Exercise Booklet

CMPE 58K, Bayesian Statistics and Machine Learning

Instructor: A. Taylan Cemgil
TAs: Arman Boyaci and Serhan Danis

October 10, 2012

List of exercises

Q1*: Quiz Question

Let x_1 and x_2 are two discrete random variables taking values in $\{-1, 1\}$. We know that $p(x_1 = -1|x_2 = -1) = 1/4$, $p(x_1 = 1|x_2 = 1) = 2/3$, $p(x_2 = -1|x_1 = 1) = 3/7$ and $p(x_2 = 1|x_1 = -1) = 2/3$. Show all your work.

1. Find the following quantities
   a) Joint: $p(x_1, x_2)$
   b) Marginals: $p(x_1)$, $p(x_2)$
   c) Max-marginal: $\max_{x_1} p(x_1, x_2)$
   d) Covariance of x_1 and x_2
2. Are x_1 and x_2 independent ? Why or why not?

Return to List of exercises. Return to List of exercises.

Q2*: Coin

Suppose a biased coin with $p(\text{head}) = \pi$ is thrown N times. The number of times head shows up is 4. Assume all π and N are a-priori equally likely. Find analytically or via computation

1. the most likely value of N as a function of π.
2. the marginal distribution of N.

Return to List of exercises. Return to List of exercises.

Q3*: `logsumexp`

Implement a function in MATLAB with the following specification:

```
%LOG_SUM_EXP Numerically stable computation of log(sum(exp(X), dim))
% [r] = log_sum_exp(X, dim)
%
% Inputs :
%     X : Array
%     dim : Sum Dimension <default = 1>
%           Row vector sums should be calculated
%           by transposing or specifying dim=2
%
% Outputs:
%   r : log(sum(exp(X), dim))
%
% Usage Example : [s] = log_sum_exp([-10 -9]');
%   log(sum(exp([-1213 -1214])))
%     Warning: Log of zero.
%
%   log_sum_exp([-1213 -1214], 2)
%    ans = -1.2127e+003
```

Return to List of exercises. Return to List of exercises.

Q4*: `randgen`

Implement a function in MATLAB that generates independent random samples from a specified distribution:

```
%RANDGEN Random samples with replacement from a specified distribution
%   Y = RANDGEN(S, Siz, P) returns a weighted sample, using positive
%   weights P. P is often a vector of probabilities but can be unnormalised.
%   If P is absent we assume a uniform distribution
%
%   Example
%   -------
%   Generate a random sequence of the characters ACGT, with
%   replacement, according to specified probabilities.
%   R = randgen('ACGT',48, [0.15 0.35 0.35 0.15])
%
%   Example
%   -------
%   Generate a random 3 by 3 matrix with independent
%   entries from S = [1 2 5] according to specified weights.
%   R = randgen([1 2 5], [3 3], [2 2 1])
%   So on average there should be about twice as many one's as five's.
```

Don't use MATLAB *statistics toolbox function* `randsample` *as a subroutine. However, you are welcome to read the source code and use the ideas in your implementation.*

Return to List of exercises. Return to List of exercises.

Q5*: Graphical Models

Consider the following probability model

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\phi_1(x_1, x_2)\phi_2(x_2, x_3)\phi_3(x_3, x_4)$$

1. Draw the associated undirected graphical model
2. Draw the associated factor graph
3. Suppose, each variable has two states. How many free parameters do we have?
4. Describe an efficient algorithm to compute Z
5. Describe an efficient algorithm to compute the marginals $p(x_i)$.
6. Sketch a variational Bayes algorithm for computing the approximate marginals and a lower bound for Z?

Return to List of exercises. Return to List of exercises.

Q6*: Medical Expert

This question aims at demonstrating the conceptual difficulties one is faced when trying to compile verbose and vague prior knowledge into a consistent probability model.

Suppose we wish to diagnose if a person has swine flu and the probability that she/he survives. Other possible diseases we wish to consider are regular seasonal flu, bronchitis, and other diseases.

The probability of survival is high if an infected person doesn't develop one or more of the following possible complications:

- pneumonia (an infection of the lungs),
- difficulty breathing, and
- dehydration.

All diseases can generate fever, but swine flu generates almost always fewer and causes on average higher temperature than the other diseases. Other possible symptoms of swine flu or regular flu are

- unusual tiredness,
- headache,
- runny nose,
- sore throat,
- shortness of breath or cough,
- loss of appetite,
- aching muscles,
- diarrhoea or vomiting.

The symptoms of bronchitis are

- A cough that is frequent and produces mucus
- A lack of energy
- A wheezing sound when breathing, which may or may not be present
- A fever, which may or may not be present

Other diseases may cause any of these symptoms but less likely all of them simultaneously. If the person has already swine flu and

- has a serious existing illness that weakens the immune system, such as cancer,
- is pregnant,

- is a child under age one,
- the condition suddenly gets much worse, or
- the condition is still getting worse after seven days

A person is very high risk if he/she has

- one or more chronic diseases (heart, kidney, liver, lung or neurological disorders include motor neurone disease, multiple sclerosis and Parkinson's disease),
- immunosuppression (whether caused by disease or treatment)
- diabetes mellitus.

Also at risk are:

- patients who have had drug treatment for asthma within the past three years,
- pregnant women,
- people aged 65 and older, and
- young children under five.

People under risk, if infected with the virus, have less probability of survival. Assume that for detecting zoonotic pathogens, such as the current strain of swine influenza H1N1, there are two tests: $T1$ and $T2$. $T1$ is cheap but it is not very reliable as it can not distinguish swine flu from seasonal flu. The other test is expensive but is more reliable.

1. Define the appropriate random variables to represent this scenario. You are not allowed to use more than 16 random variables so define your random variables and their state spaces carefully.
2. Draw the graphical model.

Return to List of exercises. Return to List of exercises.

Q7*: Bayes Theorem

Suppose that we have three coloured boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes.

1. If a box is chosen at random with probabilities p(r) = 0.2, p(b) = 0.2, p(g) = 0.6, and a piece of fruit is removed from the box (with equal probability of selecting any of the items in the box), then what is the probability of selecting an apple?
2. If we observe that the selected fruit is in fact an orange, what is the probability that it came from the green box?
3. Choose the appropriate random variables and draw a directed graphical model for this problem.

Return to List of exercises. Return to List of exercises.

Q8***: Game Show

On a game show, a contestant is told the rules as follows: There are three doors, labelled 1, 2, 3. A single prize has been hidden behind one of them. You get to select one door. Initially your chosen door will not be opened. Instead, the gameshow host will open one of the other two doors, and he will do so in such a way as not to reveal the prize. For example, if you first choose door 1, he will then open one of doors 2 and 3, and it is guaranteed that he will choose which one to open so that the prize will not be revealed. At this point, you will be given a fresh choice of door: you can either stick with your first choice, or you can switch to the other closed door. All the doors will then be opened and you will receive whatever is behind your final choice of door.

