Bayesian Classifier

Bayesian Classifier

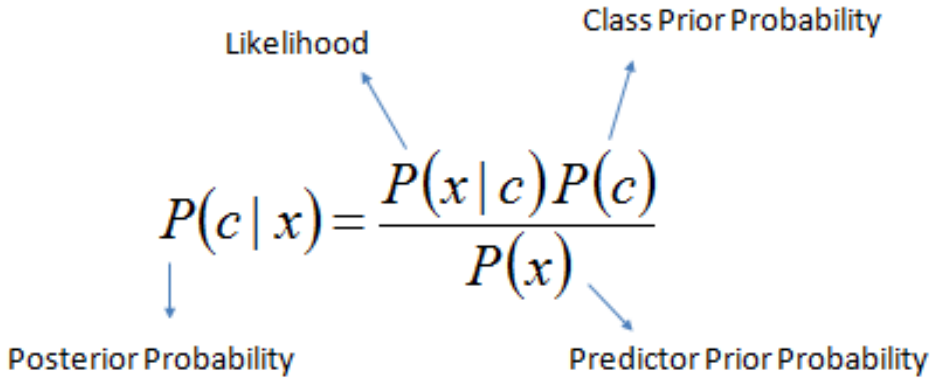
- A statistical classifier
  - Performs probabilistic prediction, i.e., predicts class membership probabilities
- Foundation
  - Based on Bayes' Theorem
- Core Idea:
  - Calculate the probability of each class given the features of the input, and assign the class with the highest probability.
- Assumptions
  1. The classes are mutually exclusive and exhaustive.
  2. The attributes are independent given the class (often unrealistic).
- Called Naïve classifier because of these assumptions.

Bayesian Classifier

- In many applications, the relationship between the attribute set and the class variable is non-deterministic (not fixed/certain).
  - In other words, a test cannot be classified to a class label with certainty.
  - In such a situation, the classification can be achieved probabilistically.
- The Bayesian classifier is an approach for modelling probabilistic relationships between the attribute set and the class variable.
- More precisely, Bayesian classifier uses Bayes' Theorem of Probability for classification.
- Note
  - A deterministic relationship means that given a particular set of attributes (input features), there is only one possible class label (output).

Bayes' Theorem

- Bayes' theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).
  - The posterior probability refers to the conditional probability of an event c occurring, given that another event x has already occurred.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- ✓ P(c|x) - posterior probability of class (c, target) given feature (x, attribute).
- ✓ P(x|c) - likelihood of feature x given class c.
- ✓ P(c) - prior probability of class c.
- ✓ P(x) - prior probability of feature x.

Naïve Bayes' Classifier (most common Bayesian classifier)

Assumes: All features are independent given the class label.

Step 1**:**

Compute prior probabilities: P(c) for each class.

Step 2:

Compute likelihood: $P(x_1 \mid c)$, $P(x_2 \mid c)$,…for each feature.

Step 3:

Use Bayes' Theorem to compute P(c | x) for all classes.

Step 4:

Choose class with the highest posterior probability.

Example (Text Classification)

- Suppose we want to classify an email as Spam or Not Spam based on the words it contains.

We compute:

- P(Spam), P(Not Spam) → prior probabilities
- $P(word_i \mid Spam)$, $P(word_i \mid Not\ Spam)$ → likelihood
- Then apply Bayes' Theorem to compute which class is more likely.

Example (Text Classification)

Pros	Cons	Solutions / Workarounds
1. **Simple and Fast** – Easy to implement and computationally efficient.	1. **Strong Independence Assumption** – Often unrealistic in real data.	Use feature selection to reduce redundancy, or try semi-naive methods.
2. **Performs Well with Small Datasets** – Needs less training data.	2. **Zero Probability Problem** – Unseen features lead to zero probability.	Apply **Laplace Smoothing** to avoid zero probabilities.
3. **Effective in High Dimensions** – Ideal for text and document classification.	3. **Ignores Feature Interactions** – Cannot capture feature dependencies.	Use models like Decision Trees or Random Forests if interaction is key.
4. **Robust to Irrelevant Features** – Unhelpful features don't degrade performance much.	4. **Assumes Distribution for Continuous Data** – E.g., Gaussian assumption may not hold.	Use **kernel density estimation** or discretize continuous features.
5. **Handles Missing Data Well** – Can classify even with incomplete input.	5. **Underperforms on Complex Tasks** – May lag behind advanced models.	Compare with models like **SVM**, **Random Forest**, or **XGBoost**.

