

Scribe Note for Machine Learning (CS60050),  
Spring 2020-2021

Topic : Bayesian Learning

Date : 21<sup>st</sup> January, 2021

Instructor : Prof. Aritra Hazra

Asmita Nandy

## 1 Summary of Last Class

We have started with the probability overview but the two important things in probability we need to know is:

1. **Conditional probability:** It says about how we define the  $P(A|B)$  in mathematical formulation of  $\frac{P(A \wedge B)}{P(B)}$  and chain rule accordingly.
2. **Bayes' rule:** It says about where we could define the conditional probability in a flipped way to know  $P(A|B)$ , based on our knowledge of prior probability and that of marginal probability. This is very important to know for learning.

All these probability notions sum up into a table, known as Joint Probability Distribution (JDT). If we have the knowledge of the full set of distribution and their probability in the JDT, then we can know arbitrarily anything about the probabilistic nature of any event and their composition, by summing up the rows in case of conditional probability just by taking a ratio of the summed over rows. This is what we need in the probability counterpart.

Then we revisited our learning problem which tries to hypothesize an unknown function. It knows only that training set of examples, set of attributes leading to a classification value of Y. It can view only that and can find out the unknown function. We have seen that for concept learning and decision tree learning however, we change our premise a little bit by saying that - why not we also try to find out what is the  $P(Y|X)$  . Let us say that it is

Figure 1: Slide Reference - from 00:11 to 04:05

▶ Probability Overview:  
 ↳ Conditional Prob,  $P(A|B) = \frac{P(A \wedge B)}{P(B)}$   
 ↳ Bayes Rule,  $P(A|B) = \frac{P(B|A) P(A)}{P(B)}$

SUMMARY of Previous Lect  
 Joint Probability Distribution  
 $\sum_{\text{rows}} (\text{Events})$

▶ Learning Problem:  
 ↳  $f: X \rightarrow Y$  (unknown)  
 ↳ Concept Learning  
 ↳ Decision Tree Learning  
 ↳ Prob (Y | X) → Bayesian Learning (Classification)

▶ Probability Estimations:  
 ↳ Maximum Likelihood Estimation (MLE) #Ex: Coin Flip  
 choose  $\theta$  that maximizes  $\text{Prob}(\text{Data} | \theta)$   
 $\hat{\theta}_{MLE} = \underset{\theta}{\text{argmax}} (\text{Prob}(\text{Data} | \theta))$   
 ↳ Maximum A Posteriori (MAP)  
 choose  $\theta$  that maximizes  $\text{Prob}(\theta | \text{Data})$

Smart Estimations

a classification problem and what is the probability for Y being positive and what is the probability of Y being negative discrete values of Y, classes given the attributes. So we call such kind of a learning as Bayesian learning where we try to use the Bayes' rule or kind of conditional probability, by seeing that - given a new set of attributes, which class is more probable. Then we will classify to that class.

Joint Probability Distribution is very good, however to our utter surprise we see that but we cant use it in practice for data sparsity. So we have to think about smart probability estimation. For example, one of the smart probability estimation that we derive from our common knowledge of coin flipping is that, if we have a large set of data in our hand then we will try to see that, how can we make my estimation more accurate towards the data. So for that, we need to choose our estimated probability in such a way that it maximises the probability of that choice given, data that is given based on the choice that I make.

And in coin flipping example we have seen that our common sense of basic intuition matches with what we want to make as a likelihood estimation.

We also had a note in the last class that - we made another type of estimation where our data is not that much strong or there is less amount of data in our hand, so we cannot try to estimate as we do in Maximum likelihood i.e. estimation will not follow the data because data may carry now misinformation due to a very less amount of availability. So for that we started to think about maximum a posteriori analysis.

## 2 Today's Class

### 2.1 Slide Reference 2 given in Figure 2

Figure 2: Slide Reference - from 04:05 to 15:39

MAP: choose  $\theta$  that  $\max. \text{Prob}(\theta | \text{data})$

$$\hat{\theta}_{\text{MAP}} = \underset{\theta}{\text{argmax}} \left[ \text{Prob}(\theta | \text{data}) \right]$$

*likelihood*

$$= \underset{\theta}{\text{argmax}} \left[ \frac{\text{Prob}(\text{data} | \theta) \cdot P(\theta)}{P(\text{data})} \right]$$

*Prior*

*max*

#Ex: Coin flip:  $\hat{\theta} = \frac{\alpha_H + \# \text{PreH}}{(\alpha_H + \# \text{PreH}) + (\alpha_T + \# \text{PreT})}$

*marginal*

*Conj Prior*

$$P(\text{data} | \theta) = \theta^{\alpha_H} (1-\theta)^{\alpha_T}$$

$$P(\theta) = \frac{\theta^{\beta_H-1} (1-\theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim \text{Beta}(\theta, \beta_H, \beta_T)$$

$$\frac{\partial}{\partial \theta} \left[ \theta^{\alpha_H + \beta_H - 1} (1-\theta)^{\alpha_T + \beta_T - 1} \right] = 0$$

$$\Rightarrow \hat{\theta}_{\text{MAP}} = \frac{\alpha_H + (\beta_H - 1)}{(\alpha_H + (\beta_H - 1)) + (\alpha_T + (\beta_T - 1))}$$

