Machine Learning CS60050
Instructor: Dr. Aritra Hazra

Scribe By: Thakur Rishabh Sanjeev (20CS60R38),
Akash Singh Sant (20CS60R40)

Lecture Date: 24 March 2021

1 Recap of Bayesian Networks

Lets recaps the Bayesian network where we try to learn certain probability distributions. Consider an arbitrary Bayesian network shown in figure 1. Let F denotes Flu,A denotes Allergy, S denotes Sinus which depends on Flu and Allergy. As an outcome of the Sinus one may get Headache(H) or Running Nose (N). This is typical Bayesian network that we usually see, that we have a depen-

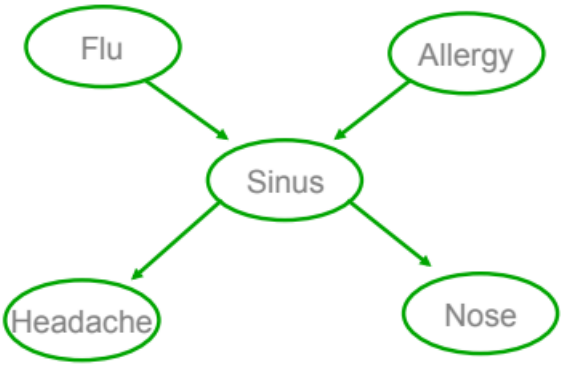

Figure 1: Bayesian Network Example

dence or we can say a directed acyclic graph whichrepresents the dependence of certain events w.r.t. to the previous events.
Now it would be absolutely great if we can have the conditional probability distribution table (CPDT) for every node. e.g for F=1, F=0 and so on. If CPDT is given then our job of finding joint Probability distribution table (JPDT) is very easy we can directly use the CPDT and use Bayesian Theorem to get the respective entry of JPDT.
e.g we want to get $P(f, a, s, h, n)$ where F=f,A=a,S=s,H=hand N=n. So we can

use the Bayesian network here and give the below equation.

$$P(f, a, s, h, n) = P(f)P(a)P(s|f, a)P(h|s)P(n|s)$$

Thus we can predict the JPDT. Thus we can conclude that if Bayesian Network + CPDT is given we can compute JPDT or viceversa.

Now from the learning prospective we can formulate the problem in four different types.

2 Type 1: Bayesian Network and Fully Observable JPDT is given

First, a JPDT is said fully observable if values of if there is no missing values is a row for any particular column. If some column values are missing then such type of JPDT is called as partially observable JPDT.

So moving forward toward our case where Baysian Net as well as a Fully observable JPDT is given. Task is to learn the CPDT for ssy Sinus(S). i.e $\theta(S = 1|F = i, A = j)$ this needs to be learn given Bayesian Network and fully observable JPDT.
So to solve this problem we will use the maximim likelihood estimation (MLE). Thus ,

$$\Theta_{s|ij} = \frac{\sum_k \delta(f_k=i, a_k=j, s_k=1)}{\sum_k \delta(f_k=i, a_k=j)}$$

where δ is the binary function if the field present and 0 otherwise.Thus,
$P[Data|\theta] = \prod_k P(f_k, a_k, s_k, h_k, n_k)$

$\frac{\delta}{\delta\theta_{s|ij}}[\log P(Data|\theta)] = \frac{\delta}{\delta\theta_{s|ij}} \sum[\log P(f_k) + \log P(a_k) + \log P(s_k|f_k, a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)]$

$\frac{\delta}{\delta\theta_{s|ij}}[\log P(Data|\theta)] = \frac{\delta}{\delta\theta_{s|ij}}[\log P(s_k|f_k, a_k)]$

3 Type 2: Bayesian Network and Partially Observable JPDT is given

As the given dataset will give us the partially observable JPDT, those missing values can be either 1 or 0 as its a binary function .
Therefore, when we will compute the MLE as we did in case 1 we will take the contribution of the column multiplied with the expected value of that column be 1 also when the expected value of the column be 0.

e.g. Consider a row where column S = 1, then for S=1 its contribution is 100 % and for S=0 its contribution is 0 %.