Prior and Posterior Probabilities

- P(A) and P(B) are called prior probabilities
- P(A|B), P(B|A) are called posterior probabilities

Example: Prior versus Posterior Probabilities

- This table shows that the event *Y* has two outcomes namely *A* and *B*, which is dependent on another event *X* with various outcomes like x_1, x_2 and x_3.
- **Case1:** Suppose, we don't have any information of the event *A*. Then, from the given sample space, we can calculate $P(Y = A) = \frac{5}{10} = 0.5$
- **Case2:** Now, suppose, we want to calculate $P(X = x_2 | Y = A) = \frac{2}{5} = 0.4$.

The later is the conditional or posterior probability, where as the former is the prior probability.

X	Y
x_1	A
x_2	A
x_3	B
x_3	A
x_2	B
x_1	A
x_1	B
x_3	B
x_2	B
x_2	A

Example-1

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

$$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$$

Example-1

$$P(PlayTennis = yes) = 9/14 = .64$$

$$P(PlayTennis = no) = 5/14 = .36$$

Outlook	Y	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5
Tempreature		
hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

Humidity	Y	N
high	3/9	4/5
normal	6/9	1/5
Windy		
Strong	3/9	3/5
Weak	6/9	2/5

Example-1

$$\langle Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong \rangle$$

$$v_{NB} = \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

$$= \underset{v_j \in \{yes, no\}}{\operatorname{argmax}} P(v_j) \quad P(Outlook = sunny | v_j) P(Temperature = cool | v_j)$$
$$\cdot P(Humidity = high | v_j) P(Wind = strong | v_j)$$

Example-1

$$v_{NB}(yes) = P(yes)\ P(sunny|yes)\ P(cool|yes)\ P(high|yes)\ P(strong|yes) = .0053$$

$$v_{NB}(no) = P(no)\ P(sunny|no)\ P(cool|no)\ P(high|no)\ P(strong|no) = .0206$$

- Normalizing the probabilities

$$v_{NB}(yes) = \frac{v_{NB}(yes)}{v_{NB}(yes)+v_{NB}(no)} = 0.205 \qquad v_{NB}(no) = \frac{v_{NB}(no)}{v_{NB}(yes)+v_{NB}(no)} = 0.795$$

Example-2

- Dataset

No	Color	Legs	Height	Smelly	Species
1	White	3	Short	Yes	M
2	Green	2	Tall	No	M
3	Green	3	Short	Yes	M
4	White	3	Short	Yes	M
5	Green	2	Short	No	H
6	White	2	Tall	No	H
7	White	2	Tall	No	H
8	White	2	Short	Yes	H

- New instance

(Color=Green, legs=2, Height=Tall, and Smelly=No)

Example-2

$$P(M) = \frac{4}{8} = 0.5 \quad P(H) = \frac{4}{8} = 0.5$$

Color	M	H
White	2/4	3/4
Green	2/4	1/4

Legs	M	H
2	1/4	4/4
3	3/4	0/4

Height	M	H
Tall	3/4	2/4
Short	1/4	2/4

Smelly	M	H
Yes	3/4	1/4
No	1/4	3/4

Example-2

$$p(M|New\ Instance) = p(M) * p(Color = Green|M) * p(Legs = 2|M) * p(Height = tall|M) * p(Smelly = no\ |M)$$

$$p(M|New\ Instance) = 0.5 * \frac{2}{4} * \frac{1}{4} * \frac{3}{4} * \frac{1}{4} = 0.0117$$

$$p(H|New\ Instance) = p(H) * p(Color = Green|H) * p(Legs = 2|H) * p(Height = tall|H) * p(Smelly = no\ |H)$$

$$p(H|New\ Instance) = 0.5 * \frac{1}{4} * \frac{4}{4} * \frac{2}{4} * \frac{3}{4} = 0.047$$

$$p(H|New\ Instance) > p(M|New\ Instance)$$

Hence the new instance belongs to Speices H

Example-3

- Dataset

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

- New instance

$$(Red, SUV, Domestic)$$

Example-3

$$p(Yes) = \frac{5}{10} = 0.5$$

$$p(No) = \frac{5}{10} = 0.5$$

Color	Yes	No
Red	3/5	2/5
Yellow	2/5	3/5

Type	Yes	No
Sports	4/5	2/5
SUV	1/5	3/5

Origin	Yes	No
Domestic	2/5	3/5
Imported	3/5	2/5

Example-3

$$P(Yes|New\ Instance) = p(Yes) * P(Color = Red|Yes) * P(Type = SUV|Yes) * P(Origin = Domestic|Yes)$$

$$P(Yes|New\ Instance) = \frac{5}{10} * \frac{3}{5} * \frac{1}{5} * \frac{2}{5} = \frac{3}{125} = 0.024$$

$$P(No|New\ Instance) = p(No) * P(Color = Red|No) * P(Type = SUV|No) * P(Origin = Domestic|No)$$

