Naive Bayes - Quiz Prep Summary (with Detailed Answers)

Ailin Ghoreishi

Overview

- **Naive Bayes** is a probabilistic classifier that applies **Bayes' Theorem** with the *naive assumption* that all features are **independent given the class**.
- It is simple, fast, and works well for **text classification**, **spam filtering**, and other **high-dimensional problems**.
- The Gaussian version models numeric features using **normal distributions**.

Discriminant Function Formula

$$\delta_k(x) \propto \log \left[\pi_k \times \prod f_{kj}(x_j)\right]$$

This expression scores each class `k` based on the **prior probability** (π_k) and the **likelihood** of each feature ($f_{kj}(x_j)$). Under the **Gaussian assumption**, each feature is modeled as a normal distribution:

$$\delta_k(x) = -\frac{1}{2} \sum \left[(x_j - \mu_{kj})^2 / \sigma_{kj}^2 + \log(\sigma_{kj}^2)\right] + \log(\pi_k)$$

Where:

- μ_{kj}: Mean of feature `j` in class `k`
- σ_{kj}^2: Variance of feature `j` in class `k`
- π_k: Prior probability of class `k`
- x_j: Feature value for the input instance
- **p**: Number of features

Sample Quiz Questions (Extended Answers)

🔹 Conceptual Questions

Q1: What is the naive assumption in Naive Bayes?
A: The naive assumption is that all features (e.g., x_1, x_2, ..., x_p) are **conditionally independent given the class label**.
This simplifies the math drastically — instead of needing the joint probability of all features together, we just multiply individual feature probabilities. While this independence often isn't true in real life, the classifier can still work surprisingly well.

Q2: When is Naive Bayes especially useful?
A: Naive Bayes shines in scenarios like:

- **High-dimensional data** (e.g., text with thousands of words as features)
- **Small datasets** where complex models may overfit
- **Real-time or fast predictions** (very efficient to train and use)
- It's a good **baseline model** for classification tasks.

Q3: Why does Naive Bayes often work well despite unrealistic assumptions?
A: Even when features aren't truly independent, Naive Bayes can still perform well because:

- The errors from independence cancel out or don't affect class ranking much.
- It focuses on **which class has the highest probability**, not the exact values.
- The simplicity helps avoid overfitting, especially with limited data.

🔹 Mathematical Questions

Q4: What does the formula $\delta_k(x)$ represent?
A: It represents a **score** (called a discriminant function) for class `k`, based on the **log of the posterior probability**.
You compute $\delta_k(x)$ for each class and choose the class with the **highest score**.
It avoids multiplying many tiny probabilities (which can cause underflow) by using logarithms to work with **sums** instead of products.

Q5: Why do we take the logarithm of the product in the formula?
A: Taking the logarithm:

- **Converts multiplication into addition**: Easier and faster to compute.

- **Prevents numerical underflow**: Small probabilities multiplied together can become extremely close to zero.

- Keeps the math more **numerically stable** and scalable for high-dimensional data.

Q6: What do μ_{kj}, σ_{kj}^2, and π_k represent in the model?
A:

- **μ_{kj}**: The **mean** of feature `j` for all training examples in class `k` (center of the Gaussian distribution).

- **σ_{kj}^2**: The **variance** of feature `j` in class `k` (spread of the distribution).

- **π_k**: The **prior probability** of class `k` (how common class `k` is in the training data).

These are all estimated from the training data and used to compute the class score for new inputs.

🔹 Practical Questions

Q7: Can Naive Bayes handle categorical variables?
A: Yes! For categorical variables:

- You can use **frequency counts** (how often a value occurs in each class).

- **Multinomial Naive Bayes** is often used for **text data**, modeling word counts.
- **Bernoulli Naive Bayes** is used for binary features (e.g., word present or not).

Q8: What's the difference between Gaussian and Multinomial Naive Bayes?
A:

- **Gaussian Naive Bayes**: Assumes **continuous** features follow a **normal distribution**. Good for things like age, height, exam scores, etc.
- **Multinomial Naive Bayes**: Used for **count data** (e.g., number of times a word appears). Common in **text classification**.

Q9: Can Naive Bayes handle mixed data types (numerical + categorical)?
A: Yes. You can model each feature according to its type:

- Use **Gaussian** distributions for numeric features.
- Use **categorical/multinomial** distributions for discrete features. This flexibility makes Naive Bayes suitable for mixed datasets with different variable types.

✅ Key Takeaways

- Naive Bayes is **fast**, **simple**, and often **very accurate** in practice.
- Even if its assumptions (like independence) are unrealistic, it still performs well.
- It's widely used in **spam detection**, **text classification**, **medical diagnosis**, and more.
- Makes predictions using **log-likelihoods** and **Bayesian probabilities**.
-