S21CSE471: Statistical Methods in AI **Spring 2021**

Naive Bayes Classification (04/03/2021)

Prepared by: Shubham Singh, Shivam Puri, Shivam Rai

1 Generative Classification

In Generative classification, we fit each class with a probability distribution and to assign labels to data, we find probability of that data belonging to each class and assign the label with maximum probability. Mathematically, for a test data x, class of x is given by -

$$class_x = argmax(p(y_i|x)) \; \forall y_i \in classes \tag{1}$$

where $p(y_i|x)$ is probability that given data x belongs to class y_i.

1.1 Naive Bayes Classification

The Naive Bayes Classifier uses Bayes Theorem to calculate $p(y_i|x)$.
Assumption by Naive Bayes Algorithm -

- It assumes that the features of the sample are independent.
- It also assumes that the features equally contribute to outcome.

Since, these assumptions are not true in many real world scenarios, hence the term 'Naive'.

Bayes Theorem is defined for two events A and B as,

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

Applying Bayes Theorem to calculate $p(y_i|x)$, we get

$$p(y_i|x) = \frac{p(x|y_i)p(y_i)}{p(x)} \tag{2}$$

where

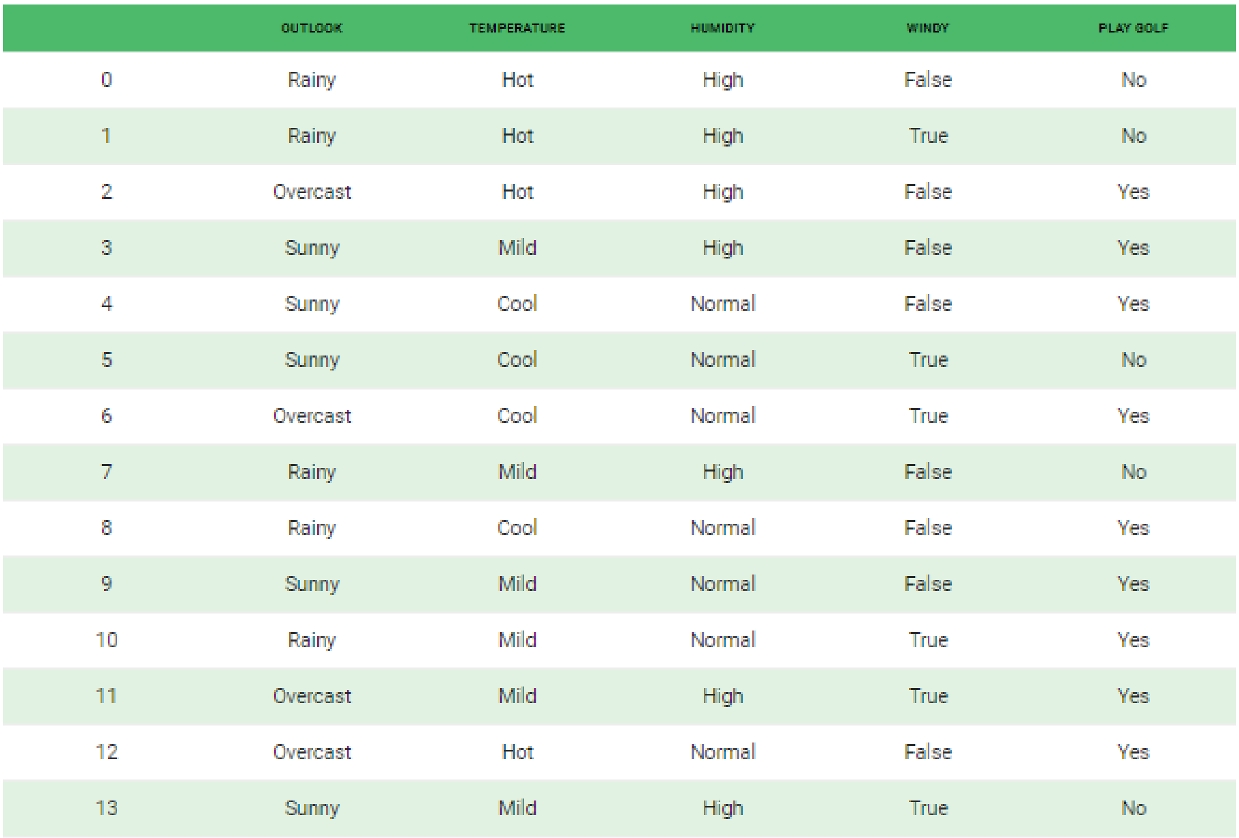
$$p(x|y_i) = p(x_1|y_i).p(x_2|y_i).p(x_3|y_i).........p(x_d|y_i) = \prod_{j=1}^{d} p(x_j|y_i) \text{ for a d-dimensional x.}$$