Now for those columns where the value of S is missing, we cannot take the contribution as it is weighted by the expected value of that particular value to be 1 or it can be weighted by the expected value of that column to be 0.

Therefore that problem is that if we wish to compute

$$\hat{\theta} = \text{argmax}_\theta \prod_k P(f_k, a_k, s_k, h_k, n_k)$$
here s_k field is partially missing

So we classify columns as fully observable set X = F,A,H,N and partially observable set Z = S
Now the problem is that we cannot do the maximum likelihood estimation because of the partially observable columns.

So for such cases we use an iterative approach:

1. Will try to fix expected value or the expectation i.e. $E_{Z|X,\theta}$ where θ is the assumed probability distribution of Z. For the first iteration we begin with some arbitrary value of θ

2. We will find the expected log likelihood to be maximize and update the θ

   $\hat{\theta} = \text{argmax}_\theta \, E_{Z|X,\theta} \log[P(X, Z|\theta)]$

In this iterative process for the missing column values we are computing the expectation and the expectation of the log likelihood.
We start by assuming an arbitrary value of the θ
Ans this value is updated in the $2^(nd)$ step. This process continues till the value of the θ converges. This process is called as the **Expexation-Maximization** Algorithm or in short EM algorithm

Formally, In setp 1, Calculate the expectation i.e. the probability of tthe unobserved variable given the observed variable and θ

$$E(P(Z|X, \theta)$$

In the setp 2, we now learn the value of $\theta_{s|ij}$. As some value of S is missing we will need to get the expectation.

$$E_{s_k=1} = \frac{P(s_k=1|f_k,a_k,h_k,n_k,\theta)}{P(s_k=1|f_k,a_k,h_k,n_k,\theta)+P(s_k=0|f_k,a_k,h_k,n_k,\theta)}$$

$$E_{s_k=1} = \frac{\theta(f_k)\theta(a_k)\theta(s_k=1|f_k,a_k)\theta(h_k|s_k)\theta(n_k|s_k)}{\theta(f_k)\theta(a_k)\theta(s_k=1|f_k,a_k)\theta(h_k|s_k)\theta(n_k|s_k)+\theta(f_k)\theta(a_k)\theta(s_k=0|f_k,a_k)\theta(h_k|s_k)\theta(n_k|s_k)}$$

Now in the $\theta_{s|ij}$ for the $s_k = 1$ for known value of the s we will take that value and for the unknown value of s we will take the value calculated from the expectation.
In 2nd step,

$$\theta_{s|ij} = \frac{\sum_k \delta(f_k=i, a_k=j) E(s_k)}{\sum_k \delta(f_k=i, a_k=j)}$$

For the known columns of s_k $E(s_k) = 1$ for other we have calculated the value of s_k i.e. some expected value , we will use that value.

The process will be repeated till the value θ converges. This will result is some local maxima. Thus to get a better value we us multiple initialization and take the best value.

For Bayesian classifier

Bayesian classifier is a special case of the Bayesian network assuming the conditional independence.

Consider a Bayesian classifier with attributes $x_i, \cdots, x_n$a dn y as the target attribute.

Thus in the E-Step,

$$E_{p(y|x_i \cdots x_n)}[y_k = 1] = \frac{P[y_k=1] \prod P(x_{i_k}|y=1)}{\sum_{j=0}^{1} P[y_k=j] \prod P(x_{i_k}|y=j)}$$

Now in M step,

$$\theta_{x_i=j|y=m} = \sum_k P(y_k = m|x_i(k) \cdots x_n(k)) \delta(s_i(k) = j) \sum_k P(y_k = m|x_i(k) \cdots x_n(k))$$

Thus EM algorithm is applicable in Bayesian classifier as well, and is used many times as many times the data is not fully observable.

4 Type 3: Bayesian Network is unknown and Fully Observable JPDT is given

This kind of problem is solved by using **Chow-Lin** Algorithm. This uses KL divergence approach and forms a tree structure for the Bayesian Network.

