Bayesian Concept Learning

Introduction

- **Principles of probability for classification** are **an important area of machine learning algorithms**.
- In **our practical life**, our **decisions are affected by our prior knowledge or belief about an event**.
- Thus, **an event that is otherwise very unlikely to occur may be considered by us seriously to occur** in certain situations **if we know that in the past, the event had certainly occurred when other events were observed**.
- The **same concept is applied in machine learning using Bayes' theorem.**
- **18th century mathematician Thomas Bayes** developed the foundational mathematical principles, known as **Bayesian methods**.
- **These methods describe the probability of events**, and **how probabilities should be revised when there is additional information available**.
- **Bayesian learning algorithms**, like the naive Bayes classifier, are **highly practical approaches** to certain types of learning problems **as they can calculate explicit probabilities for hypotheses**.
- **In many cases, they are equally competitive or even outperform the other learning algorithms**, such as **decision tree and neural network algorithms**.

Bayesian classifiers

- They use a **simple idea** that **the training data are utilized to calculate an observed probability of each class based on feature values**.
- **When the same classifier is used later for unclassified data**, **it uses the observed probabilities to predict the most likely class for the new features**.
- The application of the observations from the training data can also be thought of as **applying our prior knowledge or prior belief to the probability of an outcome, so that it has higher probability of meeting the actual or real-life outcome**.
- **This simple concept is used in for training a machine in machine learning terms**.
- **Bayesian classifiers utilize all available parameters to subtly change the predictions**, while many other algorithms tend to ignore the features that have weak effects.
- **Bayesian classifiers assume** that **even if few individual parameters have small effect on the outcome, the collective effect of those parameters could be quite large**.
- **For such learning tasks, the naive Bayes classifier is the most effective**.

Real-life uses of Bayesian classifiers

- Text-based classification such as spam or junk mail filtering, author identification, or topic categorization
- Medical diagnosis such as given the presence of a set of observed symptoms during a disease, identifying the probability of new patients having the disease
- Network security such as detecting illegal intrusion or anomaly in computer networks
- …….

Summary - Features of Bayesian learning methods

1. Prior knowledge of the candidate hypothesis is combined with the observed data for arriving at the final probability of a hypothesis. So, two important components are the prior probability of each candidate hypothesis and the probability distribution over the observed data set for each possible hypothesis.
2. The Bayesian approach to learning is more flexible than the other approaches because each observed training pattern can influence the outcome of the hypothesis by increasing or decreasing the estimated probability about the hypothesis, whereas most of the other algorithms tend to eliminate a hypothesis if that is inconsistent with the single training pattern.
3. Bayesian methods can perform better than the other methods while validating the hypotheses that make probabilistic predictions. For example, when starting a new software project, on the basis of the demographics of the project, we can predict the probability of encountering challenges during execution of the project.
4. Through the easy approach of Bayesian methods, it is possible to classify new instances by combining the predictions of multiple hypotheses, weighted by their respective probabilities.
5. IIn some cases, when Bayesian methods cannot compute the outcome deterministically, they can be used to create a standard for the optimal decision against which the performance of other methods can be measured.

Challenges in Bayesian Learning

- The success of the Bayesian method largely depends on the availability of initial knowledge about the probabilities of the hypothesis set.
- If these probabilities are not known to us in advance, we have to use some background knowledge, previous data or assumptions about the data set, and the related probability distribution functions to apply this method.
- Involves high computational cost to arrive at the optimal Bayes hypothesis.

Concept learning

Let us take an example of how a child starts to learn meaning of new words, e.g. ‘ball’.

The **child is provided with positive examples of ‘objects’ which are ‘ball’.**

At first, the child may be confused with many different colours, shapes and sizes of the balls and may also get confused with some objects which look similar to ball, like a balloon or a globe.

The child’s parent continuously feeds her positive examples like ‘that is a ball’, ‘this is a green ball’, ‘bring me that small ball’, etc.

Seldom there are negative examples used for such concept teaching, like ‘this is a non-ball’, but the parent may clear the confusion of the child when it points to a balloon and says it is a ball by saying ‘that is not a ball’.

It is observed that the learning is most influenced through positive examples rather than through negative examples

The expectation is that the child will be able to identify the object ‘ball’ from a wide variety of objects and different types of balls kept together once the concept of a ball is clear to her.

