Naïve Bayes Classifier

Inferring Parameters of the model

- We have data X and we assume it comes from some distribution
- How do we figure out the parameters that 'best' fit that distribution?

- **Maximum Likelihood Estimate (MLE)**: It produces the choice most likely to have generated the observed data [Frequentist Approach]
- **Maximum a posteriori (MAP)**: A MAP estimate is the choice that is most likely given the observed data [Bayesian Approach]

Joint Probability

- Joint Probability is the likelihood of more than one event occurring at the same time

Conditions:

- One is that events X and Y must happen at the same time. **Example:** *Throwing two dice simultaneously.*

- The other is that events X and Y must be independent of each other. That means the outcome of event X does not influence the outcome of event Y.
  Example: *Rolling two Dice.*

- If the above conditions meet, then **$P(A \cap B) = P(A) * P(B)$.**

What will happen if we find the joint probability of two dependent events?

Conditional Probability

- The conditional probability of an **event *B*** is the probability that the event will occur given the knowledge that an **event *A* has already occurred**. It is denoted by **P(B|A).**

- The joint probability of two dependent events then becomes **P(A and B) = P(A)P(B|A)**

Note:

✓ Two events A and B are independent if **P(AB) = P(A) P(B)**

✓ Two events A and B are conditionally independent given C if they are independent after conditioning on C

$$P(AB|C) = P(B|AC)P(A|C) = P(B|C)P(A|C)$$

Conditional Probability

- Joint Probability of two dependent events:

  P(A and B) = P(A)P(B|A) and **P(B and A) = P(B)P(A|B)**

- When we equate this, we will get, **P(A)P(B|A) = P(B)P(A|B),** then

$$\mathbf{P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}}$$

This is the Bayes theorem

- **It tells us:** how often A happens given that B happens, written **P(A|B)**,
- **When we know:** how often B happens given that A happens, written **P(B|A)**

  and how likely A is on its own, written **P(A)**

  and how likely B is on its own, written **P(B)**

Hypothesis provided Evidence

$$\mathbf{P(A|B)} = \frac{\mathbf{P(A)P(B|A)}}{\mathbf{P(B)}}$$

- Change **A** to **Hypothesis** & **B** to **Evidence**

$$\Longrightarrow \mathbf{P(H|E)} = \frac{\mathbf{P(H)P(E|H)}}{\mathbf{P(E)}} = \frac{\mathbf{P(H)P(E|H)}}{\mathbf{P(E|H) \times P(H) + P(E|\neg H) \times P(\neg H)}}$$

- This relates the probability of the hypothesis before getting the evidence **P(H) — prior probability,** to the probability of the hypothesis after getting the evidence **P(H|E) — posterior probability.**

- The factor that relates the two, **P(E|H) / P(E)**, is called the **likelihood ratio**.

- **Bayes Theorem** states that **"*The posterior probability equals the prior probability times the likelihood ratio*".**

Prior & Posterior Probability

- **Prior probability** is the probability an event will happen ***before*** you **take** any **new evidence** into account.

- **Posterior probability** is the probability an event will happen **after all evidence** has been **taken** into account.

- You can think of **posterior probability** as an **adjustment** on the **prior probability.**

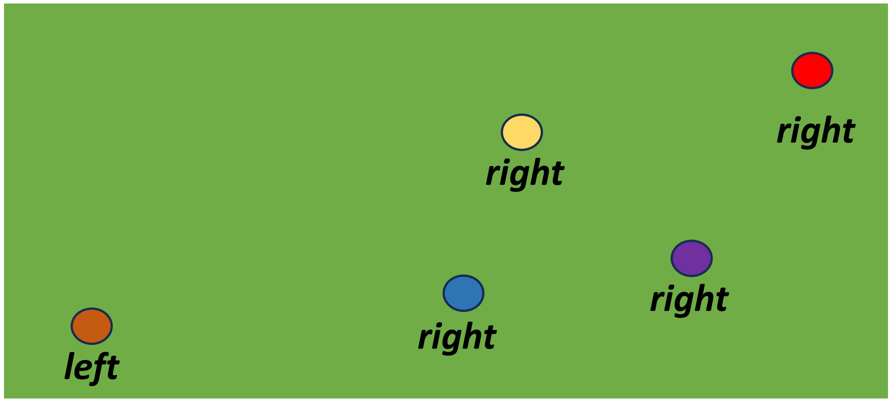
Bayesian Approach

- I have placed a white ball on the table. Find the location of the white ball
- Prior: Take a random guess

Step 1: I start placing new colour balls on the table and tell you they are to the right or left of the white ball

