#### CUNY School of Professional Studies

#### SPS.CUNY.EDU

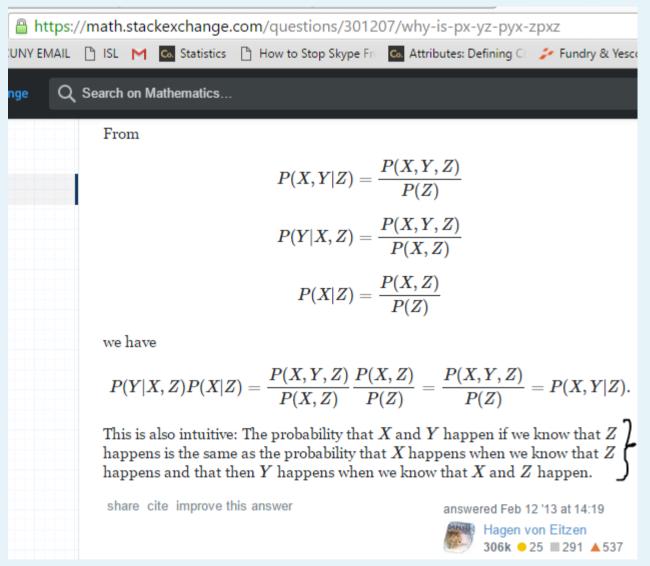
Lecture 08
2020 Spring Data-622
Naive Bayes
Raman Kannan

Instructor Email Address: Raman.Kannan@sps.cuny.edu

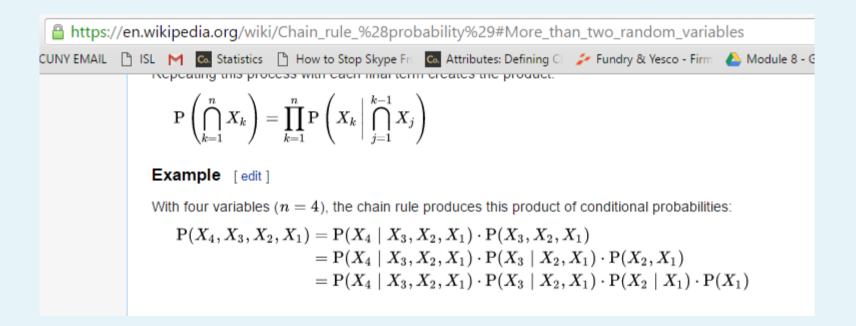
Acknowledgements:

Generous support from IBM Power Systems Academic Initiative IBM PSAI provides computing infrastructure for free

### Multivariate Conditional Probability



# Multivariate Joint Probability AKA Chain Rule



### Conditional Independence

https://www.probabilitycourse.com/chapter1/1\_4\_4\_conditional\_independence.php

▣

67%

HOME VIDEOS CALCULATOR COMMENTS COURSES FOR INSTRUCTORS

Si.

- 0 Preface
- 1 Basic Concepts
- 1.0 Introduction
- 1.1 Introduction
- 1.2 Review of Set Theory
- 1.3 Random Experiments and Probabilities
- 1.4 Conditional Probability
- 1.4.0 Conditional Probability
- 1.4.1 Independence
- 1.4.2 Law of Total Probability
- 1.4.3 Bayes' Rule
- 1.4.4 Conditional Independence

#### 1.4.4 Conditional Independence

As we mentioned earlier, almost any concept that is defined for probability can also be extended to conditional probability. Remember that two events A and B are independent if

$$P(A \cap B) = P(A)P(B)$$
, or equivalently,  $P(A|B) = P(A)$ .

We can extend this concept to conditionally independent events. In particular,

#### Definition 1.2

Two events A and B are **conditionally independent** given an event C with P(C)>0 if

$$P(A \cap B|C) = P(A|C)P(B|C) \tag{1.8}$$

Recall that from the definition of conditional probability,

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

if P(B)>0. By conditioning on C, we obtain

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

if  $P(B|C), P(C) \neq 0$ . If A and B are conditionally independent given C, we obtain

