Naive Bayes

Prof. G Panda, FNAE,FNASc,FIET(UK)
IIT Bhubaneswar

Outline

- Introduction to Bayesian Classification
  - Bayes Theorem
  - Naive Bayes classifier
  - Classification example
- Applications
- Building a Text classifier
- Pros and Cons

Classification

In machine learning and statistics, classification is a **supervised learning** approach in which the computer program learns from the data input and class labels given to it and then uses this learning to classify new observation.

For example in filtering emails need classify **‘Spam’** or **‘not Spam’**

Types of Classification algorithms

1. Linear Classifiers: Naive Bayes Classifier, Logistic Regression
2. Support Vector Machines
3. Decision Trees
4. Random Forest
5. Neural Networks
6. Nearest Neighbor

Introduction to Bayesian classification

- **What is it ?**
  - Statistical method for classification.
  - Supervised Learning Method.
  - Assumes an underlying probabilistic model, the Bayes theorem.
  - Can solve problems involving both categorical and continuous valued attributes.
  - Named after Thomas Bayes, who proposed the Bayes Theorem

What is Naive Bayes?

- It is a classification technique based on Bayes theorem with an assumption of **independence** among predictors.
- In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Why it is Naive?

For **example**, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. According to Naive Bayes,

P(Apple) = P(red and round and 3 inches) [here,P( ) indicates probability]

= P(red) P(round) P(3 inches)

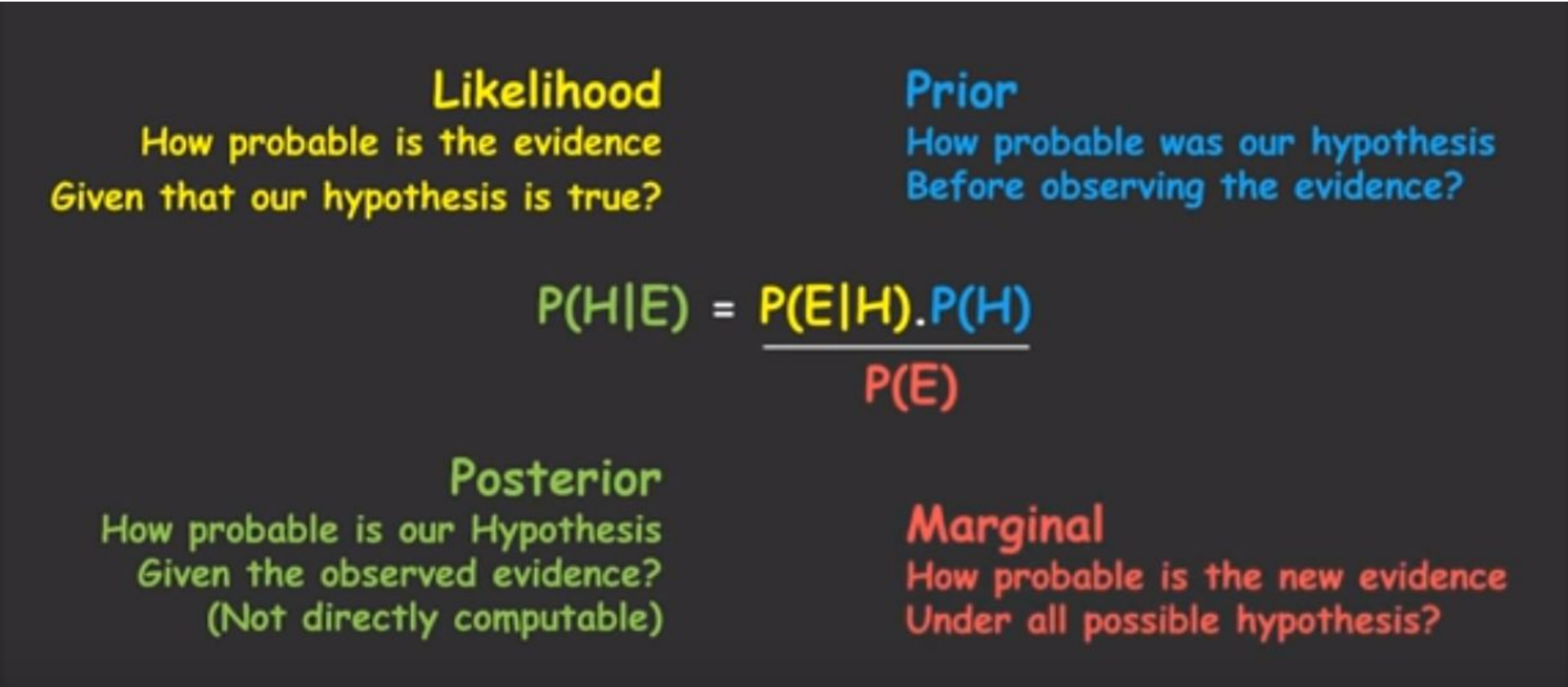
Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Bayes Theorem