This can be extended to explain how we can expect machines to learn through the feeding of positive examples, which forms the basis for concept learning.

Concept Learning using Baye's Rule

Let us define a concept set C and a corresponding function $f(k)$.

We also define $f(k) = 1$, when k is within the set C and $f(k) = 0$ otherwise.

Our aim is to learn the indicator function f that defines which elements are within the set C.

So, by using the function f, we will be able to classify the element either inside or outside our concept set.

In Bayes' theorem, we will learn how to use standard probability calculus to determine the uncertainty about the function f, and we can validate the classification by feeding positive examples.

Baye's rule - Recap

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

where *A* and *B* are conditionally related events and *p*(*A*|*B*) denotes the probability of event *A* occurring when event *B* has already occurred.

Prior and Posterior Probabilities

Assume that we have a training data set *D* where we have noted some observed data. Our task is to determine the best hypothesis in space *H* by using the knowledge of *D*.

The prior knowledge or belief about the probabilities of various hypotheses in *H* is called Prior in context of Bayes' theorem.

For example, if we have to determine whether a particular type of tumour is malignant for a patient, the prior knowledge of such tumours becoming malignant can be used to validate our current hypothesis and is a prior probability or simply called Prior.

$P(h)$ is the initial probability of a hypothesis '*h*' that the patient has a malignant tumour based only on the malignancy test, without considering the prior knowledge of the correctness of the test process or the so-called training data.

Similarly, $P(T)$ is the prior probability that the training data will be observed or, in this case, the probability of positive malignancy test results.

Let $P(T|h)$ be the probability of observing data *T* in a space where '*h*' holds true, which means the probability of the test results showing a positive value when the tumour is actually malignant.

Prior and Posterior Probabilities

The probability that a particular hypothesis holds for a data set based on the Prior is called the posterior probability or simply Posterior.

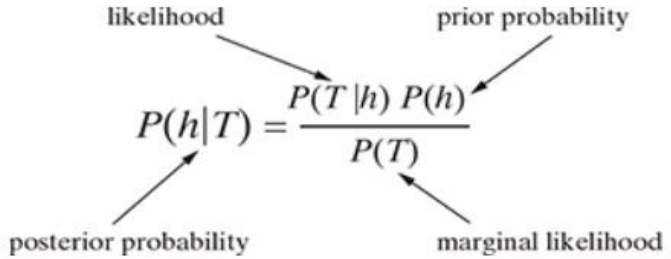
In the above example, the probability of the hypothesis that the patient has a malignant tumour considering the Prior of correctness of the malignancy test is a posterior probability.

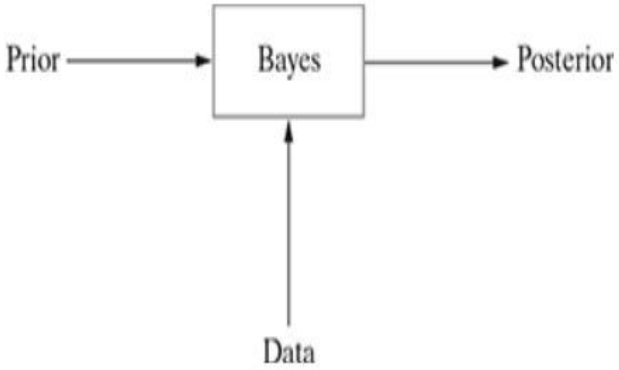
$P(h|T)$, which means whether the hypothesis holds true given the observed training data T. This is called the posterior probability or simply Posterior in machine learning language.

So, the prior probability $P(h)$, which represents the probability of the hypothesis independent of the training data (Prior), now gets refined with the introduction of influence of the training data as $P(h|T)$.

Prior, Posterior & Baye's Theorem

Bayes' theorem combines the prior and posterior probabilities together.





$P(h|T)$ increases as $P(h)$ and $P(T|h)$ increases and also as $P(T)$ decreases.

When there is more probability that T can occur independently of h, then it is less probable that h can get support from T in its occurrence.

Naive Baye's Classifier

From Bayes rule, we observed that

1. A prior probability of hypothesis *h* or *P*(*h*) is the probability of an event or hypothesis before the evidence is observed.