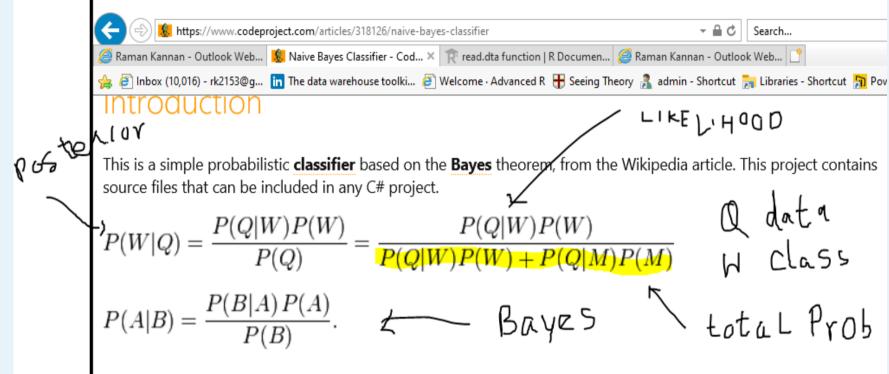
$$\begin{split} P(A|B,C) &= \frac{P(A \cap B|C)}{P(B|C)} \\ &= \frac{P(A|C)P(B|C)}{P(B|C)} \\ &= P(A|C). \end{split}$$

Thus, if  ${m A}$  and  ${m B}$  are conditionally independent given  ${m C}_t$  then

$$P(A|B,C) = P(A|C) \tag{1.9}$$

Thus, Equations  $\underline{1.8}$  and  $\underline{1.9}$  are equivalent statements of the definition of conditional independence. Now let's look at an example.

# Bayes → Naive



The **Bayes**ian **Classifier** is capable of calculating the most probable output depending on the input. It is possible to add new raw data at runtime and have a better probabilistic **classifier**. A **naive Bayes classifier** assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of other features, a **naive Bayes classifier** considers all of these properties to independently contribute to the probability that this fruit is an apple.

That is why the model is naive...

### A simple NB example

#### Sex classification

Problem: classify whether a given person is a male or a female based on the measured features. The features include height, weight, and foot size.

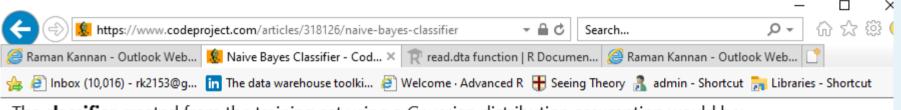
#### **Training**

Example training set is shown below.

dftrain<-data.frame(gender=c('M','M','M','M','F','F','F','F	-')
height=c(6,5.92,5.58,5.92,5,5.5,5.42,5.75),	-
weight=c(180,190,170,165,100,150,130,150),	
foot=c(12,11,12,10,6,8,7,9),stringsAsFactors=F)	

sex	height (feet)	weight (lbs)	foot size (inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

