Assignment 2

Includes methods of Directed Graphical Models, Dynamic programming,
Variational Inference and Expectation Maximisation

DD2434 Advanced Machine Learning

Filip Bergentoft, bergento@kth.se

2.1 Dependencies in a Directed Graphical Model

Question 2.1.1: In the graphical model of Figure 1, is $\mu_{r,c} \perp \mu_{r,c+1}$?

Answer: No, $\mu_{r,c}$ and $\mu_{r,c+1}$ are dependent.

Question 2.1.2: In the graphical model of Figure 1,
is $X_{r,c} \perp X_{r,c+1} | \{\mu_{r,c}, \mu_{r,c+1}\}$?

Answer: Yes, $X_{r,c}$ and $X_{r,c+1}$ are d-separated and thus independent.

Question 2.1.3: Give a minimal set of variables A such that $X_{r,c} \perp \mu_0 | A$ in Figure 1

Answer: The minimal set of variables are given by$A = \{\mu_{r,c}, \mu_{r-1,c}, \mu_{r+1,c}, \mu_{r,c-1}, \mu_{r,c+1}\}$

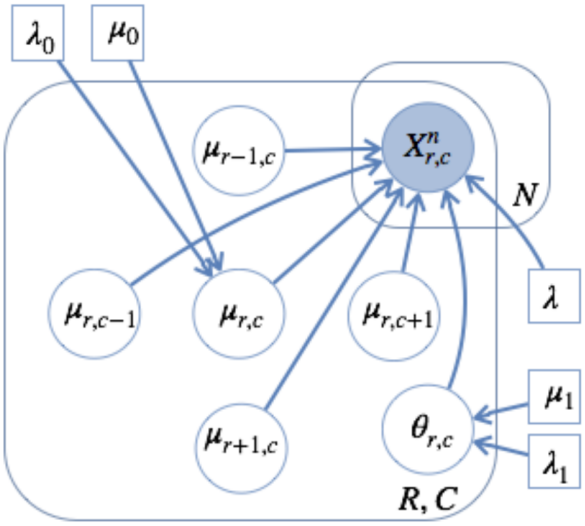

Figure 1

Question 2.1.4: In Figure 2, is $Z \perp X | C$ where $Z = \{Z^n_m : n \in [N], m \in [M]\}$, $X = \{X^n_m : n \in [N], m \in [M]\}$ and $C^n = \{C^n : n \in [N]\}$?

Answer: No, Z and X are not d-separated given C and are thus dependent.

Question 2.1.5: In Figure 2, is $A \perp e | B$ where $A = \{A^k_{i,j} : k \in [K], i, j \in [I]\}$, $e = \{e^k_{i,r} : k \in [K], i \in [I], r \in [R]\}$ and $B = \{Z^n_m : m \in [M], m \text{ odd}\} \cup \{X^n_m : m \in [M], m \text{ even}\}$?

Answer: No, A and e are not d-separated given B and are thus dependent.

Question 2.1.6: In Figure 2, give a minimal set of variables B such that $A \perp X|B$ where $A = \{A^k_{i,j} : k \in [K], i, j \in [I]\}$ and $X = \{X^n_m : n \in [N], m \in [M]\}$

Answer: The minimal set of variables are given by

$$B = \{Z^n_m : n \in [N], m \in [M]\} \cup \{C^n : n \in [N]\}$$

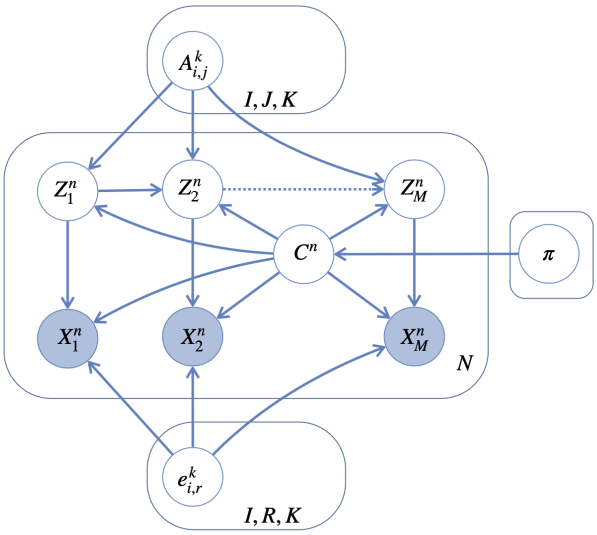

Figure 2

2.2 Likelihood of a Tree Graphical Model

Question 2.2.7: Implement a dynamic programming algorithm that, for a given T, Θ and β computes $p(\beta|T, \Theta)$

