Bayes Classification and Its Approximation – K Nearest Neighbors (KNN)

August 31, 2020

Why can we use *K Nearest Neighbor* (KNN) to obtain the classification? What is the math behind KNN?

In this document, we start with the Bayes classification to describe the best classification based on the statistical point of view. We then discuss the approximation of Bayes classification, i.e. KNN and show the relationship between those two models.

1 Bayes Classification

Bayes classification refers to a set of classification algorithms, which are all based on the Bayes Theorem. In this section, we first discuss the Bayes Theorem. We will then discuss one Bayes classification, Naive Bayes classification, by using the theorem directly. Next, we discuss the Bayes error for Bayes classification. Finally, we provide the drawback of Naive Bayes classification and the hardness of applying Bayes theorem.

1.1 Probability Background for Bayes Classification

1.1.1 Conditional Probability

We first provide the definition of conditional probability. Conditional probability is the probability of one event occurring given that one or more other events are happening. If we use A to refer to event A and B to refer to event B, then the conditional probability of event B occurring given event A is happening can be written as $P(B|A)$. The conditional probability can be calculated through the following equation

$$P(B|A) = \frac{P(AB)}{P(A)}, \tag{1}$$

where $P(AB)$ means the probability of both events A and B occurring and $P(A)$ is the probability of event A occurring.

For example, 70% of your friends like chocolate and 35% like chocolate and like blueberry. Now we calculate the percentage of those who like chocolate also like blueberry. Let event A and B refer to your friends who like chocolate and blueberry, respectively. Then we have $P(B|A) = \frac{P(AB)}{P(A)} = \frac{35\%}{70\%} = 50\%$, which means that 50% of your friends who like chocolate also like blueberry.

Now, we reorganize the the equation and obtain the probability as follows.

$$P(AB) = P(A)P(B|A). \tag{2}$$

If A is a universal event, $P(B) = P(AB) = P(B|A)$ since $P(A) = 1$.

We can extend the above equation to calculate the chance of all N events occurring.

$$P(A_1A_2A_3...A_N) = P(A_1)P(A_2|A_1)P(A_3|A_1A_2)...P(A_N|A_1A_2A_3...A_{N-1}) \tag{3}$$

For example, we have three blue marbles and five red marbles in the bag and we want to calculate the chance of drawing 3 red marbles. We use A_1, A_2, and A_3 to refer to the event that the first, second, and third drawing of red marble, respectively. We have the probability of A_1 as $P(A_1) = \frac{5}{8}$. Since we have already taken out one red marble, the probability that we draw another red marble can be calculated as $P(A_2|A_1) = \frac{4}{7}$. Similarly, we have $P(A_3|A_1A_2) = \frac{3}{6}$. Therefore, based on the chain of probability, we can calculate the chance of drawing 3 red marbles as $P(A_1A_2A_3) = P(A_1)P(A_2|A_1)P(A_3|A_1A_2) = \frac{5}{8} \cdot \frac{4}{7} \cdot \frac{3}{6} = \frac{5}{28}$.

1.1.2 Bayes Theorem

From Eq. (2) , we first provide the theorem of total probability.

THEOREM 1 (THEOREM OF TOTAL PROBABILITY). Suppose A is a universal event and partitioned into N disjoint events $A = A_1 \cup A_2 \cup \cdots \cup A_N$, then the possibility of event B can be calculated as $P(B) = \sum_{i=1}^{N} P(B|A_i)P(A_i)$.

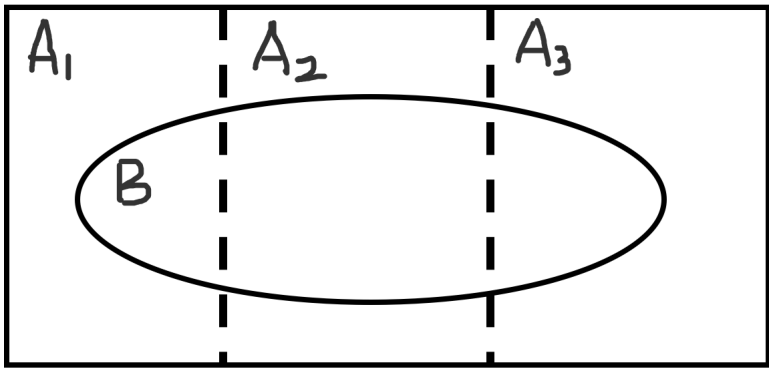

Figure 1: Illustration of Theorem 1

Figure 1 shows an illustration of theorem 1 in which A is a universal event and partitioned into 3 disjoint events. Correspondingly, Event B is partitioned into 3 parts and the total probability of event B can be calculated as $P(B) = P(B|A_1)P(A_1) + P(B|A_2)P(A_2) + P(B|A_3)P(A_3)$.

