DD2434/FDD3434 Machine Learning, Advanced Course Exercises (DGM, HMM, EM & VI)

Negar Safinianaini, Antonio Matosevic, and Hazal Koptagel *

November 2020

Abstract

This document consists of the theory and exercises for DD2434/FDD3434 Machine Learning Advanced Course, covering DGM, HMM, EM, and VI. Parts of the exercises are from year 2018 by Borja Rodriguez Galvez and Elin Samuelsson.

Contents

*KTH Royal Institute of Technology, Stockholm, Sweden

1 Directed Graphical Models (DGM) – Theory

1.1 D-separation rules

If two nodes are d-separated, they are independent. In Figure 1, S and R are independent from each other.

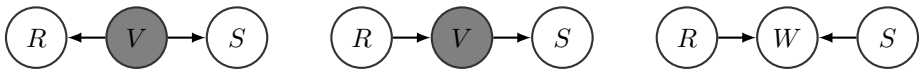


Figure 1: S and R are d-separated. Note that W is not observed and no descendants of W is observed either.

[Pearl 1988] example:
d-separated: dead battery → car won't start ← no gas
not d-separated: dead battery → **car won't start** ← no gas

A charged battery does not give information about the gas, but, a charge battery become informative (dependency) after observing that the car won't start (because then we know that gas was empty from knowing the battery was full and car didn't start)

Two methods to determine if variables are independent or not:

1.1.1 Method 1

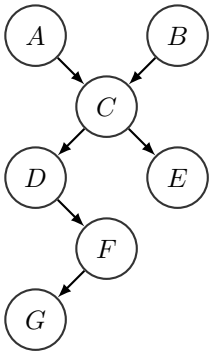
This method is efficient when the dependencies are being examined among certain variables and not all the variables in the graph. We follow this procedure [3]:

- 1. Construct the "ancestral graph" of all variables mentioned in the probability expression. This is a reduced version of the original net, consisting only of the variables **mentioned** and all of their **ancestors** (parents, parents' parents, etc.)
- 2. "Moralize" the ancestral graph by "marrying" the parents. For each pair of variables with a common child, draw an undirected edge (line) between them. (If a variable has more than two parents, draw lines between every pair of parents.)
- 3. "Disorient" the graph by replacing the directed edges (arrows) with undirected edges (lines).
- 4. Delete the givens and their edges. If the independence question had any given variables, erase those variables from the graph and erase all of their connections, too.
- 5. Read the answer off the graph.
  - If the variables are disconnected in this graph, they are guaranteed to be independent.
  - If the variables are connected in this graph, they are not guaranteed to be independent.* Note that "are connected" means "have a path between them," so if we have a path X-Y-Z, X and Z are considered to be connected, even if there's no edge between them.

– If one or both of the variables are missing (because they were givens, and were therefore deleted), they are independent.

* We can say "the variables are dependent, as far as the Bayes net is concerned" or "the Bayes net does not require the variables to be independent," but we cannot guarantee dependency using d-separation alone, because the variables can still be numerically independent (e.g. if $P(A|B)$ and $P(A)$ happen to be equal for all values of A and B).

Example Q: Are A and B conditionally independent, given D and F?



A: They are not required to be conditionally independent, using method 1 as below:

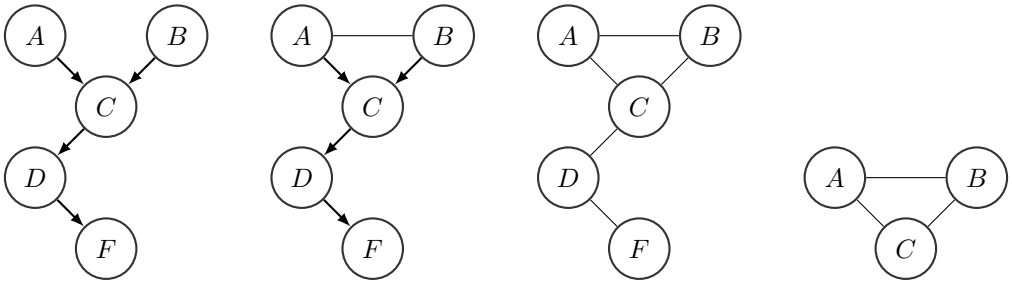


Figure 2: From left to right: 1. Draw ancestral graph 2. Moralize 3. Disorient 4. Delete givens

1.1.2 Method 2

This method is efficient when examining the dependence over the whole graph. We put a block symbol on the graph wherever we detect the d-separation using the rules in Figure 1. Then we can answer whether variables are d-separated or not by following the path between them i.e. if the path contains a block they are obviously d-separated. **Example** The blocks are illustrated as red point over the graph in Figure 3; the table shows the answers to d-separation which is gained by following the path between each pair of i and j.

i, j	d-separated
A, D	yes
A, B	no
D, G	yes
D, H	no
D, E	yes
D, F	no
B, H	yes

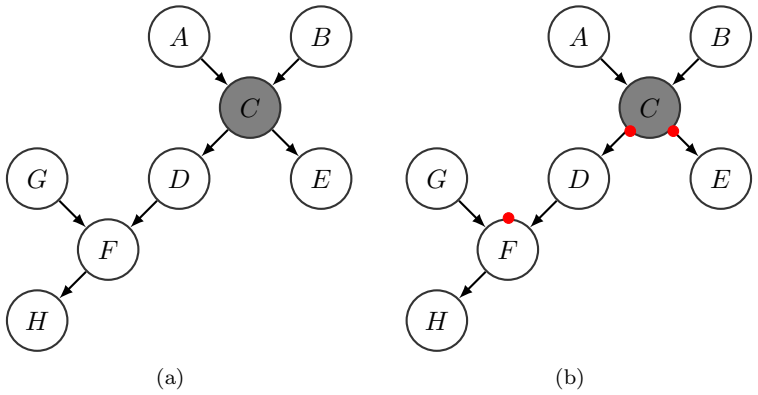
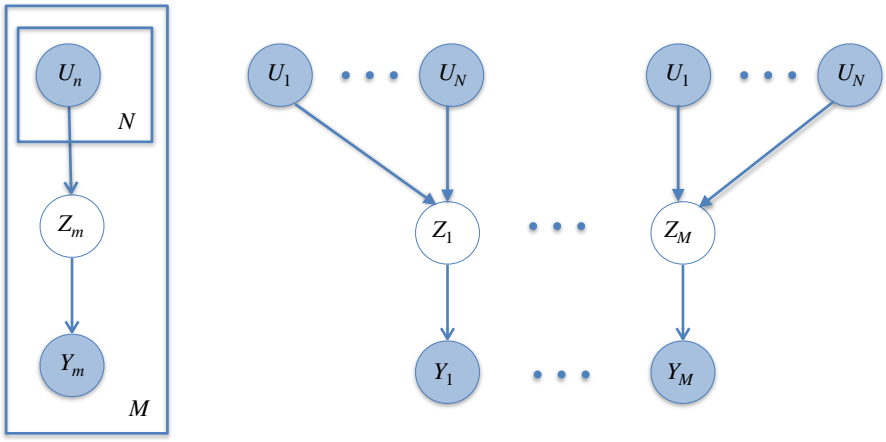


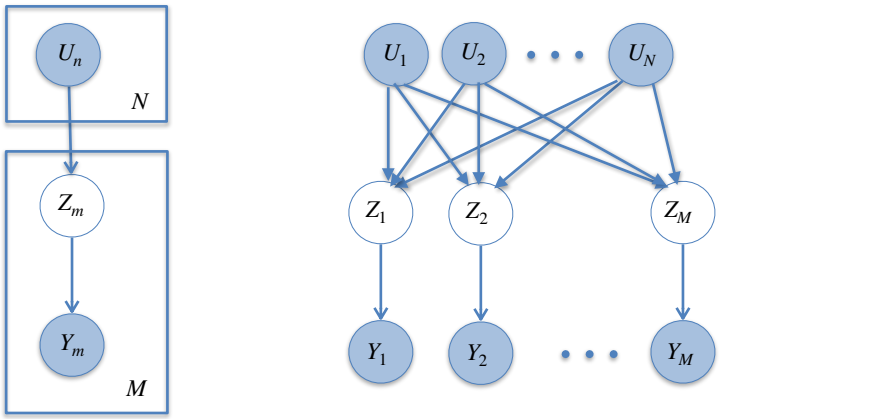
Figure 3: Blocks are shown in red on the graph in (b).

1.2 Plate notation

In plate notation, the node/graph is replicated N times (when N is written in the plate); moreover, any edge that crosses a plate boundary is replicated once for each subgraph repetition. The plate notation is used to illustrate a compact representation of the graphical model. Using such a simplified notation allows to have an overview of model parameters and variables. However, it is sometimes useful to unfold the plate, e.g., when evaluating the conditional dependencies. In Fig. 4, two examples of unfolding a plate graph are illustrated.



The plate graph The corresponding unfolded graph



The plate graph The corresponding unfolded graph

Figure 4: Two examples of unfolding a plate DGM.

