Navie Bayes & Support Vector Machine

Jerry Peng

1 Laplace Smoothing

Laplace smoothing (also known as add-one smoothing) is a technique to prevent conditional probabilities from being 0 in Naive Bayes. Its main purpose is to address the issue of zero probabilities when calculating probabilities. This situation is especially common in text classification tasks, for example, when a word has never appeared in a category in the training data. The basic idea of Laplace smoothing is to add a positive value (usually 1) to the frequency of each word to avoid situations where the probability is 0. When using Laplace smoothing, the formula for calculating conditional probability is:

$$P(w|c) = \frac{count(w,c)+1}{count(c)+|V|} \tag{1}$$

where:

- $P(w|c)$ is the conditional probability of word w given category c.
- $count(w,c)$ is the occurrence count of word w in category c.
- $count(c)$ is the total number of words in category c.
- $|V|$ is the vocabulary size in the training data.

In this way, even if a word has never appeared in a certain category, its conditional probability will not be 0. This is because the numerator (word frequency) is at least 1, and the denominator is the total number of words plus the vocabulary size (ensuring the denominator is greater than the numerator, so the probability value is between 0 and 1).

2 Navie Bayes

Naive Bayes is a set of supervised learning algorithms based on Bayes' theorem, used for classification problems. The "naive" in Naive Bayes comes from its assumption that features are mutually independent. Although this assumption rarely holds in real-world applications, Naive Bayes performs well in many practical problems, especially in text classification. The formula of Navie Bayes is defined as:

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

In classification problems, the composition of the aforementioned definition is as follows:

- $P(A|B)$ is posterior probability: It represents the probability of event A occurring given the feature B.
- $P(B|A)$ is likelihood: It represents the probability of feature B occurring given the category A.
- $P(A)$ is the prior probability, representing the probability of category A occurring.
- $P(B)$ is the evidence or marginal probability, representing the probability of feature B occurring.

In Naive Bayes classification, what we are looking for is the posterior probability of each category given feature X. Then, the sample is assigned to the category with the highest posterior probability. For example, suppose there is a feature set $X = \{x_1, x_2, \ldots, x_n\}$, then, the independence assumption of Naive Bayes is:

$$P(X|A) = P(x_1|A) \times P(x_2|A) \times \cdots \times P(x_n|A)$$

Here are the steps of how to calculate Naive Bayes:

1. Calculate prior probabilities:
   $P(y = c)$ for each class c:

$$P(y = c) = \frac{\text{Nums of samples in class } c}{\text{Totoal nums of samples}}$$

2. Calculate likelihood:
   $P(x_i|y = c)$ for each feature x_i and class c:

$$P(x_i|y = c) = \frac{\text{Nums of samples in class } c \text{ with } x_i}{\text{Nums of samples in class } c}$$

3. Calculate posterior probabilities:
   For new data point with features $x_1, x_2, \ldots, x_n$ calculate the posterior probability for each class c:

$$P(y = c|x_1, x_2, \ldots, x_n) \propto P(y = c) \times P(x_1|y = c) \times P(x_2|y = c) \times \cdots \times P(x_n|y = c)$$

4. Choose the class c that has the highest posterior probability.

3 Support Vector Machine

SVM (Support Vector Machine) is a supervised learning algorithm primarily used for classification and regression tasks. Its fundamental idea is to find a hyperplane (in high-dimensional space) that maximizes the margin between two classes. Figure 1 is a simple SVM

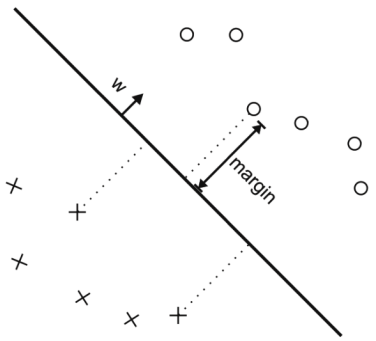

Figure 1: Simple SVM

Goal In a dataset with N dimensions (where N represents the number of features), the goal is to find the optimal decision boundary, also called a hyperplane, that accurately separates data points of different classes.

Support Vectors Support vectors are the core concept in the SVM algorithm. They are those data points in the dataset that lie close to the decision boundary defined by the SVM. Specifically, support vectors are the data points that determine the position of the maximum margin hyperplane. Only support vectors contribute to the decision function of the SVM model. This means that if you remove other points in the dataset but keep the support vectors, the SVM decision boundary will not change. However, if you remove or alter a support vector, the decision boundary might be affected.