8 data points,2 classes, 3 predictors

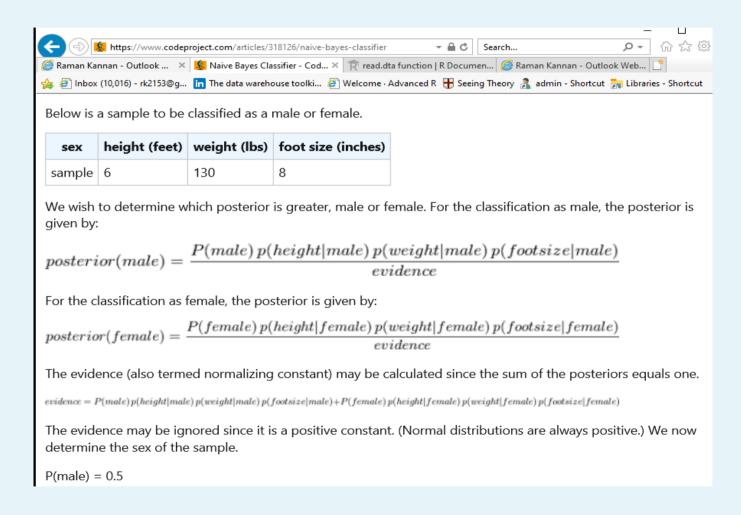


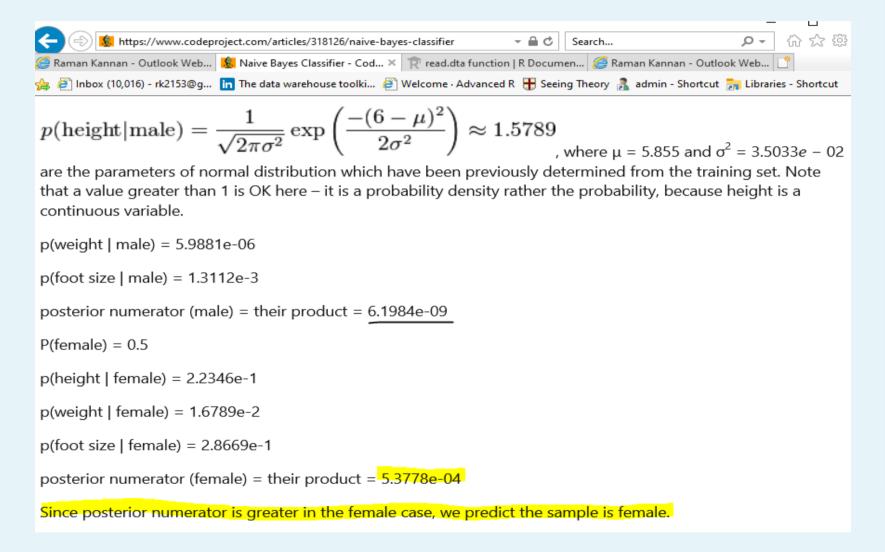
The classifier created from the training set using a Gaussian distribution assumption would be:

sex	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033e-02	176.25	1.2292e+02	11.25	9.1667e-01
female	5.4175	9.7225e-02	132.5	5.5833e+02	7.5	1.6667e+00

Let's say we have equiprobable classes so P(male) = P(female) = 0.5. There was no identified reason for making this assumption so it may have been a bad idea. If we determine P(C) based on frequency in the training set, we happen to get the same answer.

```
male_mean<-unlist(apply(dftrain[dftrain$gender=='M',2:4],2,mean))
male_mean
male_sd<-unlist(apply(dftrain[dftrain$gender=='M',2:4],2,sd))
female_sd<-unlist(apply(dftrain[dftrain$gender=='F',2:4],2,sd))
female_mean<-unlist(apply(dftrain[dftrain$gender=='F',2:4],2,mean))
dfparameters<-data.frame(rbind(male_mean,male_sd))
dfparameters<-rbind(dfparameters,female_mean,female_sd)
rownames(dfparameters)<-c("male_mean","male_sd","female_mean","female_sd")
dfparameters<-cbind(dfparameters,gender=c('M','M','F','F')
```





```
given<-c(6,130,8)
as.character(unique(dfparameters$gender)[[which.max(
unlist(lapply(lapply(
lapply(1:2,FUN=function(x,v=given,df=dfparameters){
cid=x
meanrow=(x-1)*2+1
sdrow=(x-1)*2+2
prob<-unlist(lapply(1:length(v),FUN=function(x,vi=v,dfi=df)
value=vi[[x]]
mu=dfi[meanrow,x]
sigma=dfi[sdrow,x]
rval<-(1/sqrt(2*22*sigma^2/7))*exp((-1*(value-mu)^2)/(2*sigma^2))
rval
}))
prob}
,cumprod),FUN=function(x)x[[length(x)]])))]])
```

```
given<-c(6,130,8)
as.character(unique(dfparameters$gender)[[which.max(
unlist(lapply(lapply(
lapply(1:2,FUN=function(x,v=given,df=dfparameters){
cid=x
meanrow=(x-1)*2+1
sdrow=(x-1)*2+2
prob<-unlist(lapply(1:length(v),FUN=function(x,vi=v,dfi=df)
value=vi[[x]]
mu=dfi[meanrow,x]
sigma=dfi[sdrow,x]
rval<-(1/sqrt(2*22*sigma^2/7))*exp((-1*(value-mu)^2)/
(2*sigma^2))
rval
}))
prob}
,cumprod),FUN=function(x)x[[length(x)]])))]])
```

```
> given < -c(6,130,8)
> as.character(unique(dfparameters$gender)[[which.max(
+ unlist(lapply(lapply(
+ lapply(1:2,FUN=function(x,v=given,df=dfparameters){
+ cid=x
+ meanrow = (x-1)*2+1
+ sdrow=(x-1)*2+2
+ prob<-unlist(lapply(1:length(v), FUN=function(x, vi=v, dfi=df)
+ value=vi[[x]]
+ mu=dfi[meanrow,x]
+ sigma=dfi[sdrow,x]
+ rval<-(1/sgrt(2*22*sigma^2/7))*exp((-1*(value-mu)^2)/(2*sigma^2))
+ rval
+ 1))
+ prob}
+ )
+ , cumprod), FUN=function(x)x[[length(x)]])))]])
```

How can we iteraate through all the observations in dftrain?