Given a Hypothesis **H** and Evidence **E**, Bayes theorem states that the relation between the probability of hypothesis before getting the evidence P(H) and the probability of Hypothesis after getting the Evidence P(H | E) is,

$$P(H \mid E) = \frac{P(E \mid H) \text{ x } P(H)}{P(E)}$$

Bayes Theorem



Likelihood

How probable is the evidence
Given that our hypothesis is true?

Prior

How probable was our hypothesis
Before observing the evidence?

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

Posterior

How probable is our Hypothesis
Given the observed evidence?
(Not directly computable)

Marginal

How probable is the new evidence
Under all possible hypothesis?

Example

Cards Game:

P(King | Face) = ?

P(King) = 4/52 = 1/13

P(Face | King) = 1

P(Face) = 12/52 = 3/13

$$P(\text{King} \mid \text{Face}) = \frac{P(\text{Face} \mid \text{King}) \times P(\text{King})}{P(\text{Face})} = 1/3$$

Naive Bayes classifier

- Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes’ Theorem to predict the tag of a text (like a piece of news or a customer review).
- They are probabilistic, which means that they calculate the probability of each tag for a given text, and then output the tag with the highest one.
- The way they get these probabilities is by using Bayes’ Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature.

Maximum a Posterior(MAP)

Based on Bayes Theorem, we can compute the Maximum A Posterior (MAP) hypothesis for the data.

We are interested in the best hypothesis for some space H given observed training data D

$$
\begin{aligned}
h_{MAP} &= \underset{h \in H}{\operatorname{argmax}} \, P(h \mid D) \\
&= \underset{h \in H}{\operatorname{argmax}} \, \frac{P(D \mid h) \text{ x } P(h)}{P(D)} \\
&= \underset{h \in H}{\operatorname{argmax}} \, P(D \mid h) \text{ x } P(h)
\end{aligned}
$$

Note: we can drop P(D) as the probability of the data is constant (and independent of the hypothesis)

Bayes Classifier

- The classification problem may be formalized using a-posterior probabilities:

  P(C|X) = probability that the sample tuple X is of class C.

- E.g. P(class=No | outlook= sunny, windy=true,…)
- **Idea:** assign to sample X the class label C such that P(C|X) is maximal

Naive Bayes classifier

Let, Data D: set of Tuples

Each Tuple X is a n-dimensional attribute vector.
X : $<x_1, x_2, x_3, x_4, \ldots\ldots\ldots x_n>$

Where x_i is the value of attribute A_i

Let, there are ‘m’ classes : $C_1, C_2, C_3, C_4, \ldots\ldots\ldots C_m$.

Then Posterior probability,

$$P(C_i \mid X) = \frac{P(X \mid C_i) \times P(C_i)}{P(X)}$$

Naive Bayes classifier(continued)

Where,

- $P(C_i \mid X)$ is the posterior probability of *class* (C_i i.e. *label*) given *predictor* (X i.e. *attributes*).
- $P(C_i)$ is the prior probability of *class*.
- $P(X \mid C_i)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(X)$ is the prior probability of *predictor*.

Bayes Classifier Predicts ‘X’ belongs to class ‘C_i’ iff

$$\mathbf{P(C_i \mid X) > P(C_j \mid X) \qquad \text{for } 1 \leq j, i \leq m \text{ and } j \neq i}$$

Naive Bayes classifier

- Naive assumption of **class conditional independence**

$$P(X \mid C_i) = P(x_1, x_2, x_3, x_4, \ldots\ldots x_n \mid C_i)$$

$$= P(x_1 \mid C_i) x P(x_2 \mid C_i) x P(x_3 \mid C_i) \ldots\ldots x P(x_n \mid C_i)$$

$$= \prod_{K=1}^{n} P(x_k \mid C_i)$$

- For maximum posterior probability $P(C_i \mid X)$ need to find maximum

$P(X \mid C_i)$ x $P(C_i)$ as $P(X)$ is constant.