So, as we know that in maximum posteriori analysis, what we want to do? We want to choose a  $\theta$  that maximizes  $P(\theta | \text{data})$ . Which means

that, in mathematical sense, my MAP estimate that will come up, will be  $\arg \max_{\theta}[P(\theta|data)]$ . What we will want to happen is :

MAP : Choose  $\theta$  that maximizes  $P(\theta|data)$

$$\hat{\theta}_{MAP} = \arg \max_{\theta}[P(\theta|data)] \implies \arg \max_{\theta}\left[\frac{P(data|\theta)P(\theta)}{P(data)}\right] \quad (1)$$

Now you can see that this factor ( $P(data|\theta)$ ) is our likelihood. By saying this, I meant about how we interpret this one?  $P(\theta)$  is our prior knowledge and so I know that - since I was trying to estimate - that's why some prior knowledge about the distribution of  $\theta$  will be needed to be known which we called as a prior and we call  $P(data)$  probability term as a marginal. However to get an argmax of that, we can easily understand that numerator is only dependent on  $\theta$ , and denominator is not. So, our only concern will be to get maximum out of the numerator part. Now, since I have less amount of data so this prior plays a very pivotal role here. We need to know some information about the prior knowledge of our estimation. So if we take the example of coin flipping, we had an algorithm to find this and we say that the estimate that we give about heads and tails is nothing but as follows:

$$\hat{\theta} = \frac{(\alpha_1 + \#preH)}{(\alpha_1 + \#preH) + (\alpha_0 + \#preT)} \quad (2)$$

where preH = some number of precomputed heads and preT = some number of precomputed tails (as given in Fig. 2).

This is done because this acts as a prior to us, because this preH over precomputed heads plus preT over precomputed tails are the prior knowledge about coin biases because if we donot have any data, that means our  $\alpha_1$  flips of head is equal to 0 and also  $\alpha_0$  flips of tails is equal to 0. Then our prior knowledge dominates, whatever be our estimate. Now gradually if we have more and more data - let us say  $\alpha_1 = 1000$  and  $\alpha_0 = 1000$  or 1 lakh kind of data, then our alpha component dominates and our prior cancels out. So that's why this is an online learning algorithm, since by intuition we sort out that we will go on flipping and try to estimate using this rule. Now exactly this will come if we can leverage our prior to unsuitable manner by maximising this as per our equation. So let us assume that our, as u know already from the maximum likelihood that we have this as :

$$P(data|\theta) = \theta^{\alpha_1}(1 - \theta)^{\alpha_0} \quad (3)$$

Let us assume the prior as a beta distribution. It says that suppose my prior knowledge follows a distribution of coin flipping which is a beta distribution. This is of the form like this:

$$P(\theta) = \frac{\theta^{\beta_H-1}(1-\theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \quad (4)$$

This is a Binomial distribution, but we call it a beta distribution of  $\theta$  given this  $Beta(\theta, \beta_H, \beta_T)$ . Since we don't know the exact bias, so we assume such distribution:

$$P(\theta) = \frac{\theta^{\beta_H-1}(1-\theta)^{\beta_T-1}}{B(\beta_H, \beta_T)} \sim Beta(\theta, \beta_H, \beta_T) \quad (5)$$

If we know that our prior coin flipping falls into such kind of a distribution, which is my maximum likelihood thing, then again we are trying to maximise product of  $P(\text{data}|\theta)$  and  $P(\theta)$  into partial derivative, because I need to maximize the numerical term only. So I am trying to maximize :

$$\frac{\partial \ln[\theta^{\alpha_H+\beta_H-1}(1-\theta)^{\alpha_T+\beta_T-1}]}{\partial \theta} = 0 \quad (6)$$

So previously we considered  $\alpha_H$  as  $\alpha_1$  and  $\alpha_T$  as  $\alpha_2$ , so :

$$\hat{\theta}_{MAP} = \frac{\alpha_H + \beta_H - 1}{(\alpha_H + \beta_H - 1) + (\alpha_T + \beta_H - 1)} \quad (7)$$

Now what does this signify? See it signifies the exact intuition that we have in our coin flipping example. The square box bracketed is our hallucinated pre thought about the distribution prior:

$$\hat{\theta}_{MAP} = \frac{\alpha_H + [\beta_H - 1]}{(\alpha_H + [\beta_H - 1]) + (\alpha_T + [\beta_H - 1])} \quad (8)$$

