Naive Bayes

1 Introduction

Naive Bayes is a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. It is highly scalable, requiring a number of parameters linear in the number of features in a learning problem.

2 Bayes' Theorem

Bayes' theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event. The theorem is stated mathematically as:

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

where:

- $P(A|B)$ is the posterior probability of class A given predictor B.
- $P(B|A)$ is the likelihood which is the probability of predictor B given class A.
- $P(A)$ is the prior probability of class A.
- $P(B)$ is the prior probability of predictor B.

3 Naive Assumption

The "naive" assumption of Naive Bayes is that all features are independent of each other given the class. This simplifies the computation of the posterior probability by assuming:

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

4 Types of Naive Bayes Classifiers

There are several types of Naive Bayes classifiers, including:

- **Gaussian Naive Bayes**: Assumes that the continuous values associated with each feature are distributed according to a Gaussian (normal) distribution.
- **Multinomial Naive Bayes**: Used for discrete counts, such as word counts in text classification.
- **Bernoulli Naive Bayes**: Used for binary/boolean features.

5 Algorithm

Algorithm 1: Training a Naive Bayes Classifier

Input: Training data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
Output: A Naive Bayes model
for *each class c in the dataset* **do**
  Calculate the prior probability $P(c)$
end
for *each feature f* **do**
  for *each class c* **do**
    Calculate the likelihood $P(f|c)$
  end
end

Algorithm 2: Inference with a Naive Bayes Classifier

Input: A Naive Bayes model, new data point x
Output: Predicted class $\hat{y}$
for *each class c* **do**
  Initialize $P(c|x) = P(c)$ **for** *each feature f in x* **do**
    Update $P(c|x)$ with $P(f|c)$
  end
end
return *class c with highest* $P(c|x)$

6 Applications

Naive Bayes classifiers are used in various applications such as:

- Spam filtering
- Text classification
- Sentiment analysis
- Recommender systems

7 Conclusion

Naive Bayes is a simple yet powerful algorithm for classification tasks. Its strong independence assumptions may not always hold, but it often performs surprisingly well in practice, especially for text classification problems.