Step 2: Update your belief about the white ball location based on data

- Current: The ball is more likely to be on the left side than on the right



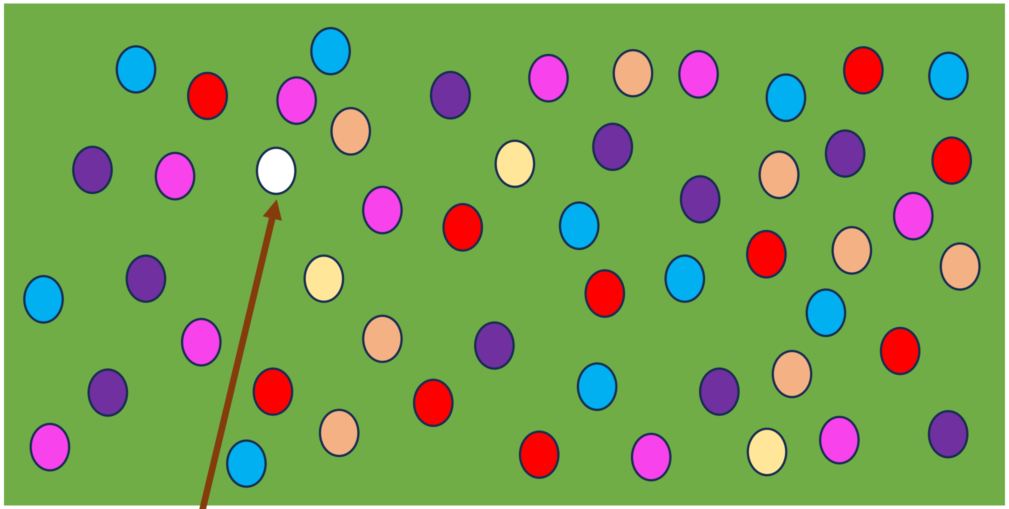
The Bayesian approach is all about updating one's beliefs based on data.

Bayesian Approach

Step 3: After randomly placing 50 colour balls, you find:

- 9 out of 50 colour balls, or 18% are to the left of the white ball
- 42 of 50, or 82% are to the right

- The more data you will have, the more precise will be your estimation.



This is the posterior belief, which is the prior belief after it has evolved based on data

Let us understand using an example!

- **One fine day Allen felt sick. He went to a doctor.**
- **Doctor suggested a test.**
- **Test result came positive.**

- **Doctor: "Allen, you have a very rare disease which affects 0.1% of the population in the world."**
- **Allen: "How certain is that I have the disease."**
- **Doctor: "The test accurately finds 99% of the people who have the disease. False alarm rate of the test is 1%."**

Has Allen got disease?

- **What is the chance that Allen actually has the disease?**
- **Is it 99%? (Because that's the accuracy of the test.)**

- **Good news for Allen: that's not actually correct!**
- **Thanks to Bayes Theorem !!!**

Bayes - Theorem

$$\mathbf{P(H|E)} = \frac{\mathbf{P(E|H) * P(H)}}{\mathbf{P(E|H) * P(H) + P(E|\neg H) * P(\neg H)}}$$

- H: Actually have the disease
- E: Test Result is positive

- **Doctor:** **"Allen, you have a very rare disease which affects 0.1% of the population in the world."** $\mathrm{P(H)} = 0.001 \Rightarrow \mathrm{P(\neg H)} = 0.999$

- **Allen:** **"How certain is that I have the disease."** $\mathrm{P(H|E)}$

- **Doctor:** **"The test accurately finds 99% of the people who have the disease. False alarm rate of the test is 1%."**

$$\mathrm{P(E|H)} = 0.99 \quad \Rightarrow \mathrm{P(E|\neg H)} = 0.01 \text{ (False Alarm Rate)}$$

Chance of Allen having disease

- Plugging all the numbers, we get

$$\mathbf{P(H|E)} = \frac{\mathbf{0.99 \times 0.001}}{\mathbf{0.99 \times 0.001 + 0.01 \times 0.999}} = \mathbf{0.09}$$

- Only 9% chance that Allen is having the disease!

Generalized Bayes Theorem

- Let $E_1, E_2, \ldots\ldots, En$ be (pairwise) mutually exclusive events
- We have $E_1 \cup E_2 \cup \ldots \cup En = S$, where S denotes the sample space.
- Let F be an events such that P(F) ≠ 0, Then

$$P(E_1|F) = \frac{P(F|E_1)P(E_1)}{P(F|E_1)P(E_1) + P(F|E_2)P(E_2) + \cdots + P(F|E_n)P(E_n)}$$

Naïve Bayes Classifier

Predicting whether to play given the weather conditions

Naïve Bayes Classifier

- The Naïve Bayes Classifier technique is based on the Bayesian theorem and is mainly suited when the dimensionality of the inputs is high.

