Naïve Bayes Classifier

15 January 2026

Alex Lyman

Steve

An individual has been described by a neighbor as follows, "Steve is very shy and withdrawn, invariably helpful but with little interest in people or in the world of reality. A meek and tidy soul, he has a need for order and structure, and a passion for detail."

Is Steve more likely to be a librarian or a farmer?

Tversky and Kahneman, 1974

Steve

Available data makes Steve sound more likely to be a librarian than a farmer.

- There are about 2,600,000 farmers in the US and about 125,000 librarians.
- This means there are 20 farmers per librarian in the U.S. today.
- What happens if we incorporate that info?

Tversky and Kahneman, 1974

Two Approaches to Statistics

- Frequentist
  - All about probability in the long run
    - ie. Flipping a coin in the long run is 50% heads
  - Assume a null hypothesis
  - Collect data, analyze, and ask "How surprising is my result?"
  - Is it surprising enough to reject the null hypothesis?
  - Confidence intervals, P values
  - ‘Available data suggests Steve is a librarian’
- Bayesian
  - Data is fixed, parameters and hypotheses are probability distributions
  - The probability distribution that summarizes what is known before is the prior distribution or just 'the prior'
  - We start with prior beliefs, and we update our beliefs based on new information
  - Bayesian prior overrides data, guess Steve as farmer.

An Analogy

- I have misplaced my phone somewhere in my home. I can use my Watch to activate the beeping on my phone.
  - Problem: Which area of my home should I search?
- Frequentist Reasoning
  - I can hear the phone beeping. I also have a mental model which helps me identify the area from which the sound is coming. Therefore, upon hearing the beep, I infer the area of my home I must search to locate the phone.
- Bayesian Reasoning
  - I can hear the phone beeping. Apart from a mental model which helps me identify the area from which the sound is coming from, I also know the locations where I have misplaced the phone in the past. So, I combine my inferences using the beeps and my prior information about the locations I have misplaced the phone in the past to identify an area I must search to locate the phone. When I find the phone, I naturally update my mental model of locations that I have misplaced my phone.
    - Location in my home is **actually a probability distribution** (prior)

Some Background Math

- A, C are random variables
  - We will often use C to represent the Class or label
  - A random variable is a mathematical function that assigns a numerical value to each outcome of a random event
- Joint Probability: P(A,C) *can be written: P(A ∩ C)*
- Conditional Probability: P(C|A)
- Relationship between Joint and Conditional
  - $P(A, C) = P(A \mid C) * P(C) = P(C \mid A) * P(A)$
- Assuming A is a set of Independent Attributes
  - $P(A) = P(a_1) * P(a_2) * P(a_3) * \ldots * P(an)$
  - $P(A \mid C) = P(a_1 \mid C) * P(a_2 \mid C) * P(a_3 \mid C) * \cdots * P(a_n \mid C)$

Some Background Math

- A, B are random variables
  - A = Probability it snows
  - B = Probability of a car accident
- Joint Probability: P(A,B)
  - Probability it snows **and** there is a car accident
- Conditional Probability: P(B|A)
  - Probability there is a car accident **given it snows**
- Relationship between Joint and Conditional
  - $P(A, B) = P(A \mid B) * P(B) = P(B \mid A) * P(A)$
  - Probability it snows **and** there is a car accident = the probability there is a car accident **given** it snows **times** the probability it snows.

Some Background Math With Example

- A, B are random variables
  - A = Probability it snows
  - B = Probability of a car accident
- Joint Probability: P(A,B) *can be written: P(A ∩ B)*
  - Probability it snows **and** there is a car accident
- Conditional Probability: P(B|A)
  - Probability there is a car accident **given it snows**
- Relationship between Joint and Conditional
  - $P(A, B) = P(A \mid B) * P(B) = P(B \mid A) * P(A)$
  - Probability it snows **and** there is a car accident = the probability there is a car accident **given** it snows **times** the probability it snows.

Imagine winter lasts **100 Days**.