Naive Bayes classifier

To compute $P(x_k \mid C_i)$ for,

- **categorical attribute A_k** :

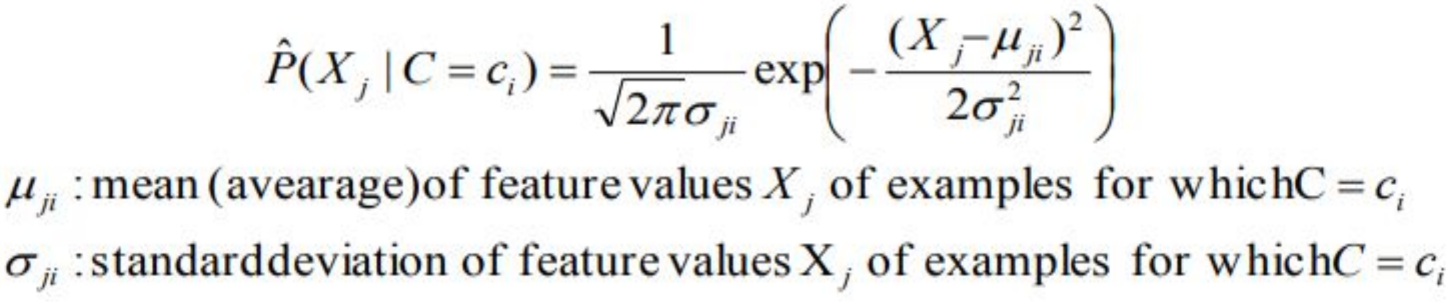
$$P(x_k \mid C_i) = \frac{\text{number of tuples of class } C_i \text{ in D having the value } x_k \text{ for } A_k}{\text{number of tuples of class } C_i \text{ in D}}$$

$$P(C_i) = \frac{\text{number of tuples of class } C_i \text{ in D}}{\text{Total number of tuples}}$$

Naive Bayes classifier

To compute P(x_k | C_i) for,

- **Continuous attribute A_k** :
  A continuous valued attribute is typically assumed to have a Gaussian distribution with a mean **μ** and standard deviation ***σ***.

