Scaler Companion beta

Class Revision Notes on Naive Bayes

This document provides a comprehensive summary of the Naive Bayes class, including definitions, calculations, concepts, and computing aspects discussed during the session.

Overview

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that the presence of a feature in a class is independent of any other feature. It's widely used for text classification due to its simplicity and efficiency, but it's called "naive" because it assumes conditional independence, which is not always the case .

Naive Bayes Classification

Key Concepts

- **Bayes Theorem:** Forms the foundation for Naive Bayes. For a given set of words {w1, w2, ..., wd} in a document: $P(y = 1|w_1, w_2..w_d) = \frac{P(y=1) \cdot P(w_1|y=1) \cdot P(w_2|y=1) \cdot \ldots \cdot P(w_d|y=1)}{P(w_1) \cdot P(w_2) \cdot \ldots \cdot P(w_d)}$ During classification, we focus on the numerator since the denominator is consistent across classes .
- **Conditional Independence**: The assumption that features are conditionally independent given the class label. In practice, words in documents aren't entirely independent, but the model often provides good results regardless .

Example Classification

1. **Training Phase:**
   - Create vocabulary of unique words from the dataset.
   - Compute prior probabilities for each class based on their frequency in the training dataset.

2. **Testing Phase**:
   - For a new text, compute score for each class by summing up the log probabilities: $log\,P(y/text) = log\,P(y) + \sum log\,P(w_i/y)$
   - Assign text to class with higher score .
   - Typically involve techniques like Laplace Smoothing to eliminate zero probabilities .

Common Issues

- **Underflow**: When the product of many small probabilities becomes too small to represent. Addressed by summing logs of probabilities instead of multiplying them directly .
- **Imbalance**: Imbalanced data can skew predictions towards the majority class. Sampling techniques or adjusting class priors can mitigate this .

Computational Aspects

- **Training Complexity**: Process involves iterating over each document and word to set up the dictionaries of word frequencies etc. .
- **Test Complexity**: Typically O(K) where K is the number of features in the text .

Feature Importance and Handling

- Features (words) with high conditional probabilities are important for the classification decision. Words with high occurrence in spam, for example, are regarded as strong indicators for spam classification .

Variants and Extensions

- **Multinomial Naive Bayes**: Considers not just presence but the frequency of words in documents. Useful in scenarios where word frequency matters significantly .
- **Gaussian Naive Bayes**: Handles continuous data by assuming the features are normally distributed .

The Naive Bayes algorithm is highly effective for text classification tasks despite its naive feature independence assumption. It is efficient, interpretable, and works well with relatively small datasets or as a baseline for text classification models .