T is a binary tree with a vertex set $V(T)$ and a leaf set $L(T)$. For each vertex $v \in V(T)$ there is an associated random variable $X_v \in [K]$ with a corresponding CPD $\theta_v = p(X_v|x_{pa(v)})$ which is a categorical distribution. β is defined as the set of values of all leafs in T such that $\beta = \{x_l : l \in L(T)\}$.

In order to compute $p(\beta|T, \Theta)$ we need to find an expression that can be used for dynamic programming, I.E. splitting up the full problem into smaller subproblems. By looking at the definition of s in equation (1)

$$s(u, i) = p(X_{Observed \cap \downarrow u}|X_u = i) \tag{1}$$

and letting the root node of the tree being denoted by r, one can use that if u is chosen as the root r we get the following expression

$$s(r,i) = p(X_{Observed \cap \downarrow r} | X_r = i) = \left\{ X_{Observed \cap \downarrow r} = \beta \right\} = p(\beta | X_r = i, T, \Theta)$$

We can then marginalise this using Bayes' theorem in the following manner

$$p(\beta|T,\Theta) = \sum_i p(\beta, X_r = i|T,\Theta) = \sum_i p(\beta|X_r = i, T, \Theta) p(X_r = i) \tag{2}$$

$$= \sum_i s(r,i) p(X_r = i) \tag{3}$$

Using that T is a binary tree and thus if v, w are children to a node u then

$$\begin{aligned} s(u,i) &= p(X_{Observed \cap \downarrow u} | X_u = i) \\ &= p(X_{Observed \cap \downarrow v} | X_v = i) p(X_{Observed \cap \downarrow w} | X_w = i) \\ &= \left(\sum_j s(v,j) p(X_v = j | x_u = i) \right) \left(\sum_j s(w,j) p(X_w = j | x_w = i) \right) \end{aligned} \tag{4}$$

A special case is when the node u is a leaf node, then the following holds

$$s(u,i) = \begin{cases} 1, & X_u = i \\ 0, & otherwise \end{cases} \tag{5}$$

Equation (3) can then be computed using dynamic programming by starting at the leaf nodes using equation (5) and then traversing up the nodes in the tree to the root using equation (4) one level at a time and storing the achieved probabilities s along the way.

Question 2.2.8: Apply your algorithm to the graphical model and data provided separately

The following likelihoods were achieved when applying my implementation of the dynamic programming algorithm on the given trees.

Tree sample:	0	1	2	3	4
Small tree	0.016	0.015	0.011	0.007	0.041
Medium tree	$4.336 \cdot 10^{-18}$	$3.094 \cdot 10^{-20}$	$1.050 \cdot 10^{-16}$	$6.585 \cdot 10^{-16}$	$1.488 \cdot 10^{-18}$
Large tree	$3.288 \cdot 10^{-69}$	$1.109 \cdot 10^{-66}$	$2.522 \cdot 10^{-68}$	$1.242 \cdot 10^{-66}$	$3.535 \cdot 10^{-69}$

2.3 Simple Variational Inference

Question 2.3.9: Implement the VI algorithm for the variational distribution in Equation (10.24) in Bishop.