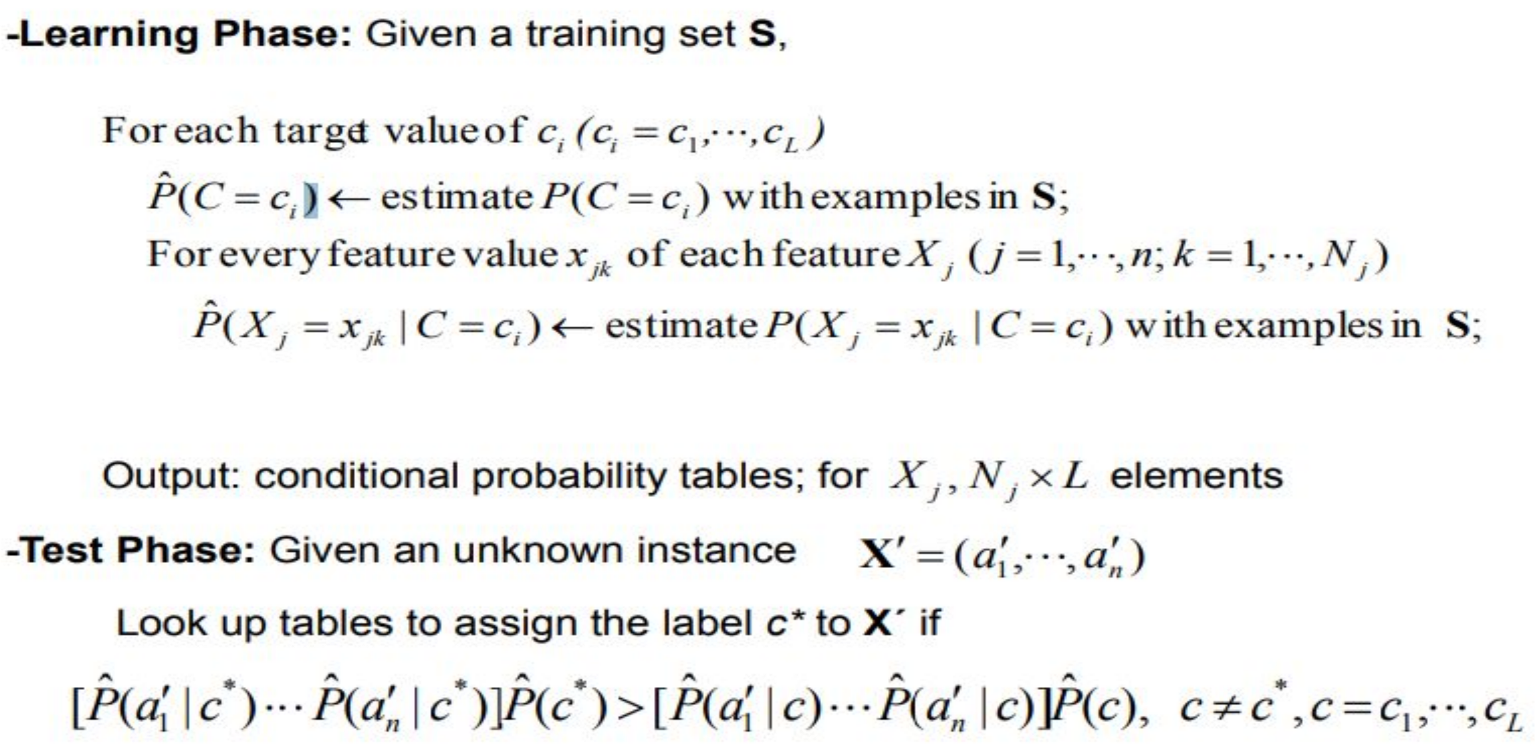


$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(- \frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

μ_{ji} : mean (avearage)of feature values X_j of examples for whichC $= c_i$

σ_{ji} : standarddeviation of feature values X$_j$ of examples for which$C = c_i$

Algorithm for Categorical(Discrete) features



-Learning Phase: Given a training set **S**,

```
For each target value of $c_i$ $(c_i = c_1, \cdots, c_L)$
    $\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in $\mathbf{S}$;
    For every feature value $x_{jk}$ of each feature $X_j$ $(j = 1, \cdots, n; k = 1, \cdots, N_j)$
        $\hat{P}(X_j = x_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in $\mathbf{S}$;
```

