Bayesian methods

Luis Antonio Ortega Andrés

May 21, 2021

This document contains notes and exercises done during the course of Bayesian methods. All the following theory is explained from a non-variational point of view, this includes the EM algorithm.

1 Introduction to Bayesian networks

Given a set of variables $\boldsymbol{X} = (X_1, \dots, X_N)$, *Bayesian networks* might be defined either as a probability distribution of a certain form or a DAG whose nodes represent these variables and links an independence constraint. Both ideas are present in the following definition.

Definition. A *belief network or Bayesian network* is a pair (G, P) formed by a DAG G and joint probability distribution P such that there is a correspondence between variables and nodes verifying:

$$P(x_1, \dots, x_N) = \prod_{n=1}^{N} P(x_n \mid pa(x_n)).$$

Remark. A Bayesian network might be given as a distribution from which the DAG can be constructed or a DAG which represents the distribution. For example in Figure 1, given the DAG one could easily define the joint distribution and conversely.

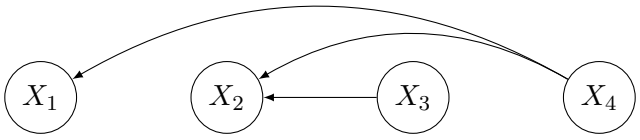


Figure 1: Bayesian Network factorizing $P(x_1, x_2, x_3, x_4) = P(x_1 \mid x_4)P(x_2 \mid x_3, x_4)P(x_3)P(x_4)$.

Any probability distribution can be written as a Bayesian network, even though it may end up being a fully-connected "cascade"[1] DAG, which means that each variable X_n is a parent of any X_m with $m > n$. This is because any distribution satisfies:

$$P(x_1, \dots, x_N) = P(x_1) \prod_{n=2}^{N} P(x_n \mid x_1, \dots, x_{n-1})$$

[1] This term comes from the visual structure of the graph.

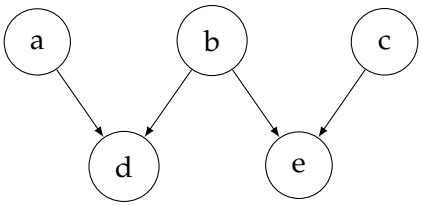


Figure 2: D-separation example

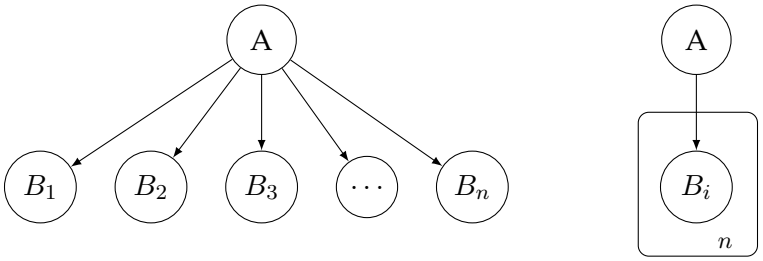


Figure 3: Plate notation example. Standard notation on the left and plate on the right.

1.1 D-separation and D-connection

There are two central concepts that determine conditional independence in any Bayesian network, these are *d-connection* and *d-separation*.

Definition. Let G be a DAG where $\boldsymbol{X}, \boldsymbol{Y}$ and $\boldsymbol{Z}$ are disjoint sets of vertices. We say that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *d-connected* by $\boldsymbol{Z}$ if and only if there exists an undirected path U from any vertex in $\boldsymbol{X}$ to any vertex in $\boldsymbol{Y}$ such that:

- For any collider C, itself or any it's descendants is in $\boldsymbol{Z}$.
- No non-collider on U is on $\boldsymbol{Z}$.

That is, there exists a path where $\boldsymbol{Z}$ contains all of its colliders and their descendants, and no other node from the path.