The following problem is stated in Bishop. Given a data set $D = \{x_1, ..., x_N\}$ of observed values x that are drawn from a Gaussian distribution we want to infer a posterior distribution using variational inference. The likelihood function is given by

$$p(D|\mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{ -\frac{\tau}{2} \sum_{n=1}^{N} (x_n - \mu)^2 \right\}$$

The conjugate prior distributions for μ and τ are given by

$$\begin{aligned} p(\mu|\tau) &= \mathcal{N}\big(\mu; \mu_0, (\lambda_0 \tau)^{-1}\big) \\ p(\tau) &= \Gamma(\tau; a_0, b_0) \end{aligned}$$

where $\mu_0, \lambda_0, a_0, b_0$ are hyperparameters. The factorised variational approximation of the posterior is given by

$$q(\mu, \tau) = q_\mu(\mu) q_\tau(\tau)$$

qqq: add what kind of distributions they become

which gives the following update formulas

$$\mu_N = \frac{\lambda_0 \mu_0 + N\overline{x}}{\lambda_0 + N} \tag{6}$$

$$\lambda_N = (\lambda_0 + N)\mathbb{E}_\tau[\tau] = (\lambda_0 + N)\frac{a_N}{b_N} \tag{7}$$

$$a_N = a_0 + \frac{N}{2} \tag{8}$$

$$\begin{aligned} b_N &= b_0 + \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n - \mu)^2 + \lambda_0(\mu - \mu_0)^2\right] \\ &= b_0 + \frac{1}{2}\left[\sum_{n=1}^{N} x_n^2 + N\mathbb{E}_\mu[\mu^2] - 2N\overline{x}\mathbb{E}_\mu[\mu] + \lambda_0(\mathbb{E}_\mu[\mu^2] + \mu_0^2 - 2\mu_0\mathbb{E}_\mu[\mu])\right] \\ &= \left\{\mathbb{E}_\mu[\mu] = \mu_N,\, \mathbb{E}_\mu[\mu^2] = V_\mu(\mu) + \mathbb{E}_\mu[\mu]^2 = \lambda^{-1} + \mu_N^2\right\} \\ &= b_0 + \frac{1}{2}\left[\sum_{n=1}^{N} x_n^2 + (N + \lambda_0)(\lambda^{-1} + \mu_N^2) - 2\mu_N(N\overline{x} + \mu_0\lambda_0) + \lambda_0\mu_0^2\right] \end{aligned} \tag{9}$$

The VI algorithm is then implemented by updating the parameters in the order of the equations given above.

Question 2.3.10: What is the exact posterior?

The exact posterior can be computed using the likelihood of the data and the priors. Since the two priors for μ and τ are conjugate priors to the likelihood we know that the posterior will be on the form of a Gaussian-Gamma distribution.

$$
\begin{aligned}
p(\mu,\tau|D) &\propto p(D|\mu,\tau)p(\mu|\tau)p(\tau) \\
&\propto \tau^{\frac{N}{2}}\tau^{\frac{1}{2}}\tau^{a_0-1}e^{-b_0\tau}\exp\Bigg\{-\frac{\tau}{2}\Bigg[\sum_{n=1}^{N}(x_n-\mu)^2+\lambda_0(\mu-\mu_0)^2\Bigg]\Bigg\} \\
&\propto \tau^{\frac{N}{2}}\tau^{\frac{1}{2}}\tau^{a_0-1}e^{-b_0\tau}\exp\Bigg\{-\frac{1}{2}\tau(N+\lambda_0)\Bigg(\mu-\frac{N\overline{x}+\lambda_0\mu_0}{N\lambda_0}\Bigg)^2\Bigg\}\times \\
&\quad \exp\Bigg\{-\frac{\tau}{2}\Bigg[-\frac{(N\overline{x}+\lambda_0\mu_0)^2}{N+\lambda_0}+\sum_{n=1}^{N}x_n^2+\lambda_0\mu_0^2\Bigg]\Bigg\} \\
&\propto \tau^{\frac{N}{2}}\tau^{a_0-1}e^{-b_0\tau}\mathcal{N}\Bigg(\mu;\frac{N\overline{x}+\lambda_0\mu_0}{N\lambda_0},\frac{1}{\tau(N+\lambda_0)}\Bigg)\times \\
&\quad \exp\Bigg\{-\frac{\tau}{2}\Bigg[-\frac{(N\overline{x}+\lambda_0\mu_0)^2}{N+\lambda_0}+\sum_{n=1}^{N}x_n^2+\lambda_0\mu_0^2\Bigg]\Bigg\} \\
&\propto \mathcal{N}\Bigg(\mu;\frac{N\overline{x}+\lambda_0\mu_0}{N\lambda_0},\frac{1}{\tau(N+\lambda_0)}\Bigg)\times \\
&\quad \Gamma\Bigg(\tau;a_0+\frac{N}{2},b_0+\frac{1}{2}\Big(\sum_{n=1}^{N}x_n^2+\lambda_0\mu_0^2-\frac{(N\overline{x}+\lambda_0\mu_0)^2}{N+\lambda_0}\Big)\Bigg) \qquad (10)
\end{aligned}
$$

The exact posterior is thus given by the expression in equation (10) where $\Gamma(\tau;\alpha,\beta)$ denotes the gamma distribution with shape α and rate β as parameters.