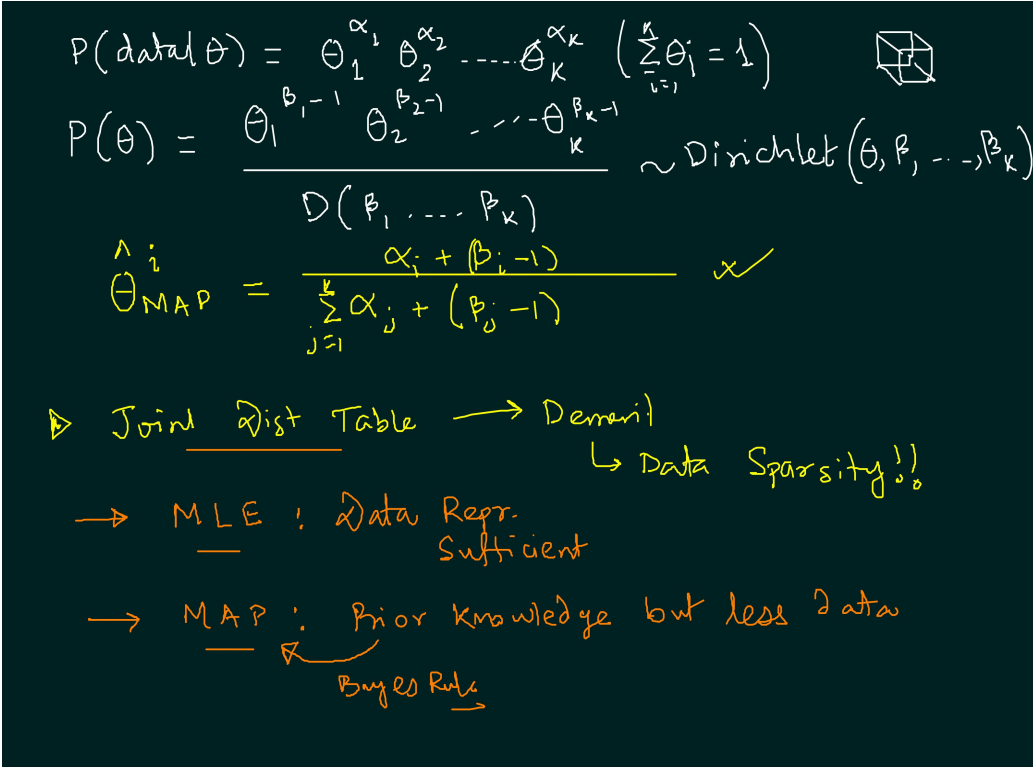
So therefore we can easily understand that if we don't have much data or say if we have ample number of data, our online algorithm in both the cases (in MLE and MAP estimates) is a good estimate. This MAP estimate is a good estimate because if we have more data we can follow this one, where this one will dominate our prior knowledge and automatically data will be more likely. If we have less data then our prior knowledge has to dominate because unless we have something we don't learn, we don't estimate anything. So that's why if we take beta distribution we can see that our common sense falling up and why beta distribution? There is something in statistics which we call conjugate prior, where our final answer is of the same structural type

as our prior, then we call it as a conjugate prior. We see this likelihood and prior is of same type (that is Eqn. 3 and Eqn. 4 respectively).

Also when we multiply it and try to find our final  $P(\theta|\text{data})$ , it is of same type because it is coming of the same style of expression. So, in common sense, we call it as a conjugate prior in statistics and usually when people try to estimate such kind of a thing, we will try to find out some prior which resembles my likelihood data. That is why we took this kind of a beta distribution in two part classification , i.e. heads or tails.

## 2.2 Slide Reference 3 given in Figure 3

Figure 3: Slide Reference - from 15:39 to 25:20



$$P(\text{data}|\theta) = \theta_1^{\alpha_1} \theta_2^{\alpha_2} \dots \theta_k^{\alpha_k} \left( \sum_{i=1}^k \theta_i = 1 \right)$$

$$P(\theta) = \frac{\theta_1^{\beta_1-1} \theta_2^{\beta_2-1} \dots \theta_k^{\beta_k-1}}{D(\beta_1, \dots, \beta_k)} \sim \text{Dirichlet}(\theta, \beta_1, \dots, \beta_k)$$

$$\hat{\theta}_{MAP}^i = \frac{\alpha_i + \beta_i - 1}{\sum_{j=1}^k \alpha_j + (\sum_{j=1}^k \beta_j - 1)}$$

▶ Joint Dist Table → Demand → Data Sparsity!!  
 → MLE : Data Repr. Sufficient  
 → MAP : Prior Knowledge but less data  
     ↖ Bayes Rule →

And if we try to look around in a more generic sense, we will use  $\theta_1^{\alpha_1}, \theta_2^{\alpha_2}$ , and so on... ,  $\theta_k^{\alpha_k}$  (if instead of a coin flip, we use a six sided die, there will be multiple outcomes instead of a head or tail)

$$P(\text{data}|\theta) = \theta_1^{\alpha_1}, \theta_2^{\alpha_2}, \dots, \theta_k^{\alpha_k} \quad (9)$$

$$\sum_{j=1}^k \theta_j = 1$$

If we try to generalize this in terms of k classifiers, we take the prior as a Dirichlet prior. We can easily understand about what we do is we get it as a Dirichlet distribution of theta with  $\beta_1$  upto  $\beta_k$  .

