ECSE 6810 Assignment-4 Report

Shiuli Subhra Ghosh

---

1 Introduction

In this assignment I have performed BN parameter learning using the maximum likelihood estimation (MLE) method and maximum posterior probability (MAP) method. For MAP learning, I have also use different prior hyper-parameters as determined by $\alpha_{ijk} = \frac{\alpha}{|x_i| \times |\pi(x_i)|}$ by varying the value of α. While selecting the α value we have ensured $\alpha_{ijk} + N_{ijk} > 1$.

2 Theory and Equations

We perform parameter learning for a BN under complete data in both cases for MLE and MAP parameter learning. We assume that our data, i.e. training samples, are drawn from an underlying probability distribution P^*. The goal is to estimate parameters θ^* which best approximates underlying P^* [1].

2.1 Maximum Likelihood Estimation

Maximum Likelihood Estimation(MLE) is the most common method to find θ parameters for our goal. MLE can be formulated as

$$\theta^* = \arg\max_{\theta} LL(D:\theta) = \arg\max_{\theta} \log p(D|\theta)$$

Under the assumption that training samples are indenpendent and identically distributed, after some rearrangement we write

$$P(D|\theta) = \prod_{m=1}^{M} P(X_1(m), .., X_N(m)|\theta) = \prod_{m=1}^{M}\prod_{n=1}^{N} P(X_n(m)|\pi(X_n(m))$$

$$\rightarrow \theta^* = \arg\max_{\theta} \prod_{m=1}^{M}\prod_{n=1}^{N} P(X_n(m)|\pi(X_n(m)) = \arg\max_{\theta} \sum_{n=1}^{N} p(D|\theta_n)$$

where M is number of data samples, N is number of nodes. This means that we can estimate θ_n independently, i.e.

$$\arg\max_{\theta_n^*} = \arg\max_{\theta_n} LL(D:\theta_n)$$

Let's see discrete BN case first. Assuming node X_n can take K discrete values and its parents can have J different configurations, we denote $\theta_n = Q_{n1}, \ldots, \theta_{nJ}$. Then θ_{nj} is denoted as $\theta_{nj1}, \ldots, \theta_{njK}$ where $\sum_{k=1}^{K} \theta_{njk} = 1$. After plugging the expression for probability of node X_n given its parents, we get

$$LL(D:\theta) = \prod_{m=1}^{M} \log \prod_{j=1}^{J}\prod_{k=1}^{K} \theta_{njk}^{I(\pi(x_n(m)=j \& x_n(m)=k)}$$

If we denote the number of samples of ${}_X n = k$ when its parents have jth configuration with N_{njk}, then

$$LL(D:\theta) = N_{njk} \log \theta_{njk} + N_{njK} \log(1 - \sum_{l=1}^{K-1} \theta_{njl})$$

Setting $\frac{\partial LL(D:\theta)}{\partial \theta_{njk}} = 0$ and normalizing yields

$$\theta_{njk} = \frac{N_{njk}}{\sum_{l=1}^{K} N_{njl}}$$

for $k = 1, 2 \ldots, K-1$. We use this equation to estimate parameters of the discrete BN provided.

2.2 Maximum A Posteriori Estimation

In this section, I will discuss MAP estimation method for finding θ by maximizing the log posterior probability of θ given a dataset D.

$$\theta_{MAP} = \arg\max_{\theta} \log p(\theta|D)$$

The likelihood function $p(D|\theta)$ can be constructed for computing the posterior probability as,

$$p(\theta|D) = \alpha p(\theta) p(D|\theta)$$

Then after maximizing the Log posterior probability, we can estimate θ_{MAP}.

$$\begin{aligned} p(\theta|D) &= \frac{p(D|\theta)p(\theta)}{p(D)} \\ \log p(\theta|D) &\propto \log p(D|\theta) + \log p(\theta) \\ &= \sum_{n=1}^{N} \{LL(\theta_n : D) + \log p(\theta_n)\} \\ &= \sum_{n=1}^{N} \log p(\theta_n|D) \end{aligned} \tag{1}$$