Margin The minimum distance between the support vectors and the hyperplane is referred to as the margin.

3.1 Calculate the Hyperplane

Suppose we have a binary classification task. We have a dataset containing n labeled data points, each denoted as (x_i, y_i). Each $x_i \in R^d$ is a feature vector, and each $y_i \in -1, 1$ is the corresponding label. Therefore, we can compute the hyperplane using the following formula:

$$f(x) = \mathbf{w}x + b = 0 \tag{2}$$

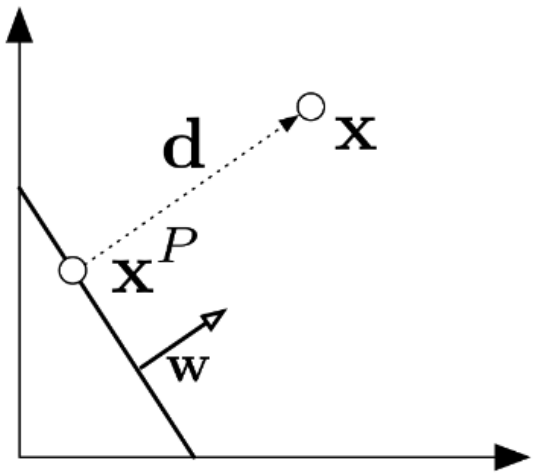

Figure 2: calculate margin

3.1.1 Margin

Before computing the specific Hyperplane, we need to calculate the Margin. Taking Figure 2 as an example. Suppose d is the vector of minimum length between hyperplane H and x, and x^P is a projection of x on H. Then, we can derive the formula: $x^P = x - d$. Since d is parallel to $\mathbf{w}$, we have $d = \alpha\mathbf{w}$ for some $\alpha \in R$. Based on equation 2, we can derive $\mathbf{w}^T x + b = \mathbf{w}^T(x - d) + b = \mathbf{w}^T(x - \alpha\mathbf{w}) + b = 0$. Rearranging this expression, we can obtain:

$$\alpha = \frac{\mathbf{w}^T x + b}{\mathbf{w}^T \mathbf{w}} \tag{3}$$

Thus, we can determine the length of d, which is:

$$\|d\|_2 = \sqrt{d^T d} = \sqrt{\alpha^2 d^T d} = \frac{|\mathbf{w}^T x + b|}{\|\mathbf{w}\|_2} \tag{4}$$

Hence, the margin of H respect to D is

$$\gamma(\mathbf{w}, b) = \min_{x \in D} \frac{|\mathbf{w}^T x + b|}{\|\mathbf{w}\|_2} \tag{5}$$

3.1.2 Max Margin Classifier

We can formulate the search for the maximum margin separating hyperplane as a constrained optimization problem. The objective is to maximize the margin under the constraint that all data points must lie on the correct side of the hyperplane:

$$\max_{\mathbf{w}, b} \gamma(\mathbf{w}, b) \text{ such that } \forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0$$

Substituting $\gamma(\mathbf{w}, b)$ from Expression 5, we obtain:

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2} \min_{x \in D} |\mathbf{w}^T x + b| \text{ s.t. } \forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0$$

Since the scale of the hyperplane is always invariant, we can arbitrarily fix the scale of $\mathbf{w}$ and b. Therefore, we set the scale such that the following expression holds:

$$\min_{x \in D} |\mathbf{w}^T x + b| = 1$$

Substituting this expression into the previously defined formula, we obtain:

$$\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2} \times 1 = \min_{\mathbf{w},b} \|\mathbf{w}\|_2 = \min_{\mathbf{w},b} \mathbf{w}^T \mathbf{w}$$

Thus, the updated optimization problem can be formulated as:

$$\min_{\mathbf{w},b} \mathbf{w}^T \mathbf{w} \text{ s.t. } \forall i \; y_i(\mathbf{w}^T x_i + b) \geq 0 \text{ and } \min_{x \in D} |\mathbf{w}^T x + b| = 1$$

The aforementioned optimization problem can be succinctly written as:

$$\min_{\mathbf{w},b} \mathbf{w}^T \mathbf{w} \text{ s.t. } \forall i \; y_i(\mathbf{w}^T x_i + b) \geq 1$$

References