Comprehensive Revision Notes on Naive Bayes and Laplace Smoothing

Introduction to Naive Bayes

Naive Bayes is a straightforward and powerful algorithm for classification problems, particularly in the context of text classification like spam detection. The fundamental premise of Naive Bayes is based on Bayes' Theorem, with a strong assumption that features are conditionally independent given the class label 【4:18†typed.md】.

Bayes Theorem

Bayes' Theorem provides a way to calculate the probability of a hypothesis given our prior knowledge. The theorem is stated as:

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

Where $P(y|X)$ is the posterior probability, $P(X|y)$ is the likelihood, $P(y)$ is the prior, and $P(X)$ is the evidence 【4:3†typed.md】.

Naive Assumption

The naive aspect stems from assuming that all features (words in the context of spam detection) are independent of each other given the class label (spam or not spam). This allows the simplification of the likelihood to a product of individual probabilities:

$$P(X|y) = \prod_{i=1}^{d} P(w_i|y)$$

Despite being a strong assumption, it simplifies computation and works effectively in practice【4:16†typed.md】.

Class Priors

Class priors are the probabilities $P(y = 1)$ and $P(y = 0)$, calculated as the ratio of the samples in each class to the total samples in the dataset:

$$P(y = 1) = \frac{Number\ of\ samples\ with\ y=1}{Total\ number\ of\ samples}$$

Similarly, for $P(y = 0)$ 【4:3†typed.md】.

Likelihood Calculation

The likelihood of each word given a class is calculated using:

$$P(w_j|y = 1) = \frac{Count\ of\ occurrences\ of\ w_j\ in\ class\ y=1}{Total\ count\ of\ words\ in\ class\ y=1}$$

This is done for both classes (spam and not spam) 【4:6†typed.md】.

Training and Testing in Naive Bayes

Training Time Complexity

Training involves calculating the likelihoods for each feature (word) for both classes. As such, the training time complexity is $O(nd)$, where n is the number of samples and d is the number of features 【4:4†typed.md】.

Test Time Complexity

During testing, for each input text, we calculate the posterior probabilities for each class and select the class with the higher probability. This gives a complexity of $O(d)$ for each test example 【4:4†typed.md】.

Laplace Smoothing

set wasn't seen during training, its probability would be zero, making the product of probabilities zero and rendering classification impossible. To combat this, a small constant is added:

$$P(w_j/y = 1) = \frac{n_{j1}+\alpha}{n_1+C\alpha}$$

Where α is the smoothing constant, and C is the number of classes. Typically, α is set to 1【4:11†typed.md】【4:12†transcript.txt】.

Applying Laplace Smoothing

When Laplace smoothing is applied, it doesn't just affect unseen words but is uniformly applied to all likelihoods to ensure consistency across the dataset. This method maintains mathematical integrity while practically solving the problem of zero probability【4:11†typed.md】.

The use of this smoothing technique is crucial in real-world applications where certain words might not appear during training but are significant during testing【4:1†transcript.txt】.

Conclusion

Naive Bayes, complemented by Laplace Smoothing, is a powerful method for tackling text classification tasks. Its strength lies in the simplicity and effectiveness of applying Bayes' Theorem under conditional independence assumptions. Despite its simplicity, it has proved to be robust in applications like spam detection. Understanding these foundations allows for practical implementation and further exploration into more advanced topics like dealing with imbalanced data and fine-tuning models【4:9†typed.md】【4:14†typed.md】.