Lab 9 – Parameter estimation in Bayesian Networks with known structure

In a previous laboratory we explained that there are 3 main challenges in working with Bayesian Networks:

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

In the previous laboratory we dealt with the Variable Elimination algorithm as an example inference method.

This laboratory 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) or Maximum Aposteriori (MAP)
  - incomplete: non-linear parametric optimization (e.g. gradient descent, Expectation Maximization)
- discrete vs continuous data
  - discrete: Multinomial Distributions (frequentist approach)
  - continuous: e.g. Gaussian distributions where one estimates the *mean* and the *variance* $N(\mu, \sigma)$

In this lab we are considering the simpler case of complete data observation (i.e. we *are able to* observe values *for each* variable in the modeled bayesian network).

In this case, the simplest criterion to estimate the value of each parameter (i.e. CPD table) is that of Maximum Likelihood Estimation (MLE).

Let *D* be a set of observations for the variables $v \in V$ of a network. Let us denote by $\theta_{v,\, pa(v)}$ the parameter corresponding to variable *v* (i.e. the conditional probability distribution $p(v|pa(v))$).

The MLE estimate is then written as: $\max_{\Theta} p(D;\Theta) = \max_{\theta_{v,pa(v)}} \prod_{d \in D} \prod_{v \in V} p(v^d | pa(v^d), \theta_{v,pa(v)})$

The MLE is decomposable such that each $\theta_{v,\, pa(v)}$ is computable independantly under the form:

$ML(\theta_{v,pa(v)}) = \frac{N_{v,\mathbf{pa}(v)}}{\sum_{v} N_{v,\,\mathbf{pa}(v)}}$ *where* $N_{v,\,\mathbf{pa}(v)}$ are counts of the co-appearance of values for *v* and the parents of *v* in the network.

To put it into perspective, an example computation of a probability estimation by counting is:

$$P(A=a|B=b,C=c)=\frac{N(A=a,B=b,C=c)}{\sum_{a\in A} N(A=a,B=b,C=c)}=\frac{N(A=a,B=b,C=c)}{N(B=b,C=c)}$$

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:*

$$P(A=a|B=b,C=c)=\frac{N(A=a,B=b,C=c)+\alpha}{\sum_{a\in A} N(A=a,B=b,C=c)+|A|\cdot\alpha}$$

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

General Algorithm for parameter estimation

Considering what has been discussed above, the general algorithm for ML estimation of bayesian network parameters is:

- **Phase 1: Count**
  - For each $example^i$ in dataset
    - For each variable *v in V*
      - increment $count_{\theta_{v,\,pa(v)}}(example^i_{pa(v)}, v^i)$
- **Phase 2: Normalize**
  - For each variable *v* and local assignment (i.e. possible value combinations of the parent variables)
    - set $\theta_{v,\,pa(v)}=p(v|pa(v))\propto count_{\theta_{v,\,pa(v)}}(pa(v),v)$ using Laplace smoothing