$$P(data|\theta) = \frac{\theta_1^{\beta_1-1}, \theta_2^{\beta_2-1}, \dots, \theta_k^{\beta_k-1}}{D(\beta_1, \dots, \beta_k)} \sim Dirichlet(\theta, \beta_1, \dots, \beta_k) \quad (10)$$

Then also if we try to make the map estimate, and prior knowledge, we will get the same thing ( $\hat{\theta}_{MAP}^i$ (lets say in a dice the ith number occurs) ):

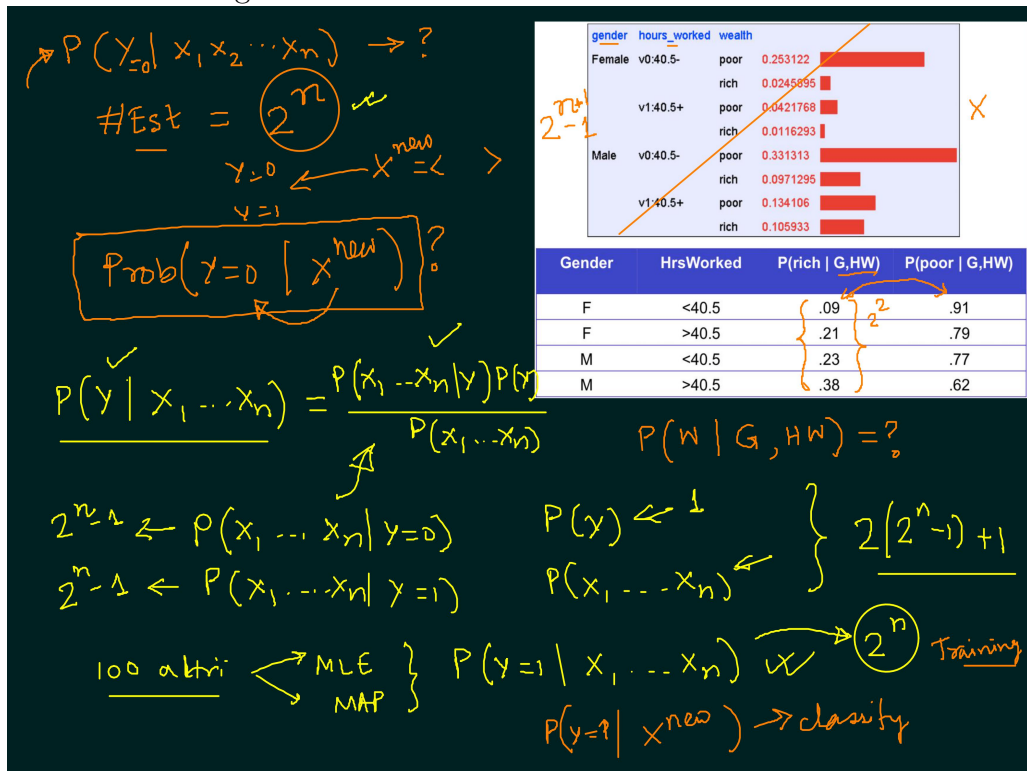
$$\hat{\theta}_{MAP}^i = \frac{\alpha_i + (\beta_i - 1)}{\sum_{j=1}^k \alpha_j + (\beta_j - 1)} \quad (11)$$

Just like the two flipping case, we just declare here like this way (the above estimation is for multi-valued attribute). It is nothing much to think on this because it is just a different distribution since we now have different classification classes rather than only 2. So the main point is that JDT has the demerit of data sparsity and as the name 'data sparsity' says - we can understand that it means we have less amount of data for all possible columns. Now if we have sufficient represented data then we go for MLE, may not be all data as - in JDT we have all  $2^n$  entries but there are sufficient representation. Where we have less amount of data but we have a prior knowledge, we go for MAP estimate which says Maximum Aposteriori. The name comes from the fact that we have a prior knowledge and we are predicting a posterior knowledge based on the Bayes' rule. That's what the smart estimation thing means. We have to be either smarter in estimating from less amount of data with a prior knowledge or have to be smarter from estimating whatever data we r given, even if knowledge is full. So, these are some of the ingredients that we need to know for our learning algorithm because I can now again post the same question of learning.

