KRR - Learning the parameters of a Bayesian Network

Tudor Berariu, Alexandru Sorici

December 2018

In working with Bayesian Networks there are three main challenges (presented in order of difficulty):

- *inference* (computation of different probability values, given a network structure and its parameters)
- *estimating parameters* (the tables for conditional probability distributions)
- *estimating network structure* (the number of variables and the direct links between them)

In previous lab work we dealt with exploiting network structure to assess conditional independence properties of random variables, as well as to design efficient inference algorithms that exploit the network structure (e.g. Junction Trees).

This practical work focuses on the task of *parameter estimation* given an existing network structure.

Parameter Estimation

Parameter estimation methods depend on the nature of the data being observed:

- *complete vs. incomplete*
  - complete: parameter estimation using Maximum Likelihood Estimation (MLE)
  - incomplete: approximation methods that try to optimize expected likelihood (e.g. Expectation Maximization)
- *discrete vs continous data*
  - discrete: Multinomial Distributions (frequentist approach)
  - continous: e.g. Gaussian distributions where one estimates the mean and the variance $N(\mu, \sigma)$.

We are focusing first on the simpler case of *complete observation* of discrete random variables (i.e. we are able to observe values for each variable in the modeled bayesian network).

We will be approach the problem by **Maximum Likelihood Estimation**, implemented in two ways: (a) using Lagrange multipliers to obtain a closed form optimization solution and (b) using Stochastic Gradient Descent over a parametrization of the Conditional Probability Distributions (CPD).

Formally, our problem is expressed in the following way. We assume there is a real distribution $P_r(\mathbf{X})$ we do not have access to. Instead we either have a collection of samples from that distribution, or we are able to sample from it. We are concerned with learning some parametric model $P_\theta(\mathbf{X})$ that models as good as possible the real distribution $P_r(\mathbf{X})$. We also assume the structure of a Bayesian Network that represents $P_\theta(\mathbf{X})$ to be known. We are left to learn the CPDs for each variable $X \in \mathbf{X}$. In what follows we will use $\mathbf{Y} \overset{not.}{=} pa(X)$ to denote the parents of X.

1 MLE by Closed Form Solution of Constrained Optimization

Let D be a set of observations for the variables $X_i \in \mathbf{X}$ of a network. Let us denote by $\theta_{X_i, pa(X_i)}$ the parameters corresponding to the the CPD $p(X_i|pa(X_i))$. We can further make $\theta_{X_i, pa(X_i)}$ more explicit under the form $\theta_{i,j,k} = p(X_i = k|pa(X_i) = j)$, where $k \in K_i$ is the set of possible values for variable X_i.

The joint probability of a Bayesian Network $p(\mathbf{X}) = p(X_1, X_2, ..., X_N)$ factorizes into a product of its CPD $p(X; \Theta) = \prod_{i=1}^{N} p(X_i|pa(X_i)) = \prod_{i=1}^{N} \theta_{i,pa(X_i),X_i}$, where $\Theta = (\theta_{i,j,k})_{i,j,k}$

We denote $L(\mathbf{X}; \Theta) = log(p(\mathbf{X})) = \sum_{i=1}^{N} log(\theta_{i,pa(X_i),X_i})$ the log-likelihood of an *observation* of the variables in the BN.

The log-likelihood of the whole dataset of obsevations D can then be expressed as:

$$L(D; \Theta) = \sum_{d=1}^{card(D)} L(\mathbf{X}^{(d)}; \Theta) = \sum_{d=1}^{card(D)} \sum_{i=1}^{N} log(\theta_{i,pa(X_i^{(d)}),X_i^{(d)}})$$

We can now make explicit that this sum depends on the parameters $\theta_{i,j,k}$ by observing that many terms are the same in the sum, i.e. $log(\theta_{i,pa(X_i^{(d)}),X_i^{(d)}}) = \sum_{j,k} \mathbf{I}_{pa(X_i^{(d)})=j, X_i^{(d)}=k} log(\theta_{i,j,k})$.

It is thus possible to rexpress the sum as:

$$L(D; \Theta) = \sum_{i,j,k} log(\theta_{i,j,k}) N_{i,j,k}$$

where $N_{i,j,k} = \sum_{d=1}^{card(D)} \mathbf{I}_{pa(X_i^{(d)})=j, X_i^{(d)}=k}$ *counts* the number of records in D where the value of node X_i is k and the set of values of its parent nodes $pa(X_i)$ is j.