- Naïve Bayes classifier can handle an arbitrary number of independent features, whether continuous or categorical.

- Given a set of features, $X = \{x_1, x_2, \ldots, x_N\}$, we want to construct the posterior probability for the event/class C_j among the set of possible outcomes, $C = \{C_1, C_2, \ldots, C_d\}$.

- Using Bayes Rule: $p(C_j | x_1, x_2, \ldots, x_N) = \frac{p(x_1, x_2, \ldots, x_N | C_j) p(C_j)}{p(x_1, x_2, \ldots, x_N)}$, where

  $p(C_j | x_1, x_2, \ldots, x_N)$ is the posterior probability of the class membership, i.e., the probability that X belong to C_j.

Naïve Bayes Classifier

$$p\left(C_j \middle| x_1, x_2, \ldots, x_N\right) = \frac{p\left(x_1, x_2, \ldots, x_N \middle| C_j\right) p(C_j)}{p(x_1, x_2, \ldots, x_N)},$$

- Since Naïve Bayes assumes that the conditional probabilities of the independent variables are statistically independent, we can decompose the likelihood to a product of terms:

$$p\left(x_1, x_2, \ldots, x_N \middle| C_j\right) \propto \prod_{k=1}^{N} p(x_k | C_j)$$

- The posterior can be rewritten as : $p\left(C_j \middle| x_1, x_2, \ldots, x_N\right) \propto p(C_j) \prod_{k=1}^{N} p(x_k | C_j)$

- Using Bayes' rule above, we label a new case X with a class level C_j that achieves the highest posterior probability.

Weather Data Record of 14 Days

Day	Outlook	Humidity	Wind	Play
1	Sunny	High	Week	No
2	Sunny	Normal	Strong	Yes
3	Overcast	High	Week	Yes
4	Rain	Normal	Strong	No
5	Rain	Normal	Week	No
6	Sunny	High	Strong	Yes
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Week	Yes
9	Overcast	High	Strong	Yes
10	Rain	Normal	Week	No
11	Overcast	High	Strong	No
12	Rain	High	Week	No
13	Sunny	Normal	Week	No
14	Overcast	High	Strong	Yes

Naïve Bayes Classifier

- Play = Yes ⇒ Y
- Play = No ⇒ N
- Outlook ⇒ O
- Humidity ⇒ Y
- Wind ⇒ Y

Step 1: Find the posterior probabilities as follows:

$$P(Y|O,H,W) = \frac{P(Y)P(O,H,W|Y)}{P(O,H,W)} = \frac{P(Y)P(O|Y)P(H|Y)P(W|Y)}{P(O,H,W)}$$

$$P(N|O,H,W) = \frac{P(N)P(O,H,W|N)}{P(O,H,W)} = \frac{P(N)P(O|N)P(H|N)P(W|N)}{P(O,H,W)}$$

- $P(O,H,W) = P(O,H,W|Y) \times P(Y) + P(O,H,W|N) \times P(N)$

Classification Rule:

- **Decide to Play if**

$$P(Y|O,H,W) > P(N|O,H,W)$$

OR

$$P(Y)P(O|Y)P(H|Y)P(W|Y) > P(N)P(O|N)P(H|N)P(W|N)$$

- **Decide Not to Play if**

$$P(N|O,H,W) > P(Y|O,H,W)$$

OR

$$P(N)P(O|N)P(H|N)P(W|N) > P(Y)P(O|Y)P(H|Y)P(W|Y)$$

- Play = Yes ⇒ Y
- Play = No ⇒ N
- Outlook ⇒ O
- Humidity ⇒ Y
- Wind ⇒ Y

Quantities defining the classifier

- $\mathbf{P(Y) = 0.5}$
- $\mathbf{P(N) = 0.5}$

- $\mathbf{P(O|Y) = P(O = Sunny|Y) + P(O = Overcast|Y) + P(O = Rain|Y)}$
- $\mathbf{P(O|N) = P(O = Sunny|N) + P(O = Overcast|N) + P(O = Rain|N)}$

- $\mathbf{P(H|Y) = P(H = High|Y) + P(H = Normal|Y)}$
- $\mathbf{P(H|N) = P(H = High|N) + P(H = Normal|N)}$

- $\mathbf{P(W|Y) = P(W = Weak|N) + P(W = Strong|N)}$
- $\mathbf{P(W|N) = P(W = Weak|N) + P(W = Strong|N)}$