- **10 Days** are Snowy
- **90 Days** are Clear
- On those **10 Snowy Days**, there are **5 Crashes**
- On the **90 Clear Days**, there are **5 Crashes**.

$$P(\text{Snow, Crash}) = \frac{\text{Days with Snow AND Crash}}{\text{Total Days}} = \frac{5}{100} = 5\%$$

$$P(\text{Crash} \mid \text{Snow}) = \frac{\text{Days with Snow AND Crash}}{\text{Total SNOW Days}} = \frac{5}{10} = 50\%$$

Does this math check out?
Do on Board

Conditional Probability

Probability of *A* and *B*

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

Probability of *A* given *B*

Probability of *B*

Conditional Probability for Two Events

We can express P(B|A):

$P(B|A) = P(A \cap B) / P(A)$

Rearranging:

$P(A \cap B) = P(A|B)\, P(B)$

$P(A \cap B) = P(B|A)\, P(A)$

Setting these equal:

$P(A|B)\, P(B) = P(B|A)\, P(A)$

Rearrange to express P(A|B):

$P(A|B) = P(B|A)\, P(A) / P(B)$

This is **Bayes' Rule**

Bayes Classifier

- A **probabilistic framework** for solving classification problems
- **Calculate the probability** of each class and **then pick the one with highest probability**
- **Bayes Theorem**:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Bayes Classifier - Vocab and Intuition