Question 2.3.11: Compare the inferred variational distribution with the exact posterior. Run the inference on data points drawn from iid Gaussians. Do this for three interesting cases and visualize the results. Describe the differences.

The following plots were obtained when comparing the inferred posterior to the real posterior.

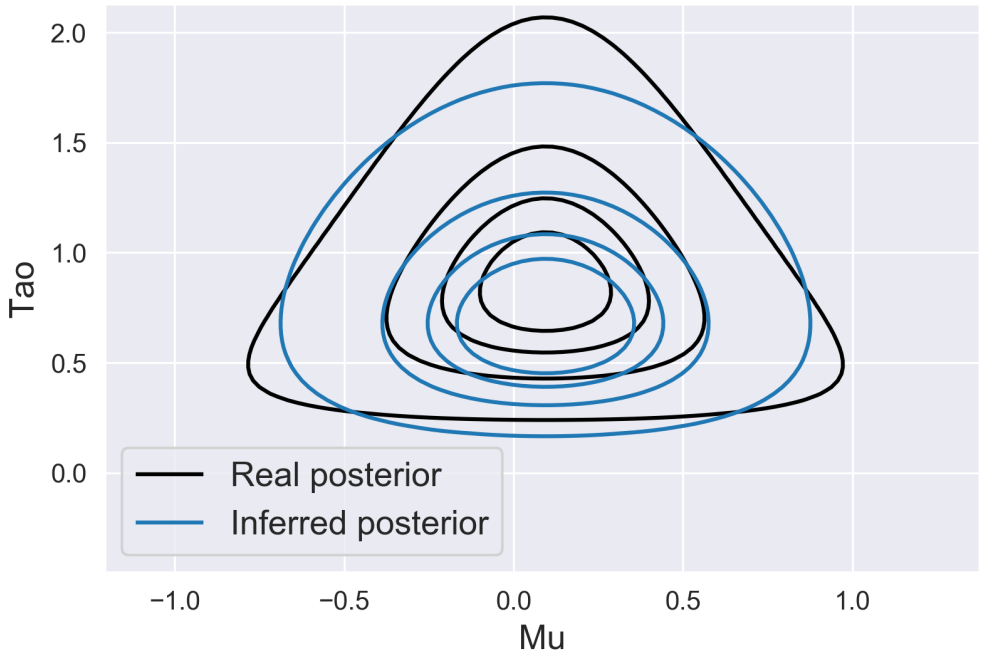

Figure 3: 10 data points drawn from $\mathcal{N}(0, 1)$, all hyperparameters set to 0

In Figure 3 one can see a similar behaviour to the book as the hyperparameters are set the same. The real target for the parameters are $(\mu, \tau) = (0, 1)$ which the real posterior seem to be a bit closer to but considering the low number of data points both distributions perform quite well. When comparing the shape of the the two posteriors one can see that the real posterior is closer to a triangle than the inferred posterior but in general the inferred posterior is close to the real posterior. Note that the uncertainty is about the same for both μ and τ.