Definition. Let G be a DAG where $\boldsymbol{X}, \boldsymbol{Y}$ and $\boldsymbol{Z}$ are disjoint sets of vertices. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are *d-separated* by $\boldsymbol{Z}$ if and only if they are not d-connected by $\boldsymbol{Z}$ in G. That is, for any undirected path from $\boldsymbol{X}$ to $\boldsymbol{Y}$, either there is a collider or a descendant of it not in $\boldsymbol{Z}$ or there is a non-collider in $\boldsymbol{Z}$.

For example, in Figure 2, d d-separates a and c (e is a collider in the path that is not in $\{d\}$), and $\{d, e\}$ d-connect them.

Theorem (Spirtes *et al.* (2000), Th 3.3). Let G be a DAG where $\boldsymbol{X}, \boldsymbol{Y}$ and $\boldsymbol{Z}$ are disjoint sets of vertices. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ if and only if they are independent conditional on $\boldsymbol{Z}$ in all probability distributions that G may represent.

In cases where the Bayesian networks contains i.i.d nodes that are essentially the same but repeated a number of times, the *plate notation* is commonly used to represent this nodes in a compacted format (Figure 3).

2 Maximum likelihood learning in Bayesian networks

Given a parameterized model, that is, a set of variables $X_1, \ldots, X_N$ given that their distribution is governed by a set of parameters Θ. Maximum likelihood estimation consists on, given a set of observations $\mathcal{D}$, find

$$\Theta^{ML} = argmax_{\Theta}\mathcal{L}(\Theta, \mathcal{D}) = argmax_{\Theta}P(\mathcal{D} \mid \Theta).$$

In simple cases, this can be approached deriving the corresponding expression and finding those minimals analytically. In this section we are reviewing this exact approach in Bayesian networks.

Consider a scenario where a disease D and two habits A and B are being studied. Consider the following i.i.d variables $\{A_1, \ldots, A_N\}$, $\{B_1, \ldots, B_N\}$ and $\{D_1, \ldots, D_N\}$ governed by the parameters θ_A, θ_B and $\boldsymbol{\theta}_D$ as shown in figure 4. Let $N = 7$ be the number of observations of the variables as shown in Table 5 and $\boldsymbol{x} = \{(a_n, b_n, d_n),\ n = 1, \ldots, N\}$ the set of observations.

All the variables are binary satisfying

$$P(A_n = 1 \mid \theta_A) = \theta_A, \quad P(B_n = 1 \mid \theta_B) = \theta_B \quad \forall n = 1, \ldots, N,$$

$$P(D_n = 1 \mid A_n = 0, B_n = 1, \boldsymbol{\theta}_D) = \theta_1, \quad \forall n = 1, \ldots, N,$$

$$\boldsymbol{\theta}_D = (\theta_0, \theta_1, \theta_2, \theta_3).$$

Where a binary to decimal transformation between the states of A and B and the sub-index of θ is being used.

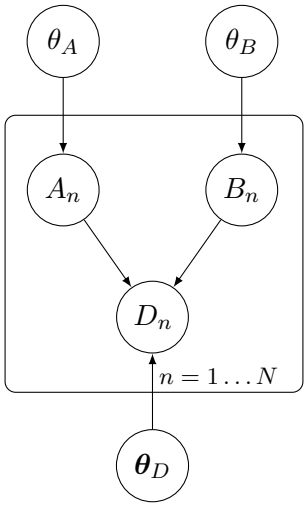


Figure 4: Bayesian network of disease example

The graph gives the following joint probability distribution the variables:

$$P(a_n, b_n, d_n, \theta_A, \theta_B, \theta_D) = P(d_n \mid a_n, b_n, \theta_D)P(a_n \mid \theta_A)P(b_n \mid \theta_A).$$

A prior distribution must be specified and since dealing with multidimensional continuous distributions is computationally problematic, it is usual to use uni-variate distributions.

2.1 Learning binary variables

The simplest cases to continue are $P(\theta_A \mid \boldsymbol{x}_A)$ and $P(\theta_B \mid \boldsymbol{x}_B)$ since they require only a uni-variate prior distribution $P(\theta_A)$ or $P(\theta_B)$. The procedure is shown using θ_A and it is analogous when using θ_B.

