CAP 4630 – Naïve Bayes

Instructor: Aakash Kumar

University of Central Florida

What is Naive Bayes?

- **Naive Bayes** is a **classification algorithm** based on **probability**.
- It’s called “Naive” because it makes a simplifying assumption that all features are independent of each other.
- Why use Naive Bayes?
  - Fast and easy to implement.
  - Works well for large datasets and high-dimensional data (many features).
  - Effective for text classification tasks like spam detection, sentiment analysis, and document classification.

Real-World Applications of Naive Bayes

- Email Spam Detection:
  - Predicts whether an email is spam or not based on the words in the email.
- Sentiment Analysis:
  - Classifies movie or product reviews as "positive" or "negative" based on the words used.
- Medical Diagnosis:
  - Classifies whether a patient has a disease based on various symptoms.
- Text Classification:
  - Automatically categorizes documents into topics (e.g., news articles, emails).

Naive Bayes Classifier: Variables and Features

- Features Representation:

$$X = (X_1, X_2, \ldots, X_k)$$

- **k** represents the number of features.
- **Y** is the class label with K possible values (classes).
- Probabilistic Perspective

$X_i \in X$ and Y are treated as **random variables**.

- Specific Values:

The value of X_i is x.

The value of Y is y.

Naive Bayes Classifier: Making Predictions

- Goal:

  Use X to predict Y.

- Problem:

  Given a data point $X = (x_1, x_2, \ldots, x_n)$, what are the odds of Y being y?

- Mathematical Representation:

$$P(Y = y | X = (x_1, x_2, \ldots, x_n))$$

Bayes Theorem

- **Bayes Theorem** provides a way of computing posterior probability P(A | B) from P(A), P(B) and P(B | A).

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

- **P(A | B) is Posterior probability**: Probability of hypothesis A on the observed event B.
- **P(B | A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.
- **P(A) is Prior Probability:** Probability of the hypothesis before observing the evidence.
- **P(B)** is Marginal Probability: **Probability of Evidence.**

Bayes Theorem in Action

- Up to this point, Bayes' theorem hasn't been applied. Now, it's time to use it:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)} \longrightarrow \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

- Why This is Important:
  - This is a simple transformation, but it bridges the gap between what we **want** to compute and what we **can** compute.
  - **Key Insight**: We can't directly calculate P(Y|X) but we can compute P(X|Y) and P(Y) from the training data.

Bayes Theorem in Action

- Bayes' theorem allows us to reverse conditional probabilities.

$$P(Y = y \mid X = (x_1, x_2, \ldots, x_n)) = \frac{P(X = (x_1, x_2, \ldots, x_n) \mid Y = y) \cdot P(Y = y)}{P(X = (x_1, x_2, \ldots, x_n))}$$

$P(Y = y \mid X = (x_1, x_2, \ldots, x_n))$ is the **posterior probability**: the probability that the class label is y given the features.

$P(X = (x_1, x_2, \ldots, x_n) \mid Y = y)$ is the **likelihood**: the probability of the features X occurring given that the class label is y.

$P(Y = y)$ is the **prior probability**: the probability of the class label $Y = y$ occurring without any feature information.

$P(X = (x_1, x_2, \ldots, x_n))$ is the **evidence**: the overall probability of the features occurring across all classes.

Independence in Probability

- **Definition**: Two events are **independent** if the occurrence of one event does not affect the other.
- Example: Flipping two coins. The result of one coin flip does not influence the other.
- Mathematical Formula:

$$P(A \text{ and } B) = P(A) \cdot P(B)$$

Conditional Independence in Naive Bayes

- **Conditional Independence**: In Naive Bayes, features are assumed to be **independent of each other, given the class**.
- **Why It's Important**: This assumption simplifies the problem by allowing us to multiply individual probabilities instead of calculating a joint probability for all features.
- **Mathematical Formula:**

$$P(X_1, X_2, \ldots, X_n \mid Y = y) = P(X_1 \mid Y = y) \cdot P(X_2 \mid Y = y) \cdot \cdots \cdot P(X_n \mid Y = y)$$

Simplification of Baye’s Theorem

- The key simplification of the Naive Bayes classifier is the assumption that all features are **conditionally independent** given the class label. This means that:

$$P(X = (x_1, x_2, \ldots, x_n) \mid Y = y) = P(x_1 \mid Y = y) \cdot P(x_2 \mid Y = y) \cdot \cdots \cdot P(x_n \mid Y = y)$$

- This assumption greatly reduces the complexity of the computation, because instead of needing to estimate the joint probability distribution, we can estimate the individual conditional probabilities for each feature separately.