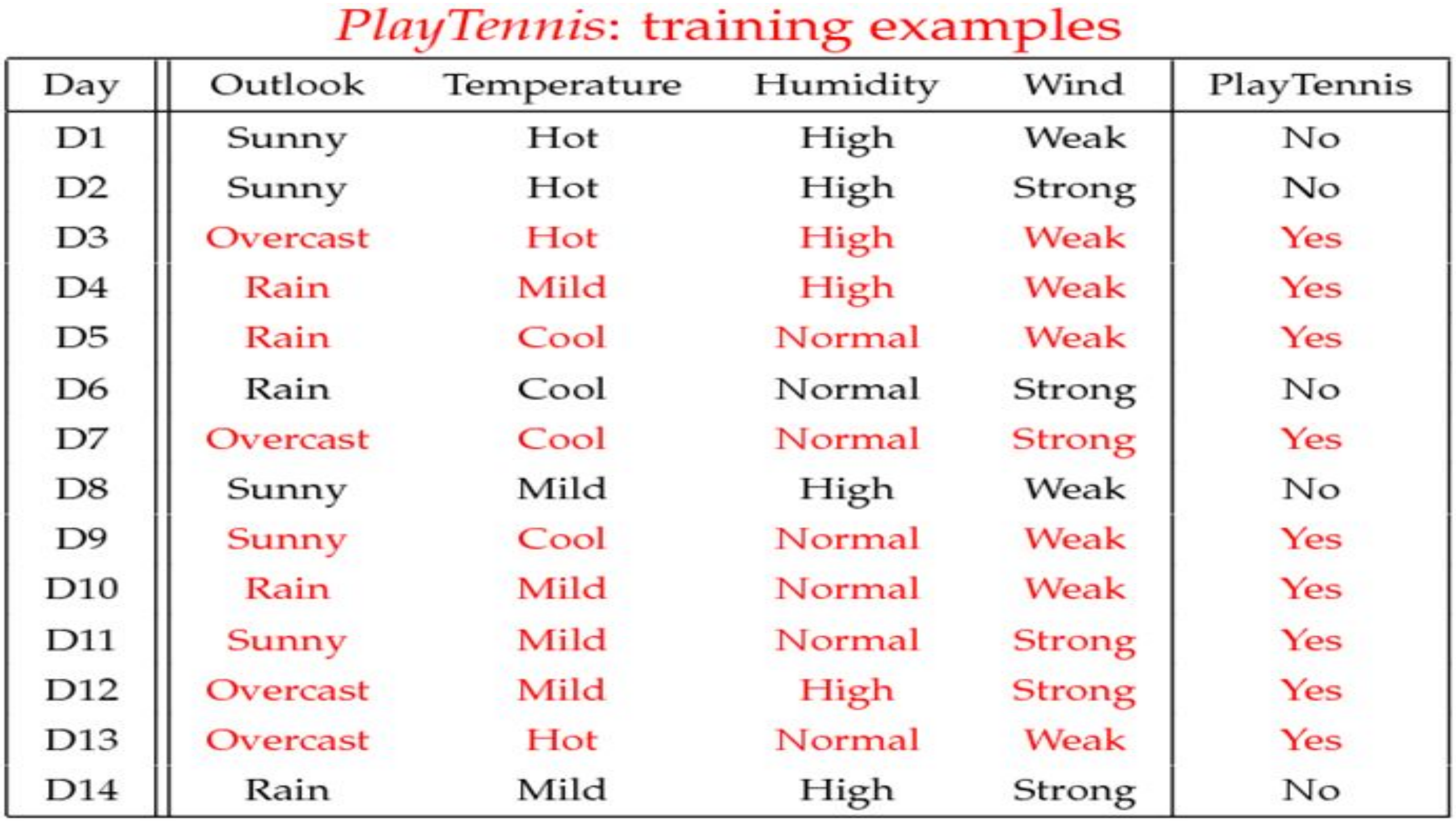
Output: conditional probability tables; for X_j, $N_j \times L$ elements

-Test Phase: Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$

Look up tables to assign the label *c** to **X**´ if

$$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c) \cdots \hat{P}(a'_n \mid c)]\hat{P}(c), \;\; c \neq c^*, c = c_1, \cdots, c_L$$

Example on Categorial-valued features



PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example on Categorical-valued features

D: Given set of 14 tuples (Days)

X: tuple with 4 attributes (Outlook, Temperature, Humidity, Wind)

C: 2 classes, C_1 = yes(play Tennis) and C_2 = no(don’t play Tennis)

Question : Given a day with Outlook is Sunny, Temperature is Cool, Humidity is High, Wind is Strong. Whether to play Tennis or not to play on the day?

Learning Phase

Build Frequency tables and corresponding Likelihood tables

Frequency Table

Outlook	**Play = Yes**	**Play = No**
Sunny	2	3
Overcast	4	0
Rain	3	2
Total	9	5

Likelihood Table

Outlook	**Play = Yes**	**Play = No**
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Learning Phase

Frequency Table

Temperature	Play = Yes	Play = No
Hot	2	2
Mild	4	2
Cool	3	1
Total	9	5

Likelihood Table

Temperature	Play = Yes	Play = No
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Learning Phase

Frequency Table

Humidity	**Play = Yes**	**Play = No**
High	3	4
Normal	6	1
Total	9	5

Likelihood Table

Humidity	**Play = Yes**	**Play = No**
High	3/9	4/5
Normal	6/9	1/5

Training Phase

Frequency Table

Wind	**Play = Yes**	**Play = No**
Strong	3	3
Weak	6	2
Total	9	5

Likelihood Table

Wind	**Play = Yes**	**Play = No**
Strong	3/9	3/5
Weak	6/9	2/5

Test Phase

Given,

X = (Outlook = Sunny, Temperature = Cool, Humidity = High, Wind = Strong)