$$P(No|New\ Instance) = \frac{5}{10} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5} = \frac{9}{125} = 0.072$$

$$New\ Instance = (Red, SUV, Domestic) \qquad No$$

Air-Traffic Data

Days	Season	Fog	Rain	Class
Weekday	Spring	None	None	On Time
Weekday	Winter	None	Slight	On Time
Weekday	Winter	None	None	On Time
Holiday	Winter	High	Slight	Late
Saturday	Summer	Normal	None	On Time
Weekday	Autumn	Normal	None	Very Late
Holiday	Summer	High	Slight	On Time
Sunday	Summer	Normal	None	On Time
Weekday	Winter	High	Heavy	Very Late
Weekday	Summer	None	Slight	On Time

Cond. to next slide…

Air-Traffic Data

Cond. from previous slide…

Days	Season	Fog	Rain	Class
Saturday	Spring	High	Heavy	Cancelled
Weekday	Summer	High	Slight	On Time
Weekday	Winter	Normal	None	Late
Weekday	Summer	High	None	On Time
Weekday	Winter	Normal	Heavy	Very Late
Saturday	Autumn	High	Slight	On Time
Weekday	Autumn	None	Heavy	On Time
Holiday	Spring	Normal	Slight	On Time
Weekday	Spring	Normal	None	On Time
Weekday	Spring	Normal	Heavy	On Time

Air-Traffic Data

- In this database, there are four attributes

  A = [Day, Season, Fog, Rain]

  with 20 tuples.
- The categories of classes are:

  C= [On Time, Late, Very Late, Cancelled]

- Given this is the knowledge of data and classes, we have to find most likely classification for any other unseen instance, for example:

Week Day	Winter	High	None	???

- Classification technique eventually to map this tuple into an accurate class.

Naïve Bayesian Classifier

- Tabulate all the posterior and prior probabilities as shown below.

		Class			
Attribute		On Time	Late	Very Late	Cancelled
Day	Weekday	9/14 = 0.64	½ = 0.5	3/3 = 1	0/1 = 0
	Saturday	2/14 = 0.14	½ = 0.5	0/3 = 0	1/1 = 1
	Sunday	1/14 = 0.07	0/2 = 0	0/3 = 0	0/1 = 0
	Holiday	2/14 = 0.14	0/2 = 0	0/3 = 0	0/1 = 0
Season	Spring	4/14 = 0.29	0/2 = 0	0/3 = 0	0/1 = 0
	Summer	6/14 = 0.43	0/2 = 0	0/3 = 0	0/1 = 0
	Autumn	2/14 = 0.14	0/2 = 0	1/3= 0.33	0/1 = 0
	Winter	2/14 = 0.14	2/2 = 1	2/3 = 0.67	0/1 = 0

Naïve Bayesian Classifier

Attribute		Class: On Time	Class: Late	Class: Very Late	Class: Cancelled
Fog	None	5/14 = 0.36	0/2 = 0	0/3 = 0	0/1 = 0
	High	4/14 = 0.29	1/2 = 0.5	1/3 = 0.33	1/1 = 1
	Normal	5/14 = 0.36	1/2 = 0.5	2/3 = 0.67	0/1 = 0
Rain	None	5/14 = 0.36	1/2 = 0.5	1/3 = 0.33	0/1 = 0
	Slight	8/14 = 0.57	0/2 = 0	0/3 = 0	0/1 = 0
	Heavy	1/14 = 0.07	1/2 = 0.5	2/3 = 0.67	1/1 = 1
Prior Probability		14/20 = 0.70	2/20 = 0.10	3/20 = 0.15	1/20 = 0.05

Naïve Bayesian Classifier

Instance:

Week Day	Winter	High	Heavy	???

Case1: Class = On Time : $0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$

Case2: Class = Late : $0.10 \times 0.50 \times 1.0 \times 0.50 \times 0.50 = 0.0125$

Case3: Class = Very Late : $0.15 \times 1.0 \times 0.67 \times 0.33 \times 0.67 = 0.0222$

Case4: Class = Cancelled : $0.05 \times 0.0 \times 0.0 \times 1.0 \times 1.0 = 0.0000$

Case 3 is the strongest, therefore, the correct classification is **Very Late**

Pros

- It is easy and fast to predict class of test data set. It also performs well in multi class prediction
- Simple to Implement. The conditional probabilities are easy to evaluate.
- Very fast – no iterations since the probabilities can be directly computed. So this technique is useful where speed of training is important.
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It performs well in case of categorical input variables compared to numerical variable(s).
  - For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction.
  - This is often known as Zero Frequency. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called **Laplace Estimation**.

- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Laplace Smoothing

- It is introduced to solve the problem of zero probability
- If that word has never appeared in training data for that class, its probability becomes zero.
  - When using models that multiply probabilities (like Naive Bayes), this zero wipes out the entire result—even if the rest of the words are a good match.
- Laplace smoothing prevents zero probabilities by assuming every possible event has occurred at least once.
- It ensures that posterior probabilities are never zero.
  - We add 1 to the numerator, and k (number of different features/values) to the denominator.
  - So, in the case that we don't have a particular ingredient in our training set, the posterior probability comes out to 1 / N + k instead of zero.

Example

- **Test data= x_1x_2x'**

- Let a test sample has three words, where we assume x_1 and x_2 are present in the training data but not x'. The probability equation becomes:

  - **P(positive/review) = P(x_1/positive)*P(x_2/positive)*P(x'/positive)*P(positive)**

  - **P(negative/review) = P(x_1/negative)*P(x_2/negative)*P(x'/negative)*P(negative)**

- In the likelihood table, the value of P(x_1/positive), P(x_2/positive) and P(positive) are present but P(x'/positive) is not present since x' is not present in our training data. Thus, there is no value for the probability.

Example

- Using Laplace smoothing, we can represent P(x'|positive) as,

  - **P(x'/positive) = (number of reviews with x' and target_outcome = positive + α) / (N+ α*k)**

    - Here, alpha (α) represents the smoothing parameter,
    - k represents the dimensions (no. of features) in the data,
    - N represents the number of reviews with target_outcome = positive

- If we choose a value of α != 0 (not equal to 0), the probability will no longer be zero even if a word is not present in the training dataset.

Example

🔷 Formula:

$$P(x' \mid \text{positive}) = \frac{\text{count}(x' \text{ in positive reviews}) + \alpha}{N + \alpha \cdot k}$$

🔷 Explanation of terms:

Symbol	Meaning
x'	A feature (typically a word or token)
positive	Target class (e.g., positive sentiment)
α	Smoothing parameter (typically 1 for Laplace smoothing)
N	Total number of words (tokens) in all reviews labeled as **positive**
k	Vocabulary size (number of unique words)

Example

Email	Content	Label
1	"win money"	Spam
2	"meeting tomorrow"	Non-Spam

🔷 Vocabulary (V)

All unique words: **{win, money, meeting, tomorrow}**

So:

- $k = 4$ (vocabulary size)

Example

🔷 Word Counts

Spam:

- Words: "win", "money"
- Total words $N_{\text{spam}} = 2$
- Counts: win = 1, money = 1

Non-Spam:

- Words: "meeting", "tomorrow"
- Total words $N_{\text{nonspam}} = 2$
- Counts: meeting = 1, tomorrow = 1

Example

🔷 Apply Laplace Smoothing (α = 1)

Formula:

$$P(x' \mid \text{class}) = \frac{\text{count}(x') + \alpha}{N + \alpha \cdot k}$$

Example:

P("win" | Spam)

$$= \frac{1+1}{2+1\cdot 4} = \frac{2}{6} = 0.333$$

P("meeting" | Spam)

$$= \frac{0+1}{2+4} = \frac{1}{6} = 0.167$$

P("meeting" | Non-Spam)

$$= \frac{1+1}{2+4} = \frac{2}{6} = 0.333$$

Example

🧠 Use Case: New Email = `"win meeting"`

Calculate probabilities:

For Spam:

$$P(\text{"win"} \mid \text{Spam}) \cdot P(\text{"meeting"} \mid \text{Spam}) = 0.333 \cdot 0.167 = 0.0556$$

For Non-Spam:

$$P(\text{"win"} \mid \text{Non-Spam}) = \frac{0+1}{2+4} = 0.167$$

$$P(\text{"meeting"} \mid \text{Non-Spam}) = 0.333$$

$$0.167 \cdot 0.333 = 0.0556$$

So both classes score the **same**, meaning the classifier is **uncertain** without additional data or prior probabilities.

Example

- Let's say the occurrence of word x is 3 with target_outcome=positive in training data. Assuming we have 4 features in our dataset, i.e., k = 4 and N = 200 (total number of positive reviews). Then

- **$P(x'/positive) = 3 + \alpha / 200 + 4*\alpha$**
  - **Case 1** when alpha = 1 => $P(x' \mid positive) = 4/204 = 0.02$
  - **Case 2** when alpha = 100 => $P(x' \mid positive) = 103/600 = 0.17$
  - **Case 3** when alpha=1000 => $P(x' \mid positive) = 1003/4200 = 0.24$
  - **Case 4** when alpha=5000 => $P(x' \mid positive) = 5003/20200 = 0.25$

- As alpha increases, the likelihood probability drives towards uniform distribution i.e. the probability value will be 0.5

Thanks.