2 Directed Graphical Models (DGM) – Exercises

2.1 Bayes Ball

Question: List all variables that are independent of A given evidence on the shaded node for each of the DGMs a), b) and c) below.

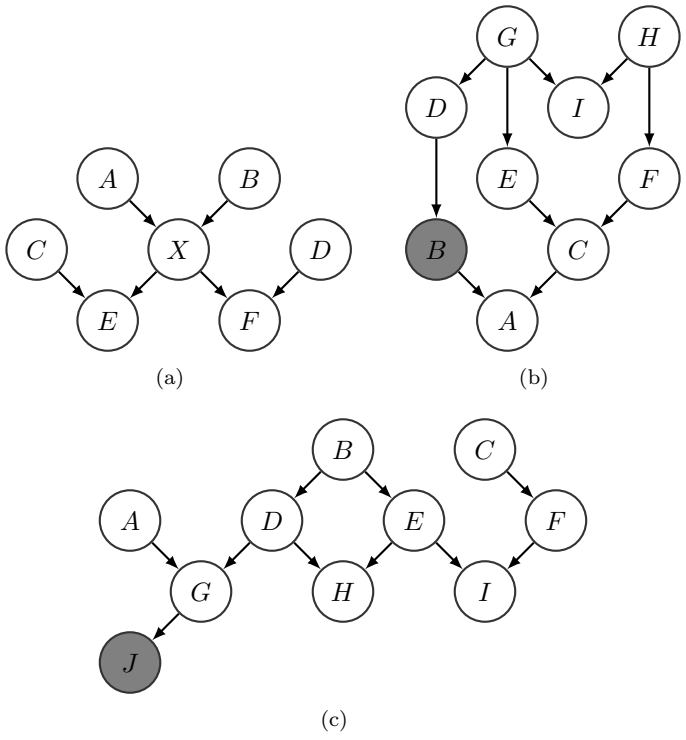


Figure 5: Some DGMs.

Solution: a) See Figure 6, b) See Figure 7, c) See Figure 8.

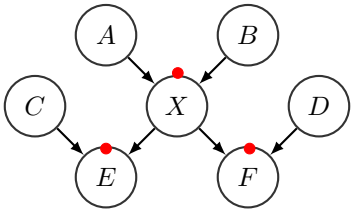


Figure 6: There is no conditioning on any node. We use method 2, So we just look for head-to-heads (the right most rule in Figure 1). Putting the blocks between the head-to-heads, we see that B,C, and D are separated from A or independent of A.

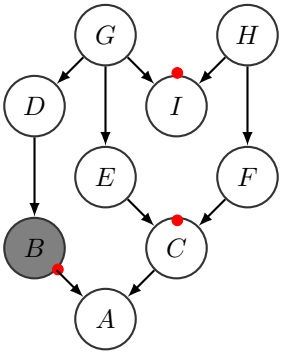


Figure 7: Following method 2, there is a path to A from all variables hence no variable is independent of A.

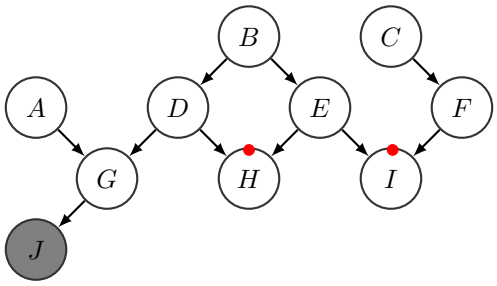


Figure 8: Following method 2, there is a path to A from all variables except F and C. Therefore, only C and F are independent of A.

2.2 Bayes nets for a rainy day (Exercise 10.5 from Murphy [4])

Question: (Source: Nando de Freitas) In this question you must model a problem with 4 binary variables: G = "$gray$", V = "$Vancouver$", R = "$rain$" and S = "sad". Consider the directed graphical model describing the relationship between these variables shown in Figure 9 (and the probability tables shown in Table 1).

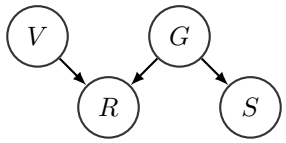


Figure 9: Bayesian net for a rainy day

Table 1: Probability tables of Bayes net for a rainy day

$V=0$	$V=1$
δ	$1-\delta$

$G=0$	$G=1$
α	$1-\alpha$

	$S=0$	$S=1$
$G=0$	γ	$1-\gamma$
$G=1$	β	$1-\beta$

	$R=0$	$R=1$
$VG=00$	0.6	0.4
$VG=01$	0.3	0.7
$VG=10$	0.2	0.8
$VG=11$	0.1	0.9

a. Write down an expression for $P(S=1|V=1)$ in terms of $\alpha, \beta, \gamma, \delta$.

b. Write down an expression for $P(S=1|V=0)$. Is this the same or different to $P(S=1|V=1)$? Explain why.

c. Find maximum likelihood estimates of α, β, γ using the following data set, where each row is a training case. (You may state your answers without proof.)

V	G	R	S
1	1	1	1
1	1	0	1
1	0	0	0

Solution: (The solution is taken from Elin Samuelsson's notes from December 2018 and modified slightly.)

a. Write down an expression for $P(S = 1|V = 1)$ in terms of $\alpha, \beta, \gamma, \delta$.

$$
\begin{aligned}
P(S=1|V=1) &= \frac{P(S=1,V=1)}{P(V=1)} \\
&= \frac{1}{P(V=1)} \sum_{g=0}^{1} \sum_{r=0}^{1} P(S=1,V=1,R=r,G=g) \\
&= \frac{1}{P(V=1)} \sum_{g=0}^{1} \sum_{r=0}^{1} P(S=1,V=1,R=r|G=g)P(G=g) \\
&\quad \{G \text{ is tail-to-tail and blocks the path} \to S \perp\!\!\!\perp \{R,V\}|G\} \\
&= \frac{1}{P(V=1)} \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \sum_{r=0}^{1} P(V=1,R=r|G=g) \\
&= \frac{1}{P(V=1)} \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \sum_{r=0}^{1} P(R=r|V=1,G=g)P(V=1) \\
&= \frac{P(V=1)}{P(V=1)} \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \sum_{r=0}^{1} P(R=r|V=1,G=g) \\
&= \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \sum_{r=0}^{1} P(R=r|V=1,G=g) \\
&\quad \{\sum_{r=0}^{1} P(R=r|V=1,G=g) = 1, \text{ regardless of the value of } G \text{ and } V\} \\
&= \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \\
&= P(S=1|G=0)P(G=0) + P(S=1|G=1)P(G=1) \\
&= \alpha(1-\gamma) + (1-\alpha)(1-\beta) \\
&= 1 - \beta + \alpha\beta - \alpha\gamma
\end{aligned}
\tag{1}
$$

b. Write down an expression for $P(S = 1|V = 0)$. Is this the same or different to $P(S = 1|V = 1)$? Explain why.

$$
\begin{aligned}
P(S=1|V=0) &= \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \sum_{r=0}^{1} P(R=r|V=0,G=g) \\
&= \sum_{g=0}^{1} P(S=1|G=g)P(G=g) \\
&= P(S=1|V=1)
\end{aligned}
\tag{2}
$$

since $\sum_{r=0}^{1} P(R = r|V = v, G = g) = 1$, regardless of the value of G and V.

c. Find maximum likelihood estimates of α, β, γ using the following data set, where each row is a training case. (You may state your answers without proof.)

V	G	R	S
1	1	1	1
1	1	0	1
1	0	0	0

Notation:

$$\mathbf{1}(a = b) = \begin{cases} 1, \text{ if } a = b \\ 0, \text{ if } a \neq b \end{cases} \tag{3}$$

Maximum likelihood estimates:

$$\hat{\alpha} = P(G = 0) = \frac{\sum_{n=1}^{N} \mathbf{1}(G_n = 0)}{N} = \frac{1}{3} \tag{4}$$

$$\hat{\beta} = P(S = 0|G = 1) = \frac{P(S = 0, G = 1)}{P(G = 1)} = \frac{\sum_{n=1}^{N} \mathbf{1}(S_n = 0, G_n = 1)}{\sum_{n=1}^{N} \mathbf{1}(G_n = 1)} = \frac{0}{2} \tag{5}$$

$$\hat{\gamma} = P(S = 0|G = 0) = \frac{P(S = 0, G = 0)}{P(G = 0)} = \frac{\sum_{n=1}^{N} \mathbf{1}(S_n = 0, G_n = 0)}{\sum_{n=1}^{N} \mathbf{1}(G_n = 0)} = \frac{1}{1} \tag{6}$$

An Alternative Solution

The more general way to find the MLE is i) write the likelihood (or log-likelihood) ii) take derivative w.r.t the parameter of interest and set it to zero iii) check whether the value actually maximizes the likelihood (by looking at the second derivative [5] is negative or not).