Impact on Complexity:

- Joint Probability Complexity:
  - For n features, estimating the joint probability involves calculating 2^n probabilities, leading to an exponential increase in complexity as the number of features grows.
- Naive Bayes Simplification:
  - By assuming conditional independence, we can decompose the joint probability into the product of individual conditional probabilities for each feature.
  - This reduces the number of calculations to n conditional probabilities, resulting in linear complexity O(n).

Simplification of Baye's Theorem

- Hence, we can convert following

$$P(Y = y \mid X = (x_1, x_2, \ldots, x_n)) = \frac{P(X = (x_1, x_2, \ldots, x_n) \mid Y = y) \cdot P(Y = y)}{P(X = (x_1, x_2, \ldots, x_n))}$$

- Using this condition of indepdence

$$P(X = (x_1, x_2, \ldots, x_n) \mid Y = y) = P(x_1 \mid Y = y) \cdot P(x_2 \mid Y = y) \cdot \cdots \cdot P(x_n \mid Y = y)$$

- Into following

$$P(Y = y \mid X = (x_1, x_2, \ldots, x_n)) = \frac{P(Y = y) \cdot \prod_{i=1}^{n} P(x_i \mid Y = y)}{P(X = (x_1, x_2, \ldots, x_n))}$$

How This Simplifies Computation

- The evidence term is typically difficult to compute directly, as it involves summing over all possible classes.

- However, in practice, we don't need to compute this term for classification because it is the same for all class labels. Instead, we compute the unnormalized posterior:

$$P(Y = y \mid X = (x_1, x_2, \ldots, x_n)) \propto P(Y = y) \cdot \prod_{i=1}^{n} P(x_i \mid Y = y)$$

- This gives us a score for each class y. To determine the final classification, we compare these scores for each possible class and pick the class with the highest score:

$$y_{\text{pred}} = \arg\max_{y} P(Y = y) \cdot \prod_{i=1}^{n} P(x_i \mid Y = y)$$

Steps in Naive Bayes Classification

- **Training Phase**:
  - Estimate the **prior probabilities** for each class from the training data.

$$P(Y = y)$$

  - Estimate the **conditional probabilities** for each feature and each class label.

$$P(x_i \mid Y = y)$$

    - For continuous features, we often assume the features follow a normal distribution, and we estimate the mean and variance from the training data.
    - For categorical features, we estimate the probabilities as the frequency of each feature value in the training data.
- **Prediction Phase:**
  - For a new instance with features, compute **the posterior probability** for each class.
  - Select the class that maximizes this posterior probability.

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

New Instance

x′=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)

Example

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Outlook	Play=*Yes*	Play=*No*
Sunny		
Overcast		
Rain		

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Example

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Temperature	Play=*Yes*	Play=*No*
Hot		
Mild		
Cool		

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Example

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Humidity	Play=*Yes*	Play=N*o*
High		
Normal		

Humidity	Play=*Yes*	Play=N*o*
High	3/9	4/5
Normal	6/9	1/5

Example

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Wind	Play=*Yes*	Play=*No*
Strong		
Weak		

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Example

- Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$ $\quad$ $P(\text{Play}=No) = 5/14$

Example

- Test Phase
  - Given a new instance,

    $\mathbf{x}'$=(Outlook=*Sunny*, Temperature=*Cool*, Humidity=*High*, Wind=*Strong*)
  - Look up tables

P(Outlook=*Sunny*|Play=*Yes*) = 2/9	P(Outlook=*Sunny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play=*Yes*) = 3/9	P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*Yes*) = 3/9	P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*Yes*) = 3/9	P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*Yes*) = 9/14	P(Play=*No*) = 5/14

  - MAP rule

$P(Yes|\mathbf{x}')$: $[P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]P(Play=Yes) = 0.0053$

$P(No|\mathbf{x}')$: $[P(Sunny|No)\ P(Cool|No)P(High|No)P(Strong|No)]P(Play=No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Handling Zero Frequency in Naive Bayes

- Zero Frequency Problem:
  - If a categorical variable in the test dataset contains a category that was not observed in the training dataset, the Naive Bayes classifier assigns a zero probability to that category.
  - This means the model will be unable to make a prediction because any multiplication involving zero will result in zero.
- What Causes Zero Frequency?
  - This happens when the training data is not comprehensive enough and misses certain categories or feature combinations.
  - For example, in text classification, if a certain word appears in the test data but was not present in the training data, the model will assign it a zero probability.