1. Imagine that the contestant chooses door 1 first; then the gameshow host opens door 3, revealing nothing behind the door, as promised. Should the contestant (a) stick with door 1, or (b) switch to door 2, or (c) does it make no difference?
2. Imagine that the game happens again and just as the gameshow host is about to open one of the doors a violent earthquake rattles the building and one of the three doors flies open. It happens to be door 3, and it happens not to have the prize behind it. The contestant had initially chosen door 1. Repositioning his toupée, the host suggests, 'OK, since you chose door 1 initially, door 3 is a valid door for me to open, according to the rules of the game; I'll let door 3 stay open. Let's carry on as if nothing happened.' Should the contestant stick with door 1, or switch to door 2, or does it make no difference? Assume that the prize was placed randomly, that the gameshow host does not know where it is, and that the door flew open because its latch was broken by the earthquake.
3. A similar alternative scenario is a gameshow whose confused host forgets the rules, and where the prize is, and opens one of the unchosen doors at random. He opens door 3, and the prize is not revealed. Should the contestant choose what's behind door 1 or door 2? Does the optimal decision for the contestant depend on the contestant's beliefs about whether the gameshow host is confused or not?
4. Formally derive the results defining the appropriate random variables and using the Bayes rule.

Return to List of exercises. Return to List of exercises.

Q9*: Twenty-Faced Dice

A die is selected at random from two twenty-faced dice on which the symbols 1–10 are written with nonuniform frequency as follows.

Symbol	1	2	3	4	5	6	7	8	9	10
Number of faces of die A	6	4	3	2	1	1	1	1	1	0
Number of faces of die B	3	3	2	2	2	2	2	2	1	1

1. The randomly chosen die is rolled 7 times, with the following outcomes:

   5, 3, 9, 3, 8, 4, 7.

   What is the probability that the die is die A?

2. Assume that there is a third twenty-faced die, die C, on which the symbols 1–20 are written once each. As above, one of the three dice is selected at random and rolled 7 times, giving the outcomes: 3, 5, 4, 8, 3, 9, 7.
   What is the probability that the die is die A, die B or die C?

3. Choose the appropriate random variables and draw directed graphical models for both problems.

Return to List of exercises. Return to List of exercises.

Q10***: Sums of Random Variables

1. Two ordinary dice with faces labelled $1 \ldots 6$ are thrown. What is the probability distribution of the sum of the values? What is the probability distribution of the absolute difference between the values?
2. One hundred ordinary dice are thrown. What, roughly, is the prob- ability distribution of the sum of the values? Sketch the probability distribution and estimate its mean and standard deviation.

   This exercise is intended to help you think about the central-limit theorem, which says that if independent random variables $x_1, \ldots x_N$ have means μ_n and finite variances σ_n^2, then, in the limit of large N, the sum $\sum_n x_n$ has a distribution that tends to a normal (Gaussian) distribution with mean $\sum_n \mu_n$ and variance $\sum_n \sigma_n^2$.

Return to List of exercises. Return to List of exercises.

Q11**: Jacobians

Consider a probability density $p_x(x)$ of a continuous random variable x. Suppose we make a nonlinear change of variable using $x = g(y)$, so that the density transforms according to

$$\begin{aligned} p_y(y) &= \left|\frac{dx}{dy}\right| p_x(x) \\ &= |g'(y)| p_x(g(y)) \end{aligned} \tag{1}$$

1. By differentiating Eq.1, show that the location y^* of the maximum of the density (in y) is **not** in general related to the location x^* of the maximum of the density over x by the simple functional relation $x^* = g(y^*)$.

   Note: This as a consequence of the Jacobian factor. This shows that the maximum of a probability density (in contrast to a simple function) is dependent on the choice of variable.

2. Verify that, in the case of a linear transformation, the location of the maximum transforms in the same way as the variable itself.

Return to List of exercises. Return to List of exercises.

Q12*: Covariance

We are given two random variables x and y

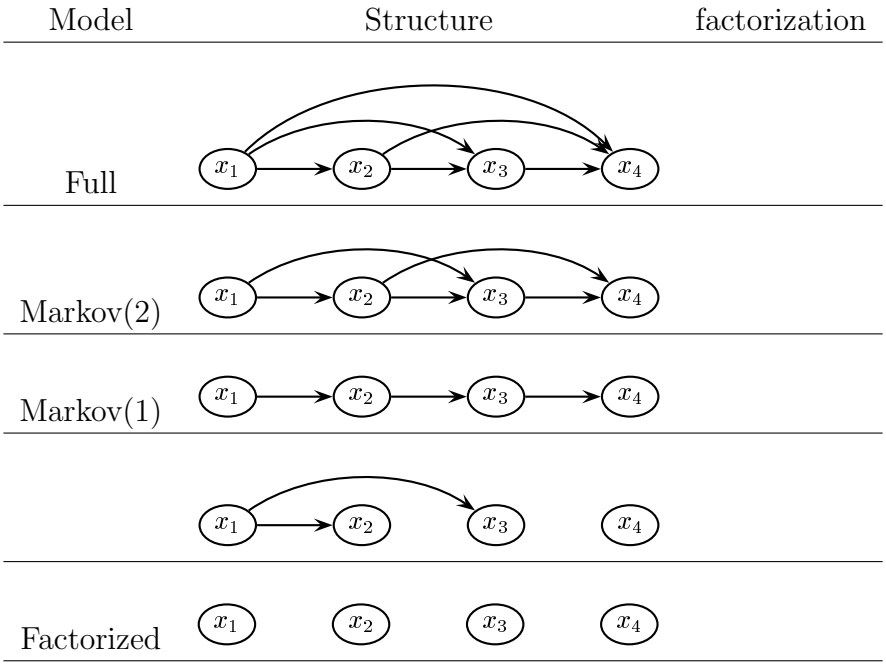
1. Show that if x and y are independent, then their covariance is zero.
2. Give an example joint density $p(x, y)$ where the covariance is zero but the variables are not independent (i.e. observing one gives information about the other).

Return to List of exercises. Return to List of exercises.

Q13**: Counting States

Suppose x_i for $i = 1 \dots 4$ are discrete random variables, each with 10 states.

1. For each of the below graphical models, specify the implied factorisation of the joint distribution $p(x_1, x_2, x_3, x_4)$ and calculate the number of free parameters one should specify *Be picky*