The likelihood of the data is:

$$\begin{aligned}
\mathcal{L} &= P(D|\Theta) \\
&= P(D_1, \ldots, D_N|\Theta) \\
&= \prod_{n=1}^{N} P(D_n|\Theta) \\
&= \prod_{n=1}^{N} P(V_n|\delta)P(G_n|\alpha)P(S_n|G_n, \beta, \gamma)P(R_n|V_n, G_n) \\
&= P(V_n = 1|\delta)^3 P(G_n = 0|\alpha)P(G_n = 1|\alpha)^2 P(S_n = 0|G_n = 0, \beta, \gamma)P(S_n = 1|G_n = 1, \beta, \gamma)^2 \\
&\quad P(R_n = 0|V_n = 1, G_n = 0)P(R_n = 0|V_n = 1, G_n = 1)P(R_n = 1|V_n = 1, G_n = 1) \\
&= (1 - \delta)^3 \alpha(1 - \alpha)^2 \gamma(1 - \beta)^2 \times 0.2 \times 0.1 \times 0.9 \\
&\propto (1 - \delta)^3 \alpha(1 - \alpha)^2 \gamma(1 - \beta)^2
\end{aligned} \tag{7}$$

Let's look at the likelihood (see Figure 10). For the first subplot, I fixed $\alpha, \gamma, \beta \in (0, 1)$ and ranged $\delta \in [0, 1]$. The subplot shows how the likelihood changes w.r.t δ. I repeated the same method for the rest of the parameters. From the figure, we can clearly see which values of the parameters maximize the likelihood ($\hat{\delta} = 0, \hat{\alpha} = 0.\overline{33}, \hat{\gamma} = 1, \hat{\beta} = 0$).

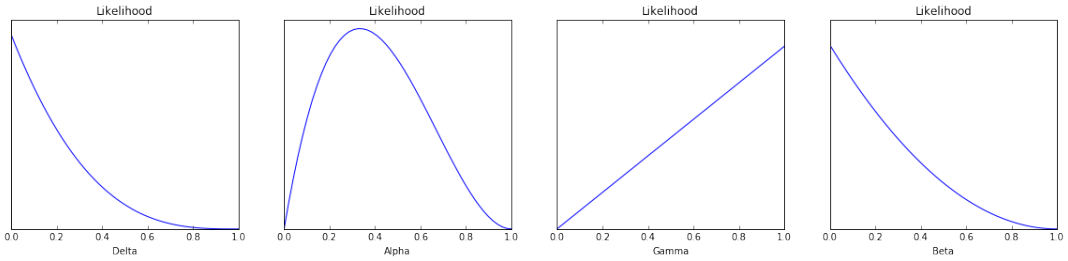


Figure 10: Likelihood of rainy day example with varying parameters

Now, let's show these with the derivatives. First, consider α. Take the first derivative of the likelihood w.r.t α.

$$\begin{aligned}
\frac{\partial}{\partial\alpha}\mathcal{L} &= \frac{\partial}{\partial\alpha}\alpha(1-\alpha)^2C \\
&= (1-\alpha)^2C - 2\alpha(1-\alpha)C \\
&= (1-\alpha)(1-\alpha-2\alpha)C \\
&= (1-\alpha)(1-3\alpha)C \\
&= 0 \\
\alpha &= 1 \text{ or } \alpha = \frac{1}{3}
\end{aligned} \tag{8}$$

where C is a temporary variable that I used to represent all the other terms in the likelihood beside the parameter of interest (notice that C is non-negative). There are two α values which either maximize or minimize the likelihood. We need to check the second derivative of the likelihood w.r.t α:

$$\begin{aligned}
\frac{\partial^2}{\partial\alpha^2}\mathcal{L} &= \frac{\partial}{\partial\alpha}\alpha(1-\alpha)^2C \\
&= -(1-3\alpha)C - 3(1-\alpha)C \\
&= (-4+6\alpha)C
\end{aligned} \tag{9}$$

When $\alpha = 1$, the second derivative becomes $2C$, which is non-negative, which means $\alpha = 1$ is not a maximizer of the likelihood. When $\alpha = \frac{1}{3}$, the second derivative becomes $-2C$, which is negative, which means $\hat{\alpha} = \frac{1}{3}$ is the maximum likelihood estimator. We can confirm this result with Figure 10.

Now, we move on to β. We re-write the likelihood as $\mathcal{L} = (1-\beta)^2C$. It is clear that the $\hat{\beta}$ which maximizes the likelihood must be $\hat{\beta} = 0$ (since C is non-negative and $\mathcal{L}$ gets the highest value, which is $1C$, when $\beta = 0$). Let's look at take the derivatives.

$$\begin{aligned}
\frac{\partial}{\partial\beta}\mathcal{L} &= \frac{\partial}{\partial\alpha}(1-\beta)^2C \\
&= -2(1-\beta)C \\
&= 0 \\
\beta &= 1
\end{aligned} \tag{10}$$

Now, check the second derivative:

$$\begin{aligned}\frac{\partial^2}{\partial\beta^2}\mathcal{L} &= \frac{\partial}{\partial\alpha}(1-\beta)^2 C \\ &= 2C \\ &> 0\end{aligned} \tag{11}$$

Since the second derivative is always non-negative, $\beta = 1$ is the minimizer of the likelihood. Notice that we were unable to find the β which maximizes the likelihood with this approach. Why? In our data D, we don't have any samples where $S = 0$ and $G = 1$.

Finally, let's re-write the likelihood in terms of γ; $\mathcal{L} = \gamma C$. It is clear that the $\hat{\gamma}$ which maximizes the likelihood must be $\hat{\gamma} = 1$ (because the likelihood is a linearly increasing function of γ).

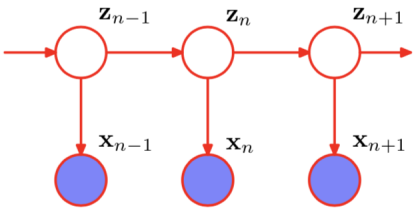


Figure 11: An HMM unfolded across the sequence (or time sometimes), i.e., trellis diagram.
from Bishop [1]

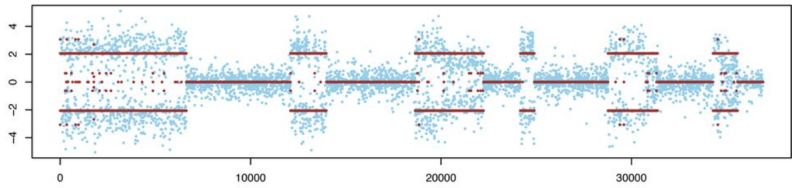


Figure 12: An example of HMM with continuous data (observable R.V.)
accessed at https://www.biorxiv.org/content/10.1101/410845v1.full

3 Hidden Markov Models (HMM) – Theory

A Hidden Markov Model (HMM), illustrated in Fig. 3, is a simple and rich probabilistic graphical model, widely used for sequential or temporal real world data analysis, see example of biological application in Fig. 12. In an HMM, a sequence of symbols/observable R.V.'s (observation sequence colored in blue in the figure) is emitted/generated from a sequence of states (state sequence). The state sequence comprises of hidden discrete variables and the sequence follows a Markov structure, i.e., the next state in the state sequence only depends on the current state. For a finite HMM, we assume that the input sequences have length N. The observation sequence is denoted as $X = \{x_1, \ldots, x_N\}$ and the state sequence is denoted as $Z = \{z_1, \ldots, z_N\}$. Each state in Z takes a value j for j = 1,...,J. The value of each element in X can be discrete, continuous, or multi dimensional. An HMM is parameterized by initial probabilities, $p(z_1)$, transition probabilities, $p(z_n|z_{n-1})$, and emission probabilities, $p(x_n|z_n)$. We term these as θ. These are short notations, and, the proper one for transition probabilities is for instance: $p(z_n = j|z_{n-1} = i)$, i.e., the probability of moving from state i to j where i and j take values in $\{1, ..., J\}$. The emission probability is either pdf or pmf of the observable variable, depending if it is continuous or discrete. The transition probabilities form a matrix, called the transition matrix with shape $J \times J$; each row on this matrix forms a probability distribution, hence the probabilities sum to one.
The joint distribution of all of the random variables in the HMM (X's and Z's) factors as the following; the key point in HMMs is the factorization property which makes inference tractable and easy.

$$p(x_{1:N}, z_{1:N}) = p(z_1)p(x_1|z_1) \prod_{n=2}^{N} p(z_n|z_{n-1})p(x_n|z_n)$$

3.1 Marginal distribution of the hidden variables of HMM

For learning and inference in HMMs (e.g. EM), we need to calculate marginal distribution of the hidden variables, such as $p(z_n|x_{1:N}, \theta)$. So we assume to know the model parameters and we have observed the data. For readability, we write $p(z_n|x_{1:N})$. We could calculate this by performing marginalization over all the hidden variables, but, that is costly (we would have to calculate N nested sum with each iterating over J possible values of the hidden variable i.e. $O(J^N)$). Instead we use an approach which takes $O(NJ^2)$ number of calculations. We write the probability as $\frac{p(x_{1:N}|z_n)p(z_n)}{p(x_{1:N})}$ due to Bayes, and then, because of conditional independence, we can write $p(x_{1:N}|z_n)$ as $p(x_{1:n}|z_n)p(x_{n+1:N}|z_n)$. We have now split the problem into two problems which are solved by recursion (this whole process is referred to as dynamic programming). We can rewrite the problem, finally, as: $\frac{p(x_{1:n}, z_n)p(x_{n+1:N}|z_n)}{p(x_{1:N})}$ or $\frac{\alpha(z_n)\beta(z_n)}{\sum_{z_N}\alpha(z_N)}$. In the next two sections we calculate the probabilities $\alpha(z_n)$ and $\beta(z_n)$, forward and backward, respectively.
For calculation of the marginal of the two consecutive hidden variables, see [1].