From Eq. (1) and Eq. (2) , we have the equation, i.e. $P(AB) = P(A)P(B|A) = P(B)P(A|B)$. Therefore, the Bayes theorem can be described in Theorem 2.

THEOREM 2 (BAYES THEOREM). Given the probability of $P(A)$, $P(B)$, and $P(B|A)$, the conditional probability $P(A|B)$ is calculated as $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$.

Although Bayes theorem is quite straightforward, it is very useful since we always utilize the theorem in the scenario that it is very easy to obtain $P(B|A)$ but it is very hard to get $P(A|B)$ directly.

We use COVID-19 PCR test as an example. We know every medical test can have false positives and false negatives. Assume that the correct rate of PCR test is 99% and the false positive rate is as low as 5%. Assume the COVID-19 disease incidence is 0.001, which means that it will have 1 person get COVID-19 among 1,000 persons in average. You may think that since PCR test has 99% correctness, you will get COVID-19 if your test result is positive. Is it true? Let us see.

First of all, what does 99% mean to us? Suppose we use A to refer to an event that a person gets COVID-19 and B to refer to the event that person gets positive result at the PCR test. Therefore, 99% means that the person gets positive results if that person really obtains COVID-19. That is to say, $P(B|A) = 99\%$.

Secondly, we know that $P(A) = 0.001$. Through Theorem 1, we can also obtain $P(B) = P(B|A)P(A) + P(B|\overline{A})P(\bar{A})$ where $\overline{A}$ refers to the event that a person does not get COVID-19. It is easy to know that $P(\overline{A}) = 1 - P(A) = 0.999$. Since we have 5% as the false positive rate, we obtain $P(B|\overline{A}) = 5\%$. Therefore, $P(B) = 0.99 \times 0.001 + 0.05 \times 0.999 = 0.05094$.

Thirdly, we obtain the probability that the person gets COVID-19 if her/his test shows positive result based on the Bayes theorem, which is $P(A|B) = \frac{P(A)P(B|A)}{P(B)} = \frac{0.001 \times 0.99}{0.05094} = 1.9\%$. This number is quite low (only 1.9%) and if we just based on the test results, we may get a lot of misdiagnosis.

Now, you may ask how to increase the diagnosis correction. The answer is also inside the Bayes theorem. If we can increase $P(A)$, then we can get larger $P(A|B)$. Therefore, we need the doctor to apply their experience on the diseases. The other way, which we use quite often and also very efficient, is that we test all positive cases again. Both ways can increase $P(A)$.

1.2 Naive Bayes Classifier

The Naive Bayes classifier is a very simple classification algorithm based on the Bayes theorem. The key idea behind the classifier is that for a given unclassified data, we put it into the class which has the highest probability based on the attribute values. The algorithm can be described as follows:

Algorithm 1 NaiveBayes(D, Y, X)
/* Input:
* D: available data point set with assigned label.
* Y: available label set $\{y_1, y_2, ..., y_k\}$
* X: a data point with m attribute value $\{a_1, a_2, ..., a_m\}$ needed to be classified.
* Naive Bayes assumes that all attributes are independent to each other.
* Output:
* one label $y_l \in Y$ for data X
*/
1: Calculate the following probability through the dataset D:
$P(y_1), P(y_2), ..., P(y_k)$
2: Calculate the following conditional probability through the dataset D:
$P(a_1|y_1), P(a_2|y_1), ...P(a_m|y_1); P(a_1|y_2), P(a_2|y_2), ...P(a_m|y_2); ...; P(a_1|y_k), P(a_2|y_k), ...P(a_m|y_k)$
3: Calculate the following conditional probability based on the Bayes theorem:
$P(y_i|X) = P(y_i|\{a_1, a_2, ..., a_m\}) = \frac{P(\{a_1,a_2,...,a_m\}|y_i)P(y_i)}{P(X)}$ for all $1 \leq i \leq k$
$P(y_i|X)P(X) = P(\{a_1, a_2, ..., a_m\}|y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P(a_j|y_i)$ for all $1 \leq i \leq k$
4: Classify X to y_l if $P(y_l|X)P(X) = \max_{1\leq i\leq k} P(y_i|X)P(X)$

Note that the probability calculation in the Steps 1 and 2 can be approximately obtained through the frequency in the dataset D. The second line in Step 3 is based on the Bayes theorem and the last line in Step 3 is true since all attributes are independent. In Steps 3 and 4, we do not need to calculate $P(X)$.