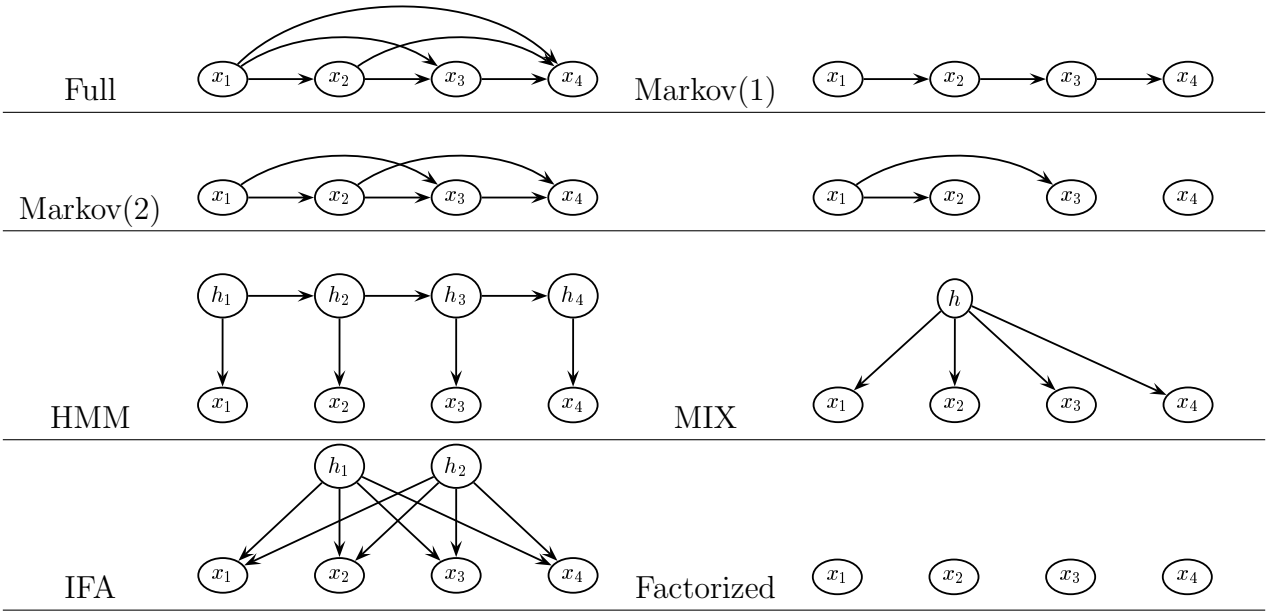
   and calculate a minimal parametrisation. For example, if x_1 would be independent from the rest, $p(x_1)$ has only 9 free parameters.

2. For each model, draw an associated factor graph and an equivalent undirected graphical model.

Return to List of exercises. Return to List of exercises.

Q14**: Models

For the following Graphical models, write down the factors of the joint distribution and plot an equivalent factor graph and an undirected graph.



Return to List of exercises. Return to List of exercises.

Q15**: Model Construction

Part I

We want to model a domain where we want to model a troubleshooter for a printer. A printer can print successfully a page or not. There are possible reasons for failure. The driver is corrupt, the printer is not plugged to the computer, the printer may be out of paper, if the printer is a network printer, there might be a problem with the network software. Another possibility is that there is no power. If there is no power the lights in the room are also off.

1. Carefully define the appropriate random variables to represent this scenario.
2. Draw the graphical model including model parameters and denote the conditional probability tables.
3. Suppose we wish to find a MAP estimate of the parameters. Write down the loglikelihood function that needs to be optimised with respect to the parameters given the data set. Remember, unknown variables need to be integrated over.

Part II

Suppose we have a dataset of the exam grades of 200 of students in 3 different subjects: Sports, Maths and History. For each subject we have 2 exam results. Suppose we believe that there are three types of orientations : Science, Sports and Arts. We believe that a student can be either science or art oriented but not both. He or she can be also sports oriented independent of being science or art oriented. Given the orientation of the student, the grade obtained from a subject by a student is assumed to be a random variable. The grade distributions have the same parameters for all students and exams of the same subject. Grade distributions have different parameters for different subjects. Moreover, each student can be ill during an examination, independent of other students and other examinations. If a student is ill during an examination, this would only affect the students performance for that examination.

1. Carefully define the appropriate random variables to represent this scenario.
2. Draw the graphical model including model parameters and denote the conditional probability tables.

Return to List of exercises. Return to List of exercises.

Q16*: Time Series Modeling

In the following figures, observations y_t from two processes are given as a function of time index t. Observations are known to be discrete with $y_t \in \{1, \ldots, 30\}$. For each realisation, define a plausible process that would generate similar realisations. Define the appropriate latent variables (if you use any), draw the graphical model and provide the conditional probability tables **and/or** state transition diagrams.

1.

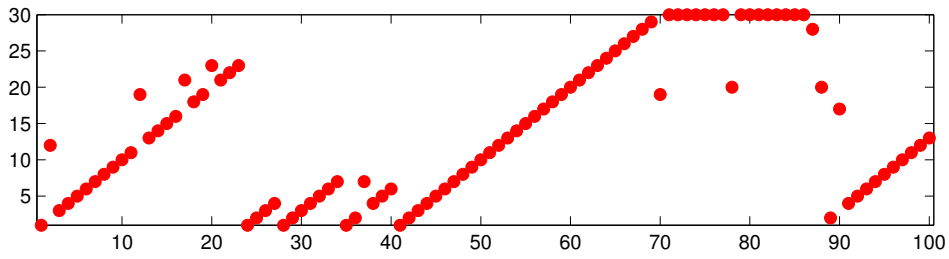


Figure 1: Process 1

2.

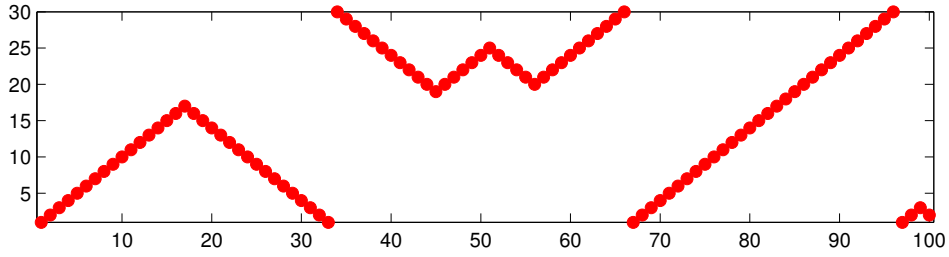


Figure 2: Process 2

3.

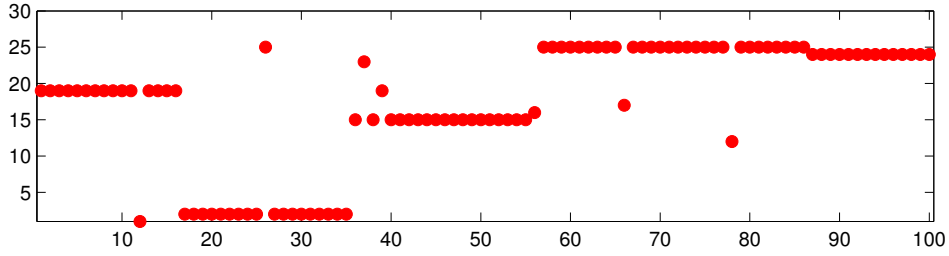


Figure 3: Process 3

4.

abcdeabcaabcdeeeeabcababcdabc

Figure 4: Process 4. $x_t \in \{a, b, c, d, e\}$

5.

1110001111000111100001110001111000011110001111

Figure 5: Process 5. $x_t \in \{1, 0\}$

Return to List of exercises. Return to List of exercises.

Q17**: Counting DAGs

This is a tedious exercise but should give an idea about the search space when learning the model structure from data. You need a large piece of paper. Let us call the set of all directed acyclic graphs with N nodes DAG(N).