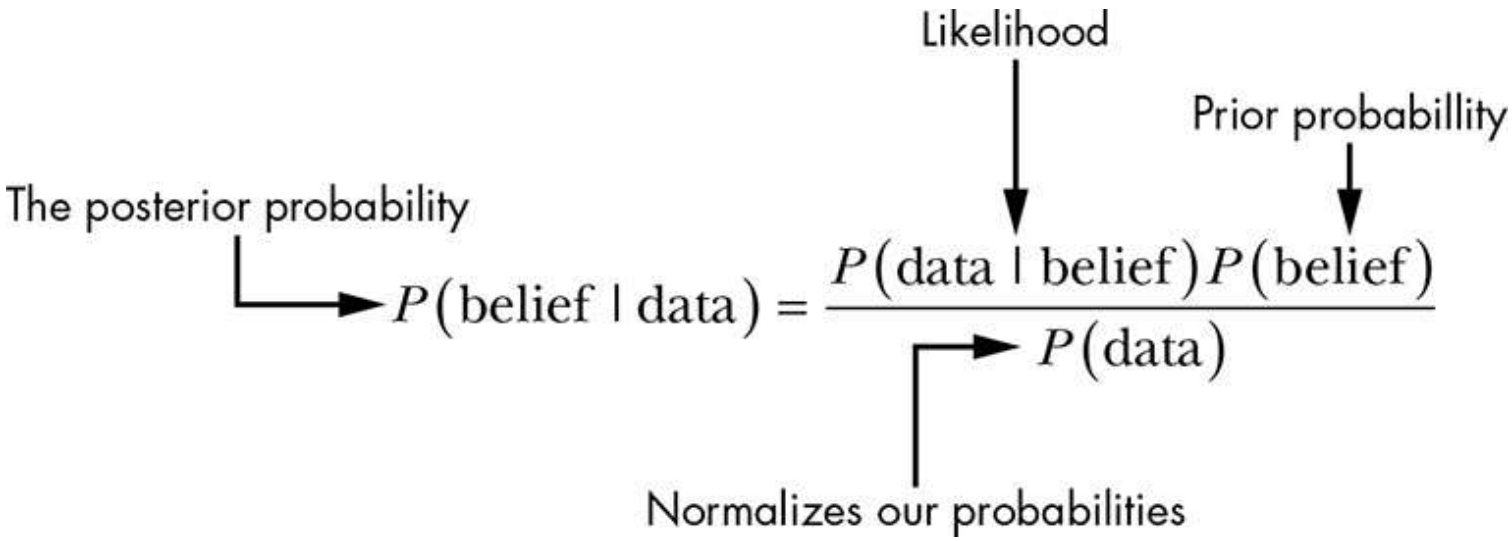


Figure credit: Peter Baumgartner

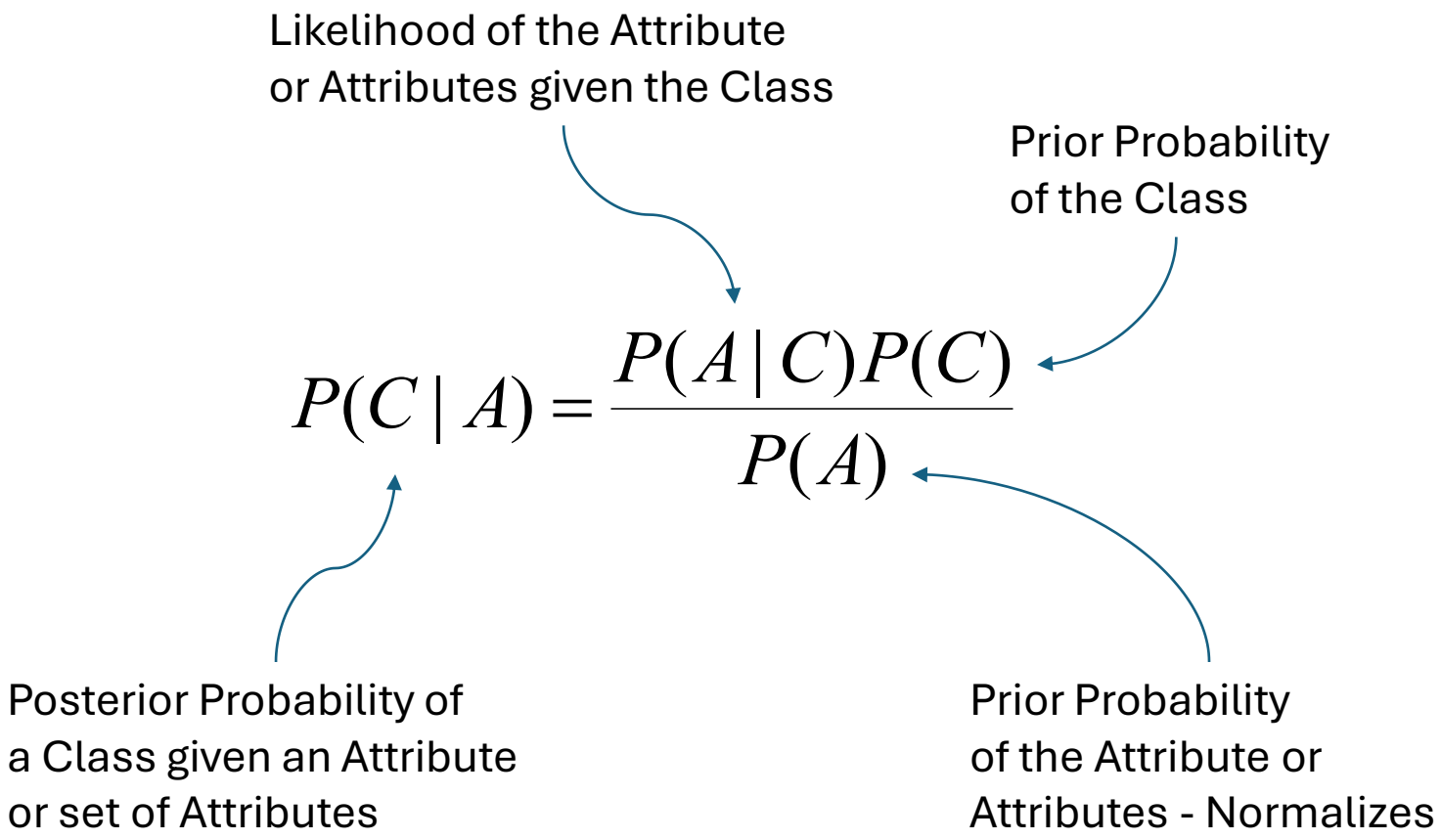
Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayes Theorem to Bayes Classifier



Note: The right side of the equation is based on our current data

Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)

- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?
  - (probability of **Class** given the **Attributes** of the record)

Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of C that maximizes
    $P(C \mid A_1, A_2, \ldots, A_n)$
  - Equivalent to choosing value of C that maximizes
    $P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$
    - We don't actually have divide by $P(A_1, A_2, \ldots A_n)$ if we are just maximizing
    - We are always dividing by the same thing

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

- Assume **independence** among attributes A_i
  - $P(A_1, A_2, \ldots, A_n|C_j) = P(A_1|C_j)\, P(A_2|C_j) \cdots P(A_n|C_j)$
  - We can now estimate $P(A_i|\, C_j)$ for all A_i and C_j.
  - New point X is classified to C_j if

$$P(C_j|X) = P(C_j) \prod_i P(A_i|C_j)$$

is maximal.