1.2.1 Example: Real/Fake Account Classification

In this example, we choose 10,000 twitter accounts to form the training set, in which 8900 with label 1 (real account) and the other 1100 with label 0 (fake account). Let label set $\{0, 1\}$ represent the possible labels where $y = 0$ means the fake account and $y = 1$ means the real account. Therefore, $P(y = 0) = 1100/10000 = 11\%$ and $P(y = 1) = 8900/10000 = 89\%$.

Associate with each data, we select three attributes

-active ratio a_1: the number of days a user login over the total registered days. a_1 has three ranges, i.e. $\{a_1 \leq 0.1$, $0.1 < a_1 \leq 0.3$, and $a_1 > 0.3\}$.

-average daily friends a_2: the number of daily friends over the total registered days. a_2 has three ranges, i.e. $\{a_2 \leq 1$, $1 < a_2 \leq 10$, and $a_2 > 10\}$.

-real image associated with the account a_3. a_3 has two discrete values, i.e. $\{a_3 = 0$ (no or fake image); $a_3 = 1$ (real image)$\}$.

The probability needed for classification can be obtained through the training dataset shown in the following table.

We can calculate the following probability as follows:

$P(a_1 \leq 0.1|y = 0) = 8000/10000 = 0.8$, $P(0.1 < a_1 \leq 0.3|y = 0) = 1000/10000 = 0.1$, $P(a_1 > 0.3|y = 0) = 1000/10000 = 0.1$;

Table 1: Frequency obtained through the training dataset

freq.	$a_1 \leq 0.1$	$0.1 < a_1 \leq 0.3$	$a_1 > 0.3$	$a_2 \leq 1$	$1 < a_2 \leq 10$	$a_2 > 10$	$a_3 = 0$	$a_3 = 1$
y=0	8000	1000	1000	7000	2000	1000	9000	1000
y=1	3000	5000	2000	1000	7000	2000	2000	8000

$P(a_1 \leq 0.1|y = 1) = 3000/10000 = 0.3$, $P(0.1 < a_1 \leq 0.3|y = 1) = 5000/10000 = 0.5$, $P(a_1 > 0.3|y = 1) = 2000/10000 = 0.2$;

$P(a_2 \leq 1|y = 0) = 7000/10000 = 0.7$, $P(1 < a_2 \leq 10|y = 0) = 2000/10000 = 0.2$, $P(a_2 > 10|y = 0) = 1000/10000 = 0.1$;

$P(a_2 \leq 1|y = 1) = 1000/10000 = 0.1$, $P(1 < a_2 \leq 10|y = 1) = 7000/10000 = 0.7$, $P(a_2 > 10|y = 1) = 2000/10000 = 0.2$;

$P(a_3 = 0|y = 0) = 9000/10000 = 0.9$, $P(a_3 = 1|y = 0) = 1000/10000 = 0.1$;

$P(a_3 = 0|y = 1) = 2000/10000 = 0.2$, $P(a_3 = 1|y = 1) = 8000/10000 = 0.8$.

Now we have the value we want and we can apply the above value to classify the new data $X = \{a_1 = 0.2, a_2 = 3, a_3 = 0\}$.

Based on step 3 in the algorithm, we calculate $P(y = 0|X)P(X)$ as follows.

$P(y = 0|X)P(X) = P(\{a_1, a_2, a_3\}|y = 0)P(y = 0)$

$= P(y = 0)P(0.1 < a_1 \leq 0.3|y = 0)P(1 < a_2 \leq 10|y = 0)P(a_3 = 0|y = 0)$

$= 0.11 \times 0.1 \times 0.2 \times 0.9 = 0.00198$.

Similarly, we can calculate $P(y = 1|X)P(X)$ as follows.