To maximize the likelihood of the datast we need to solve *a set* of *constrained optimization problems* of the form:

$$max \sum_{k} log(\theta_{i,j,k}) N_{i,j,k} \text{ subject to } \sum_{k} \theta_{i,j,k} = 1$$

Each constrained optimization problem is solved with a *Lagrange Multiplier* method, which yields the closed form solution:

$$\hat{\theta}_{i,j,k} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}}$$

which counts the *empirical frequency* of $X_i = k$ when $pa(X_i) = j$.

Laplace smoothing
The above frequentist method works well when there are enough examples of each possible combination. However, if a particular value combination is never observed in the dataset, the value of the corresponding parameter (by counting) will be 0.

This is why we introduce *Laplace smoothing*:

$$\hat{\theta}_{i,j,k} = \frac{N_{i,j,k} + \alpha}{\sum_k N_{i,j,k} + |K| \cdot \alpha}$$

Note: in the case of binary variables α can be set to 1, while $|K| = 2$.

2 MLE by Stochastic Gradient Descent

Another way to ensure that we have good representations of the probabilities is learning the set of parameters $\theta_{i,pa(X_i),X_i}$ such that $p(X_i|pa(X_i)) = \sigma\left(\theta_{i,pa(X_i),X_i}\right)$ where $\sigma(x) = (1 + e^{-x})^{-1}$ is the *sigmoid* function. The sigmoid function is continous and differentiable in $\mathbb{R}$ and has a nice derivative $\sigma'(x) = \sigma(x)(1 - \sigma(x))$.

Learning the model of a distribution. A common metric between distributions is the KL divergence. We therefore might use it to perform stochastic optimization of the parameters θ in order to increase the cross-entropy between the two distributions.

$$\theta^* = \underset{\theta}{\text{argmin}}\, KL(p_r \,||\, p_\theta) = \underset{\theta}{\text{argmin}}\, \mathbb{E}_{\mathbf{X}\sim p_r}\left[\log\left(\frac{p_r(\mathbf{X})}{p_\theta(\mathbf{X})}\right)\right] = \underset{\theta}{\text{argmin}}\, -\mathbb{E}_{\mathbf{X}\sim p_r}\left[\log\left(p_\theta(\mathbf{X})\right)\right] \tag{1}$$

Stochastic Optimization. We start with some random parameters $\theta^{(0)}$ and for each observed sample $\mathbf{x}^{(t)}$ we move in the direction opposed to the gradient in order to minimize our cost function.

$$KL\left(p_r \mid\mid p_\theta\right) \approx -\sum_{\mathbf{X}\sim p_r} \log p_\theta\left(\mathbf{X}\right) \approx -\log p_\theta\left(\mathbf{X}^{(t)}\right) \tag{2}$$

$$\theta^{(t+1)}_{i,pa(X_i),X_i} \leftarrow \theta^{(t)}_{i,pa(X_i),X_i} + \eta \cdot \nabla_{\theta_{i,pa(X_i),X_i}} \log p_\theta\left(\mathbf{X}=\mathbf{x}^{(t)}\right) \tag{3}$$

Since the joint probability $p_\theta\left(\mathbf{X}\right)$ is just a product of all CPDs, its logarithm becomes a sum.

$$\log P_\theta\left(\mathbf{X}\right) = \sum_{X_i\in\mathbf{X}} \log P_\theta\left(X_i \mid pa\left(X_i\right)\right) \tag{4}$$

Case of binary random variables (events)

For some specific parameter $\theta_{i,pa(X_i),X_i}$:

$$\nabla_{\theta_{i,pa(X_i),X_i}} \log P_\theta\left(\mathbf{X}\right) = \nabla_{\theta_{i,pa(X_i),X_i}} \log P_\theta\left(X_i|pa(X_i)\right) = \begin{cases} \frac{\sigma'\left(\theta_{i,pa(X_i),X_i}\right)}{\sigma\left(\theta_{i,pa(X_i),X_i}\right)} = 1-\sigma\left(\theta_{i,pa(X_i),X_i}\right) & \text{if } X_i=1 \\ \frac{-\sigma'\left(\theta_{i,pa(X_i),X_i}\right)}{1-\sigma\left(\theta_{i,pa(X_i),X_i}\right)} = -\sigma\left(\theta_{i,pa(X_i),X_i}\right) & \text{if } X_i=0 \end{cases} = X_i - \sigma\left(\theta_{i,pa(X_i),X_i}\right) \tag{5}$$