2. A posterior probability of *h* or *P*(*h*|*D*) is the probability of an event after the evidence is observed within the population *D*.

$$\textbf{Posterior probability} = \frac{\textbf{(Prior probability} \times \textbf{Conditional Probability)}}{\textbf{Evidence}}$$

Naive Baye's Classifier

"What is the probability that a particular object belongs to class *i* given its observed feature values?"

Naive Baye's classifier is a probabilistic classifier.

We take a learning task where each instance *x* has some attributes and the target function ($f(x)$) can take any value from the finite set of classification values *C*.

We also have a set of training examples for target function, and the set of attributes $\{a1, a2, ..., an\}$ for the new instance are known to us.

Our task is to predict the classification of the new instance.

Naive Baye's Classifier

According to Bayes' theorem, the classification of the new instance is performed by assigning the most probable target classification C_{MAP} on the basis of the attribute values of the new instance $\{a_1, a_2, \ldots, a_n\}$. So,

$$C_{\text{MAP}} = \underset{c_i \in C}{\text{argmax}} \sum_{h_i \in H} P(c_i | a_1, a_2, \ldots, a_n)$$

which can be rewritten using Bayes' theorem as

$$C_{\text{MAP}} = \underset{c_i \in C}{\text{argmax}} \sum_{h_i \in H} \frac{P(a_1, a_2, \ldots, a_n | c_i) P(c_i)}{P(a_1, \ldots a_n)}$$

As the combined probability of the attributes defining the new instance fully is always 1, the denominator becomes 1

$$C_{\text{MAP}} = \underset{c_i \in C}{\text{argmax}} \sum_{h_i \in H} P(a_1, a_2, \ldots, a_n | c_i) P(c_i)$$

Naive Baye's Classifier

Naïve Bayes classifier makes a simple assumption that the attribute values are conditionally independent of each other for the target value.

For a target value of an instance, the probability of observing the combination $a_1, a_2, ..., a_n$ is the product of probabilities of individual attributes $P(a_i|c_j)$.

$$P(a_1, a_2, ..., a_n|c_j) = \prod_i P(a_i|c_j)$$

The approach for the Naïve Bayes classifier is:

$$C_{NB} = \underset{c_i \in C}{\operatorname{argmax}} \sum_{h_i \in H} P(c_i) \prod_i P(a_i|c_j)$$

Naive Baye's Classifier

Strengths	Weakness
Simple and fast in calculation but yet effective in result	The basis assumption of equal importance and independence often does not hold true
In situations where there are noisy and missing data, it performs well	If the target dataset contains large numbers of numeric features, then the reliability of the outcome becomes limited
Works equally well when smaller number of data is present for training as well as very large number of training data is available	Though the predicted classes have a high reliability, estimated probabilities have relatively lower reliability
Easy and straightforward way to obtain the estimated probability of a prediction	

Example

Predict the outcome of a football world cup match on the basis of the past performance data of the playing teams. We have training data available for actual match outcome.

4 parameters are considered – Weather Condition (Rainy, Overcast, or Sunny), how many matches won were by this team out of the last three matches (one match, two matches, or three matches), Humidity Condition (High or Normal), and whether they won the toss (True or False).

Using Naïve Bayesian,

1. Classify the conditions when this team wins
2. Predict the probability of this team winning a particular match when Weather Conditions = Rainy, they won two of the last three matches, Humidity = Normal and they won the toss in the particular match.

Weather Condition	Wins in last 3 matches	Humidity	Win toss	Won match?
Rainy	3 wins	High	FALSE	No
Rainy	3 wins	High	TRUE	No
OverCast	3 wins	High	FALSE	Yes
Sunny	2 wins	High	FALSE	Yes
Sunny	1 win	Normal	FALSE	Yes
Sunny	1 win	Normal	TRUE	No
OverCast	1 win	Normal	TRUE	Yes
Rainy	2 wins	High	FALSE	No
Rainy	1 win	Normal	FALSE	Yes
Sunny	2 wins	Normal	FALSE	Yes
Rainy	2 wins	Normal	TRUE	Yes
OverCast	2 wins	High	TRUE	Yes
OverCast	3 wins	Normal	FALSE	Yes
Sunny	2 wins	High	TRUE	No

Solution

Step 1: First construct a frequency table. A frequency table is drawn for each attribute against the target outcome.

From the given table, various attributes are (1) Weather Condition, (2) How many matches won by this team in last three matches, (3) Humidity Condition, and (4) whether they won the toss and the target outcome is will they win the match or not?