$P(y = 1|X)P(X) = P(\{a_1, a_2, a_3\}|y = 1)P(y = 1)$

$= P(y = 1)P(0.1 < a_1 \leq 0.3|y = 1)P(1 < a_2 \leq 10|y = 1)P(a_3 = 0|y = 1)$

$= 0.89 \times 0.5 \times 0.7 \times 0.2 = 0.0623$.

Since $0.0623 > 0.00198$, this account is a real account based on Naive Bayes Classifier. Although the user does not provide the real image for the account, the Naive Bayes classifier still prefer to classify this account as a real account. This example demonstrates that if the dataset contains many attributes, the Naive Bayes can exclude the noise coming from a single attribute.

1.3 Bayes Error

To make our discussion simple, we first discuss the Bayes error for only two classes (0 or 1). As for data point X, we define a risk as the probability that we assign a sample to a wrong class. The risk R of the Bayes classifier for two classes can be determined as $R(X) = min\{P(X|y = 1)P(y = 1), P(X| = 0)P(y = 0)\}$. If the region is continuous, we can obtain the expectation of the risk as

$$E[R(X)] = \int_X R(X)p(X)dX = p(y = 0)\int_{L_1} p_0(x)dx + p(y = 1)\int_{L_0} p_1(x)dx, \tag{4}$$

where L_1 is the region where we assign instance to class 1, p_0 is the probability density function for random variable $x = 0$, L_0 is the region where we assign instance to class 0, and p_1 is the probability density function for random variable $x = 1$.

In general, the total Bayes error can be calculated as

$$1 - E(\max_j P_r(y = j|X)), \tag{5}$$

where $E(\cdot)$ is the expectation function and $P_r(\cdot)$ is the probability. This is the error for every Bayes theorem based classifiers and we cannot avoid it. For example, for the continuous dataset, Bayes error is around 0.1304, which is larger than 0, because we have some overlap among the distribution of different classes.

1.4 Problems with Bayes Classification

How do we deal with $P(a|y) = 0$? If the size of the dataset is small, some attribute values may have no data point in the dataset. However, if this happens, Naive Bayes may not be trained for this case which will lead to lower the quality of classification. In order to solve this problem, we first need to enlarge the size of the training dataset. We then add 1 to those attribute values and therefore, we avoid $P(a|y) = 0$ but instead $P(a|y) = \varepsilon > 0$ where ε is very small.

How do we deal with the attributes if those attributes are not independent to each other? We can use a graph to show the relationship among attributes and apply Eq. (3) to calculate the probability at the step 2 in the algorithm. Therefore, we can easily extend Naive Bayes classification to Bayesian Network classification.

2 K Nearest Neighbors (KNN)

Here, we introduce K Nearest Neighbors (KNN) algorithm, which is simple yet surprisingly efficient algorithm. In this subsection, we discuss the process of KNN and show KNN is approximation of Bayes classifier.

2.1 Process of KNN

The process of KNN is quite straightforward.

Step 1. For any unclassified data, find the closest K classified data.

Step 2. Among those K data points, take the majority label among those labels and put that label to the unclassified data.

Through this process, we do not need to count the frequency of each label and calculate the conditional probability.

As for KNN, we have solve two key things. First, we have to define the distance so that we can find the closest data points. We have two common distance functions: Euclidian distance and Manhattan distance. Suppose there are two points $X(x_1, x_2, \cdots, x_n)$ and $Y(y_1, y_2, \cdots, y_n)$, the distance can be calculated as follows.

Euclidian distance: $d(X, Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$; and

Manhattan distance: $d(X, Y) = \sum_{i=1}^{n}|x_i - y_i|$.

Second, there is a key parameter, K, needed to be determined before we classify the data. An example will be shown here to demonstrate that different choice of K will have different classification. We will discuss how to select the proper K in another document with the introduction of common tool, i.e. cross-validation.