Here, assuming θ_n are independent. The likelihood function for discrete BN follows multinomial distribution for node i and its j^{th} parent configuration.

$$\begin{aligned} L(\theta_{ij}|D) &= \frac{\sum_k N_{ijk}!}{\prod_k N_{ijk}!} \prod_k \theta_{ijk}^{N_{ijk}} \\ &= \propto \pi_k \prod_k \theta_{ijk}^{N_{ijk}} \end{aligned} \tag{2}$$

Conjugate prior for discrete BN is Dirichilet prior with hyperparameters $\alpha_{ij1}, \alpha_{ij2}, ..., \alpha_{ijK}$.

$$p(\theta_{ij}) \propto \prod_{k=1}^{K} \theta_{ijk}^{\alpha_{ijk}-1}$$

The posterior has the same form, with hyperparameters $\alpha_{ij1} + N_{ij1}, \alpha_{ij2} + N_{ij2}, ..., \alpha_{ijK} + N_{ijK}$.

$$\begin{aligned} p(\theta_{ij}|D) &\propto p(\theta_{ij}) p(D|\theta_{ij}) \\ &\propto \prod_{k=1}^{K} \theta_{ijk}^{\alpha_{ijk}-1} \prod_{k=1}^{K} \theta_{ijk}^{N_{ijk}} \\ &\propto \prod_{k=1}^{K} \theta_{ijk}^{\alpha_{ijk}+N_{ijk}-1} \end{aligned} \tag{3}$$

$$\theta^*_{ijk} = \arg\max_{ijk} \log p(\theta_{ijk}|D) \quad \text{s.t} \quad \sum_{k=1}^{K} \theta_{ijk} = 1$$

Hence,

$$\theta^*_{ijk} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_k N_{ijk} + \sum_k \alpha_{ijk} - K}$$

In practice, α_{ijk} is approximated as $\frac{\alpha}{|x_i| \times |\pi(x_i)|}$. Alpha is the scale factor or prior strength to be tuned.

3 Experiment and Results

3.1 Learned CPTs

Data set is already provided. Hence, MLE estimate is straightforward. The results for the MAP estimates are as follows,

A=1	A=2
0.411	0.589

B=1	B=2	B=3
0.287	0.4215	0.2915

Parents: A	C=1	C=2	C=3
A = 1	0.2737	0.2579	0.4684
A = 2	0.0458	0.1969	0.7572

Parents: A,B	D=1	D=2
A= 1, B=1	0.0495	0.9505
A= 2, B=1	0.2358	0.7642
A= 1, B=2	0.9292	0.0708
A= 2, B=2	0.5210	0.4790
A= 1, B=3	0.3519	0.6481
A= 2, B=3	0.2971	0.7029

Parents: C,F	E=1	E=2
C= 1, F=1	0.0136	0.9864
C= 2, F=1	0.3235	0.6765
C= 3, F=1	0.1085	0.8915
C= 1, F=2	0.3106	0.6894
C= 2, F=2	0.9208	0.0792
C= 3, F=2	0.5900	0.4100

Parents: A,D	F=1	F=2
A= 1, D=1	0.9516	0.0484
A= 2, D=1	0.4299	0.5701
A= 1, D=2	0.1134	0.8866
A=2, D=2	0.3001	0.6999

Table 1: Learned CPT's (MLE Estimate)

A=1	A=2
0.411	0.589

B=1	B=2	B=3
0.287	0.4215	0.2915

Parents: A	C=1	C=2	C=3
A = 1	0.2732	0.2575	0.4669
A = 2	0.0462	0.1968	0.7552

Parents: A,B	D=1	D=2
A= 1, B=1	0.0504	0.9473
A= 2, B=1	0.2358	0.7627
A= 1, B=2	0.9292	0.0708
A= 2, B=2	0.5204	0.4785
A= 1, B=3	0.3515	0.6464
A= 2, B=3	0.2970	0.7016