Look up Tables achieved in Training Phase

$P(\text{Outlook=Sunny} \mid \text{Play=Yes}) = 2/9$	$P(\text{Outlook=Sunny} \mid \text{Play=No}) = 3/5$
$P(\text{Temperature=Cool} \mid \text{Play=Yes}) = 3/9$	$P(\text{Temperature=Cool} \mid \text{Play=No}) = 1/5$
$P(\text{Humidity=High} \mid \text{Play=Yes}) = 3/9$	$P(\text{Humidity=High} \mid \text{Play=No}) = 4/5$
$P(\text{Wind=Strong} \mid \text{Play=Yes}) = 3/9$	$P(\text{Wind=Strong} \mid \text{Play=No}) = 3/5$
$P(\text{Play=Yes}) = 9/14$	$P(\text{Play=No}) = 5/14$

Test Phase

Decision Making with MAP rule

$P(Yes|X) \approx P(X \mid Yes)P(Yes)$

$\approx [P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]P(Yes) = 0.0053$

$P(No|X) \approx P(X \mid Yes)P(Yes)$

$\approx [P(Sunny|No)\ P(Cool|No)P(High|No)P(Strong|No)]P(No) = 0.0206$

Given the fact P(Yes|X) < P(No|X), we label X to be “No”.

Prediction: Should **not play** on the day with Outlook is Sunny, Temperature is Cool, Humidity is High and Wind is Strong.

Zero Frequency Problem

- If no example contains the feature value, i.e. if test set has new attribute value not present in training set
  - In this circumstance, we face a zero conditional probability problem during test .

    $P(x_1 \mid C_i)xP(x_2 \mid C_i)xP(x_3 \mid C_i)........xP(x_n \mid C_i) = 0$ if some $P(x_i \mid C_i) = 0$

    where x_i is value of attribute a_i

Zero Frequency Problem

To avoid this problem conditional probability is re-estimated with

$$P(a_i \mid C_i) = \frac{n_c + mp}{n + m}$$

n_c = number of training examples of value x_i with attribute a_i and class C_i

n = number of training examples of class C_i

p = prior estimate(usually, $p = 1/t$ for t possible values of x_i

m = Weight to prior (number of “virtual” examples, $m >= 0$

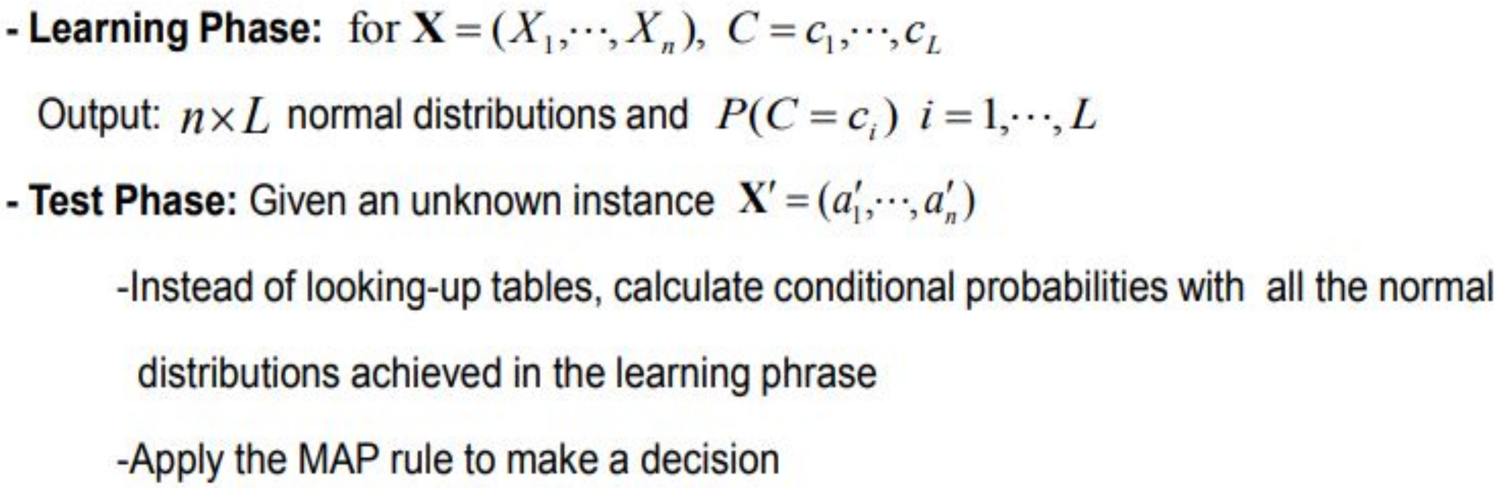
Zero Frequency Problem

- In previous Example: P(outlook=overcast | no)=0 in the play-tennis dataset
    - Adding m “virtual” examples (m: tunable but up to 1% of #training examples)
    - In this dataset, # of training examples for the “no” class is 5.
    - Assume that we add m=1 “virtual” example in our m-estimate treatment.
    - The “outlook” feature can takes only 3 values. So p=1/3.
    - Re-estimate P(outlook | no) with the m-estimate