3.1.1 Forward pass

$$
\begin{aligned}
&\alpha(z_n) = p(x_{1:n}, z_n) = p(x_{1:n}|z_n)p(z_n) = p(x_n|z_n)p(x_{1:n-1}|z_n)p(z_n) = \\
&p(x_n|z_n)p(x_{1:n-1}, z_n) = p(x_n|z_n)\sum_{z_{n-1}} p(x_{1:n-1}, z_n, z_{n-1}) = \\
&p(x_n|z_n)\sum_{z_{n-1}} p(x_{1:n-1}, z_n|z_{n-1})p(z_{n-1}) = \\
&p(x_n|z_n)\sum_{z_{n-1}} p(x_{1:n-1}|z_{n-1})p(z_{n-1})p(z_n|z_{n-1}) = \\
&p(x_n|z_n)\sum_{z_{n-1}} p(x_{1:n-1}, z_{n-1})p(z_n|z_{n-1}) = p(x_n|z_n)\sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1}) \\
&, where \quad \alpha(z_1) = p(x_1|z_1)p(z_1)
\end{aligned}
$$

Note that we have $\alpha(z_1)$ from the initializations (we initialize the initial and emission probabilities, thus we have $\alpha(z_1)$).

3.1.2 Backward pass

$$
\begin{aligned}
&\beta(z_n) = p(x_{n+1:N}|z_n) = \sum_{z_{n+1}} p(x_{n+1:N}, z_{n+1}|z_n) = \\
&\sum_{z_{n+1}} p(x_{n+1:N}|z_{n+1}, z_n)p(z_{n+1}|z_n) = \\
&\sum_{z_{n+1}} p(x_{n+1:N}|z_{n+1})p(z_{n+1}|z_n) = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1})p(x_{n+2:N}|z_{n+1})p(z_{n+1}|z_n) = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1})\beta(z_{n+1})p(z_{n+1}|z_n) \\
&, where \quad \beta(z_N) = 1
\end{aligned}
$$

Example of an input-output hidden Markov model. In this case, both the emission probabilities and the transition probabilities depend on the values of a sequence of observations $\mathbf{u}_1, \ldots, \mathbf{u}_N$.

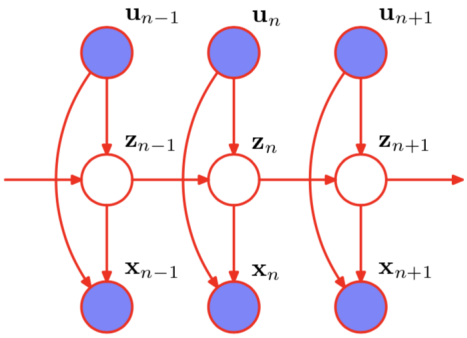


Figure 13: IOHMM
from Bishop [1]

4 Hidden Markov Models (HMM) – Exercises

4.1 Marginal distribution of the hidden variable of IOHMM

Find the marginal distribution of one hidden variable, given the observed input and output (shaded in Fig. 13.), i.e., $p(z_n|x_{1:N}, u_{1:N})$.
In the below we use $\alpha(z_n)$ and $\beta(z_n)$ as in the previous section, but, it is not the same probability, see the new definitions below. Here, due to the graph structure in Fig. 13, we have an extra component/variable u in addition to the standard HMM. So, we need to condition on this extra component; e.g., the new $\alpha(z_n)$ is the HMM's $\alpha(z_n)$ with conditioning on u. **Note that the emission and transition probabilities are now conditioned on u too, i.e., transition probability is $p(z_n|z_{n-1}, u_n)$ and emission probability is $p(x_n|z_n, u_n)$ which can be noticed in Fig. 13.**

$$p(z_n|x_{1:N}, u_{1:N}) = \frac{p(x_{1:N}, z_n|u_{1:N})}{p(x_{1:N}|u_{1:N})} = \frac{p(x_{1:n}, z_n|u_{1:n})p(x_{n+1:N}|z_n, u_{n+1:N})}{p(x_{1:N}|u_{1:N})} = \frac{\alpha(z_n)\beta(z_n)}{\sum_{z_N}\alpha(z_N)}$$

$$\begin{aligned}
\alpha(z_n) &= p(x_{1:n}, z_n|u_{1:n}) = \sum_{z_{n-1}} p(x_{1:n}, z_n, z_{n-1}|u_{1:n}) = \sum_{z_{n-1}} p(x_{1:n}|z_n, z_{n-1}, u_{1:n})p(z_n, z_{n-1}|u_{1:n}) \\
&= p(x_n|z_n, u_n) \sum_{z_{n-1}} p(x_{1:n-1}|z_n, z_{n-1}, u_{1:n})p(z_n, z_{n-1}|u_{1:n}) \\
&= p(x_n|z_n, u_n) \sum_{z_{n-1}} p(x_{1:n-1}|z_n, z_{n-1}, u_{1:n})p(z_n|z_{n-1}, u_{1:n})p(z_{n-1}|u_{1:n}) \\
&= p(x_n|z_n, u_n) \sum_{z_{n-1}} p(x_{1:n-1}|z_{n-1}, u_{1:n})p(z_n|z_{n-1}, u_n)p(z_{n-1}|u_{1:n}) \\
&= p(x_n|z_n, u_n) \sum_{z_{n-1}} \frac{p(x_{1:n-1}, z_{n-1}|u_{1:n})}{p(z_{n-1}|u_{1:n})} p(z_n|z_{n-1}, u_n)p(z_{n-1}|u_{1:n}) \\
&= p(x_n|z_n, u_n) \sum_{z_{n-1}} p(x_{1:n-1}, z_{n-1}|u_{1:n})p(z_n|z_{n-1}, u_n) = p(x_n|z_n, u_n) \sum_{z_{n-1}} \alpha(z_{n-1})p(z_n|z_{n-1}, u_n)
\end{aligned}$$

$$
\begin{aligned}
&\beta(z_n) = p(x_{n+1:N}|z_n, u_{n+1:N}) = \\
&\sum_{z_{n+1}} p(x_{n+1:N}, z_{n+1}|z_n, u_{n+1:N}) = \\
&\sum_{z_{n+1}} p(x_{n+1:N}|z_{n+1}, z_n, u_{n+1:N}) p(z_{n+1}|z_n, u_{n+1:N}) = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1}, u_{n+1}) p(x_{n+2:N}|z_{n+1}, u_{n+1:N}) p(z_{n+1}|z_n, u_{n+1:N}) = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1}, u_{n+1}) p(x_{n+2:N}|z_{n+1}, u_{n+1:N}) \underbrace{p(z_{n+1}|z_n, u_{n+1})}_{\text{see text below}^*} = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1}, u_{n+1}) p(x_{n+2:N}|z_{n+1}, u_{n+2:N}) p(z_{n+1}|z_n, u_{n+1}) = \\
&\sum_{z_{n+1}} p(x_{n+1}|z_{n+1}, u_{n+1}) \beta(z_{n+1}) p(z_{n+1}|z_n, u_{n+1})
\end{aligned}
$$

* z_{n+2} d-separates z_{n+1} from $u_{n+2:N}$ due to v-structure (the car example in section 1)

5 Expectation-Maximization – Theory

General setup Let $\mathbf{x} = \{x_1, \ldots, x_N\}$ be a set of observations (of corresponding random variables (RVs) $X_1, \ldots, X_N$) and $\mathbf{Z} = \{Z_1, \ldots, Z_K\}$ a set of latent RVs. Oftentimes one constructs a model where $\mathbf{x}$ and $\mathbf{Z}$ are directly connected and parameterises it (i.e. distributions involved) by a set of parameters θ. The goal is then to infer θ given data $\mathbf{x}$.

Maximum likelihood estimation A common tool for estimating distribution parameters is the maximum likelihood estimation (MLE). In its essence it seeks to find the point estimate θ^* of θ so that the probability density of observing exactly this sample $\mathbf{x}$ of $X_1, \ldots, X_n$ is maximized. This probability density is referred to as data likelihood and denoted as $p(\mathbf{x}; \theta)$, which is just uncluttered way of writing $p(X_1 = x_1, \ldots, X_N = x_n; \theta)$, henceforth used for all distributions.

MLE for latent variables models In the context of latent variable models, data likelihood can be written using the law of total probability as

$$p(\mathbf{x}; \theta) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta), \tag{12}$$

for **all** possible combinations $\mathbf{z} = \{z_1, \ldots, z_K\}$ that $Z_1, \ldots, Z_K$ can attain. It is obvious that this sum[1] is computationally exponential, often referred to as intractable[2].