1. How many directed acyclic graphs are there with 3 nodes ?
2. Draw each graph in DAG(3) and write down the corresponding factorisation of a probability distribution for x_1, x_2 and x_3.
3. Assume each random variable x_i has the same number of states. Find the partial ordering, where the binary relation for ordering two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in DAG(3) is defined if the factorisation corresponding to $\mathcal{G}_1$ is a special case of the one corresponding to $\mathcal{G}_2$. For example, $p(x_1)p(x_2)$ is a special case of $p(x_1|x_2)p(x_2)$, whereas $p(x_1|x_2)p(x_2)p(x_3)$ and $p(x_1|x_3)p(x_2)p(x_3)$ are not comparable.
4. Draw the Hasse diagram. (See partially ordered set entry in wikipedia.)

Return to List of exercises. Return to List of exercises.

Q18*: Chest Clinic

A distribution factorises according to the following factorisation

$$p(A, B, D, F, T, L, M, X) = p(F|T, L)p(M)p(T|A)p(B|M)p(X|F)p(L|M)p(D|F, B)p(A)$$

1. Draw the corresponding directed graphical model
2. Draw an equivalent factor graph and undirected graphical model
3. If all the variables have N states, compute the space to store the model specification.
4. Verify the following conditional independence statements using d-separation. State if they are true or false and explain why.
   a) $A \perp\!\!\!\perp M|\emptyset$
   b) $A \perp\!\!\!\perp M|X$
   c) $T \perp\!\!\!\perp L|X$
   d) $X \perp\!\!\!\perp L|F$
   e) $X \perp\!\!\!\perp L|D$

Return to List of exercises. Return to List of exercises.

Q19*: Hierarchical Hidden Markov Model

A process is given by the following specification

$$\begin{aligned} x_0 &\sim p(x_0) \\ z_0 &\sim p(z_0) \\ x_k &\sim p(x_k|x_{k-1}) \\ y_k &\sim p(y_k|x_k) \\ z_k &\sim p(z_k|z_{k-1}, y_k) \end{aligned}$$

1. Draw the corresponding directed graphical model
2. Draw an equivalent factor graph and undirected graphical model

Return to List of exercises. Return to List of exercises.

Q20**: The Gamma Function

The Gamma function is defined by the Integral:

$$\Gamma(x) \equiv \int_0^\infty u^{x-1} e^{-u} du$$

1. Show that $\Gamma(1) = 1$
2. Using integration by parts, show that

$$\Gamma(x+1) \quad = \quad x\Gamma(x)$$

Informally, integration by part follows from the chain rule as

$$\begin{aligned} (uv)' &= u'v + uv' \\ \int (uv)' &= \int u'v + \int uv' \\ \int u'v &= uv - \int uv' \end{aligned}$$

Return to List of exercises. Return to List of exercises.

Q21***: `log(gamma)` versus `gammaln`

In numeric computations, we almost always work with the logarithm of the gamma function $\log(\Gamma(x))$, which is computed without explicit reference to $\Gamma(x)$ to avoid overflow. In matlab, this function is `gammaln`. Using the `gammaln` function, write functions to evaluate the logarithms of $\mathcal{G}, \mathcal{IG}$ and $\mathcal{B}$ densities.

Return to List of exercises. Return to List of exercises.

Q22*: Exponential Distrubition

The *exponential* distribution is defined as

$$\mathcal{E}(v;\lambda) \quad = \quad \frac{1}{\lambda}\exp(-\frac{v}{\lambda})$$

Verify that the exponential distribution is a special case of the Gamma distribution. Find the shape and scale parameters of the corresponding gamma distribution.

Return to List of exercises. Return to List of exercises.

Q23**: Gamma and Inverse Gamma

Let

$$\begin{aligned} z &\sim \mathcal{G}(v; a, 1) \\ v &= bz \\ \lambda &= 1/v \end{aligned}$$

where $a, b > 0$ are known positive constants.

Using the transformation formula Eq.1, derive the marginal distributions $p(v)$ and $p(\lambda)$ and if possible express the result as known distributions.

Return to List of exercises. Return to List of exercises.

Q24***: Generalized Gamma

The *Generalised gamma* distribution is a three parameter family defined as (Stacey and Mihram 1965, Johnson and Kotz pp.393)

$$\mathcal{GG}(v;\alpha,\beta,c) = \frac{|c|}{\Gamma(\alpha)\beta^{c\alpha}} v^{c\alpha-1}\exp(-(v/\beta)^c)$$

Here, α is the shape, β is the scale and c is the power parameter.

1. Is the Generalised Gamma distribution an exponential family? If so, give the canonical parameters and the sufficient statistics.
2. Verify that the inverse Gamma distribution $\mathcal{IG}(v;a_i,b_i)$ and Gamma distribution $\mathcal{G}(v;a_g,b_g)$ are special cases. Give the corresponding settings of the power parameter.
3. Show that if

$$\begin{aligned} v &\sim \mathcal{GG}(v;\alpha,\beta,c) \\ z &= (v/\beta)^c \end{aligned}$$

   then, z has the standard $\mathcal{G}(z;\alpha,1)$ distribution. Using this fact, and a function that samples from standard gamma, implement a function generates random samples from a generalised Gamma distribution. The matlab statistics toolbox function `gamrnd(a, 1)` samples from the standard Gamma distribution.

Return to List of exercises. Return to List of exercises.

Q25*: Expectations

You are probably familiar with the idea of computing the expectation of a function of x,

$$\langle f(x) \rangle = \sum_x P(x) f(x).$$

Maybe you are not so comfortable with computing this expectation in cases where the function $f(x)$ depends on the probability $P(x)$. The next few examples address this concern.

1. Let $p_a = 0.1$, $p_b = 0.2$, and $p_c = 0.7$. Let $f(a) = 10$, $f(b) = 5$, and $f(c) = 10/7$. What are $\langle f(x) \rangle$ and $\langle 1/P(x) \rangle$?
2. For an arbitrary ensemble, what is $\langle 1/P(x) \rangle$?
3. Let $p_a = 0.1$, $p_b = 0.2$, and $p_c = 0.7$. Let $g(a) = 0$, $g(b) = 1$, and $g(c) = 0$. What is $\langle g(x) \rangle$?
4. Let $p_a = 0.1$, $p_b = 0.2$, and $p_c = 0.7$. What is the probability that $P(x) \in [0.15, 0.5]$? What is

$$P\left(\left| \log \frac{P(x)}{0.2} \right| > 0.05 \right)?$$

Return to List of exercises. Return to List of exercises.

Q26**: Entropies and Expectations

The expectation of a function of a discrete random variable is denoted as

$$\langle f(x) \rangle \equiv \sum_{x \in X} f(x) p(x)$$

Similarly, for a pair of random variables, we have the expectation

$$\langle f(x,y) \rangle \equiv \sum_{x \in X} \sum_{y \in Y} f(x,y) p(x,y)$$

The variance is defined as

$$\text{Var}\{f(x)\} = \left\langle (f(x) - \langle f(x) \rangle)^2 \right\rangle$$

It is a measure of spread. For a pair of random variables, the covariance is

$$\text{Cov}\left[f(x), g(y)\right] = \langle (f(x) - \langle f(x) \rangle)(g(y) - \langle g(y) \rangle) \rangle$$

The covariance gives information about the dependence between $f(x)$ and $g(y)$.