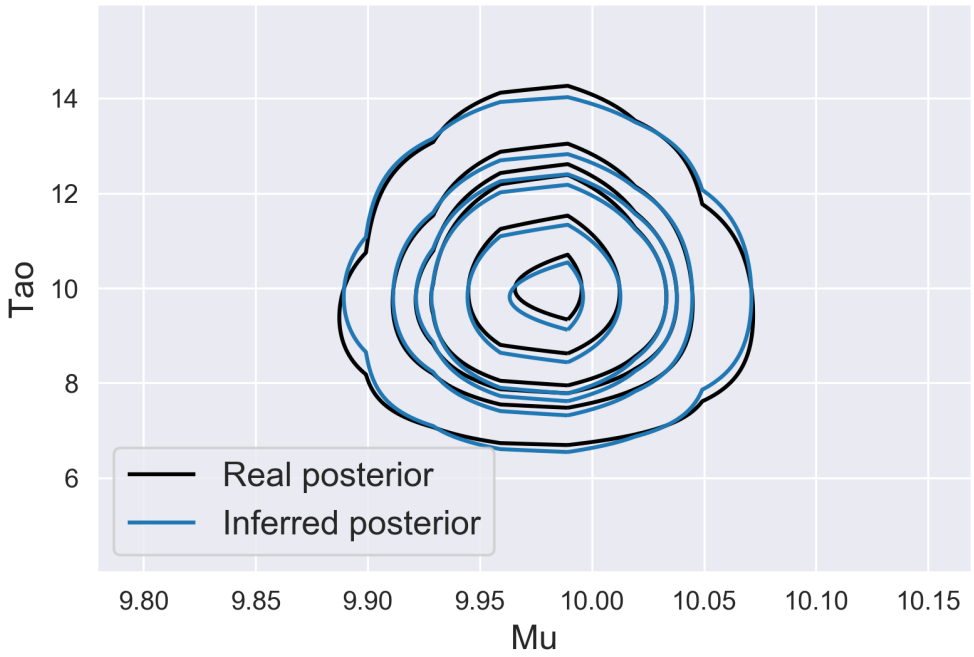

Figure 4: 100 data points drawn from $\mathcal{N}(10, \frac{1}{10})$, all hyperparameters set to 0

In Figure 4 the number of data points has increased to 100 and the distribution they are generated from has changed to $\mathcal{N}(10, \frac{1}{10})$. Noticing that the axis scales are different here the conclusions from Figure 3 still holds except that the uncertainty in τ is almost two orders of magnitude larger than the uncertainty for μ.

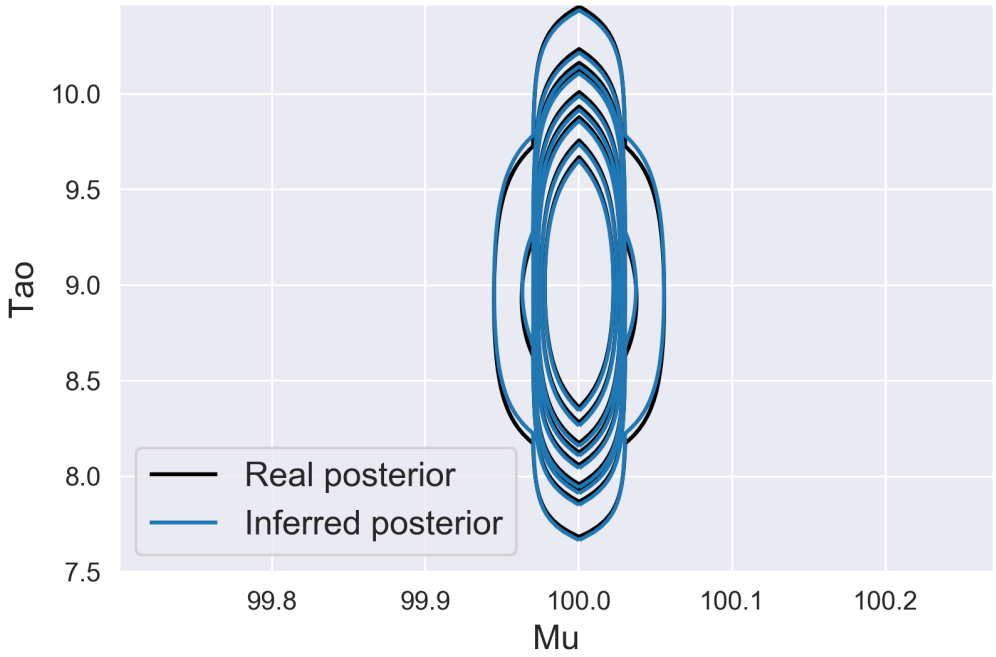

Figure 5: 1000 data points drawn from $\mathcal{N}(100, \frac{1}{100})$, all hyperparameters set to 0