### 2.3 Slide Reference 4 given in Figure 4

Now, We have a JDT and all that we know is conditional probability, so now coming back to our learning question - let us say we are classifying with the set of n attributes  $P(Y|x_1 x_2 x_3 \dots x_n)$  , so this is what we want to answer now. So as we can see that if u have a JDT, it is pretty bad because we need to have  $2^n$  rows if there are n attributes. For example, in this case what

Figure 4: Slide Reference - from 25:20 to 34:32



we are trying to learn is that for probability of wealth given the gender and given the hours of working i.e.  $P(W|G, HW)$ , this is what we are trying to learn. So if we have the full distribution table, we need to estimate  $2^n - 1$  no. of cases because one case we can eliminate, i.e. the whole sum. So why  $n$ ?  $n$  Because I have  $n$  no. of attributes. Say there is  $n+1$  no. of space, so in the total distribution table it is  $2^{n+1}-1$  (as given in Fig. 5) or  $2^8$  estimation needing due to these 3 values.

By this way, let us say we compute  $P(\text{rich}|\text{gender and hours work})$ . So we only need to square number of estimations to make. This is nothing but a

Table 1:  $2^2$  datapoints

Gender	Hours_Worked	$P(\text{rich} G, HW)$	$P(\text{poor} G, HW)$
F	<40.5	0.09	0.91
F	>40.5	0.21	0.79
M	<40.5	0.23	0.77
M	>40.5	0.38	0.62



Figure 5:  $2^{n+1}-1$  no. of cases

GENDER	HOURS_WORKED	WEALTH		
Female	v0:40.5-	poor	0.253122	
		rich	0.0245895	
	v1:40.5+	poor	0.0421768	
		rich	0.0116293	
Male	v0:40.5-	poor	0.331313	
		rich	0.0971295	
	v1:40.5+	poor	0.134106	
		rich	0.105933	

conjugate of this because :

$$P(\text{rich}|G, HW) + P(\text{poor}|G, HW) = 1 \text{ (From Table 1)}$$

As I am considering two classes – rich and poor. So  $P(\text{poor}|gender, hours) = 1 - P(\text{rich}|gender, hours)$ .

So this is redundant. So for each of the combination or values since gender takes 2 values and hour work is thresholded by 40.5, it can take 2 values, so we are getting this many number of computation (Fig 5).

So therefore if there are n attributes so the number of estimates that we require is  $2^n$  which we can easily see just by computing it. By estimate, we are not confining to only MLE or MAP, any kind of estimate. If we want to make it from n attributes, u need to compute these many :- atleast 2n number of entries to find out the classified value . Say , JDT is not visible for us, estimation is needed for us, so for that MLE and MAP is needed. But my question is if we need to determine the value classified and probability value say  $P(Y = 0|X_1 X_2 \dots X_n)$ , we need  $2^n$  estimates to confirm because each estimate will give such kind of an answer so that should be  $2^n$  estimates.

So if we get a new attribute of a problem, if we know these  $2^n$  estimates, then I can classify it whether it is  $Y=0$  or  $Y=1$ , which probability is more. Now moving towards the learning problem, what we need to know is to classify  $P(Y = 0| \text{given a new set of attributes})$ . So this is our learning problem. To know this, what we need to know is  $2^n$  estimates. For many large set of attributes we may not have all informations but our MLE or MAP estimates can give us these values. Now how many estimates do we need to judge any

new attribute falling into  $Y=0$  or  $1$  category? We need to know  $2^n$  estimates. Thus if Bayes' rule learned can help, let us see if our  $Y$  (say  $Y=0$  or  $1$  or anything more than  $2$  class) is given by my bayes' rule:

$$P(Y|X_1 X_2 \dots X_n) = \frac{P(X_1 X_2 \dots X_n|Y)P(Y)}{P(X_1 X_2 \dots X_n)} \quad (12)$$

If we want to find out the class of  $Y$  given these attributes, how many estimations to make here if instead of this formula we have used bayes' rule . This formula didn't help because we need to have all the estimates for this viz.  $P(X_1 X_2 \dots X_n|Y = 0)$  and all the estimates for this  $P(X_1 X_2 \dots X_n|Y = 1)$  because these two donot sum up to  $1$ . Also we need to have estimates of  $P(Y)$  and  $P(X_1 X_2 \dots X_n)$ . And therefore for this, we need  $2^n - 1$  estimates for  $P(X_1 X_2 \dots X_n|Y = 0)$  and  $P(X_1 X_2 \dots X_n|Y = 1)$  , even if we precompute  $P(Y)$  or  $P(X_1 X_2 \dots X_n)$ . Let us say one estimate for this, so effectively we are needing  $2(2^n - 1) + 1$ , even if thinking that, this is precomputed(all the compositions). So it is even worse than this form :

$$\begin{aligned} P(Y = 0|X_1 X_2 \dots X_n) \\ \#Estimations = 2^n \end{aligned}$$

So directly applying baye's rule doesn't help. Firstly, Problem is that we donot have joint distribution. we have a subset of the data, i.e. a very low subset of the data, as we can see if we have  $100$  attributes and most of the columns in the JDT are unknown so for that we can use MLE estimate or MAP estimate depending on the data and prior knowledge availability, but we can come up with these classes  $P(Y = 1|X_1 X_2 \dots X_n)$  for the given data. We can come up with these values for a given data. So if we need to compute all such, then directly we are computing it in  $2^n$  steps. If we use bayes' rule, then it doesn't help because it is doing worse.