Expectation-maximization algorithm To alleviate this problem, one can optimize something close to likelihood, but tractable. Let us present one way to do this by maximizing its lower bound instead. Returning to Eq. 12 we can, thanks to monotonicity, equivalently maximize data likelihood in the log-space i.e. we have that

$$\log p(\mathbf{x}; \theta) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta). \tag{13}$$

Now, there is a result called Jensen's inequality stating that for a convex function $f(\cdot)$ of a RV Ψ it holds that $f(\mathbb{E}[\Psi]) \geq \mathbb{E}[f(\Psi)]$. Since the logarithm conveniently appearing in Eq. 13 is convex, we can perhaps use this inequality to somehow simplify the initial problem. It only remains to rewrite Eq. 13 into an expectation so that the inequality can be applied. The trick is a common mathematical manipulation of multiplying and dividing by the same thing, essentially not changing the expression, but making it possible to manipulate the expression

[1] Or integral in case of continuous latent RVs.

[2] This terms also applies to integrals without closed form in the continuous case.

in a favourable way. Following this, we can write

$$\log p(\mathbf{x};\theta) = \log \sum_{\mathbf{z}} p(\mathbf{x},\mathbf{z};\theta) \tag{14}$$

$$= \log \sum_{\mathbf{z}} p(\mathbf{x},\mathbf{z};\theta)\frac{q(\mathbf{z})}{q(\mathbf{z})} \tag{15}$$

$$= \log \mathbb{E}_q\left[\frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z})}\right] \tag{16}$$

$$\geq \mathbb{E}_z\left[\log \frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z})}\right] \tag{17}$$

$$= \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z})}, \tag{18}$$

for any $q(\mathbf{z})$. Now that we have a lower bound to the true log-likelihood, it remains to choose $q(\mathbf{z})$ wisely so that the lower bound is as tight as possible. By rewriting Eq. 18 as

$$\mathcal{L}(q,\theta) = \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z})} = -\mathbb{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x};\theta)) + \log p(\mathbf{x};\theta) \tag{19}$$

one can notice (ignoring the second term since it does not depend on $q(\mathbf{z})$) that the inequality becomes equality for $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x};\theta)$. However, θ parameterising this $q(\mathbf{z})$ is exactly what we wanted to estimate from the very beginning and now we need it to compute the lower bound, which we need to estimate θ, which we need to compute lower bound etc. This cyclic reasoning implies an iterative algorithm that will at one stage (E-step) re-estimate

$$q^t = \arg\max_q \mathcal{L}(q,\theta^t),$$

where θ^t is fixed and comes from the previous iteration[3], and at the next stage (M-step) re-estimate

$$\theta^{t+1} = \arg\max_\theta \mathcal{L}(q^t,\theta),$$

where now q^t is fixed and comes from the current iteration. This algorithm is better known as expectation-maximization (EM), although it is here actually presented as maximization-maximization. *Note:* Now that we have introduced an iterative scheme, we simply have that $q^t = p(\mathbf{z}|\mathbf{x};\theta^t)$, which can somewhat simplify the expression for $\mathcal{L}(q,\theta)$ at the M-step. By plugging q^t in $\mathcal{L}(q,\theta)$, one can easily show that

$$\mathcal{L}(q^t,\theta) = Q(\theta,\theta^t) + c, \tag{20}$$

where $Q(\theta,\theta^t) = \mathbb{E}_{q^t}[\log p(\mathbf{x},\mathbf{z};\theta)]$, also called expected complete log-likelihood[4], and c is the entropy of q^t which is constant w.r.t. θ[5]. Hence, it is actually enough to maximize only $Q(\theta,\theta^t)$ with respect to θ at the M-step instead of the whole $\mathcal{L}(q^t,\theta)$.

[3] At the first iteration we initialize θ^t.

[4] It is this expectation that motivates the name E-step.

[5] Here it is crucial to distinguish between fixed θ^t from the previous iteration that q^t depends on and θ as a variable parameter to be maximized w.r.t.

The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.

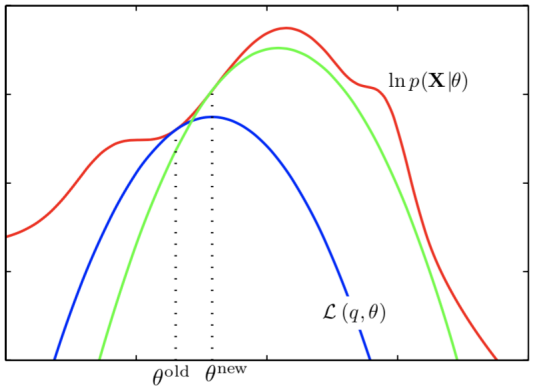


Figure 14: Maximizing the lower bound of likelihood by using EM
from Bishop [1]

The pseudo code of the EM algorithm is shown in Alg. 1. And, Fig. 14 shows how EM is used as maximizing the lower bound of likelihood, instead of maximizing the likelihood. Note the following:
Convergence condition: not increasing the likelihood or reaching the max number of iterations.
Convergence: local maximum.
Initialization is important: can start from a poor point (initialization) and reach a poor local maximum.

Algorithm 1 EM

procedure LEARN(X):
Initialise θ
  repeat
    E-step: calculate lower bound: $Q(\theta, \theta^{old}) = E_{Z|X,\theta^{old}}\big[P(X, Z|\theta)\big]$
    M-step: update θ by maximizing the lower bound: $\arg\max_\theta Q(\theta, \theta^{old})$
  until convergence
return θ

6 Expectation-Maximization – Exercises

6.1 Mixture of scale mixtures

Usual Gaussian mixture model (GMM) is represented as a weighted sum of k components, each being a Gaussian distribution with some mean and variance. The goal in such model is to estimate the mixture weights, as well as all means and variances. One can extend this model to be more flexible by making each component a mixture on its own. Namely, instead of components being Gaussian distributions, we could instead model them as mixtures of Gaussians with the same mean, but different variances (so-called scale mixtures). Graphical model in plate notation for this setup is given in Figure 15.

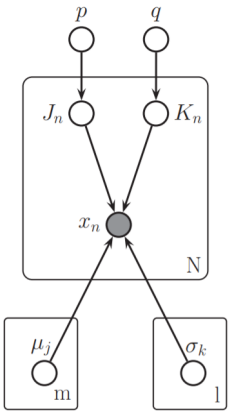


Figure 15: Mixture of scale mixtures

We can interpret J_n as the latent variable specifying which mean to use, and analogously K_n which variance to use. With this notation we can formally define the scale mixture as

$$p(x_n|J_n = j) = \sum_{k=1}^{l} q_k \mathcal{N}(\mu_j, \sigma_k^2), \tag{21}$$

where $q_j \coloneqq p(K_n = k)$, and the mixture of scale mixtures as

$$p(x_n) = \sum_{j=1}^{m} p_j \left[\sum_{k=1}^{l} q_k \mathcal{N}(\mu_j, \sigma_k^2) \right], \tag{22}$$

where $p_j \coloneqq p(J_n = j)$, for all n.[6]

Task 1. *Given the observations* $\mathbf{x} = \{x_n\}_{n=1}^N$ *estimate using EM algorithm parameters* $\theta = \{p_1, \ldots, p_m, \mu_1, \ldots, \mu_m, q_1, \ldots, q_l, \sigma_1^2, \ldots, \sigma_l^2\}$.

[6]Note that Eq. 21 and 22 are simple applications of marginalization and conditioning, e.g. for 21 $p(x_n|J_n = j) = \sum_{k=1}^{l} p(x_n, K_n = k|J_n = j) = \sum_{k=1}^{l} p(x_n|J_n = j, K_n = k)p(K_n = k)$.