The posterior is

$$P(\theta_A \mid \boldsymbol{x}_A) = \frac{1}{P(\boldsymbol{x}_A)} P(\theta_A) \theta_A^{\#(A=1)} (1-\theta_A)^{\#(A=0)}.$$

The most convenient choice for the prior is a Beta distribution as conjugacy will hold:

$$\theta_A \sim \text{Beta}(\alpha_A, \beta_A) \implies P(\theta_A) = \frac{1}{B(\alpha_A, \beta_A)} \theta_A^{\alpha_A - 1} (1-\theta_A)^{\beta_A - 1}.$$

Therefore, it follows that

$$\theta_A \mid \boldsymbol{x}_A \sim \text{Beta}(\alpha_A + \#(A=1), \beta_A + \#(A=0)).$$

The predictive marginal is then

$$\begin{aligned} P(A=1 \mid \boldsymbol{x}_A) &= \frac{P(A=1, \boldsymbol{x}_A)}{P(\boldsymbol{x}_A)} = \int_{\theta_A} \frac{P(A=1, \boldsymbol{x}_A, \theta_A)}{P(\boldsymbol{x}_A)} = \int_{\theta_A} \frac{P(A=1 \mid \boldsymbol{x}_A, \theta_A) P(\boldsymbol{x}_A, \theta_A)}{P(\boldsymbol{x}_A)} \\ &= \int_{\theta_A} \frac{P(A=1 \mid \boldsymbol{x}_A, \theta_A) P(\theta_A \mid \boldsymbol{x}_A) P(\boldsymbol{x}_A)}{P(\boldsymbol{x}_A)} \\ &= \int_{\theta_A} P(\theta_A \mid \boldsymbol{x}_A) \theta_A = \mathbb{E}[\theta_A \mid \boldsymbol{x}_A] \\ &= \frac{\alpha_A + \#(A=1)}{\alpha_A + \#(A=1) + \beta_A + \#(A=0)}. \end{aligned}$$

Where the last equality is given by the expected value of a Beta distribution.

For $P(d \mid a, b)$ the situation is more complex, the simplest approach is to specify a Beta prior for each of the components of $\boldsymbol{theta}_D$. Focus on θ_2, notice the parameters α and β we used before now do depend on A and B:

$$\theta_2 \sim Beta\Big(\alpha_D(1,0) + \#(D=1, A=1, B=0),\ \beta_D(1,0) + \#(D=0, A=1, B=0)\Big).$$

Repeating the procedure we used with A we get that

$$P(D=1 \mid A=1, B=0, \boldsymbol{x}) = \frac{\alpha_D(1,0) + \#(D=1, A=1, B=0)}{\alpha_D(1,0) + \beta_D(1,0) + \#(A=1, B=0)}.$$

All hyperparameters could be set to the same value, where a complete ignorance prior would correspond to set them to 1.

There are two limit possibilities depending on the amount of data available.

- **No data**. The marginal probability corresponds to the prior, which in the last case is

  $$P(D=1 \mid A=1, B=0, \boldsymbol{x}) = \frac{\alpha_D(1,0)}{\alpha_D(1,0) + \beta_D(1,0)}.$$

  Note that equal hyperparameters would give a result of 0.5.

A	B	D
1	1	1
1	0	0
0	1	1
0	1	0
1	1	1
0	0	0
1	0	1

Figure 5: Set of observations, where 1 means true and 0 means false.

- **Infinite data**. When infinite data is available, the marginal is generally dominated by it, this corresponds to the Maximum Likelihood solution.

$$P(D = 1 \mid A = 1, B = 0, \boldsymbol{x}) = \frac{\#(D = 1, A = 1, B = 0)}{\#(A = 1, B = 0)}.$$

  This happens unless the prior has a pathologically strong effect.

Consider the data given in the table in figure 4, and equal parameters and hyperparameters 1 and a prior belief that any setting is equally probable, i.e, $P(A = 1) = 0.5$.