Now, given a probability table $p(x,y)$ specified as a matrix and respective domains of two discrete random variables $x \in X$ and $y \in Y$, write programs to calculate

1. Expectations $\langle x \rangle$, $\langle y \rangle$, $\langle y|x \rangle$, $\langle x|y \rangle$, $\text{Cov}\left[x, y\right]$
2. Joint Entropy

$$H[x,y] = -\left\langle \log p(x,y) \right\rangle_{p(x,y)}$$

3. Marginal Entropies

$$\begin{aligned} H[x] &= -\left\langle \log p(x) \right\rangle_{p(x)} \\ H[y] &= -\left\langle \log p(y) \right\rangle_{p(y)} \end{aligned}$$

4. Conditional Entropies

$$\begin{aligned} H[y|x] &= -\left\langle \log p(y|x) \right\rangle_{p(x,y)} \\ H[x|y] &= -\left\langle \log p(x|y) \right\rangle_{p(x,y)} \end{aligned}$$

5. Mutual Information

$$I(x,y) = H[x] - H[x|y] = KL(p(x,y)||p(x)p(y))$$

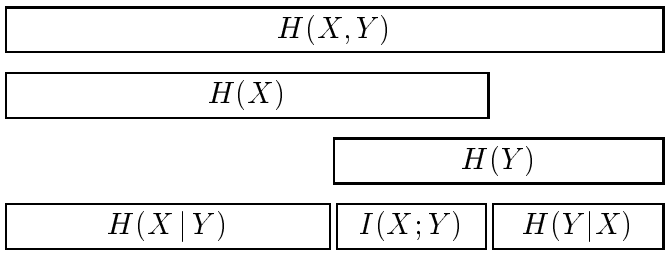
   Your program should correctly handle the limit case $0 \log 0 = 0$.

6. Test your program for the following joint probability table

$p(x,y)$	$y=-1$	$y=0$	$y=5$
$x=1$	0.3	0.3	0
$x=2$	0.1	0.2	0.1

Here, $X=\{1,2\}$ *and* $Y=\{-1,0,5\}$.

7. Verify the following picture



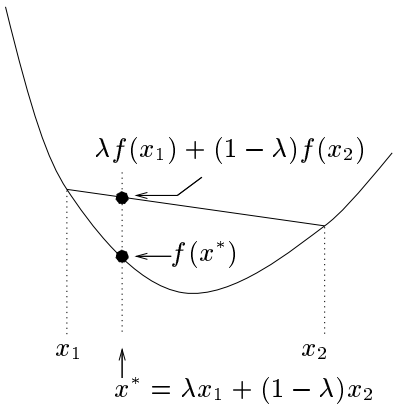
Return to List of exercises. Return to List of exercises.

Q27*: Jensen

A function $f(x)$ is convex if

$$\lambda f(x_1) + (1-\lambda) f(x_2) \quad \geq \quad f(\lambda x_1 + (1-\lambda) x_2)$$

for $\lambda \in [0,1]$. A function $f(x)$ is concave when $-f(x)$ is convex.



1. Specify if the following functions are convex, concave, both or none on the positive real numbers:

$$x^2, x^3, \log x, x \log x, e^{-x}, \log(\Gamma(x))$$

2. The celebrated Jensen's inequality states that for a convex function $f(x)$

$$\langle f(x) \rangle \geq f(\langle x \rangle)$$

By applying Jensen's inequality with $f(x) = \ln(x)$ show that the arithmetic mean of a set of real numbers is never less than their geometric mean. *In Jensen's, the direction of inequality is reversed for a concave function. For x_1, x_2, x_3, the arithmetic mean is $(x_1 + x_2 + x_3)/3$ and the geometric mean is $(x_1 x_2 x_3)^{1/3}$.*

Return to List of exercises. Return to List of exercises.

Q28*: Jensen's Inequality

Prove Jensen's inequality:
If f is a convex function and x is a random variable then $f(E[x]) \leq E[f(x)]$.

Return to List of exercises. Return to List of exercises.

Q29***: Bounds on Entropy

Prove the assertion that $H(X) \leq \log(|\mathcal{A}_X|)$ with equality iff $p_i = 1/|\mathcal{A}_X|$ for all i. ($|\mathcal{A}_X|$ denotes the number of elements in the set $\mathcal{A}_X$.) *Jensen involves both a random variable and a function, and you have quite a lot of freedom in choosing these; think about whether your chosen function f should be convex or concave.*

Return to List of exercises. Return to List of exercises.

Q30**: Differential Entropy

Given a continuous real valued random variable x with density $p(x)$, the *differential entropy* is defined by

$$\begin{aligned} H[q] &= -\left\langle \log q(x) \right\rangle_q \\ &= -\int q(x) \log q(x) \end{aligned}$$

Calculate the differential entropy of a

1. Gaussian $\mathcal{N}(x; \mu, \Sigma)$
2. Gamma $\mathcal{G}(x; a, b)$
3. Beta $\mathcal{B}(x; \alpha, \beta)$
4. Give an example where $h(X) < 0$.

Return to List of exercises. Return to List of exercises.

Q31***: Gibbs' Inequality

Prove that the relative entropy

$$D_{\mathrm{KL}}(P||Q) \quad = \quad \sum_x P(x) \log P(x)/Q(x)$$

satisfies $D_{\mathrm{KL}}(P||Q) \geq 0$ with equality only if $P = Q$.

Return to List of exercises. Return to List of exercises.

Q32**: KL Divergence

The KL (Kullback-Leibler) divergence is defined as

$$KL(P||Q) = \int p(x) \log p(x)/q(x)$$

1. Let $p(x) = \mathcal{N}(x; 0, 1)$. Find an expression for $KL(p||q)$ when $q(x) = \mathcal{N}(x; \mu, \Sigma)$.
2. Find an expression for $KL(q||p)$
3. Find expressions for $KL(p||q)$ and $KL(q||p)$ when $p(x) = \mathcal{N}(x; m, V)$ and $q(x) = \mathcal{N}(x; \mu, \Sigma)$.

Return to List of exercises. Return to List of exercises.

Q33***: Twelve Balls and Balance

You are given 12 balls, all equal in weight except for one that is either heavier or lighter. You are also given a two-pan balance (=terazi) to use. In each use of the balance you may put any number of the 12 balls on the left pan, and the same number on the right pan, and push a button to initiate the weighing; there are three possible outcomes: either the weights are equal, or the balls on the left are heavier, or the balls on the left are lighter. Your task is to design a strategy to determine which is the odd ball *and* whether it is heavier or lighter than the others *in as few uses of the balance as possible*.

While thinking about this problem, you may find it helpful to consider the following questions:

1. How can one measure *information*?
2. When you have identified the odd ball and whether it is heavy or light, how much information have you gained?
3. Once you have designed a strategy, draw a tree showing, for each of the possible outcomes of a weighing, what weighing you perform next. At each node in the tree, how much information have the outcomes so far given you, and how much information remains to be gained?
4. How much information is gained when you learn (i) the state of a flipped coin; (ii) the states of two flipped coins; (iii) the outcome when a four-sided die is rolled?
5. How much information is gained on the first step of the weighing problem if 6 balls are weighed against the other 6? How much is gained if 4 are weighed against 4 on the first step, leaving out 4 balls?