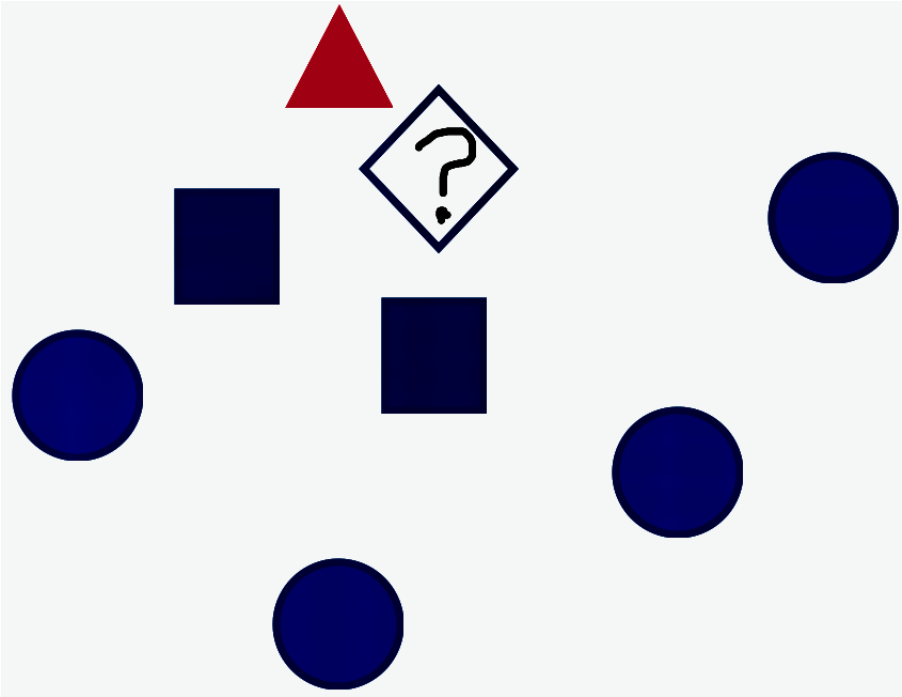

Figure 2: KNN Example

Figure 2 shows an example to demonstrate the impact of K. We use different shapes to represent different labels of the training data. Round the unclassified shape (i.e. the shape with question mark), three are three types of shapes: triangle, circle, and square. If we choose $K = 1$, the closest shape to the unclassified one will be triangle and therefore, we classify the data with the triangle label. If we choose $K = 3$, the unclassified one will have a square label due to 2 squares and 1 triangle near the unclassified data. However, if we choose $K = 7$, the unclassified one will have a circle label since the majority label is circle.

2.2 KNN - Bayes Classifier Approximation

In this subsection, we discuss the computing probability for KNN. Although KNN does not apply the whole training dataset to determine the class of unclassified data, it still classify the data based on the Bayes theorem on a subset

of the data.

Based on the Bayes theorem, the computation of KNN should include the conditional probability of the data d given the class ($P(d|y)$) , the priority probability of each class ($P(y)$), and the marginal probability of the data ($P(d)$). These properties would be computed for some small region around the sample and the size of the region will be dependent on the distribution of the test sample. Let V be the volume of the m dimensional ball around x containing the K nearest neighbors for x (where m is the number of attributes). We list all parameters in the following table for our probability computation.

Parameter	Definition
x	new data point to classify
y	class label (e.g. $y = 1$ means class 1)
V	selected ball
T	probability that a random point is in V
K	number of nearest neighbors
N	number of nearest neighbors
N_1	number of data samples from class 1
K_1	number of data samples from class 1 in K

Then, we can write the probability computing as

1) training data distribution in the small region V: $P(d) \cdot V = T = \frac{K}{N}$, where we can obtain $P(d) = \frac{K}{NV}$;

2) conditional probability: $P(d|y = 1) = \frac{K_1}{N_1 V}$; and

3) priority probability: $P(y = 1) = \frac{N_1}{N}$.

Thus, using the Bayes theorem, we obtain

$$P(y = 1|x) = \frac{P(x|y = 1)P(y = 1)}{P(x)} \cong \frac{P(d|y = 1)P(y = 1)}{P(d)} = \frac{\left(\frac{K_1}{N_1 V}\right)\left(\frac{N_1}{N}\right)}{\left(\frac{K}{NV}\right)} = \frac{K_1}{K}.$$

Similarly, the conditional probability can be obtained the conditional probability for class j through the following equation.

$$P(y = j|x) = \frac{K_j}{K}.$$

According the Bayes classifier rule, we classifies the data x to class j if and only if $P(y = j|x) = \max_{1 \le l \le M} P(y = l|x) = \max_{i \le l \le M} K_l$, where M is the total number of classes. Through this calculation example, KNN is the approximation approach of Bayes classifier.