We may illustrate the different results that are obtained using using Bayesian inference and Maximum likelihood training. The former is

$$P(A = 1 \mid \boldsymbol{x}) = \frac{1 + \#(A = 1)}{2 + N} = \frac{5}{9} \approx 0.556.$$

and the latter is $4/7 = 0.571$. In conclusion, the Bayesian result is more prudent than this one, which fits in with our prior belief.

2.2 Frequentist vs Bayesian

The main disadvantages of frequentist approaches are:

1. They assign zero probability to events just because they do not appear in the considered sample.
2. They only consider the ratio of the outcomes but not their amount.

2.3 Missing data

There are situations where maximum likelihood estimation cannot be approached as we have done before, for example, when there is missing data or latent (unobserved) variables.

There are three types of missing data:

1. **Missing completely at random**: The reason why those values are missing is independent of the values themselves and the observed ones.
2. **Missing at random**: The fact that data is missing is not completely random but can be explained given the observed ones.

3. **Missing not at random**: The reason why data is missing is related with such data.

Consider the following example with missing data: let X, Y be two random variables such that

$$P(x, y \mid \Theta) = P(x \mid \theta_x)P(y \mid x, \theta_{y|x}) = \theta_x \, \theta_{y|x}.$$

That is, we are considering a simple Bayesian network $X \to Y$, with both variables being bernoulli trials. Consider the following set of observations $\mathcal{D} = \{(?, y_0)), (x_0, y_1), (?, y_0)$. The likelihood is

$$\mathcal{L}(\Theta, \mathcal{D}) = P(y_0)^2 P(x_0, y_1) = P(y_1 \mid x_0)P(x_0) \left(\sum_x P(y_0 \mid x)P(x) \right)^2 = (\theta_{x_0}\theta_{y_0|x_0} + \theta_{x_1}\theta_{y_0|x_1})^2 \theta_{x_0}\theta_{y_1|x_0}.$$

Where its partial derivatives cannot be independently optimized.

2.4 EM algorithm

The EM algorithm performs maximum likelihood estimation in probabilistic models with missing data. Its main idea is to follow a two-step iterative process where, the first computes the expected value of the missing data and the second optimized the set of parameters.

Given a set of observed variables $\boldsymbol{X} = (X_1, \ldots, X_N)$ and a set of hidden or latent variables $\boldsymbol{Z} = (Z_1, \ldots, Z_M)$, governed by a set of parameters $\boldsymbol{\theta}$, the EM algorithm seeks to find the maximum likelihood estimate of the marginal likelihood $P(\boldsymbol{x} \mid \boldsymbol{\theta})$ of the visible variables by applying the following 2-step iterative procedure:

1. **Expectation step**: Define $Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}$ as the expected value of the log likelihood with respect to the conditional distribution of the hidden variables given the observed:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{x}, \boldsymbol{Z} \mid \boldsymbol{\theta}) \right].$$

2. **Maximization step**: Find the optimal parameters that maximize Q:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^(t)).$$

Theorem. The marginal likelihood cannot be decreased after any iteration of the expectation maximization algorithm.

Proof. For any unknown but fixed value of the hidden variables $\boldsymbol{z}$, we can write[2]

$$\log P(\boldsymbol{x} \mid \boldsymbol{\theta}) = \log P(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) - \log P(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta})$$

By taking expectations over $\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)}$, we get that

$$\begin{aligned} \log P(\boldsymbol{x} \mid \boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{x}, \boldsymbol{Z} \mid \boldsymbol{\theta}) \right] - \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}) \right] \\ &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) - \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}) \right] \end{aligned}$$

Given this quality, the increase in the marginal likelihood is

$$\begin{aligned} \log P(\boldsymbol{x} \mid \boldsymbol{\theta}) - \log P(\boldsymbol{x} \mid \boldsymbol{\theta}^{(t)}) &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \\ &\quad - \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}) \right] + \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)}) \right] \\ &= Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) + KL \left(P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{(t)}) \middle| P(\boldsymbol{Z} \mid \boldsymbol{x}, \boldsymbol{\theta}) \right) \end{aligned}$$

[2]Given that $P(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}) \neq 0$.