Naïve Bayes Classifier

- Involves a learning step in which the various $P(c_j)$ and $P(a_i|c_j)$ terms are estimated
  - (Based on frequencies in the training data)

- Steps:
  - Calculate the probability of all classes c_j
  - Calculate the conditional probability for all the attributes and classes $P(a_i|c_j)$
  - Store the values in a table
  - Use the table to perform calculations of probabilities

Naïve Bayes Classifier Example

- What types of feature are X1 and X2? (discrete or continuous?)
- Estimate the probability of *Plays Fetch = Yes*
- **Given Class = Cat:**
  - Total Cats: 3
  - Cats that Fetch: 1

$$P(\text{Fetch} \mid \text{Cat}) = 1/3 \approx 0.33$$

- **Given Class = Dog:**
  - Total Dogs: 3
  - Dogs that Fetch: 2 (Rows #4, #5)

$$P(\text{Fetch} \mid \text{Dog}) = 2/3 \approx 0.67$$

ID	Species (Y)	Plays Fetch? (X1)	Weight (X2)
1	**Cat**	No	8 lbs
2	**Cat**	Yes (rare!)	9 lbs
3	**Cat**	No	7 lbs
4	**Dog**	Yes	45 lbs
5	**Dog**	Yes	35 lbs
6	**Dog**	No (lazy)	15 lbs

$$P(x \mid C) = \frac{\text{Count of records with Feature } x \text{ and Class } C}{\text{Total Count of Class } C}$$

What about continuous data?

- For **continuous** attributes:
  - Discretize the range into bins
    - one ordinal attribute per bin
    - **violates independence** assumption
  - Two-way split: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - Probability density estimation:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

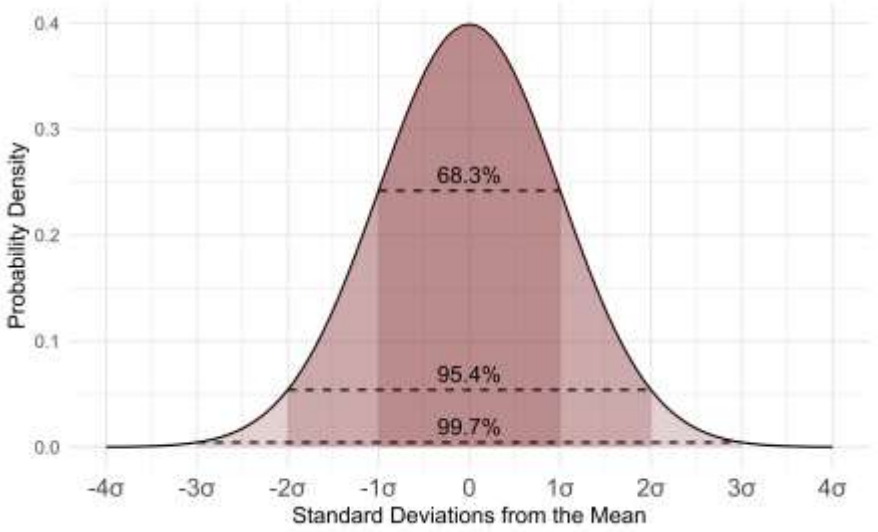
Naïve Bayes Classifier Example

- What do we do with the weight attribute???
- We need to **estimate** the **probability density**.
- To do this, we assume the data follows a **normal** (**Gaussian**) distribution (**Bell Curve**)
- This means we estimate the **Mean (μ)** and **Variance (σ^2)**
- We'll look at that next slide.

ID	Species (Y)	Plays Fetch? (X1)	Weight (X2)
1	**Cat**	No	8 lbs
2	**Cat**	Yes (rare!)	9 lbs
3	**Cat**	No	7 lbs
4	**Dog**	Yes	45 lbs
5	**Dog**	Yes	35 lbs
6	**Dog**	No (lazy)	15 lbs

Normal Distributions

- What is a **Normal** (**Gaussian**) distribution (**Bell Curve**)?
- **The "Shape" of Continuous Data** When we measure things in nature (height, weight, test scores), they usually clump around an average.
  - Most values are near the middle.
  - Extreme values (very small or very large) become rare.
- Conservation of mass
- We define this entire curve using just two numbers:
  - The **Mean (μ)** : The Center
  - The **Variance (σ^2)** : The Spread



$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

Naïve Bayes Classifier Example

- **Class = Cat Statistics:**
- Values: {8,9,7}
  - **Mean (μ):** 8.0
  - **Variance (σ^2):** 1.0 (Very consistent!)