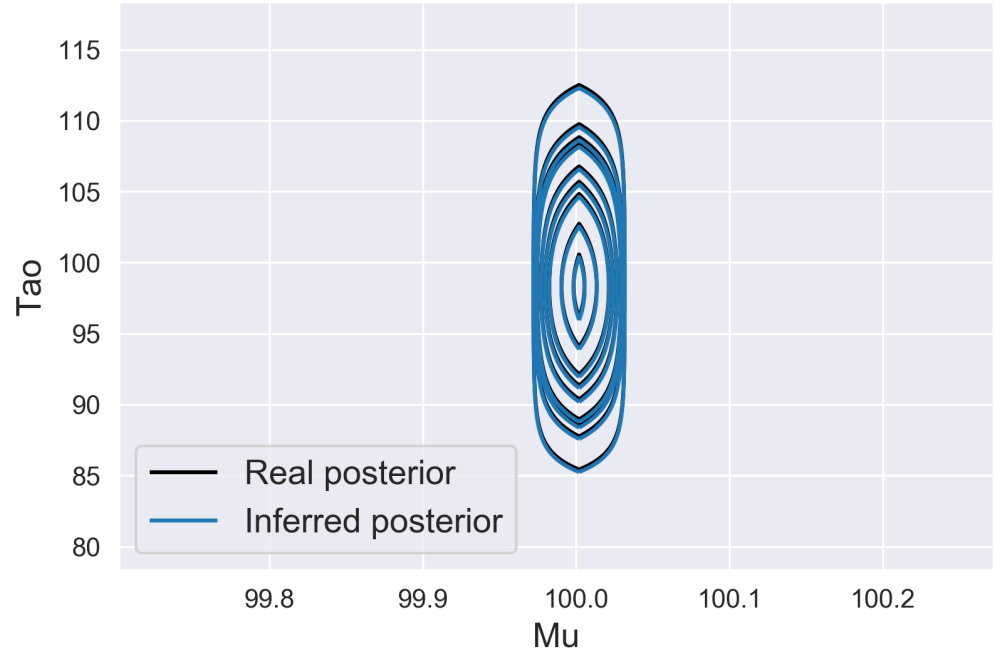

Figure 6: 1000 data points drawn from $\mathcal{N}(100, \frac{1}{100})$, all hyperparameters set to 0 except $\mu_0 = 100$

In Figures 5 and 6 the effect of the hyperparameter μ_0 is enlightened. In both examples 1000 data points are drawn from $\mathcal{N}(100, \frac{1}{100})$. In Figure 5 when $\mu_0 = 0$ both posteriors are centred around $\tau = 9$, very far from the real value of $\tau = 100$ whilst in Figure 6 both posteriors are centred around $\tau = 98$, very close to the real value. This shows how important the hyperparameters can be if one does not have a sufficient amount of data. In addition to this conclusion one can also conclude that the inferred posterior is very similar to the real posterior in this example.

2.4 Mixture of trees with observable variables

Question 2.4.12: Implement this EM algorithm.

The EM algorithm with sieving was was implemented in the following manner using the given Tree package.

Algorithm 1: EM algorithm

Input: Data samples
Output: Tree mixture

1 Compute a distance matrix of the data: $D \leftarrow \text{weighted\_distance}(Y)$
2 Compute a similarity matrix from D: $S \leftarrow \text{similarity\_matrix}(D)$
3 Compute Eigen-decomposition of S: $[D, Q] \leftarrow \text{Eig}(S)$
4 Order D in descending order of eigenvalues magnitude and Q correspondingly
5 Ensure elements of D and Q are real
6 Compute embedding: $X \leftarrow I_{2\times 101} D Q^T$

Question 2.4.13: Apply your algorithm to the provided data and show how well you reconstruct the mixtures. First, compare the real and inferred trees with the unweighted Robinson-Foulds (aka symmetric difference) metric. Do the trees have similar structure? Then, compare the likelihoods of real and inferred mixtures.

Question 2.4.14: Simulate new tree mixtures with different number of nodes, samples and clusters. Try to find some interesting cases. Analyse your results as in the previous question.