$$p(y_i) \text{ is probability of class } y_i, i.e., p(y_i) = \frac{\text{number of samples of class } i}{\text{total number of samples}}$$

$$p(x) = \sum_i p(x|y_i)p(y_i)$$

For calculating $p(x_i|y)$, we can try fitting a probability distribution on the given sample data for i^{th} dimension and then calculating $p(x_i|y)$.

Also, from (2), it is clear that the $p(y_i|x) \propto (x|y_i)p(y_i)$ as for all y_i, $p(x)$ is same.



	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Figure 1: Sample data

Example Let's see an example of how it works

From the data in Figure 1, the following values can be calculated.

- Probability of playing golf $= P(Play = Yes) = \frac{9}{14}$
- Probability of not playing golf $= P(Play = No) = \frac{5}{14}$
- $P(Outlook = Rainy|Play = Yes) = \frac{2}{9}$ $P(Outlook = Rainy|Play = No) = \frac{3}{5}$
- $P(Outlook = Sunny|Play = Yes) = \frac{3}{9}$ $P(Outlook = Sunny|Play = No) = \frac{2}{5}$
- $P(Outlook = Overcast|Play = Yes) = \frac{4}{9}$ $P(Outlook = Overcast|Play = No) = 0$
- $P(Temperature = Hot|Play = Yes) = \frac{2}{9}$ $P(Temperature = Hot|Play = No) = \frac{2}{5}$
- $P(Temperature = Mild|Play = Yes) = \frac{4}{9}$ $P(Temperature = Mild|Play = No) = \frac{2}{5}$
- $P(Temperature = Cool|Play = Yes) = \frac{3}{9}$ $P(Temperature = Cool|Play = No) = \frac{1}{5}$
- $P(Humidity = High|Play = Yes) = \frac{3}{9}$ $P(Humidity = High|Play = No) = \frac{4}{5}$
- $P(Humidity = Normal|Play = Yes) = \frac{6}{9}$ $P(Humidity = Normal|Play = No) = \frac{1}{5}$

- $P(Windy = True|Play = Yes) = \frac{3}{9}$ $P(Windy = True|Play = No) = \frac{3}{5}$
- $P(Windy = False|Play = Yes) = \frac{6}{9}$ $P(Windy = False|Play = No) = \frac{2}{5}$

Let the given test sample be Outlook=Sunny, Temperature=Hot, Humidity=Normal, Windy=False.

$P(Play = Yes|test) \propto P(test|Play = Yes) * P(Play = Yes)$
$\implies P(Play = Yes|test) \propto P(Outlook = Sunny|Play = Yes) * P(Temperature = Hot|Play = Yes) * P(Humidity = Normal|Play = Yes) * P(Windy = False|Play = Yes) * P(Play = Yes)$
$\implies P(Play = Yes|test) \propto \frac{3}{9} * \frac{2}{9} * \frac{6}{9} * \frac{6}{9} * \frac{9}{14}$
$\implies P(Play = Yes|test) \propto 0.02116$

$P(Play = No|test) \propto P(test|Play = No) * P(Play = No)$
$\implies P(Play = No|test) \propto P(Outlook = Sunny|Play = No) * P(Temperature = Hot|Play = No) * P(Humidity = Normal|Play = No) * P(Windy = False|Play = No) * P(Play = No)$
$\implies P(Play = No|test) \propto \frac{2}{5} * \frac{2}{5} * \frac{1}{5} * \frac{2}{5} * \frac{5}{14}$
$\implies P(Play = No|test) \propto 0.00457$

Since, $P(Play = Yes|test) > P(Play = No|test)$
Therefore, Output of this model for given test sample will be, Play=Yes.