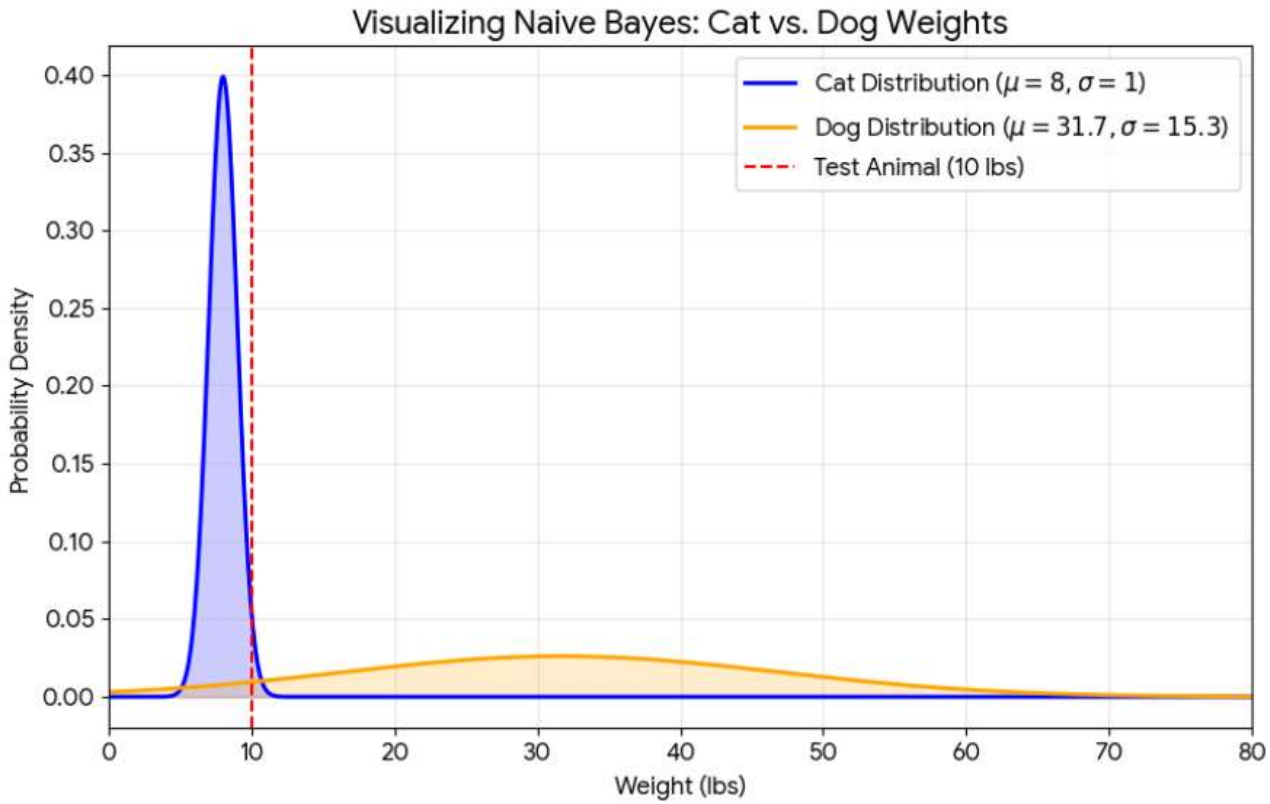
- **Class = Dog Statistics:**
- Values: {45,35,15}
  - **Mean (μ):** 31.7
  - **Variance (σ^2):** 233 (Huge spread!)

- Imagine we found an animal that weighs 10 lbs. What is it?

ID	Species (Y)	Plays Fetch? (X1)	Weight (X2)
1	**Cat**	No	8 lbs
2	**Cat**	Yes (rare!)	9 lbs
3	**Cat**	No	7 lbs
4	**Dog**	Yes	45 lbs
5	**Dog**	Yes	35 lbs
6	**Dog**	No (lazy)	15 lbs

Naïve Bayes Classifier Example

- Imagine we found an animal that weighs 10 lbs. What is it?