Using that the KL divergence is always positive, we arrive at

$$\log P(\boldsymbol{x} \mid \boldsymbol{\theta}^{(t+1)}) - \log P(\boldsymbol{x} \mid \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \geq 0.$$

Where the last inequality is given by the maximization step of the EM algorithm. ■

2.5 EM algorithm for Bayesian networks

Let us take a look at each of the steps given the Bayesian network structure, given that we now have to distinguish 3 types of variables, we are changing the notation. Let $\boldsymbol{X} = (X_1, \ldots, X_N)$ and $\boldsymbol{\theta}$ define the Bayesian network

$$P(\boldsymbol{x} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} P(x_n \mid pa(x_n), \boldsymbol{\theta}),$$

and $\boldsymbol{X} = \boldsymbol{V} \cup \boldsymbol{H}$ where $\boldsymbol{V}$ is the set of visible or observed variables and $\boldsymbol{H}$ is the set of latent variables.

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\boldsymbol{H}|\boldsymbol{v},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{x} \mid \boldsymbol{\theta})\right].$$

The first thing to notice is that

$$\log P(\boldsymbol{x} \mid \boldsymbol{\theta}) = \sum_{n=1}^{N} \log P(x_n \mid pa(x_n), \boldsymbol{\theta}).$$

Which implies

$$\mathbb{E}_{\boldsymbol{H}|\boldsymbol{v},\boldsymbol{\theta}^{(t)}} \left[\log P(\boldsymbol{x} \mid \boldsymbol{\theta})\right] = \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{H}|\boldsymbol{v},\boldsymbol{\theta}^{(t)}} \left[\log P(x_n \mid pa(x_n), \boldsymbol{\theta})\right].$$

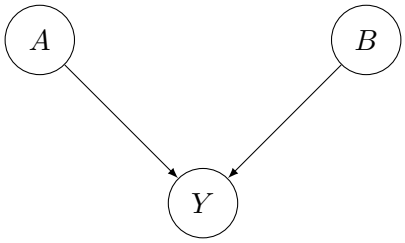
Where such expectation does only affect those terms of the network where either x_n is hidden or there is a hidden variable in $pa(x_n)$.

Assuming that each parameters models the probability of a certain factor as

$$\theta_n = P(x_n \mid pa(x_n)),$$

these can be optimized separately.

Consider a model with three binary variables following



With the following observations from A and Y: $(1,1), (0,0), (1,1), (1,0), (1,1), (1,0), (0,1)$.

Algorithm 1: Expectation Maximization Algorithm for Bayesian networks

Data: A dataset $\boldsymbol{v} = \{v^1, \ldots, v^N\}$, a distribution $P(\boldsymbol{x} \mid \boldsymbol{\theta}) = P(\boldsymbol{v}, \boldsymbol{h} \mid \boldsymbol{\theta})$ that factorizes in a Bayesian network.

Result: The maximum likelihood estimates for $\theta_{x_m} = P(x_m \mid pa(x_m)), m = 1, \ldots, M$,

Initialize $\theta^0_{x_m}, m = 1, \ldots, M$;

$t \leftarrow 0$;

while *Convergence stop criteria* **do**

  for $n = 1$ *to* N **do**

    $Q^n_t(h^n) = P(h^n \mid v^n, \boldsymbol{\theta}^t)$; `// E-step`

  end

  for $m = 1$ *to* M **do**

    $\theta^{t+1}_{x_m} = \arg\max_{\theta_{x_m}} \sum_{n=1}^{N} \mathbb{E}_{Q^t(h^n)}[\log P(x^n_m \mid pa(x^n_m))]$; `// M-step`

  end

  $t \leftarrow t + 1$;

end

return $\theta^t_{x_m}, \quad m = 1, \ldots, M$;