Note - In the above example we can see that $P(Outlook = Overcast|Play = No) = 0$, therefore whenever we have test data where $Outlook = Overcast$, the $P(Play = No|test)$ will be zero, without even considering other factors. So, to handle such situation we add some value to numerator of all probability calculations.

1.2 Types of Naive Bayes Classification

1.2.1 Gaussian Naive Bayes Classification

This is used when features of given data set are in continuous form and may follow normal/Gaussian Distribution. Also since, a lot of real word data tends to be in normal distribution, this type of classification can be applied to many data-sets.

For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance in each class .Let μ_k be the mean of the values in x associated with class C_k, let σ_k^2 be the Bessel corrected variance of the values in x associated with class Ck. Suppose we have collected some observation value v . Then, the probability density of v given a class C_k , $p(x = v \mid C_k)$ can be computed by plugging v into the equation for a normal distribution parameterized by μ_k and σ_k^2. That is,

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

1.2.2 Bernoulli Naive Bayes Classification

This classification can be applied when data-set has features in boolean form.

For example the multinomial model, this model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies. If x_i is a boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class

$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)}$ where p_{ki} is the probability of class C_k generating the term x_i. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

The multinomial naïve Bayes classifier becomes a linear classifier when expressed in log-space:-

$$\begin{aligned}\log p(C_k \mid \mathbf{x}) & \log \left(p(C_k) \prod_{i=1}^{n} {p_{k_i}}^{x_i} \right) \\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{k_i} \\ &= b + \mathbf{w}_k^\top \mathbf{x}\end{aligned}$$

1.2.3 Multinomial Naive Bayes Classification

This classification is best suited to data that has feature values in discrete count form. The most common usage of this type is in document or text classification. For text classification, the data is often represented in form of frequency (or some operation on frequency) of words.

For example With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial (p 1 , … , p n) $(p_1, \ldots, p_n)$ where p_i is the probability that event i occurs (or K such multinomials in the multiclass case). A feature vector x = (x 1 , … , x n) $\mathbf{x} = (x_1, \ldots, x_n)$ $\mathbf{x}$=$(x_1, \ldots, x_n)$ is then a histogram, with x_i counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram x is given by

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

Advantages of Naive Bayes Classification-

- These algorithms are fast and easy to implement.
- With right probability distribution, it can be used for most of the data.
- If the assumptions hold true, the amount of training data required is less compared to other classifiers.

Disadvantages of Naive Bayes Classification-

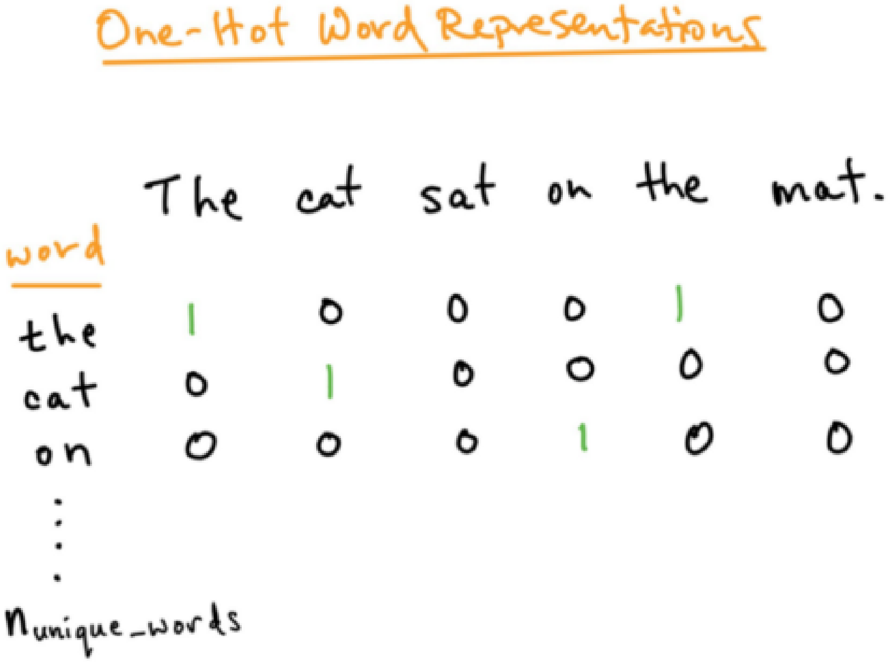
- The assumption that features are independent is not seen in a real world data. This can probably be handled by by proper pre-processing.
- Another assumption that each feature equally contributes to the outcome which may not be true.