Solution: Smoothing Techniques

- **Smoothing** is a technique used to handle **zero frequency** by adding a small value to the count of each feature/category combination.
- Laplace Estimation (Additive Smoothing):
  - One of the simplest and most commonly used smoothing techniques.
  - It adds 1 to the count of each feature/category occurrence in the training dataset to avoid zero probabilities.
- The formula for Laplace Smoothing:

$$P(X_i|C) = \frac{\text{count}(X_i \text{ in } C) + 1}{\text{count}(C) + k}$$

- X_i is a feature value.
- C is a class.
- k is the total number of unique features.

Example of Laplace Smoothing

- Consider a dataset for predicting whether to "Play Tennis" based on weather conditions.
- Let's say in the training data, we never encountered "Wind = Strong" when the class label is "Play = Yes".
- Without smoothing:

$$P(\text{Wind} = \text{Strong}|\text{Play} = \text{Yes}) = 0$$

- With Laplace smoothing:

$$P(\text{Wind} = \text{Strong}|\text{Play} = \text{Yes}) = \frac{0+1}{\text{count of 'Play} = \text{Yes'} + k}$$

Benefits of Laplace Smoothing

- **Prevents Zero Probability**: Ensures that no feature or category gets a zero probability.
- **Improves Model Robustness**: Makes the Naive Bayes model more **resilient** to missing categories.
- **Simple to Implement**: Laplace Smoothing is computationally easy to apply and widely used in practical Naive Bayes implementations.

Advantages of Naïve Bayes Classifier

- Fast and Efficient:
  - Naive Bayes is quick and easy to use for classification tasks.
  - It performs exceptionally well in multi-class prediction problems due to its simplicity.
- Performs Well with Limited Data:
  - When the independence assumption holds, Naive Bayes can outperform more complex models like logistic regression, especially when the training dataset is relatively small.
- Effective with Categorical Data:
  - Naive Bayes tends to perform better with categorical input variables compared to numerical ones, as it does not rely heavily on numerical distribution assumptions.
- Handles High-Dimensional Data:
  - It can efficiently handle high-dimensional datasets, making it ideal for tasks like text classification (e.g., spam detection).

Disadvantages of Naïve Bayes

- Assumption of Independent Predictors:
  - A key limitation is the assumption of independence between predictors.
  - In real-world scenarios, it's rare for features to be completely independent, which can reduce the model's accuracy when this assumption is violated.
- Zero Frequency Problem:
  - If a categorical variable in the test dataset contains a category not present in the training dataset, the model assigns a zero probability to this event and fails to make a prediction.
  - This is known as the Zero Frequency problem.
  - To address this issue, smoothing techniques like Laplace Smoothing (or Additive Smoothing) can be used.
- Sensitivity to Numerical Assumptions:
  - For numerical data, Naive Bayes often assumes that the features follow a normal distribution. When the data deviates from this assumption, the model's performance can suffer.

Types of Naive Bayes Algorithms

Types of Naive Bayes Algorithms

- **Gaussian Naive Bayes:**
  - Assumption: The Gaussian Naive Bayes model assumes that the features follow a normal (Gaussian) distribution.
- Use Case:
  - It is typically used when the input features are continuous.
- Explanation:
  - If the predictors take continuous values, the model assumes that these values are drawn from a Gaussian distribution (i.e., the values form a bell curve).
  - This is in contrast to other Naive Bayes models that handle discrete features.

Types of Naive Bayes Algorithms

- **Multinomial Naive Bayes:**
  - Assumption: The Multinomial Naive Bayes classifier assumes that the data follows a multinomial distribution, which is suitable for discrete count data.
- Use Case:
  - It is commonly used in document classification problems, where the task is to categorize documents into predefined categories such as Sports, Politics, Education, etc.
- Explanation:
  - The model works by using the frequency of words (or features) in a document to predict its category.
  - Each word in the document is treated as a predictor, and its count (or occurrence) is used to calculate the likelihood of the document belonging to a particular category.

Types of Naive Bayes Algorithms

- **Bernoulli Naive Bayes:**
  - Assumption: The Bernoulli Naive Bayes classifier assumes that the features are binary (Boolean), meaning each predictor represents whether a specific attribute or word is present or absent.
- Use Case:
  - It is particularly useful for document classification tasks where the presence or absence of a word (rather than its frequency) is used to predict the category.
- Explanation:
  - For example, instead of counting how many times a word appears in a document, the Bernoulli classifier checks whether the word is present or not (1 or 0).
  - It is especially effective in tasks like text classification with binary features, such as spam detection, where the existence of certain keywords is more important than their frequency.