Now for a new thing to classify  $X^{new}$ , we will just try to ensure what is the  $P(Y = 1|X^{new})$  or  $P(Y = 0|X^{new})$  and we classify it according to whichever is the highest based on my goal. Now to classify something new, I will say this is my training phase. So my training phase needs  $2^n$  estimates atleast in the way we are doing. So  $2^n$  estimates is huge, because for  $100$ , it's a huge and we usually do with  $50K$ ,  $1$  lakh, even  $1$  crore features. So we can see that if each estimate takes a fraction of second even, then also all these estimates computation to find out the classification boundary is impossible. What can we do about this, is the next question.

## 2.4 Slide Reference 5 given in Figure 6

Figure 6: Slide Reference - from 34:32 to 44:20

$$P(Y=1 | X_1, \dots, X_n) = ?$$

$$= \frac{P(X_1, \dots, X_n | Y) P(Y)}{P(X_1, \dots, X_n)}$$

assume  

$$P(X_1, \dots, X_n | Y)$$

$$= \prod_{i=1}^n P(X_i | Y)$$

$$P(Y | \langle X_1, \dots, X_n \rangle) = \frac{P(\langle X_1, \dots, X_n \rangle | Y) P(Y)}{P(X_1, \dots, X_n)}$$

$$= \frac{P(Y) \cdot \prod_{i=1}^n P(X_i | Y)}{P(X_1, \dots, X_n)}$$

$2n + 1$  (boxed)  
 $2^n$  (circled)  
 $n = 10^4$  (boxed)  
 est

Conditional Independent  
 def:  $X \perp\!\!\!\perp Y | Z$  if  
 $P(X | YZ) = P(X | Z)$   
 Ex:  $P(T | R \wedge L) = P(T | L)$

$$P(X_1 X_2 | Y) = P(X_1 | X_2 Y) P(X_2 | Y)$$

$$= P(X_1 | Y) P(X_2 | Y)$$

So the question lies in our want to classify by having training with this data  $P(Y = 1 | X_1 \dots X_n)$ , where we want to classify a new thing. What can we do or how can we train it instead of training the  $2^n$  estimates or calculating  $2^n$  estimates during training phase.

In statistics, there is a concept called conditional independence. We say that a random variable, is conditionally independent of  $y$  given  $z$ , if I can write this thing that  $P(X | YZ) = P(X | Z)$ . So this is a definition of conditional independence in statistics, that I say  $X$  is conditionally independent with  $Y$  given, if this thing happens:

$X$  is Conditionally Independent of  $Y | Z$  if

$$P(X | YZ) = P(X | Z) \quad (13)$$

For Example: We know that when you see a lightning, a thunder will come so raining doesn't have any conditional dependence with the thunder com-

ing if there is a lightning. So therefore  $P(\text{thunder}|\text{Rain} \wedge \text{Lightning}) = P(\text{Thunder}|\text{Lightning})$  because whenever lightning happens, thunder will automatically come. Whether it is raining or not, it doesn't matter when we see a lightning happen. This can be mathematically considered as:

$$P(T|R \wedge L) = P(T|L) \quad (14)$$

This is very interesting given this thing, otherwise it may not be conditionally independent. So with this definition, what does it help, let's try to see mathematically. We are trying to invoke this training  $P(Y = 1|X_1 \dots X_n)$  by using bayesian rule.

Now, what does this term say if we can make an assumption about the conditional independence of each of my attribute. So if we can choose my attributes during our classification problem in a conditionally independent way, how does it help.

Assume  $P(X_1 \dots X_n|Y)$ . So let's try to see for 2 attributes. so what does  $P(X_1X_2|Y)$  mean. Apply chain rule :

$$P(X_1X_2|Y) = P(X_1|X_2Y)P(X_2|Y)P(Y) \quad (15)$$

We assume that our attributes in the first term of RHS (i.e.  $X_1$  and  $X_2$ ) are conditionally independent, so:

$$P(X_1X_2|Y) = P(X_1|Y)P(X_2|Y)P(Y) \quad (16)$$

So therefore in general, we can write it as (if  $X_1, X_2, X_3 \dots$  all are conditionally independent) :

$$P(X_1 \dots X_n|Y) = \prod_{i=1}^n P(X_i|Y) \quad (17)$$

But in terms of learning, what does these assumptions bring? When we assume something in our learning framework, we only generalize our learning. May be some bad examples also come in, may be some good examples also come in. We donot know whether these assumptions, of making conditional independence, will affect our learning or will help our learning or not. So now our only goal is that whichever learning problem we are given to solve , we try to figure out its attributes so that it becomes conditionally independent with each other. Then it is a very good problem for bayesian example because now our estimation of  $P(Y| < X_1 \dots X_n >)$  boils down to the fact

that I have :

$$P(Y | \langle X_1 X_2 \dots X_n \rangle) = \frac{P(\langle X_1 X_2 \dots X_n \rangle | Y)P(Y)}{P(X_1 X_2 \dots X_n)} \quad (18)$$