2 Text Representations

For training a model on text data, we cannot directly feed the data to models. Therefore, data needs to be converted to in some form that can be processed by models such as vectors. In most of the cases involving textual data, the model generally has a vocabulary on which it is trained. Some of the representations are -

2.1 One-Hot Encoding

It is a 2-D array representation where one of the axis is the vocabulary and the other axis is the sentence. The advantage of this representation is that it easy to implement. But a big disadvantage is that, this representation has different shapes of array which depends on the size of sentence. This can be a big drawback as most of the models take data with a uniform shape. Another disadvantage is that this representation can take up a lot of memory.



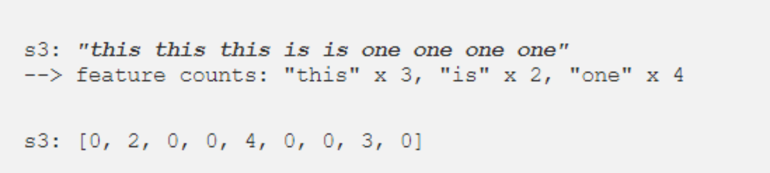
One-Hot Word Representations

word	The	cat	sat	on	the	mat.
the	1	0	0	0	1	0
cat	0	1	0	0	0	0
on	0	0	0	1	0	0
⋮						
n_{unique_words}						

Figure 2: One-Hot Encoding Example

2.2 Count Vectorizer

This representation creates a vector of frequencies of each word in vocabulary. This representation does not take into account the order of occurrence of words. The advantage of this representation is that the



```
s3: "this this this is is one one one one"
--> feature counts: "this" x 3, "is" x 2, "one" x 4

s3: [0, 2, 0, 0, 4, 0, 0, 3, 0]
```

Figure 3: Count Vector Represent

size of vector is constant and therefore, can be used for many models. The disadvantages are there is no importance to order of occurrence of words. Also, proper pre-processing of text is required so that stop words do not occur as frequency is a deciding factor.

2.3 TF-IDF

TF-IDF stands for term frequency - inverse document frequency. TF-IDF not only takes into account the frequency of word in a document but also how many documents it appears in.

$$TF - IDF = TF(w,d) * IDF(w)$$

$$TF(w,d) = \text{ frequency of word w in document d}$$

$$IDF(w) = \log(\frac{N}{df(w)})$$

where N = Total number of documents and df(w) is number of documents in which w occurs.
Advantages of TF-IDF are -

- Easy to build.
- The representation considers Inverse document frequency as well which increases the chance of a word that occurs less number of times to be an important word.

Disdvantages are -

- The order of occurrence is still not considered.
- To calculate IDF, we need a large set of documents.
- To make a generalized model, we need a diverse set of documents.

2.4 N-Grams

N-Grams are a way of representing a sequence of n words. This representation helps in maintaining a relationship between the words based on their occurrence. It stores the number of times, n words have occurred together. A big advantage of this representation is that it keeps the relationship between words and the representation can be then changed to conditional probability.
Note - For all the above text representations, pre-processing of text is an important aspect. Some ways of pre-processing are -

- Stop word Removal
- Lemmatization
- Stemming
- Converting text to lower case

References

[1] Naive Bayes Classifier: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[2] Naive Bayes Classifiers: https://www.geeksforgeeks.org/naive-bayes-classifiers/

[3] An Overview for Text Representations in NLP: https://towardsdatascience.com/an-overview-for-text-representations-in-nlp-311253730af1

[4] Introduction to Text Representations for Language Processing: https://towardsdatascience.com/introduction-to-text-representations-for-language-processing-part-1-dc6e8068b8a4

[5] Types of Naive Bayes Classificatio : https://en.wikipedia.org/wiki/Naive_Bayes_classifier