The **expectation step** of this model would compute

$$Q^1(b) = P(b \mid Y = 1, A = 1) \propto P(Y = 1 \mid A = 1, b)P(A = 1)P(b) = \begin{cases} \theta_Y^{1,0}\theta_A(1 - \theta_B) & \text{if } b = 0 \\ \theta_Y^{1,1}\theta_A\theta_B & \text{if } b = 1 \end{cases}$$

$$\vdots$$

$$Q^7(b) = P(b \mid Y = 1, A = 0) \propto P(Y = 1 \mid A = 0, b)P(A = 0)P(b) = \begin{cases} \theta_Y^{0,0}(1 - \theta_A)(1 - \theta_B) & \text{if } b = 0 \\ \theta_Y^{1,0}(1 - \theta_A)\theta_B & \text{if } b = 1 \end{cases}$$

On the other hand, the **maximization step** would optimize each parameter:

$$\begin{aligned}
\theta_A^{new} &= \arg\max_{\theta_A} \sum_{i=1}^{7} \mathbb{E}_{Q^i(B)}[\log P(a_i \mid \theta_A)] \\
&= \arg\max_{\theta_A} \sum_{i=1}^{7} \sum_{b=0,1} Q^i(b) \log P(a_i \mid \theta_A) \\
&= \arg\max_{\theta_A} \sum_{i=1}^{7} Q^i(B = 0) \log P(a_i \mid \theta_A) + Q^i(B = 1) \log P(a_i \mid \theta_A) \\
&= \arg\max_{\theta_A} \sum_{i=1}^{7} \log P(a_i \mid \theta_A) = \arg\max_{\theta_A} \log \left(\prod_{i=1}^{7} P(a_i \mid \theta_A) \right) \\
&= \arg\max_{\theta_A} \mathcal{L}(\theta_A; \mathcal{D}_A) = \frac{N_A}{N} = \frac{5}{7}
\end{aligned}$$

$$
\begin{aligned}
\theta_B^{new} &= \arg\max_{\theta_B} \sum_{i=1}^{7} \mathbb{E}_{Q^i(B)}[\log P(b_i \mid \theta_B)] \\
&= \arg\max_{\theta_B} \sum_{i=1}^{7} \left(Q^i(B=0) \log P(B=0 \mid \theta_B) + Q^i(B=1) \log P(B=1 \mid \theta_B) \right) \\
&= \arg\max_{\theta_B} \sum_{i=1}^{7} \left(Q^i(B=0) \log(1-\theta_B) + Q^i(B=1) \log \theta_B \right) \\
&= \frac{\sum_{i=1}^{7} Q^i(B=1)}{\sum_{i=1}^{7} Q^i(B=0) + \sum_{i=1}^{7} Q^i(B=1)} = \frac{\sum_{i=1}^{7} Q^i(B=1)}{N}
\end{aligned}
$$

2.6 Gaussian mixture

A Gaussian mixture model considers the following variables:

- A corresponding set of observations $\boldsymbol{x} = \{x_1, \ldots, x_N\}$.
- The cluster assignment latent variables $\boldsymbol{Z} = \{Z_1, \ldots, Z_N\}$.
- The mixture weights $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, i.e, prior probability of a particular component k.
- Each normal distribution $\mathcal{N}(\mu_k, \Sigma_k)$.

The joint probability factorizes as

$$
P(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = P(\boldsymbol{x} \mid \boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) P(\boldsymbol{z} \mid \boldsymbol{\pi}) P(\boldsymbol{\pi}) P(\boldsymbol{\mu} \mid \boldsymbol{\Sigma}) P(\boldsymbol{\Sigma}).
$$

We are now in situation to give the explicit Bayesian network for this model:

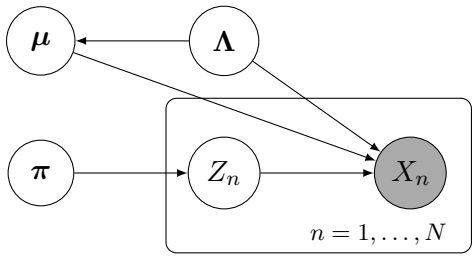


Figure 6: Gaussian mixture model. Squares represent hyper-parameters and X_n are observed.