Naïve Bayes
Revisit Bayesian Classification

- $P(c|x) = P(x|c)P(c)/P(x)$
- $P(c)$ - Prior probability of class c – How do we know?
  - Just count up and get the probability for the Training Set – Easy!
- $P(x|c)$ - Probability "likelihood" of data vector X given that the output class is c
  - We use $P(x_1, \ldots, xn|c_j)$ as short for $P(x_1 = val_1, \ldots, x_n = val_n|c_j)$
- How do we really do this?
  - If x is nominal we can just look at the training set and count to see the probability of x given the output class c
  - If x is real valued, we can use the probability distribution to estimate

Bayes Classification

- Rephrased

$$P(c|X) = \frac{P(x_1|c) * P(x_2|c) * \ \ldots * P(x_n|c) * P(c)}{P(x_1) * P(x_2) * \cdots * P(xn)}$$

- Since the denominator remains constant for all the classes

$$P(c|x_1, x_2, \ldots, xn) \approx P(c) \prod_{i=1}^{n} P(xi|c)$$

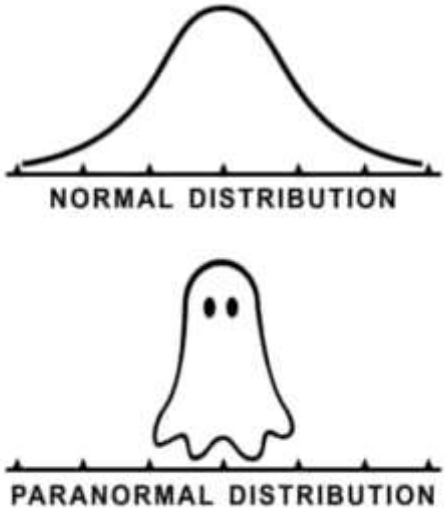
- Combine this model with a rule to pick the hypothesis that is most probable

$$c_{NB} = argmaxy\ P(c_y) \prod_{i=1}^{n} P(xi|cy)$$

Naïve Bayes Classifier

- Conditional independence may not be a reasonable assumption… (heart rate and blood pressure), but…
  - Low complexity simple approach
  - Need only store all $P(c_j)$ and $P(x_i|c_j)$ terms
  - Assume nominal features for the moment
  - Easy to calculate the $|attribute\ values| \times |classes|$ terms
  - There is often enough data to make the independent terms reasonably accurate
  - Effective and common for many large applications (Document classification, etc.)

Stretch Break (2 minutes)



Naïve Bayes (Summary)

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)
- Naïve Bayes can produce a probability estimate, but it is usually a very biased one
  - Logistic Regression is better for obtaining probabilities.

Text Classification Example

- A text classification approach
  - Want P(*class*|*document*) – How to represent document?
- We can think of our document as a *bag of words*
  - **Input 1:** "The Prince offers free money."
  - **Input 2:** "Free money offers the Prince."
  - **The "Bag":** { "Free": 1, "Money": 1, "Offers": 1, "Prince": 1, "The": 1 }
- These two messages would be treated the exact same.
- This violates independence (words are not statistically independent from one another) *but we don't care!*
- Let's do a really spam email classification example.

Text Classification Example

- **P(Spam) Prior:** 0.4 (40% of all mail is spam)
- **P(Ham) Prior:** 0.6 (60% of all mail is ham)

$$P(\text{Class}) \times P(\text{"Free"} \mid \text{Class}) \times P(\text{"Money"} \mid \text{Class})$$

- **Spam Score:**
  0.4×0.30×0.10=0.012
- **Ham Score:**
  0.6×0.01×0.02=0.00012

Word	Probability in Spam	Probability in Ham
"Free"	0.30 (30%)	0.01 (1%)
"Money"	0.10 (10%)	0.02 (2%)

Text Classification Example

- A text classification approach
  - Want P(*class*|*document*) - Use a "Bag of Words" approach – order independence assumption (valid?)
    - Variable length input of query document is fine
  - Calculate bag of words for every word/token in the language and each output class based on the training data. Words that occur in testing but do not occur in the training data are ignored.
  - Good empirical results. Can drop filler words (the, and, etc.) and words found less than *z* times in the training set.

Naïve Bayes Classifier - Problem

- If one of the conditional probability is zero, then the entire expression becomes zero
- Scenario**:** You are classifying a clear Spam email.
  - It has "Viagra" (High prob).
  - It has "Prince" (High prob).
  - But… it contains a rare word like **"Unicorn"** which we *never* saw in our Spam training data.

$$P(\text{Spam}) \times P(\text{"Viagra"}|S) \times P(\text{"Unicorn"}|S) \ldots$$

$$0.4 \times 0.5 \times \mathbf{0} = \mathbf{0}$$

  - What do we do?