$$P(\text{Overcast} \mid \text{no}) = \frac{0 + 1^{*}(⅓)}{5+1} = 1/18$$

$$P(\text{Sunny} \mid \text{no}) = \frac{3 + 1^{*}(⅓)}{5+1} = 5/9 \quad \text{and } P(\text{Rain} \mid \text{no}) = \frac{2 + 1^{*}(⅓)}{5+1} = 7/18$$

Algorithm for Continuous-valued features



- **Learning Phase:** for $\mathbf{X} = (X_1, \cdots, X_n)$, $C = c_1, \cdots, c_L$

  Output: $n \times L$ normal distributions and $P(C = c_i)$ $i = 1, \cdots, L$

- **Test Phase:** Given an unknown instance $\mathbf{X}' = (a'_1, \cdots, a'_n)$

  -Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase

  -Apply the MAP rule to make a decision

Example on Continuous-valued features

Temperature is naturally of continuous value.

Play Game		Temperature(in ºC)
Yes	:	25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8
No	:	27.3, 30.1, 17.4, 29.5, 15.1

Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \quad \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \quad \sigma_{No} = 7.09$$

Example on Continuous-valued features

Finding Posterior probability :

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

μ_{ji} : mean (avearage)of feature values X_j of examples for whichC $= c_i$

σ_{ji} : standarddeviation of feature values X$_j$ of examples for which$C = c_i$

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{2 \times 2.35^2} \right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x - 21.64)^2}{11.09} \right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{2 \times 7.09^2} \right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x - 23.88)^2}{50.25} \right)$$

Application

- Text Classification
- Weather Prediction
- Medical Diagnosis
- Hybrid Recommender System
  - Recommender Systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource
- Simple Emotion Modeling

Why Text classification

- Learning which articles are of interest
- Classify web pages by topic
- Information extraction
- Internet filters

Text classifications

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis

Examples

- CLASSES=BINARY
  - “spam” / “not spam”
- CLASSES =TOPICS
  - “finance” / “sports” / “politics”
- CLASSES =OPINION
  - “like” / “hate” / “neutral”
- CLASSES =TOPICS
  - “AI” / “Theory” / “Graphics”
- CLASSES =AUTHOR
  - “Shakespeare” / “Marlowe” / “Ben Jonson”

Naive Bayes approach for Text classification

- Build the Vocabulary as the list of all distinct words that appear in all the documents of the training set.
- Remove stop words and markings.
- The words in the vocabulary become the attributes, assuming that classification is independent of the positions of the words
- Each document in the training set becomes a record with frequencies for each word in the Vocabulary.
- Train the classifier based on the training data set, by computing the prior probabilities for each class and likelihood probability of attributes.
- Evaluate the results on Test data.

List of Stop words

{‘ourselves’, ‘hers’, ‘between’, ‘yourself’, ‘but’, ‘again’, ‘there’, ‘about’, ‘once’, ‘during’, ‘out’, ‘very’, ‘having’, ‘with’, ‘they’, ‘own’, ‘an’, ‘be’, ‘some’, ‘for’, ‘do’, ‘its’, ‘yours’, ‘such’, ‘into’, ‘of’, ‘most’, ‘itself’, ‘other’, ‘off’, ‘is’, ‘s’, ‘am’, ‘or’, ‘who’, ‘as’, ‘from’, ‘him’, ‘each’, ‘the’, ‘themselves’, ‘until’, ‘below’, ‘are’, ‘we’, ‘these’, ‘your’, ‘his’, ‘through’, ‘don’, ‘nor’, ‘me’, ‘were’, ‘her’, ‘more’, ‘himself’, ‘this’, ‘down’, ‘should’, ‘our’, ‘their’, ‘while’, ‘above’, ‘both’, ‘up’, ‘to’, ‘ours’, ‘had’, ‘she’, ‘all’, ‘no’, ‘when’, ‘at’, ‘any’, ‘before’, ‘them’, ‘same’, ‘and’, ‘been’, ‘have’, ‘in’, ‘will’, ‘on’, ‘does’, ‘yourselves’, ‘then’, ‘that’, ‘because’, ‘what’, ‘over’, ‘why’, ‘so’, ‘can’, ‘did’, ‘not’, ‘now’, ‘under’, ‘he’, ‘you’, ‘herself’, ‘has’, ‘just’, ‘where’, ‘too’, ‘only’, ‘myself’, ‘which’, ‘those’, ‘i’, ‘after’, ‘few’, ‘whom’, ‘t’, ‘being’, ‘if’, ‘theirs’, ‘my’, ‘against’, ‘a’, ‘by’, ‘doing’, ‘it’, ‘how’, ‘further’, ‘was’, ‘here’, ‘than’}