Likelihood of Outlook

$$\mathbf{P(Outlook|Play = Yes)}$$

$$\mathrm{P(Outlook = Sunny|Play = Yes)} = \frac{3}{7}$$

$$\mathrm{P(Outlook = Overcast|Play = Yes)} = \frac{4}{7}$$

$$\mathrm{P(Outlook = Rain|Play = Yes)} = \frac{0}{7} = 0$$

$$\mathbf{P(Outlook|Play = No)}$$

$$\mathrm{P(Outlook = Sunny|Play = No)} = \frac{2}{7}$$

$$\mathrm{P(Outlook = Overcast|Play = No)} = \frac{1}{7}$$

$$\mathrm{P(Outlook = Rain|Play = No)} = \frac{4}{7}$$

Weather	Play Game	
	Yes	No
Sunny	3	2
Overcast	4	1
Rain	0	4
Total	7	7

Likelihood of Humidity

$$\mathbf{P(Humidity|Play = Yes)}$$

$$\mathrm{P(Humidity = High|Play = Yes)} = \frac{5}{7}$$

$$\mathrm{P(Humidity = Normal|Play = Yes)} = \frac{2}{7}$$

$$\mathbf{P(Humidity|Play = No)}$$

$$\mathrm{P(Humidity = High|Play = No)} = \frac{3}{7}$$

$$\mathrm{P(Humidity = Normal|Play = No)} = \frac{4}{7}$$

Humidity	Play Game	
	Yes	No
High	5	3
Normal	2	4
Total	7	7

Likelihood of Wind

$$\mathbf{P(Wind|Play = Yes)}$$

$$P(\text{Wind} = \text{Weak}|\text{Play} = \text{Yes}) = \frac{2}{7}$$

$$P(\text{Wind} = \text{Strong}|\text{Play} = \text{Yes}) = \frac{5}{7}$$

$$\mathbf{P(Wind|Play = No)}$$

$$P(\text{Wind} = \text{Weak}|\text{Play} = \text{No}) = \frac{5}{7}$$

$$P(\text{Wind} = \text{Strong}|\text{Play} = \text{No}) = \frac{2}{7}$$

Wind	Play Game	
	Yes	No
Weak	2	5
Strong	5	2
Total	7	7

Classification for a Test Example

- Let's say on a given day, we observe the following: **Outlook=Sunny, Humidity=High, Wind = Strong.**

- What is the prediction of Naïve Bayes Classifier (Play or Not to Play)?

$$\mathrm{P}(\mathrm{Play} = \mathrm{Yes}|\mathrm{Outlook} = \mathrm{Sunny}, \mathrm{Humidity} = \mathrm{High}, \mathrm{Wind} = \mathrm{Strong})$$
$$\propto \mathrm{P}(\mathrm{Play} = \mathrm{Yes})\mathrm{P}(\mathrm{Outlook} = \mathrm{Sunny}|\mathrm{Play} = \mathrm{Yes})\mathrm{P}(\mathrm{Humidity} = \mathrm{High}|\mathrm{Play} = \mathrm{Yes})\mathrm{P}(\mathrm{Wind} = \mathrm{Strong}|\mathrm{Play} = \mathrm{Yes})$$
$$\propto 0.5 * \frac{3}{7} * \frac{5}{7} * \frac{5}{7}$$
$$\propto 0.11$$

➢ The classifier is predicting Play

$$\mathrm{P}(\mathrm{Play} = \mathrm{No}|\mathrm{Outlook} = \mathrm{Sunny}, \mathrm{Humidity} = \mathrm{High}, \mathrm{Wind} = \mathrm{Strong})$$
$$\propto \mathrm{P}(\mathrm{Play} = \mathrm{No})\mathrm{P}(\mathrm{Outlook} = \mathrm{Sunny}|\mathrm{Play} = \mathrm{No})\mathrm{P}(\mathrm{Humidity} = \mathrm{High}|\mathrm{Play} = \mathrm{No})\mathrm{P}(\mathrm{Wind} = \mathrm{Strong}|\mathrm{Play} = \mathrm{No})$$
$$\propto 0.5 * \frac{2}{7} * \frac{3}{7} * \frac{2}{7}$$
$$\propto 0.02$$

Types of Naïve Bayes Model

- Gaussian: It is used for numerical/continuous features. The Gaussian model assumes that the features follow a normal distribution.
- Multinomial: This classifier is used when the data is multinomial distributed. It is primarily used for document classification problems,
- Bernoulli: It works like the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document.

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.naive_bayes import GaussianNB
>>> X, y = load_iris(return_X_y=True)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,
random_state=0)
>>> gnb = GaussianNB()
>>> y_pred = gnb.fit(X_train, y_train).predict(X_test)
```

The documentation contains lot of useful details and explanations

https://scikit-learn.org/stable/modules/naive_bayes.html