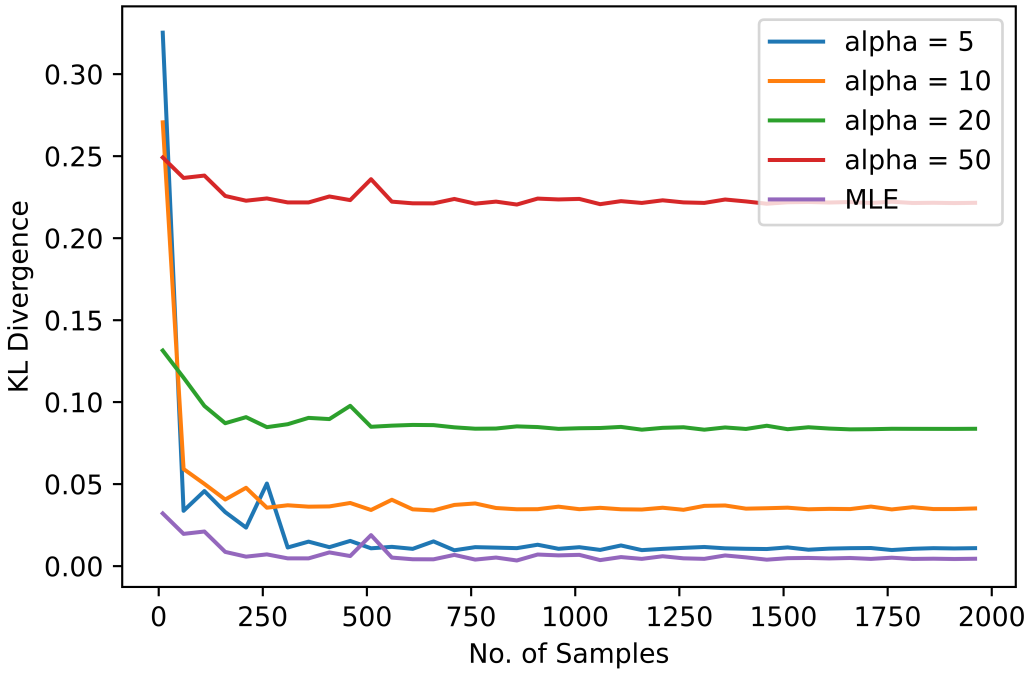
Parents: C,F	E=1	E=2
C= 1, F=1	0.0152	0.9814
C= 2, F=1	0.3232	0.6744
C= 3, F=1	0.1088	0.8902
C= 1, F=2	0.3102	0.6861
C= 2, F=2	0.9180	0.0799
C= 3, F=2	0.5896	0.4098

Parents: A,D	F=1	F=2
A= 1, D=1	0.9500	0.0489
A= 2, D=1	0.4295	0.5694
A= 1, D=2	0.1138	0.8850
A=2, D=2	0.3001	0.6993

Table 2: Learned CPT's (MAP Estimate, alpha = 5)

3.2 MLE and MAP Plots

In this section I have plotted both the MLE and MAP learning methods in terms of KL divergence between the estimated distribution and the groundtruth distribution versus the number of training samples.



As expected, with an increase in the sample size, the KL divergence is reducing significantly. alpha = 5 , seems to produce the best output. If alpha is too large, it is failing to converge even if the sample size is large enough. MLE estimate demonstrates best results as compared to MAP for different alpha values.

4 Appendix

There are two .py files in which I have implemented MLE and MAP parameter learning. main.py is the code file with MLE and MAP estimate function, and analysis.py is used to run simulation multiple times for estimating different CPTs for different sample sizes and different alpha values.

File Name	Run Time (Sec)
main.py	0.003
analysis.py	0.4519

Table 3: Runtime table

References

[1] Q. Ji. *Probabilistic Graphical Models for Computer Vision.* Elsevier Science & Technology, 2019.