Solution:

Referring now to the theory part in Section 5, we recall having the E-step and the M-step. In the E-step, we were looking for the optimal q^t, that has been shown to be $p(\mathbf{x}, \mathbf{z}; \theta^t)$, here in particular for $\mathbf{z} = \{\mathbf{J}, \mathbf{K}\}$, where $\mathbf{x} = \{x_n\}_{n=1}^N$, $\mathbf{J} = \{J_n\}_{n=1}^N$ and $\mathbf{K} = \{K_n\}_{n=1}^N$. We can even somewhat simplify this term by observing that $J_n \perp K_n$. Omitting θ^t to keep the notation uncluttered and assuming the i.i.d. data[7], we have that

$$p(\mathbf{x}, \mathbf{J}, \mathbf{K}) = p(\mathbf{x}|\mathbf{J}, \mathbf{K})p(\mathbf{J})p(\mathbf{K}) \tag{23}$$

$$= \prod_{n=1}^{N} \{p(x_n|J_n, K_n)p(J_n)p(K_n)\} \tag{24}$$

$$= \prod_{n=1}^{N} \prod_{k=1}^{l} \prod_{j=1}^{m} \{p(x_n|J_n = j, K_n = k)p(J_n = j)p(K_n = k)\}^{\mathbb{1}(J_n=j, K_n=k)}, \tag{25}$$

where $\mathbb{1}(J_n = j, K_n = k)$ is the indicator function specifying what mean and variance are being considered for a given x_n (since J_n and K_n are otherwise not observed)[8]. For simplicity of the expressions and to avoid numerical underflow in implementation, we can work in the log-space, where we have

$$\log p(\mathbf{x}, \mathbf{J}, \mathbf{K}) = \sum_{n=1}^{N} \sum_{k=1}^{l} \sum_{j=1}^{m} \mathbb{1}(J_n = j, K_n = k)\big\{ \log p(\mathbf{x}_n|J_n = j, K_n = k) \tag{26}$$

$$+ \log p(J_n = j) + \log p(K_n = k)\big\}. \tag{27}$$

Now that we have optimal q^{old}, we can compute $Q(\theta, \theta^{old})$ that we showed is enough to maximize to obtain optimal θ in the M-step. Namely, we have that

$$Q(\theta, \theta^t) = \mathbb{E}_{J,K|x,\theta^{old}}[\log p(\mathbf{x}, \mathbf{J}, \mathbf{K})] \tag{28}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{l} \sum_{j=1}^{m} \mathbb{E}_{J,K|x,\theta^{old}}[\mathbb{1}(J_n = j, K_n = k)]\big\{ \log \mathcal{N}(\mu_j, \sigma_k^2) + \log p_j + \log q_k \big\} \tag{29}$$

where

$$\mathbb{E}_{J,K|x,\theta^{old}}[\mathbb{1}(J_n = j, K_n = k)] = \sum_{j'=1}^{m} \sum_{k'=1}^{l} \mathbb{1}(J_n = j, K_n = k)p(J_n = j', K_n = k'|x_n, \theta^{old}) \tag{30}$$

$$= p(J_n = j, K_n = k|x_n, \theta^{old}) \tag{31}$$

$$= \frac{p(J_n = j, K_n = k, x_n|\theta^{old})}{p(x_n|\theta^{old})} \tag{32}$$

$$= \frac{p_j^{old} q_k^{old} \mathcal{N}(\mu_j^{old}, (\sigma_k^{old})^2)}{\sum_{j'=1}^{m} \sum_{k'=1}^{l} p_{j'}^{old} q_{k'}^{old} \mathcal{N}\left(\mu_{j'}^{old}, (\sigma_{k'}^{old})^2\right)}, \tag{33}$$

where the superscript ^{old} denotes that these values are parameter estimates from the previous iteration, reflecting that the expectation is taken w.r.t q^{old}. Let us also for convenience of

[7] Formally, it is assumed that random variables $X_1, \ldots, X_N$, whose realizations are $x_1, \ldots, x_N$, are i.i.d.

[8] Note that now the probability densities can simply be evaluated, and the expectation needed to compute Q is acting only on the indicator.

notation denote $\gamma_{j,k,n} \coloneqq \mathbb{E}_{J,K|x,\theta^{old}}[\mathbb{1}(J_n = j, K_n = k)]$. Finally, by unraveling $\log \mathcal{N}(\mu_j, \sigma_k^2)$ the Q-function can be explicitly written as

$$Q(\theta, \theta^{old}) = \sum_{n=1}^{N} \sum_{k=1}^{l} \sum_{j=1}^{m} \gamma_{j,k,n} \Big(-\frac{1}{2} \log \sigma_k^2 - \frac{1}{2} \log(2\pi) - \frac{1}{2} \frac{(x_n - \mu_j)^2}{\sigma_k^2} + \log p_j + \log q_k \Big) \tag{34}$$

Now we can find the update expressions for the parameters (i.e. perform the M-step) by maximizing the obtained Q-function. To find the parameter μ_j and σ_k^2 maximizing Q, one can take the corresponding partial derivatives and solve $\partial_{\mu_j} Q = 0$ and $\partial_{\sigma_k^2} Q = 0$. For p_j and q_k, one needs to be slightly more careful – since these parameters are in fact probabilities, one must satisfy the constraints that $\sum_{j=1}^{m} p_j = 1$ and $\sum_{k=1}^{l} q_k = 1$. How to tackle this, we will show later on. Attentive reader[9] could have also notice that we cheated a bit here; namely one should actually jointly maximize Q w.r.t. θ, and not w.r.t. each parameter separately keeping the others fixed. As it turns out, this is rather difficult, which is why it is a common practice to take the suggested approach, which is sometimes referred to as generalized EM. Let us first find the update expression for μ_j. Note that j is fixed here, hence no summation over it, so we have

$$\partial_{\mu_j} Q \overset{+}{=} \partial_{\mu_j} \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \left(-\frac{(x_n - \mu_j)^2}{2\sigma_k^2} \right) = \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \frac{x_n - \mu_j}{\sigma_k^2} = 0 \tag{35}$$

$$\iff \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \frac{x_n}{\sigma_k^2} = \mu_j \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \frac{1}{\sigma_k^2} \tag{36}$$

$$\iff \mu_j^{new} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \frac{x_n}{\sigma_k^2}}{\sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} \frac{1}{\sigma_k^2}}, \tag{37}$$

where we used $\overset{+}{=}$ to denote equality up to an additive constant. Similarly, for σ_k^2 we have

$$\partial_{\sigma_k^2} Q \overset{+}{=} \partial_{\sigma_k^2} \sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n} \left(-\frac{\log \sigma_k^2}{2} - \frac{(x_n - \mu_j)^2}{2\sigma_k^2} \right) = \sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n} \left(-\frac{1}{2\sigma_k^2} + \frac{(x_n - \mu_j)^2}{2(\sigma_k^2)^2} \right) = 0 \tag{38}$$

$$\iff \frac{1}{2\sigma_k^2} \sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n} = \frac{1}{2(\sigma_k^2)^2} \sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n} (x_n - \mu_j)^2 \tag{39}$$

$$\iff (\sigma_k^2)^{new} = \frac{\sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n} (x_n - \mu_j)^2}{\sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n}}. \tag{40}$$

Now, as mentioned, to obtain the update equations for p_j and q_k we have to solve a constrained optimization problem, where the constraints are as given before. Such problems are usually approached using Lagrangian multipliers that convert the problem to unconstrained optimization where the new objective is $\mathcal{L}(x, \lambda) = f(x) + \lambda g(x)$, where $f(x)$ is the old objective, $g(x)$ constraint and λ the Lagrangian multiplier. Since $\mathcal{L}$ depends on both x and λ, it is necessary to optimize w.r.t. to both, and this can again be done with partial derivatives.

[9] `http://www.ma.rhul.ac.uk/~uvah099/Sat/reader.html`

Applying this to our setup, we have the objective for p_j

$$\mathcal{L}(p_j, \lambda) = Q(\theta, \theta^t) - \lambda \left(-1 + \sum_{j'=1}^{m} p_{j'} \right) \tag{41}$$

$$\stackrel{+}{=} \sum_{n=1}^{N} \sum_{k=1}^{l} \{\gamma_{j,k,n} \log p_j\} - \lambda \left(-1 + \sum_{j'=1}^{m} p_{j'} \right). \tag{42}$$

Firstly, the partial derivative w.r.t. p_j gives

$$\partial_{p_j} \mathcal{L}(p_j, \lambda) = \frac{1}{p_j} \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n} - \lambda = 0 \tag{43}$$

$$p_j = \frac{\sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n}}{\lambda}. \tag{44}$$

We can see that the update of p_j depends on λ. To obtain the explicit expression for λ, we first take the partial derivative of $\mathcal{L}$ w.r.t. it

$$\partial_{\lambda} \mathcal{L}(p_j, \lambda) = 1 - \sum_{j'=1}^{m} p_{j'} = 0 \tag{45}$$

$$\sum_{j'=1}^{m} p_{j'} = 1, \tag{46}$$

which can be used to obtain

$$\lambda = \lambda \cdot 1 = \lambda \sum_{j'=1}^{m} p_{j'} = \sum_{j'=1}^{m} p_{j'} \lambda = \sum_{j'=1}^{m} \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j',k,n}, \tag{47}$$

where the last equality comes from rearranging Equation 44. Hence, the update equation for p_j is given by

$$p_j^{new} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j,k,n}}{\sum_{j'=1}^{m} \sum_{n=1}^{N} \sum_{k=1}^{l} \gamma_{j',k,n}}. \tag{48}$$

One can in the exactly same manner obtain the update equation for q_k. For brevity we omit these calculations and conclude with the update equation

$$q_k^{new} = \frac{\sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k,n}}{\sum_{k'=1}^{l} \sum_{n=1}^{N} \sum_{j=1}^{m} \gamma_{j,k',n}}. \tag{49}$$

7 Variational Inference – Theory

A deterministic approximate inference algorithm.