Which is basically applying the conditional independence, it becomes:

$$\frac{P(\langle X_1 X_2 \dots X_n \rangle | Y)P(Y)}{P(X_1 X_2 \dots X_n)} \implies \frac{\prod_{i=1}^n P(X_i | Y)P(Y)}{P(X_1 X_2 \dots X_n)} \quad (19)$$

Because the term  $P(\langle X_1 \dots X_n \rangle | Y)$  by the conditional independence assumption has reduced to this term :

$$\prod_{i=1}^n P(X_i | Y)$$

Now how many probability estimates we need? For this Eqn. 19, n probability values for  $Y=0$  and n probability values for  $Y=1$ . So we can see that  $2n$  and  $2$  (for  $P(Y)$  for  $Y=0$  and  $Y=1$ ) i.e.  $2n+2$  number of estimates during training to figure out anything after we train and then try to classify. We are not counting the probability value of  $P(X_1 \dots X_n)$  considering it to be given previously.

So we have a meaningful computational algorithm to estimate or to train a bayes' classifier. Now our training is that, given the partial set of JDT, we use MLE estimates. For eg., let us only concentrate on MLE. Suppose let us think that we are given a table with sufficient amount of data and may not be the full exponential set of data, thus we train it and build a classifier. Given a new example, we try to see what that classifier results. If our  $P(Y = 1) > 0.5$ , then classify it as +ve, if our  $P(Y = 1) < 0.5$  then classify it as -ve.

## 2.5 Slide Reference 6 given in Figure 7

We call it Naïve Bayes' Algorithm. Though it's an assumption of the conditional independence of selection of attributes, it works like magic in our spam filtering, in our News Article Separation, in our Text classification. Though in spam filtering what we do? Suppose we get such kind of message given in

Figure 7: Slide Reference - from 44:20 to 01:00:59

Naive Bayes Algorithm → Spam filtering, News Article Separation, Text Classification

$$P(Y=y_k | x_1, \dots, x_n) = \frac{P(Y=y_k) \prod_{i=1}^n P(x_i | Y=y_k)}{\sum_j P(Y=y_j) \prod_{i=1}^n P(x_i | Y=y_j)}$$

$P(B) = P(B|\bar{A}) + P(B|A) = P(B|\bar{A})P(\bar{A}) + P(B|A)P(A)$

$x_1$	$x_2$	$x_3$	$y$
good	Red	W	+
bad	Or	B	-

$P(Y=+ | B_0 + O + W) = P(Y=+) [P(B_0|Y=+) + P(O|Y=+) + P(W|Y=+)]$

$x_1$	$x_2$	$x_3$	$y$
G	R	B	-
G	O	B	-
B	R	W	+
B	O	B	+
G	R	W	+

MLE, TE,  $x_{\text{news}}$ ,  $[B_0 O W]$ ,  $y=?$

Figure. 8. See this is spam, if you look into a spam, we can easily see that there is something like the selected words in this Figure. 8.

Hence this kind of a thing is a spam. So what are our attributes here? Our attributes here are number of these words occurring here. So the words that are occurring here will act as attributes to us and in English it has been seen that there are 50K words that we colloquially use to communicate. So at max there will be 50K attributes traditionally. In general english communication, we don't use greater than 50K words. But since now 50K attributes, so if we go by our generic principle we get  $2^{50K}$  number of prior probability estimates u require. Whereas here,  $2 \times 50K$  number of estimates u require. And so, what we are trying to do in the naive bayes' classification is :

Let us say if for eg. this mail is spam or not spam is binary classification, but news separation is not a binary classification, because it could be a sports news, financial news, political news etc. So therefore I say that there are n number of classes that we want to classify and given that there are k classes (given the value of class is  $y_k$ ) and we try to make it like this way that - given

Figure 8: Spam mail



the attributes we choose, we write it using bayes' rule with the conditional independence assumption :

$$P(Y = y_k | X_1 X_2 \dots X_n) = \frac{P(Y = y_k) \prod_{i=1}^n P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_{i=1}^n P(X_i | Y = y_j)} \quad (20)$$

Denominator is expanded just like earlier we did, when we got P(B) in the denominator which can be expanded as :

$$\begin{aligned} P(B) &= P(B \wedge \bar{A}) + P(B \wedge A) \\ &= P(B|\bar{A})P(\bar{A}) + P(B|A)P(A) \end{aligned} \quad (21)$$

So for each of the values, we just expanded the summation. So here our naïve bayes classifier will just take on each of the training examples. So suppose we have values of attributes:

$$X_1 = \{\text{good(G), bore, bad}\}$$

Table 2: Training Examples

$X_1$	$X_2$	$X_3$	Y
G	R	B	-
G	O	B	-
Bore	R	W	+
Bad	O	B	+
G	R	W	+

$X_2 = \{\text{study regularly(R), study occasionally(O)}\}$

$X_3 = \{\text{perform well in exam(W), perform bad in exam(B)}\}$

$Y = \{\text{I remember ML after this course is over(+), otherwise(-)}\}$

With this attribute set, we are given suppose this set of data given in Table 2.