5 Type 4: Bayesian Network is unknown and Partially Observable JPDT is given

This is the most challenging problem and is one of the open research topic. Some approaches are there which first assumes one part and calculate other and then using the value of other part computes first part and continues this process until the values converges.

6 Mixture of Gaussian Models

It is observed that Expectation minimization poses a resemblance with the classification problem in a way that if all of the data field are unknown for outcome then its nothing but unsupervised classification problem which is tackled by applying the same expectation maximization algorithm to cluster with respect to the probabilities.
It is observed that since clustering is an unsupervised learning it mostly depends on the how the data is being originated.
For better understanding assume we have a 2D line consisting certain set of points, and these are points are drawn from two guassian models,known as mixture of guassian models.So for better/correct clustering we need to understand this mixture because any point we extract to classify because the point drawing may be dependent on both distribution.
For this mixture of guassian models the expectation minimization also applies for partial given data where a point can be a mixture of guassian model where the point (say P) is given as a vector of say n values and the probability of point P being drawn from the mixture of models is given by sum of product of Probability of selecting i^{th} guassian and the probability of drawing that point with respect to the i^{th} guassian model:

$$P(x_1, x_2.....x_N) = \sum_i P(Z = i)P(x_1, ...x_N|Z)$$

So her the unknown part is which are the guassian .Point is the observed data and "which" guassian we select to make out the point X is the unobserved part. Therefore, at each expectation step we'll try to find out the expected probability given n^{th} point ,X_n being gaussian i,j,....n :

$$P(Z(n)|X(n),\theta)$$

After getting expectation we'll try to find out the probability of this point as a mixture of guassian models.

Here to solve this we take some assumptions :
1. Assume $X =< X_1.....X_N >$, and the X_i are conditionally independent given Z.

$$P(X|Z=j)=\prod_i N(P(x_i|\mu,\sigma))$$

2. Assume 2 clusters (values of Z) and $\forall_{i,j}\sigma_{ji}=\sigma$

$$P(X)=\sum_{j=1,2} P(Z=j|\pi)\prod_i N(x_i|\mu_{ji},\sigma)$$

3.Assume $\sigma known, \pi_1....p_{i_k}, \mu_{l_i}.....\mu_{k_i}$
Observed: $X =< X1...Xn >$
Unobserved: Z
Now we'll try to maximize θ with respect to μ and π.

$$Q(\theta'|\theta)=E_{Z|x,\theta}\log[P(x,Z|\theta)]$$

$$\text{or } Q(\theta'|\theta)=E_{Z|\theta,x}[\log[P(x,Z|\theta)]+\log[P(Z|\theta)]]$$

Now we have to maximize $\mu,\pi \implies \frac{\partial}{\partial\pi}=0, \frac{\partial}{\partial\mu}=0$

$$\implies \frac{\partial}{\partial\pi}E_{Z|\theta,x}\log[P(Z|\theta)]=0$$

$$=\frac{\partial}{\partial\pi}[\log(\pi^{\sum Z(n)}(1-\pi)^{\sum(1-Z(n))}]$$

On simplifying the above equation we get :

$$=\frac{1}{N}\sum_{n=1}^{N}E[Z(n)]$$

Here we have a point P(X) and the set of gaussians $[Z_i, Z_j, Z_k.....]$ are possible candidate to generate this point, P(x) so we are trying to find which guassians are responsible for the generation of this point.So the $E[Z_i]$ represents the expectation if i^{th} gaussian actually generates this point.
Basically in the expectation step :

Calculate $P(Z(n)|(X(n),\theta)$ for each example X(n),and then use this to construct $Q(\theta'|\theta)$

And then in the maximization step we try to maximize the θ. We are trying to maximize the likelihood of this point to occur with respect to the gaussian.

$$\theta \leftarrow argmax_{\theta'}Q(\theta'|\theta)$$

And we iterate over these two steps until convergence