Naive Bayes approach

Finding Prior and likelihood probabilities using word(w) frequencies.

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Text classification Example

Given Data with 5 documents and respective tags(label)

Doc	Text	Tag
1	“A great game”	Sports
2	“The election was over”	Not Sports
3	“Very clean match”	Sports
4	“A clean but forgettable game”	Sports
5	“It was a close election”	Not Sports

Text classification Example

Given a document having “ A very close game”. To which Tag does it belongs to?

Approach:

We use **word frequencies**. That is, we ignore word order and sentence construction, treating every document as a set of the words it contains. Our features will be the counts of each of these words.

First find and remove Stop words : <A, The, was, over , very, it, but>

Text classification Example

Frequency table of remaining words:

Word	Sports	Not Sports
great	1	0
game	2	0
election	0	2
clean	2	0
match	1	0
forgettable	1	0
close	0	1
Total	7	3

Text classification Example

Likelihood table of required words(given “a very close game”) after removing stop words

Word	P(Word | Sports)	P(Word | Not Sports)
close	$\frac{0}{7}$	$\frac{1}{3}$
game	$\frac{2}{7}$	$\frac{0}{3}$

Zero Frequency Problem

P(Sports) = 3/5   P(Not Sports) = 2/5

P(“A very close game” | Sports)

= P(close | Sports) x P(game | Sports)

- However, we run into a problem here: “close” doesn’t appear in any *Sports* text!.
- That means that P(close | Sports) = 0.
- This makes whole value equals 0, since it is a multiplication.
- To avoid this we use Laplace smoothing.

Zero Frequency Problem

Laplace Smoothing: we add 1 to every count so it’s never zero. To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1.

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'}) + B'},$$

Tct – Number of particular word in particular class
Tct’ – Number of total words in particular class
B’ – Number of distinct words in all class

Text classification Example

Re- computed Likelihood table

Word	P(Word | Sports)	P(Word | Not Sports)
close	$\frac{0 + 1}{7 + 7}$	$\frac{1 + 1}{3 + 7}$
game	$\frac{2 + 1}{7 + 7}$	$\frac{0 + 1}{3 + 7}$

Text classification Example

From the Question we need to find tag of “A very close game”

P(“A very close game” | Sports)

= P(close | Sports) x P(game | Sports) [∵ <a, very> are stop words]

= (1/14)*(3/14) = 0.0153

P(“A very close game” | Not Sports)

= P(close | Not Sports) x P(game | Not Sports)

= (2/10) * (1/10) = 0.02

Text classification Example

P(Sports | “A very close game”) = P(A very close game | Sports) *P(Sports)
= 0.0153*(3/5)
= 0.00918

P(Not Sports | “A very close game”) = P(A very close game | Not Sports) *P(Not Sports)
= 0.02*(2/5)
= 0.008

Since **P(Sports | “A very close game”) > P(Not Sports | “A very close game”)**

Our classifier gives **Sports** tag to the document “a very close game”

Pros and Cons

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction.
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Pros and Cons

Cons:

- Limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

  Eg: hospitals: patients: Profile: age, family history, etc. Symptoms: fever, cough etc., Disease: lung cancer, diabetes, etc.

References

- Data Mining: Concepts and Techniques, 3rd Edition, Han & kamber & Pei ISBN: 9780123814791.
- Building a Naive Bayes Text Classifier and Accounting for Document Length, Dwight Sunanda, 2017 ISBN:1521523800, 9781521523803
- Naive Bayes Classifier, Text classification.Edureka.
- http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch6.pdf.

Thank you