So these are the data given to u. This is not a full data available. Suppose only this much is my set of training examples, so what we need to do? MLE estimates is needed because Bayes' Algorithm based on MLE estimates. So you need to first find out probabilities  $P(Y=+)$  and  $P(Y=-)$ . From this above given training dataset:

$$P(Y=+) = \frac{3}{5}$$

$$\text{where } P(Y=+) = \frac{\alpha_1}{\alpha_1 + \alpha_2}$$

$$\text{And } P(Y=-) = \frac{2}{5}$$

Then we will try to estimate what if  $X_1$  is good, given that Y is +, which is  $P(G|+)$  and  $P(G|-)$ ,  $P(Bore|+)$ , etc. How do you compute  $P(G|+)$  only from the table? So we see which entries are good with + along all the plus entries because it's a conditional probability so it will be  $\frac{1}{3^{\text{rd}}}$  among three + Y's, only the  $X_1$  of last column is good. So  $P(G|+) = \frac{1}{3}$ . So in this way we compute all of them assuming whether the subject is good, whether I read it or not, whether I remember. Usually we are assuming these three as conditionally independent ( $X_1$ ,  $X_2$  and  $X_3$ ) variables. So when we get all these informations, that is our training.

Now what will happen, suppose we get out of the box one example, say:



$$\begin{aligned}
X_1 &= B_o \\
X_2 &= O \\
X_3 &= W \\
Y &= ?
\end{aligned}$$

This is my  $X^{new}$  say, how will we classify them to get our Y. For this example, we will again try to resort to this kind of probability estimates and taking the values from here we can determine the value of Y.

Suppose this is a new attribute given to us, where we say that the 3 attributes are [  $B_o$ (Bored) ,  $O$ (Occasionally Study) ,  $W$ (i.e. still remember well)], then we want to see what will be our class Y, i.e. :

$$P(Y = +|B_o + O + W) = \frac{P(Y = +)P(B_o|Y = +)P(O|Y = +)P(W|Y = +)}{\sum_j P(Y = +) \prod_{j=1}^3 P(X_i|Y = +)} \quad (22)$$

(Using Bayes' Rule) If *value*  $\geq 0.5$ , then classify this as +, else -

So, Concise Description:

1. We will use MLE for estimation during training. We have data and we will estimate every possible parameters through MLE in the data,  $P(Y=+)$  i.e. how many '+' out of how many data we have,  $P(X_i = G|Y = +)$ , etc. These many set says me that,  $Y=+$  in that given set G is only one so  $\text{Probability}=\frac{1}{3}$ .
2. In this way, we will estimate everything - single attribute given  $Y=+$  and given  $Y=-$ . That's my training done.
3. After getting a new attribute, we need to classify now with that training data where we have these all informations of miniscule probability values. We just take that computation once again with this new attribute and if the computation results of  $P(Y=+)$  is higher than  $P(Y=-)$ , we will classify this as positive otherwise negative.

So to summarize - with the MLE and MAP estimates, now our job is to learn. So when we started our learning, we said that if JDT is not available to us, we need to use the estimates that we just learned (MLE or MAP)  $2^n$  number of times to have a meaningful answer to our classification solution. However we have seen that bayes' rule directly is not well enough because it makes us more. So the earlier one is good, but it is not good because  $2^n$  is

exponential so we thought about something called conditional independence formula. And if we can select attributes which are conditionally independent then estimates come down to  $2^n$ . Estimates are these granular values -

$$\prod_{i=1}^n P(X_i|Y)$$

Now naïve bayes just exploits those. Let us say we are using MLE, just from these we are trying to say what is the  $P(X_2 = R|Y = -)$ , what is  $P(X_2 = R|Y = +)$ , all these things it concludes. Then what will happen in such a case if it computes this way? Then my training is just estimations, that's my training. For new classification, we will again follow this conditionally independent bayes' improvisation and try to see what my class is, whichever class probability is higher, that will be my class and that is all 'Be-All and End-All of naïve bayes'. Why these algorithms are popular in Machine learning? Because there is no deterministic solution we can give so the conditional independence is an assumption over the environment and the more we bring in assumptions in our algorithm, the more generalize we do or may be more specialised thing u do. But in case of problems viz spam filtering, our gmail automatically sends them into spam folder. How does they do it? Because they have certain kinds of words which are attributes, its occurrences are the probability and based on that it uses just naïve bayes'. Same thing happens when now-a-days we see that when in gmail we type and send something, if there is no attachment but we have written something like attached or attachment, then we will be notified that we have not attached anything. How does it do that? It watches a millions of mails, computes this MLE kind of estimates, the training and then gives a go at it with a conditional dependence assumption.