Return to List of exercises. Return to List of exercises.

Q34**: K-means Clustering

Consider the following clustering model:

$$\begin{aligned} x_{i,n} &\sim \mathcal{N}(x_{i,n}; \mu_{i,r_n}, \Sigma_i) \\ r_n &\sim \mathcal{M}(r_n; 1, \pi) \\ \mu_{i,k} &\sim \mathcal{U}([x_{min}, x_{max}] \times [y_{min}, y_{max}]) \end{aligned}$$

where $\mathcal{N}$, $\mathcal{M}$ and $\mathcal{U}$ are Gaussian, Multinomial and Uniform distributions, and

$$\begin{aligned} k &= 1 \dots K \\ i &= 1, 2 \\ n &= 1 \dots N \end{aligned}$$

1. Assuming that the class probabilities (π) in the Multivariate distribution are equal, generate data with the following parameters and plot the results such that the cluster centers and data points $x_{:,n} = [x_{1,n}\ x_{2,n}]^\top$ are clearly visible. Run your program several times and investigate the type of data sets generated by this generative model.

   a)

$$\begin{aligned} \Sigma_1 = \Sigma_2 &= 2 \\ \pi &= \left[\frac{1}{K}, \dots, \frac{1}{K}\right] \\ x_{min} = 0, x_{max} &= 10 \\ y_{min} = 0, y_{max} &= 10 \\ K = 3, N &= 20 \end{aligned}$$

   b)

$$\begin{aligned} \Sigma_1 = \Sigma_2 &= 0.5 \\ \pi &= \left[\frac{1}{K}, \dots, \frac{1}{K}\right] \\ x_{min} = 0, x_{max} &= 10 \\ y_{min} = 0, y_{max} &= 10 \\ K = 7, N &= 100 \end{aligned}$$

   Gaussian random numbers for $x \sim \mathcal{N}(x, \mu, \Sigma)$ can be drawn using the MATLAB *code:* `sqrt(Sigma)* randn + mu`. *Here,* `randn` *is a random number from the standard normal distribution $\mathcal{N}(x; 0, 1)$. Multinomial random variables $\mathcal{M}(r_n; 1, \pi)$ can be drawn using the* MATLAB *code:* `ceil(rand * K)` . *Here,* `rand` *is a function returning a uniform double number in* $(0, 1)$.

2. Implement the k-means algorithm described in the lecture. For the first data set you have generated, fit models with $K = 2 \ldots 5$ and plot the results.

Return to List of exercises. Return to List of exercises.

Q35**: Clustering with ICM

Consider the following clustering model:

$$
\begin{aligned}
x_n &\sim \mathcal{PO}(x_n; \lambda_{r_n}) \\
r_n &\sim \mathcal{M}(r_n; 1, \pi_{1:K}) \\
\lambda_k &\sim \mathcal{G}(\lambda_k; a, b)
\end{aligned}
$$

where

$$
\begin{aligned}
k &= 1 \dots K \\
n &= 1 \dots N
\end{aligned}
$$

and $\mathcal{PO}$, $\mathcal{M}$ and $\mathcal{G}$ are Poisson, Multinomial and Gamma distributions respectively, defined by

$$
\begin{aligned}
\mathcal{PO}(x; \lambda) &= \exp(-\lambda)\lambda^x / x! = \exp(x \log \lambda - \lambda - \log \Gamma(x+1)) \\
\mathcal{G}(\lambda; a, b) &= \exp((a-1)\log \lambda - b\lambda - \log \Gamma(a) + a \log b) \\
\mathcal{M}(r; \pi_{1:K}) &= \prod_{k=1}^{K} \pi_k^{[r=k]} \quad \text{if } r \in \{1 \dots K\} \\
\mathcal{M}(r; 1, \pi_{1:K}) &= \prod_{k=1}^{K} \pi_k^{r_k} \quad \text{if } r \in \{(1,0,\dots 0), (0,1,\dots 0), \dots, (0,0,\dots,1)\}
\end{aligned}
$$

1. Derive the iterative update equations for an Iterated Conditional Modes (ICM) algorithm to find the mode of the posterior
$$p(\lambda_{1:K}, r_{1:N} | x_{1:N})$$

2. Generate one dimensional data for the above model and plot the data similar to the example below ($K = 3, N = 100$): *use Matlab functions* `poissrnd`, `gamrnd`, `bar`, `barh`, `hist`

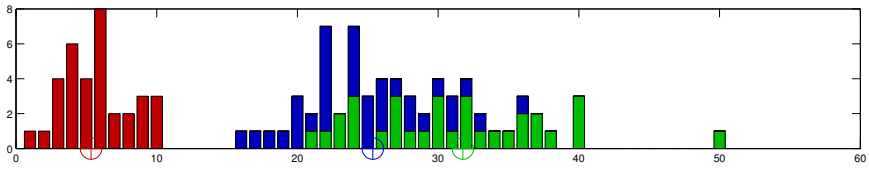


Figure 6: Histogram plot of the generated data points x_n, with the Gamma parameters $a = 2.5$ and $b = 20$.

3. Implement the ICM algorithm derived in 1., and test your code with the generated data in 2.

Return to List of exercises. Return to List of exercises.

Q36**: Biclustering via ICM

Consider the following clustering model for entries of a $I \times J$ matrix X where the element at i'th row and j'th column is denoted by $x_{i,j}$. We define indicator variables $c_i \in \{1 \dots U\}$ for $i = 1 \dots I$ and $s_j \in \{1 \dots U\}$ for $j = 1 \dots J$

$$x_{i,j} \sim \prod_u \prod_e \mathcal{PO}(x_{i,j}; \lambda_{u,e})^{[u=c_i][e=s_i]}$$

Here $\lambda_{u,e}$ denotes the element at row u and column e of a parameter matrix.

1. Write a matlab program to generate samples from this model.
2. Sketch the corresponding directed graphical model.
3. Why is this model called biclustering? (Hint: consider a scenario where i corresponds to customers and j corresponds to services. Then $x_{i,j}$ denotes the number of times that a customer i has used service j.)
4. Derive an ICM algorithm to estimate the mode of $p(c, s, \lambda|X)$

Return to List of exercises. Return to List of exercises.

