Name: Parashara Ramesh	**Student Id**: A0285647M	**NUSNET Id**: E1216292

Derivation

$$L=\sum_{n=1}^{N}\{\frac{-log(2\pi\sigma_u^2))}{2}-\frac{1}{2\sigma_u^2}*[x_{un}-(w_{u0}+\Sigma_{c\epsilon x_{\pi_u}}w_{uc}x_{ucn})]^2\}$$ (1.from the slides)

$$\frac{\partial L}{\partial w_{u0}}=\sum_{n=1}^{N}\left(x_{u,n}-\left(w_{u1}x_{u1,n}+\ \dots+\ w_{uC}x_{uC,n}+w_{u0}\right)\right)=0$$
$$\frac{\partial L}{\partial w_{u1}}=\sum_{n=1}^{N}\left(x_{u,n}-\left(w_{u1}x_{u1,n}+\ \dots+\ w_{uC}x_{uC,n}+w_{u0}\right)\right)x_{u1,n}=0$$
$$\vdots$$
$$\frac{\partial L}{\partial w_{uC}}=\sum_{n=1}^{N}\left(x_{u,n}-\left(w_{u1}x_{u1,n}+\ \dots+\ w_{uC}x_{uC,n}+w_{u0}\right)\right)x_{uC,n}=0$$

(2.from the slides, use C+1 linear equations to solve for each of the C weights and the bias)

$$\mu_u=w_{u0}+\Sigma_{c\epsilon x_{\pi_u}}w_{uc}x_{ucn}$$ (3.calling this part as the mean for the uth RV)

$$L=-0.5Nlog(2\pi)-0.5Nlog(\sigma_u^2)-\frac{0.5}{\sigma_u^2}\sum_{n=1}^{N}[x_{un}-\mu_u]^2$$ (4.Equation now changes like this)

$$\frac{\partial L}{\partial\sigma_u^2}=\frac{-N}{2\sigma_u^2}+\frac{1}{2\sigma_u^4}\sum_{n=1}^{N}[x_{un}-\mu_u]^2=0$$ (5.Taking partial derivative wrt variance and equating to 0 to find argmax)

$$\sigma_u^2=\frac{\sum_{n=1}^{N}[x_{un}-\mu_u]^2}{N}$$ (6. Re arranging terms we get this for optimal variance)

Implementation

1. **_get_learned_parameters**
- For each node in the given DGM we are trying to learn the following parameters: 1 bias term, C weight terms from its C parents, 1 variance term
- We first try to learn the C+1 weights by calling the learn_node_parameter_w and then use those weights to compute the mean μ_u using formula #3 above
- Using this mean μ_u for the x_u we can now compute variance using the formula #6 in the learn_node_parameter_var function

2. **_learn_node_parameter_w**
- We have to solve c+1 equations shown in figure #2. To do this we formulate that as Ax = b which can then be fed into np.linalg.solve(A,b) to get the weights
- We take A as matrix of values corresponding to each w_{uc} and b as the values of the constants.
- A = [equation_0, equation_1….,equation_c], b = [b0,b1,...bc]
- Each equation_i = [Vu0, Vu1,...Vuc] (where Vui is the value of the Wuc variable which can be found by using np by summing specific nodes across different axis)
- Each bi can again be computed using numpy by summing across all observations based on the formula shown in figure 2
- Refer to code comments for more details on how each Vuc , bi is computed

3. **_learn_node_parameter_var**
- Using this mean μ_u for the x_u we can now compute variance using the formula #6 in this function as per formula #6 (i.e. summing across all n observations of xu)