Is it possible to verify our implementation with standard package NB Implementation.

```
> unlist(
+ apply(dftrain[,-1],1,FUN=function(x)
+ given=x
+ as.character(unique(dfparameters$gender)[[which.max(
+ unlist(lapply(lapply(
+ lapply(1:2,FUN=function(x,v=given,df=dfparameters){
+ meanrow=(x-1)*2+1
+ sdrow=(x-1)*2+2
+ prob<-unlist(lapply(1:length(v), FUN=function(x, vi=v, dfi=df)
+ value=vi[[x]]
+ mu=dfi[meanrow,x]
+ sigma=dfi[sdrow,x]
+ rval<-(1/sqrt(2*22*sigma^2/7))*exp((-1*(value-mu)^2)/(2*sigma^2))
+ }))
+ prob}
+ ,cumprod), FUN=function(x)x[[length(x)]])))]])
   uMu uMu uMu uMu uEu uEu uEu uEu
> dftrain[,1]
   uMu uMu uMu uMu uEu uEu uEu uEu
```

```
require(e1071)
nbmodel<-naiveBayes(gender~.,data=dftrain)
nb.predicted<-predict(nbmodel,dftrain[,-1],type='raw')
nb.predicted
unlist(lapply(apply(nb.predicted,1,which.max),
FUN=function(x)names(as.data.frame(nb.predicted))
[[x]]))
#confusion matrix for training error
```

table(dftrain\$gender,unlist(lapply(apply(nb.predicted,1,w hich.max), FUN=function(x)names(as.data.frame(nb.predicted)) [[x]]))) # never seen before data dftest<-data.frame(height=6,weight=130,foot=8) nbtest.pred<-predict(nbmodel,dftest,type='raw') nbtest.pred dftest<-data.frame(height=6,weight=130,foot=8) predict(nbmodel,dftest,type='raw')

```
> require(e1071)
> nbmodel<-naiveBayes(gender~.,data=dftrain)
> nb.predicted<-predict(nbmodel,dftrain[,-1],type='raw')
> #nb.predicted
> unlist(lapply(apply(nb.predicted, 1, which.max),
+ FUN=function(x)names(as.data.frame(nb.predicted))[[x]]))
[1] nWu nWu nWu nWu nEu nEu nEu nEu
> #confusion matrix for training error
> table(dftrain$gender,unlist(lapply(apply(nb.predicted,1,which.max),
+ FUN=function(x)names(as.data.frame(nb.predicted))[[x]])))
    F M
  F 4 0
 M 0 4
> # never seen before data
> dftest<-data.frame(height=6,weight=130,foot=8)</p>
> nbtest.pred<-predict(nbmodel,dftest,type='raw')
> nbtest.pred
                          М
[1,] 0.9999885 1.152307e-05
> dftest<-data.frame(height=6,weight=130,foot=8)</p>
> predict(nbmodel,dftest,tvpe='raw')
[1,] 0.9999885 1.152307e-05
```

#### References

https://en.wikipedia.org/wiki/Chain\_rule\_%28probability%29#More\_than\_two\_random\_variables https://math.stackexchange.com/questions/301207/why-is-px-yz-pyx-zpxz https://corporatefinanceinstitute.com/resources/knowledge/other/total-probability-rule/https://www.probabilitycourse.com/chapter1/1\_4\_4\_conditional\_independence.php https://www.codeproject.com/articles/318126/naive-bayes-classifier https://rpubs.com/riazakhan94/naive\_bayes\_classifier\_e1071

#### Review

We have seen 3 different probabilistic classifiers: Logistic, LDA and NaiveBayes

Since these algorithms model a probability distribution needing Only a few parameters for the PDF, they are also called parametric models.

Is it possible to estimate class given data without assuming PDF. Our focus will be such classifiers which do not assume any PDF and calculate based on logic and instances.

Practice ... Practice ... Practice