The posterior probability can be easily derived by constructing a frequency table for each attribute against the target.

For example, frequency of Weather Condition variable with values ‘Sunny’ when the target value Won match is ‘Yes’, is, 3/(3+4+2) = 3/9.

	Won Match			Won Match	
Weather condition	Yes	No	**Humidity**	Yes	No
Sunny	3	2	High	3	4
OverCast	4	0	Normal	6	1
Rainy	2	3			
Total	9	5	Total	9	5

	Won Match			Won Match	
Wins in last 3 matches	Yes	No	Win toss	Yes	No
3 wins	2	2	FALSE	6	2
1 win	4	2	TRUE	3	3
2 wins	3	1			
Total	9	5	Total	9	5

Frequency Table

Solution

Step 2: Identify the cumulative probability for ‘Won match = Yes’ and the probability for ‘Won match = No’ on the basis of all the attributes. Otherwise, simply multiply probabilities of all favourable conditions to derive ‘YES’ condition. Multiply probabilities of all non-favourable conditions to derive ‘No’ condition.

To predict whether the team will win for given weather conditions (*a*1) = Rainy, Wins in last three matches (*a*2) = 2 wins, Humidity (*a*3) = Normal and Win toss (*a*4) = True, we need to choose ‘Yes’ from the above table for the given conditions.

$$P(\text{Win match}|a_1 \cap a_2 \cap a_3 \cap a_4) = \frac{P(a_1 \cap a_2 \cap a_3 \cap a_4|\text{Win match})P(\text{Win match})}{P(a_1 \cap a_2 \cap a_3 \cap a_4)}$$

Solution

Naïve Bayes classifier assumes independence among events.

$$P(\text{Win match}|a_1 \cap a_2 \cap a_3 \cap a_4)$$

$$= \frac{P(a_1|\text{Win match})P(a_2|\text{Win match})P(a_3|\text{Win match})P(a_4|\text{Win match})P(\text{Win match})}{P(a_1)P(a_2)P(a_3)P(a_4)}$$

$$= 2/9*4/9*6/9*9/14$$

$$= 0.014109347$$

This should be compared with

$$P(!\text{Win match}|a_1 \cap a_2 \cap a_3 \cap a_4)$$

$$= \frac{P(a_1|!\text{Win match})P(a_2|!\text{Win match})P(a_3|!\text{Win match})P(a_4|!\text{Win match})P(!\text{Win match})}{P(a_1)P(a_2)P(a_3)P(a_4)}$$

$$= 3/5*2/5*1/5*5/14$$

$$= 0.010285714$$

Solution

Step 3: Calculate probability through normalization by applying the below formula.

$$P(\text{Yes}) = \frac{P(\text{Yes})}{P(\text{Yes}) + P(\text{No})}$$

$$P(\text{No}) = \frac{P(\text{No})}{P(\text{Yes}) + P(\text{No})}$$

P(Yes) will give the overall probability of favourable condition in the given scenario.

P(No) will give the overall probability of non-favourable condition in the given scenario.

Solution

By normalizing the above two probabilities, we can ensure that the sum of these two probabilities is 1.

$$
\begin{aligned}
P(\textbf{Win match}) &= \frac{P(\textbf{Win match})}{P(\textbf{Win match}) + P(\textbf{!Win match})} \\
&= \frac{0.014109347}{0.014109347 + 0.010285714} \\
&= 0.578368999
\end{aligned}
$$

$$
\begin{aligned}
P(\textbf{!Win match}) &= \frac{P(\textbf{!Win match})}{P(\textbf{Win match}) + P(\textbf{!Win match})} \\
&= \frac{0.010285714}{0.014109347 + 0.010285714} \\
&= 0.421631001
\end{aligned}
$$

Conclusion

- This shows that there is 58% probability that the team will win if the above conditions become true for that particular day.
- Thus, Naïve Bayes classifier provides a simple yet powerful way to consider the influence of multiple attributes on the target outcome and refine the uncertainty of the event on the basis of the prior knowledge.
- It is able to simplify the calculation through independence assumption.

Few Popular Applications of Naïve Bayes classifier

- Text classification
- Spam filtering
- Hybrid Recommender Systems (used by e-retailors like eBay, Alibaba, Target, Flipkart, etc.)
- Online Sentiment Analysis