Exploration Sheet: The Mathematics of Changing Your Mind

An Introduction to Bayesian Statistics & Machine Learning

Abhishek Bhattacharjee

Introduction

Traditional (frequentist) probability often deals with fixed, long-run frequencies, like flipping a fair coin infinitely many times. Bayesian statistics, however, is the mathematics of *updating your beliefs in the face of new evidence.* In Machine Learning, this principle is the engine behind how algorithms "learn" from data—starting with a baseline guess and continuously refining it as more information streams in. In this sheet, you will explore how simple probabilistic rules scale up to power modern spam filters and autonomous decision-making.

---

1. The False Positive Paradox (Medical Testing)

Imagine a new, mysterious illness called "Mathitis" affects exactly 1% of the population. A biotech company develops a rapid test for Mathitis.

- **True Positive Rate:** If you have Mathitis, the test will correctly return positive 99% of the time.
- **False Positive Rate:** If you are perfectly healthy, the test will incorrectly return positive 5% of the time.

You take the test, and the result is **positive**.

Explore:

- **The Baseline (Prior):** Before taking the test, what was the probability that you had Mathitis? We call this the *prior probability*, denoted as $P(D)$.
- **Visualizing the Population:** Imagine a representative town of 10,000 people. How many people actually have Mathitis? How many are healthy?
- **Counting the Positives:** Out of the healthy people in your town, how many will get a false positive? Out of the sick people, how many will get a true positive?

- **The Big Reveal (Posterior):** Let $P(D|T)$ be the probability you have the disease given a positive test. Using your population counts, calculate $P(D|T)$. Are you surprised by how low the probability still is?
- **Formulating the Rule:** Generalize your logic. Let $P(T|D)$ be the probability of a positive test given you have the disease, and $P(T|\text{not } D)$ be the probability of a false positive. Derive **Bayes' Theorem**:

$$P(D|T) = \frac{P(T|D)P(D)}{P(T|D)P(D) + P(T|\text{not } D)P(\text{not } D)}$$

---

2. The Spam Filter (Naive Bayes Classification)

Machine Learning algorithms often need to classify data into categories. Let's build a simple AI to classify emails as "Spam" or "Ham" (real emails).

Suppose 20% of all emails you receive are Spam. You notice that the word **"URGENT"** appears in 80% of your Spam emails, but it only appears in 10% of your Ham emails.

Explore:

- **Single Feature Classification:** An email arrives containing the word "URGENT". Let S represent the event the email is Spam, and U represent the presence of the word "URGENT". Use Bayes' Theorem to calculate $P(S|U)$. Should your algorithm flag this email?
- **Adding Complexity:** You notice another word, **"WINNER"**. It appears in 60% of Spam, but only 1% of Ham. If an email contains *both* "URGENT" and "WINNER", how do we classify it?
- **The "Naive" Assumption:** To calculate the probability of seeing both words, Machine Learning models often assume that the occurrence of "URGENT" and "WINNER" are *independent* given the class of the email. Why is this called the "Naive" assumption? (Hint: In real human language, are words truly independent?)
- **The Math of Independence:** Assuming independence, write out the formula for $P(U \text{ and } W|S)$. Calculate the new posterior probability that the email is Spam given it contains both words.

---

3. The Robot Mouse in a Maze (Markov Localization)

A robotic mouse is placed on a 1-dimensional track with 5 distinct squares, numbered 1 through 5. The cheese is definitely located on **Square 4**, but the mouse doesn't know this.

Initially, the mouse has no idea where the cheese is, so it assigns an equal probability to all squares. This is its *prior distribution.*

The mouse has a "cheese sensor" that detects the smell of cheese. However, the sensor is noisy:

- If the mouse is exactly on the cheese square, the sensor rings with a probability of 0.8.
- If the mouse is one square away (Square 3 or 5), the sensor rings with a probability of 0.4.
- If the mouse is two or more squares away, the sensor rings with a probability of 0.1.

Explore:

- **The Uniform Prior:** Write down the initial probability array for the mouse's belief about the cheese's location: $[P_1, P_2, P_3, P_4, P_5]$.
- **The First Measurement:** The mouse starts on Square 3. Its sensor **rings**.
  - Calculate the likelihood of the sensor ringing if the cheese were on Square 1, Square 2, etc.
  - Multiply your prior probabilities by these likelihoods to get an "unnormalized" posterior belief.
- **Normalization:** Your probabilities likely no longer add up to 1. Divide each term by the sum of all terms so that they form a valid probability distribution. Which square does the mouse now *believe* is the most likely location of the cheese?
- **Recursive Updating:** The mouse moves to Square 4. The sensor **rings again**. Use your *new* posterior distribution from the previous step as your *new* prior distribution. Calculate the updated belief. How has the probability mass shifted? This iterative loop is exactly how self-driving cars track their location!