The complete log likelihood takes the form

$$
\log P(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[X_i = k] \left(\log \pi_k \log \mathcal{N}(x_n; \mu_k, \Sigma_k) \right).
$$

The expected value of the log-likelihood is

$$\begin{aligned} Q(\lambda \mid \lambda^{new}) &= \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x}}[\log P(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x}} \left[\mathbb{I}[X_i = k] \left(\log \pi_k \log \mathcal{N}(x_n; \mu_k, \Sigma_k)\right)\right] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{x}} \left[\mathbb{I}[X_i = k]\right] \left(\log \pi_k \log \mathcal{N}(x_n; \mu_k, \Sigma_k)\right) \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} P(Z_n = k \mid \boldsymbol{x}) \left(\log \pi_k \log \mathcal{N}(x_n; \mu_k, \Sigma_k)\right) \end{aligned}$$

3 Exact inference

Making bayesian inference refres to computing posterior probabilities. The main step at this task is to compute marginal probabilities

$$P(A) = \sum_{B,C} P(A, B, C)$$

Twomain algorithms to compute marginal probabilities:

- Variable elimination (VE).
- Belief propagación (Sum-product).

3.1 Variable elimination

Consider a un-normalized distribution

$$\phi(A, B, C, D) = \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(D, A)$$

Poner un grafo es un cuadrado. we aim to compute

$$\phi(A) = \sum_{B,C,D} \phi(A, B, C, D) = \sum_{A,B,C} \phi_1(A, B)\phi_2(B, C)\phi_3(C, D)\phi_4(D, A) = \sum_{D} \phi_4(D, A) \sum_{C} \phi_3(C, D) \sum_{B} \phi_1(A, B)\phi$$

To this end, we compute the following factors

$$\tau_1(A, C) = \sum_{B} \underbrace{\phi_1(A, B)\phi_2(B, C)}_{\psi_i(A,B,C)} \tau_2(A, D) = \sum_{C} \phi_3(C, D)\tau_1(A, C)\phi(C) = \sum_{D} \phi_4(D, A)\tau_2(A, D)$$

The cost is exponential in the size of the scope of the largest factor ψ_i .

Suppose each variable is discrete with K possible values, the largest factor has 3 variables, that is, the cost is K^3.

3.2 Belief propagation

Suppose we got a graph that is a tree. It allows to compute all marginals at once. (mirar algoritmo den las diapositivas).

Those graphs where this algorithm can be applied must be cluster graphs:

1. Undirected.
2. Each node $C_i \subset \mathcal{X}$ is a subset of variables.
3. Edges contain separator sets $i \to j, S_{ij} \subset C_i \cap C_j$.
4. Nodes contain an associated factor ψ_i with $scope(\psi_i) \subset C_i$.
5. Satisfies the family preservation property:
6. Satisfies the general running intersection property:

Given this, for each variable, there is an unique path that lets information flow.

For the example given before:

3.2.1 Junction tree algorithm

1. Triangulation. Ensure that every loop of length 4 or more has a chord.
2. Junction tree: Tree of max-clique of the triangulated graph.
3. Potential assignments. Assign potentials to juntion tree cliques.
4. Find a tree that goes trough all cliques.

3.2.2 Variable elimination

1. A node C_i corresponds to each intermediate factor ψ_i, $C_i = scope(\psi_i)$.
2. A edge connects C_i and C_j if τ_i is used to compute ψ_j:

$$S_{ij} = scope(\tau_i).$$

Each maximal clique form the juntion tree algorithm corresponds to an intermiate factor from the VE algorithm. This means that the resulting graphs are always similar.

References

Spirtes, Peter, Glymour, Clark N, Scheines, Richard, & Heckerman, David. 2000. *Causation, prediction, and search*. MIT press.