Q37***: Clustering Problem

Part I

We are given the following generative model:

- A set of observed samples $X = \{x_1, x_2, \ldots, x_i, \ldots x_N\}$. Here $i = 1 \ldots N$ is the sample index. In this example we assume $x_i \in \mathbb{R}$.
- In our model, we assume each data point x_i comes from one of the M "clusters". The cluster label of x_i is denoted by $r_i \in \{1, \ldots, M\}$. We assume

  $$r_i \sim p(r_i) = U[1, M]$$

  Here $U[a, b]$ is the discrete uniform distribution on integers n such that $a \leq n \leq b$.
- We assume that each cluster has a center denoted by μ_j for $j = 1 \ldots M$, and these centers come from the following Gaussian distribution with variance P

  $$\mu_j \sim p(\mu_j) = \mathcal{N}(\mu_j|0, P)$$

  Here, $\mathcal{N}(x|\mu, \Sigma)$ denotes a Gaussian density with mean μ and variance Σ.
- Given the cluster centers and the cluster label, the conditional probability density of an observation is

  $$x_i|\mu_{1:M}, r_i \sim p(x_i|\mu_{1:M}, r_i) = \prod_{j=1}^{M} p(x_i|\mu_j)^{[r_i=j]}$$

  Here $[f]$ denotes an *indicator* function defined as

  $$[f] \equiv \begin{cases} 1 & \text{if } f \text{ is true} \\ 0 & \text{if } f \text{ is false} \end{cases}$$

  We assume that x_i depends on a cluster center μ_j according to a Gaussian conditional probability density with variance Q

  $$p(x_i|\mu_j) = \mathcal{N}(x_i|\mu_j, Q)$$

1. Look at the model and answer the following:
   a) Among random variables x_i, μ_i and r_i, which are observed variables, target variables and latent (=hidden, unobserved) variables respectively?
   b) Represent this model as a Bayesian dependency graph.
   c) Write the equation for full joint probability $p(x_{1:N}, \mu_{1:M}, r_{1:N})$ for this model.
   d) Write the integration to find the joint probability $p(x_{1:N}, r_{1:N})$
   e) While deriving log probability, which coefficients or terms can be omitted, why?
2. Write and derive the log probability $logp(x_{1:N}, r_{1:N})$ that is independent from $\mu_{1:M}$.
   Make use of the normalization condition of normal distribution.

Part II

In this exercise, we will write a MATLAB program that finds most probable $r_{1:N}$ and $\mu_{1:M}$ given an input vector $x_{1:N}$ by using assumptions of the Bayesian model that we defined in the previous assignment.

We are given an input vector X_1 as $\{1, 1.1, 1.2, -1, -1.1\}$ and another input vector X_2 as $\{1, 1.2, 3, 3.2, 3.4, -4,$ The parameters of the model is taken as $P = 1$ and $Q = 0.1$.

As we now know the joint probability $p(x_{1:N}, r_{1:N})$, we can calculate the probability corresponding to a partition $r_{1:N}$. Thus, we can iterate all possible partitions to find the one with the maximum likelihood:

$$r^*_{1:N} = \arg\max_r \ log\ p(x_{1:N}, r_{1:N})$$

Assume that we chose a particular partitioning, and now we want to find the most probable values for hidden variables $\mu_{1:M}$. As we already know $x_{1:N}$ and $r_{1:N}$, we can calculate the probability. We only need to iterate through $\mu_{1:M}$ values to find the best combination:

$$\mu^*_{1:M} = \arg\max_\mu \ log\ p(\mu_{1:M}|x_{1:N}, r^*_{1:N})$$

where

$$p(\mu_{1:M}|x_{1:N}, r^*_{1:N}) \propto p(x_{1:N}|\mu_{1:M}, r^*_{1:N})p(\mu_{1:M})$$

1. Finding the best partition:
   a) Write a loop that iterates a vector r from $[1, 1, 1, 1, 1]$ to $[5, 5, 5, 5, 5]$ by counting partitions one by one: $[1, 1, 1, 1, 2]$, $[1, 1, 1, 1, 3]$, ..., $[1, 1, 1, 1, 5]$, $[1, 1, 1, 2, 1]$, $[1, 1, 1, 2, 2]$, etc.
   b) Write a function that finds r^* as the partitioning with the maximum likelihood for a given input vector $x_{1:N}$ and maximum number of clusters M. It will iterate through all possible combinations of $r_{1:N}$.
   c) Run it for two inputs: $(X_1, N = 5, M = 3)$ and $(X_2, N = 9, M = 4)$ and print your results. Note that best solution might involve less than M clusters.
   d) Plot your resulting partitions by marking input points in different clusters with o, +, etc.
   e) Draw a figure similar to slide 15 of lecture03 that shows log probability for all of the partitions.
2. Finding latent variables:
   a) Write a program to find $\mu^*_{1:M}$ as the most likely hidden variable configuration given an input vector $x_{1:N}$ and a partition $r^*_{1:N}$. It should iterate through all combinations of real values of μ_j ranging from $-3P$ to $+3P$ with short intervals (e.g. 0.1).
   b) Run it by using the solution $r^*_{1:N}$ obtained from X_1, and then with the solution obtained from X_2. Print the results.
   c) Add μ_j points on your previous plots.
3. Generated input:

a) Write a function that takes N, M, P, Q as arguments, and generates an input vector $x_{1:N}$ by randomly choosing partitions $r_{1:N}$ with centers $\mu_{1:M}$ according to the given Bayesian model.

b) Write a program that generates input vectors for $(N = 9, M = 4, P = 1, Q = 0.1)$ and feeds them to the solvers that you wrote in (1) and (2).

c) For each input, it will first plot the input points according to the maximum likelihood partitioning $r^*_{1:N}$ with the corresponding ML mean values $\mu^*_{1:M}$, and then it will plot them according to their real partitioning $r_{1:N}$ with their real mean values $\mu_{1:M}$.

d) While keeping inter-cluster variance $P = 1$ constant, increase the intra-cluster variance Q in the model to observe its effect on the accuracy of the solver on generated data. Try $Q = 0.3$, $Q = 0.5$, $Q = 0.8$, $Q = 1.3$. Comment on the results.

4. Density image of μ for a given r:

   a) In question (2), we found $\mu^*_{1:M}$ based on the probability density $p(\mu_{1:M}|x_{1:N}, r^*_{1:N})$. When M is 2, we can draw this density as an image. Write a function that evaluates this density in range $[-3P, 3P]$ and displays it as an image (you can use MATLAB function imagesc). It should show range values in horizontal and vertical coordinates.

   b) Run this function for X_1 and the corresponding $r^*_{1:N}$. It should look like a hill around $\mu^*_{1:M}$.

5. Density image of posterior of μ:

   a) In terms of log probability, we know that log of a product of probabilities becomes the sum of their log probabilities. Then, what does the log of a sum of probabilities become? The answer is, the log of the sum of exps of each of the log probabilities. Thus, we need a simple function to calculate $log(sum(exp(l)))$ of a vector of log probabilities. Implement this function.

   b) Now we would like to evaluate the posterior probability $p(\mu_{1:M}|x_{1:N})$ and draw it as an image for $M = 2$. What we have to do is, take your function in (a), modify it so that it not only uses $r^*_{1:N}$, but sums probabilities over all possible $r_{1:N}$ using your log-sum-exp function.

   c) Run your function using X_1 and show the result. It should look like two hills corresponding to two possible r values: $[1, 1, 2, 2, 2]$ and $[2, 2, 1, 1, 1]$.

   d) Modify your function so that equivalent partitions count only once. For example it will evaluate $[1, 1, 2, 3]$ and discard $[2, 2, 3, 1]$, $[3, 3, 1, 2]$, $[1, 1, 3, 2]$ etc. This modification will remove one of the two hills in your result in (c).

   e) Run your new function using inputs generated by $N = 5, M = 2, P = 1, Q = 0.1$. Increase Q to see the effect on the posterior image. Note that the posterior contains probabilities for both two and one cluster partitionings.