7.1 Motivation

We can't compute the posterior for many interesting models. For example for the Bayesian mixture of Gaussian, we draw $z_i \sim Mult(\pi)$ and $x_i \sim N(\mu_{zi}, \sigma^2)$ resulting in the following posterior [2]:

$$p(\mu_{1:K}, z_{1:n}|x_{1:n}) = \frac{\prod_k p(\mu_k) \prod_i p(z_i)p(x_i|z_i, \mu_{1:K})}{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_k p(\mu_k) \prod_i p(z_i)p(x_i|z_i, \mu_{1:K})} \tag{50}$$

The denominator is the problem; even if we take the summation outside, this is intractable when n is reasonably large.

7.2 Main idea

So far we used EM and did point estimate for the model parameters; now we want to find posterior distribution for the **unknown** model parameters and hidden variables. For a DGM with observations X, hidden variables Z and model parameters Θ, want to pick an approximation $q(Z, \Theta)$ to the distribution from some tractable family, and then to try to make this approximation as close as possible to the true posterior. This reduces inference to an optimization problem [4]. We measure the closeness of the two distributions q and p with Kullback-Leibler (KL) divergence.

$$\mathbb{KL}(q||p) = \sum_z q(Z, \Theta) log \frac{q(Z, \Theta)}{p(Z, \Theta|X)} \tag{51}$$

$$\sum_z q(Z, \Theta) log \frac{q(Z, \Theta)}{p(Z, \Theta|X)} = -E_Z\big[\log p(X, Z, \Theta)\big] + E_Z\big[\log q(Z, \Theta)\big] + \log p(X) \tag{52}$$

We cannot actually minimize KL divergence in Eq. 51 but since we have Eq. 52, we maximise lower bound of log likelihood, Evidence Lower Bound (ELBO) [2], which is (same as equation 18):

$$\text{ELBO}(q) = E_Z\big[\log P(X, Z, \theta)\big] - E_Z\big[\log q(Z, \theta)\big]. \tag{53}$$

Note that we can use ELBO for convergence test at each iteration i.e. the difference of the ELBO of current value and the previous one should be smaller than some small epsilon.

In mean field Variational Inference, we assume that the variational family factorizes,

$$q(Z_1, ..., Z_n, \Theta_1, ..., \Theta_k) = \prod_i q(Z_i) \prod_k q(\Theta_k) \tag{54}$$

Now for each update equation of $q(z_i)$ we perform the expectation, over the log of the joint distribution, w.r.t. all the hidden variables except the one we are deriving the approximate posterior for. For example, we obtain the update equation for z_j, $q(z_j)$, by calculating $E_{-z_j}\big(\log P(X, Z, \Theta)\big)$. Here we explain [2] that this is the result of the maximization of ELBO. We rewrite the ELBO and then take the partial derivative and set it to zero.

If we call the set of l latent variables and parameters, Ψ, and use the chain rule, we can write $p(X, \Psi) = p(X) \prod_j p(\Psi_j | \Psi_{1:j-1}, X)$. Also because of the independence we have: $\log q(\Psi) = \log q(\Psi_1) ... q(\Psi_l) = \sum_j q(\Psi_j)$ Now the ELBO becomes:

$$\text{ELBO}(q) = \log P(X) + \sum_j \Big(E\big[\log P(\Psi_j | \Psi_{1:j-1}, X) \big] - E_j\big[\log q(\Psi_j) \big] \Big) \tag{55}$$

Now if you write the ELBO as a function of the last variable in the chain, say Ψ_k, you get:

$$\mathcal{L}(q(\Psi_k)) = const + E\big[\log P(\Psi_k | \Psi_{-k}, X) \big] - E\big[\log q(\Psi_k) \big] = \tag{56}$$

$$\int q(\Psi_k) E_{-k}[\log P(\Psi_k | \Psi_{-k}, X)] d\Psi_k - \int q(\Psi_k) \log q(\Psi_k) d\Psi_k \tag{57}$$

Taking the partial derivative of the integrals in 57 and setting it to zero you get the update equation for each posterior during the coordinate ascent algorithm:

$$q^*(\Psi_k) \propto E_{-k}[\log P(\Psi_k | \Psi_{-k}, X)] \propto E_{-k}[\log P(\Psi, X)] \tag{58}$$

8 Variational Inference – Exercises

The Cartesian Matrix Model (CMM) is defined as follows. There are R row distributions $\{N(\mu_r, \lambda_r^{-1}) : 1 \leq r \leq R\}$, each variance λ_r^{-1} is known and each μ_r has prior distribution $N(\mu, \lambda^{-1})$. There are also C column distributions $\{N(\xi_r, \tau_c^{-1}) : 1 \leq c \leq C\}$, each variance τ_c^{-1} is known and each ξ_c has prior distribution $N(\xi, \tau^{-1})$. All hyper-parameters are known. A matrix S is generated by, for each row $1 \leq r \leq R$ and each column $1 \leq c \leq C$, setting $S_{rc} = X_r + Y_c$ where X_r is sampled from $N(\mu_r, \lambda_r^{-1})$ and Y_c from $N(\xi_r, \tau_c^{-1})$. Use Variational Inference in order to obtained a variational distribution

$$q(\mu_1, \ldots, \mu_R, \xi_1, \ldots, \xi_C) = \prod_r q(\mu_r) \prod_c q(\xi_c)$$

that approximates $p(\mu_1, \ldots, \mu_R, \xi_1, \ldots, \xi_C | S)$. Tip: what distribution do you get from the sum of two Gaussian random variables? What is the relation between the means?

Question 15: *Present the algorithm written down in a formal manner (using both text and mathematical notation, but not pseudo code).*

Figure 16: From Assignment 2, 2017

Question: See the question 15 from Figure 16.
Solution:
Since each cell of the matrix denoted as S_{rc} is a sum of two Gaussians then as we know we can get the Gaussian result where the mean is the sum of the mean of X and the mean of Y. The variance is also summed. So by having μ_r and ϵ_c, the distribution of each cell can be expressed as $\mathcal{N}(\mu_r + \epsilon_c, \lambda_r^{-1} + \tau_c^{-1})$. Below are the calculations of $q_l(\mu_l)$ and $q_t(\epsilon_t)$ according to variational approximation (l and t represent any relevant index for calculating all of the possible $q_l(\mu_l)$ and $q_t(\epsilon_t)$). The known parameters are represented with $\Theta = \{\lambda_{1:R}^{-1}, \lambda^{-1}, \mu, \tau_{1:C}^{-1}, \tau^{-1}, \epsilon\}$.

$$\ln q_l(\mu_l) = E_{\substack{\epsilon_{1:C} \\ \mu_{1:R\neg l}}}[\ln p(S, \mu_{1:R}, \epsilon_{1:C}|\Theta)] + const$$
$$\ln q_t(\epsilon_t) = E_{\substack{\mu_{1:R} \\ \epsilon_{1:C\neg t}}}[\ln p(S, \mu_{1:R}, \epsilon_{1:C}|\Theta)] + const$$

To calculate the above we first calculate $\ln\ q_l(\mu_l)$ which needs in return to first calculate $p(S, \mu_{1:R}, \epsilon_{1:C})$ (Note that we choose indexes l and later t to denote a specific index so that it is not mixed with the indexes in the sum and product terms):

$$p(S, \mu_{1:R}, \epsilon_{1:C}|\Theta) = p(S|\mu_{1:R}, \epsilon_{1:C}, \Theta)p(\mu_{1:R}, \epsilon_{1:C}|\Theta) =$$
$$\prod_{c=1}^{C}\prod_{r=1}^{R} p(S_{rc}|\mu_r + \epsilon_c, \lambda_r^{-1} + \tau_c^{-1}) \prod_{r=1}^{R} p(\mu_r|\mu, \lambda^{-1}) \prod_{c=1}^{C} p(\epsilon_c|\epsilon, \tau^{-1})$$

Notice that each μ_r and each ϵ_c are independent in the above expression, resulting in factors. Now we calculate $\ln\ q_l(\mu_l)$ by taking the expectation over the log of the expression above:

$$\ln q_l(\mu_l) = -\frac{1}{2} E_{\neg\mu_l}\left[\sum_{c=1}^{C}\sum_{r=1}^{R} \frac{(S_{rc} - (\mu_r + \epsilon_c))^2}{\lambda_r^{-1} + \tau_c^{-1}} + \frac{1}{\lambda^{-1}}\sum_{r=1}^{R}(\mu_r - \mu)^2 + \frac{1}{\tau^{-1}}\sum_{c=1}^{C}(\epsilon_c - \epsilon)^2\right] + const$$

All the components of r in $\sum_{r=1}^{R}$ are independent on μ_l except $r = l$; adding those independent terms in the above into constant results in:

$$-\frac{1}{2} E_{\epsilon}\left[\sum_{c=1}^{C} \frac{S_{lc}^2 - 2S_{lc}(\mu_l + \epsilon_c) + (\mu_l^2 + \epsilon_c^2 + 2\mu_l\epsilon_c)}{\lambda_l^{-1} + \tau_c^{-1}} + \lambda(\mu_l^2 + \mu^2 - 2\mu\mu_l) + \frac{1}{\tau^{-1}}\sum_{c=1}^{C}(\epsilon_c - \epsilon)^2\right]$$
$$+ const$$

