Performance Implementation of the Naive Bayes Algorithm in R. R package version 0.9.7, `https://CRAN.R-project.org/package=naivebayes`.