Naïve Bayes Classifier - Solution

- Smoothing! Laplace (add one to each class) or something smarter like m-estimate.

- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + N_i}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

N_i: number of attribute values for attribute A_i

p: prior probability

m: parameter

Implementation details

- Computing the conditional probabilities involves multiplication of many very small numbers
  - Numbers get very close to zero, and there is a danger of numeric instability
- How do you think we could solve this?
- We can deal with this by computing the logarithm of the conditional probability

$$\log P(C|A) \sim \log P(A|C) + \log P(A)$$
$$= \sum_i \log P(A_i|C) + \log P(A)$$

Generative vs Discriminative Models

Generative

- **Example**: Naive Bayes.
- **The Philosophy**: "I want to learn what each class looks like from the inside."
- **The Goal**: Learn the blueprint (distribution) for each class separately.
- **The Math**: Learns P(X|Y) and P(Y).
  - "Given the class (Y), what is the probability of seeing these features (X)?"

Discriminative

- **Examples**: Logistic Regression, k-Nearest Neighbors (kNN).
- **The Philosophy**: "I don't care how the data was created. I just want to know where the line is."
- **The Goal**: Find the decision boundary that separates the classes.
- **The Math**: Learns P(Y|X) directly.
  - "Given the features (X), what is the probability of the class (Y)?"

Generative vs Discriminative models

- In order to classify the language of a document, you can
  - Either learn the two languages and find which is more likely to have generated the words you see
  - Or learn what differentiates the two languages.

sklearn.naive_bayes

- sklearn has 5 different implementations
  - Based on probability distribution types

`sklearn.naive_bayes`: Naive Bayes

The `sklearn.naive_bayes` module implements Naive Bayes algorithms. These are supervised learning methods based on applying Bayes' theorem with strong (naive) feature independence assumptions.

User guide: See the Naive Bayes section for further details.

`naive_bayes.BernoulliNB`(*[, alpha, ...])	Naive Bayes classifier for multivariate Bernoulli models.
`naive_bayes.CategoricalNB`(*[, alpha, ...])	Naive Bayes classifier for categorical features.
`naive_bayes.ComplementNB`(*[, alpha, ...])	The Complement Naive Bayes classifier described in Rennie et al. (2003).
`naive_bayes.GaussianNB`(*[, priors, ...])	Gaussian Naive Bayes (GaussianNB).
`naive_bayes.MultinomialNB`(*[, alpha, ...])	Naive Bayes classifier for multinomial models.

The Complement Naive Bayes classifier described in Rennie et al. (2003).
ComplementNB implements the complement naive Bayes (CNB) algorithm. CNB is an adaptation of the standard multinomial naive Bayes (MNB) algorithm that is particularly suited for imbalanced data sets. Specifically, CNB uses statistics from the *complement* of each class to compute the model's weights. The inventors of CNB show empirically that the parameter estimates for CNB are more stable than those for MNB. Further, CNB regularly outperforms MNB (often by a considerable margin) on text classification tasks.

How to choose?

- The Naive Bayes algorithm is a probabilistic classifier based on Bayes' theorem, assuming independence between features. Scikit-learn provides several implementations of Naive Bayes, each suited for different types of data:

- Gaussian Naive Bayes:
  - This is used when features are **continuous and assumed to follow a Gaussian distribution**. It estimates the mean and variance from the training data for each class and uses these to calculate the likelihood of a data point belonging to a particular class. It is suitable for datasets where features are normally distributed, such as in many scientific or engineering applications.
- Multinomial Naive Bayes:
  - This is designed for **discrete data, specifically when representing counts or frequencies**, such as word counts in text classification. It estimates the probability of each feature given a class and is appropriate for problems like spam filtering or document categorization.
- Bernoulli Naive Bayes:
  - This is used when **features are binary** (boolean values, 0 or 1), indicating the presence or absence of a particular attribute. It is often applied in text classification tasks where the feature represents the occurrence of a word in a document (e.g., using a bag-of-words model).
- Complement Naive Bayes:
  - An adaptation of Multinomial Naive Bayes particularly suited for **imbalanced datasets**. It uses the complement of each class to compute model weights, often outperforming Multinomial NB in text classification tasks with uneven class distribution.
- Categorical Naive Bayes:
  - Designed for **categorically distributed features**, it requires encoding categorical variables into numerical format, such as using ordinal encoding, before use. It's useful for datasets with categorical features that do not follow a specific distribution.