The μ_l-independent terms in the above expression can be pushed into constant resulting in (the expectation goes through the terms as below):

$$\ln q_l(\mu_l) = -\frac{1}{2} E_{\epsilon}\left[\sum_{c=1}^{C} \frac{-2S_{lc}(\mu_l) + (\mu_l^2 + 2\mu_l\epsilon_c)}{\lambda_l^{-1} + \tau_c^{-1}} + \lambda(\mu_l^2 - 2\mu\mu_l)\right] + const$$
$$= -\frac{1}{2} E_{\epsilon}\left[\left(\lambda + \sum_{c=1}^{C}\frac{1}{\lambda_l^{-1} + \tau_c^{-1}}\right)\mu_l^2 - 2\mu_l\left(\mu\lambda + \sum_{c=1}^{C}\frac{S_{lc} - \epsilon_c}{\lambda_l^{-1} + \tau_c^{-1}}\right)\right] + const$$
$$= -\frac{1}{2}\left[\mu_l^2\left(\lambda + \sum_{c=1}^{C}\frac{1}{\lambda_l^{-1} + \tau_c^{-1}}\right) - 2\mu_l\left(\mu\lambda + \sum_{c=1}^{C} E_{\epsilon_c}\left[\frac{S_{lc} - \epsilon_c}{\lambda_l^{-1} + \tau_c^{-1}}\right]\right)\right] + const$$
$$= -\frac{1}{2}\left[\mu_l^2\left(\lambda + \sum_{c=1}^{C}\frac{1}{\lambda_l^{-1} + \tau_c^{-1}}\right) - 2\mu_l\left(\mu\lambda + \sum_{c=1}^{C}\frac{S_{lc} - E_{\epsilon_c}[\epsilon_c]}{\lambda_l^{-1} + \tau_c^{-1}}\right)\right] + const$$

By completing the square, the above expression becomes in a form of Gaussian with mean

m_l and precision p_l, where

$$p_l = \lambda + \sum_{c=1}^{C} \frac{1}{\lambda_l^{-1} + \tau_c^{-1}}$$

$$m_l = \frac{\mu\lambda + \sum_{c=1}^{C} \frac{S_{lc} - E_{\epsilon_c}[\epsilon_c]}{\lambda_l^{-1} + \tau_c^{-1}}}{p_l}$$

Similarly we can calculate $\ln q_t(\epsilon_t)$ and so it results in a Gaussian with mean m_t and precision p_t, where

$$p_t = \tau + \sum_{r=1}^{R} \frac{1}{\tau_t^{-1} + \lambda_r^{-1}}$$

$$m_t = \frac{\epsilon\tau + \sum_{r=1}^{R} \frac{S_{rt} - E_{\mu_r}[\mu_r]}{\tau_t^{-1} + \lambda_r^{-1}}}{p_t}$$

Since we have approximated $q(\mu_l)$, we have R number of estimations each having similar result as in m_l and p_l. Having $q(\epsilon_t)$ form, we have C number of estimations each having similar result as in m_t and p_t; thus, we can calculate $\prod_{r=1}^{R} q(\mu_r) \prod_{c=1}^{C} q(\epsilon_c)$.

A TL;DR

We encourage you to know the theory of each method and algorithm. We also prepared a small cheat sheet to help.

A.1 Conditional Independence in Graphical Models

(See Section 1 for details)
Use D-separation or "Method 1" to check conditional independence in graphical models. Don't forget to pay attention to the plate notation.

A.2 Hidden Markov Model

(See Section 3 for details)

- The summing, e.g. over z_{n-1}, implies $\sum_{z_{n-1}=1}^{J}$.
- The known probabilities in the inference (forward and backward) are: initial, transition, and emission probabilities.
- Equations 13.24 until 13.31 in [1] are useful to perform the forward and backward passes.
- In $\alpha(z_n)$, we mean that the z_n is observed.
- More advanced stuff: we assume a stationary distribution for HMM and that the HMM is finite in sequence length. For the definitions, see [4] section 17.2.3 Stationary distribution of a Markov chain.
- An HMM can be formulated as a GMM and vice versa. To understand this better, you need to understand mixture models, see [1, 4].
- To perform EM for HMM, you need to calculate the marginal of the two consecutive hidden variables too. See [1] for the calculation w.r.t. HMM. For IOHMM, you should derive it yourself.
- For solving IOHMM in an easier way, see factor graph representation in [1] section 13.2.3.

A.3 Expectation-Maximization (EM) Algorithm

(See Section 5 for details)
Goal: Find `point estimation` of parameters ($\theta^* = \{\theta_1^*, \ldots\}$) with an iterative process.
$X = \{X_1, \ldots, X_N\}$ are observed variables, $Z = \{Z_1, \ldots\}$ are latent variables, $\theta = \{\theta_1, \ldots\}$ are the parameters.
At each iteration

- **E-Step:**
  - Write complete-data log-likelihood ($logP(X, Z|\theta)$)
  - Write Q function ($Q(\theta, \theta^{old}) = E_{Z|X,\theta^{old}}[logP(X, Z|\theta)]$) and move E symbol inside (pay attention to the subscript of expectation, $Z|X, \theta^{old}$, it helps you to identify which terms' expectation you should calculate)

- Calculate γ value, based on previous iteration's parameters θ^{old} (i.e, $\gamma_{n,k} = E_{Z|X,\theta^{old}}[I(\ldots)]$) and put γ in the Q function

- **M-Step:**
  - Update parameters, which maximizes the Q function (i.e $\theta_1^* = argmax_{\theta_1} Q(\theta, \theta^{old})$) by simply taking partial derivative and setting it to zero (pay attention to the parameters with constraints, i.e $\sum_j \theta_{1,j} = 1$)

A.4 Variational Inference (VI)

(See Section 7 for details)
Goal: Instead of computing the exact posterior distribution of the parameters and latent variables ($P(Z, \theta|X)$) you obtain new, simpler `distributions` for parameters ($q(Z_1, \ldots, \theta_1, \ldots) = \prod_i q(Z_i) \prod_j q(\theta_j)$) with an iterative process.
$X = \{X_1, \ldots, X_N\}$ are observed variables, $Z = \{Z_1, \ldots\}$ are latent variables, $\theta = \{\theta_1, \ldots\}$ are the parameters, $\xi = \{\xi_1, \ldots\}$ are the hyperparameters (that you know). We represent the terms we are interested in as $\Psi = \{Z, \theta\}$.
At each iteration

- Write log probability of joint distribution ($logP(X, R, \theta|\xi)$)

- Find the new distribution of the parameters/latent variables one-by-one (i.e $q(\theta_j)$ or $q(Z_i)$) by
  - Take the expectation of log joint ($E_{\Psi \neg \theta_j}[logP(X, R, \theta|\xi)]$) w.r.t the rest of the parameters
  - Find a probability distribution which matches the pattern of a well-known distribution (Gaussian, Gamma etc) and identify the new hyperparameters (i.e $q(\theta_j) = Ga(\theta_j|\alpha^*, \beta^*)$)
  - If the new hyperparameter includes moment [10] of another parameter (i.e $q(\theta_k) = N(\theta_k|0.6E[\theta_j^2], \sigma^*)$), check the distribution of θ_j (i.e $Ga(\theta_j|\alpha^*, \beta^*)$) find the moment's value ($E[\theta_j^2] = \frac{\alpha^*(\alpha^*+1)}{(\beta^*)^2}$) and place it to appropriate place ($q(\theta_k) = N(\theta_k|0.6E[\theta_j^2], \sigma^*) = N(\theta_k|0.6\frac{\alpha^*(\alpha^*+1)}{(\beta^*)^2}, \sigma^*)$))

A.5 Useful Properties

- $E[X + Y] = E[X] + E[Y]$ [6]
- $E[aX] = aE[X]$ [6]
- $E[I[X = x]] = \sum_x I[X = x]P(X = x) = P(X = x)$ [7]

[10] For instance the first three moments of a Normal distribution $N(x|\mu, \sigma^2)$ are $E[X] = \mu, E[X^2] = \mu^2 + \sigma^2, E[X^3] = \mu^3 + 3\mu\sigma^2$, the first two moments of a Gamma distribution $Ga(x|a, b)$ are $E[X] = \frac{a}{b}, E[X^2] = \frac{a(a+1)}{b^2}$.

References

[1] C. Bishop. *Pattern Recognition and Machine learning*. Springer, 2006.

[2] David M. Blei. *Variational Inference*. `https://www.cs.princeton.edu/courses/archive/fall11/cos597C/lectures/variational-inference-i.pdf`. [Online; accessed 20-November-2019]. 2011.

[3] *d-separation*. `http://web.mit.edu/jmn/www/6.034/d-separation.pdf`. [Online; accessed 12-December-2019].

[4] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[5] E Weisstein. *Second derivative test. From MathWorld–A Wolfram Web Resource*. 2004.

[6] Wikipedia. *Expected value*. `https://en.wikipedia.org/wiki/Expected_value`. [Online; accessed 12-December-2019].

[7] Wikipedia. *Indicator function*. `https://en.wikipedia.org/wiki/Indicator_function`. [Online; accessed 12-December-2019].