Return to List of exercises. Return to List of exercises.

Q38***: Expectation-Maximization Derivation

AR(1) model is defined as follows

$$A \sim p(A) = \mathcal{N}(A|0, P)$$

$$R \sim p(R) = \mathcal{IG}(R|\alpha, \beta)$$

$$x_1|x_0, A, R \sim p(x_1|x_0, A, R) = \mathcal{N}(x_1|Ax_0, R)$$

Here, $\mathcal{IG}$ denotes the inverse gamma density function:

$$\mathcal{IG}(r|a, b) = \frac{1}{\Gamma(a)} \frac{r^{-(a+1)}}{b^{-a}} exp(-\frac{b}{r})$$

In the last lecture, we derived equations to implement Expectation-Maximization to maximize A parameter. Below is the derivation:

We want to find

$$A^* = \arg\max_A p(A|x) = \arg\max_A log\ p(A, x)$$

The log of posterior is tightly bounded by the function $B(A|A^{old})$, by Jensen's inequality:

$$\begin{aligned}
\mathcal{L}_x(A) &= log\ p(A, x) \\
&= log \int dR\ p(A, R, x) \\
&= log \int dR\ p(A, R, x) \frac{p(R|x, A^{old})}{p(R|x, A^{old})} \\
&\geq \langle logp(A, R, x) \rangle_{p(R|x,A^{old})} - \langle log\ p(R|x, A^{old}) \rangle_{p(R|x,A^{old})}
\end{aligned}$$

We derive the log of the joint density function:

$$\begin{aligned}
\phi = log\ p(A, R, x) &= log\ \mathcal{N}(x_1|Ax_0, R) + log\ \mathcal{N}(A|0, P) + log\ \mathcal{IG}(R|\alpha, \beta) \\
&= -\frac{1}{2} log 2\pi R - \frac{1}{2} \frac{x_1^2}{R} + \frac{x_1 A x_0}{R} - \frac{1}{2} \frac{A^2 x_0^2}{R} \\
&\quad - \frac{1}{2} log 2\pi P - \frac{1}{2} \frac{A^2}{P} - (\alpha + 1) log R - \frac{\beta}{R} - log\Gamma(\alpha) + \alpha log\beta
\end{aligned}$$

As x and A^{old} are known, $p(R|x, A^{old})$ is the full conditional that only depends on R. Thus, we derive it by choosing only terms of ϕ that depend on R:

$$
\begin{aligned}
log\ p(R|x, A^{old}) &=^{+} -\frac{1}{2}logR - (\alpha+1)logR - \frac{1}{2}\frac{(x_1 - Ax_0)^2}{R} - \frac{\beta}{R} \\
&= -(\alpha + \frac{3}{2})logR - (\frac{1}{2}(x_1 - Ax_0)^2 + \beta)\frac{1}{R} \\
&=^{+} log\ \mathcal{IG}(R; \alpha + \frac{1}{2}, \frac{1}{2}(x_1 - Ax_0)^2)
\end{aligned}
$$

We found that $p(R|x, A^{old})$ is distributed according to an inverse gamma density with known parameters. Now we return to the bounding function. We only choose terms of ϕ that depend on A:

$$
\begin{aligned}
B(A|A^{old}) &=^{+} \left\langle \frac{x_1 A x_0}{R} - \frac{A^2 x_0^2}{2R} - \frac{A^2}{2P} \right\rangle_{p(R|x,A^{old})} \\
&= (x_1 A x_0 - \frac{A^2 x_0^2}{2}) \left\langle \frac{1}{R} \right\rangle_{p(R|x,A^{old})} - \frac{A^2}{2P}
\end{aligned}
$$

Let z be the expectation of R^{-1} in the inverse gamma density:

$$
z = \left\langle \frac{1}{R} \right\rangle_{p(R|x,A^{old})}
$$

We continue:

$$
B(A|A^{old}) =^{+} (x_1 A x_0 - \frac{A^2 x_0^2}{2})z - \frac{A^2}{2P}
$$

We take derivative with respect to A and equate to zero to find A^{new}:

$$
\begin{aligned}
0 &= \frac{\delta B(A|A^{old})}{\delta A} \\
0 &= x_1 x_0 z - A^{new} x_0^2 z - \frac{A^{new}}{P} \\
A^{new} &= \frac{z x_1 x_0}{x_0^2 z + P^{-1}}
\end{aligned}
$$

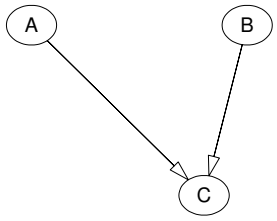
1. Make the derivations to implement EM to maximize R parameter!

$$
R^* = \arg\max_R p(R|x) = \arg\max_R log\ p(R, x)
$$

Return to List of exercises. Return to List of exercises.

Q39*: Explaining Away

Consider the following graphical model:



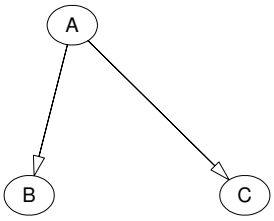
Here, all variables are binary. $p(A = 1) = 0.9$, $p(B = 1) = 0.3$, $C = A \oplus B$ where $\oplus$ is the xor (exclusive or) operation.

1. Find the following quantities:
   a) $p(C)$
   b) $p(A, B|C)$
2. Write a program that will compute above quantities for arbitrary $p(A)$, $p(B)$ and $p(C|A, B)$
3. Write a program that will generate random probability tables $p(A)$, $p(B)$ and $p(C|A, B)$. *Use the Beta distribution as a prior.*
4. Using the `randgen` subroutine you developed in the previous assignment sheet, write a program that will generate random instances from the above model.

Return to List of exercises. Return to List of exercises.

Q40*: Sensor Fusion

Consider the following graphical model:



1. How does the associated probability distribution factorise?
2. Write a program that will generate random probability tables (i.e. parameters) compatible with this graph.
3. Using the `randgen` subroutine you developed in the previous assignment sheet, write a program that will generate random instances from the above model.
4. Write a program that will compute the following quantities
   a) $p(C)$
   b) $p(A|B,C)$
   c) $p(C|B)$

Return to List of exercises. Return to List of exercises.

Q41**: One Sample Source Separation

Consider the following model

$$\begin{aligned} s_1 &\sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \\ s_2 &\sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \\ x|s_1, s_2 &\sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R) \end{aligned}$$

We will use the following parameters: $\mu_1 = 3$, $\mu_2 = 5$, $P_1 = P_2 = 0.5$ and $R = 0.3$.

1. Draw the graphical model
2. Find $p(x)$,
3. Find $p(s_1, s_2|x)$, $p(s_1|x)$
4. Find $p(s_1|s_2, x)$ and $p(s_2|s_1, x)$
5. Suppose we observe $x = 9$. Find $p(s_1, s_2|x = 9)$ analytically. Plot the posterior.

Return to List of exercises. Return to List of exercises.

Q42***: AR Model

Part I

Consider the following model:

$$\begin{aligned} A &\sim \mathcal{N}(A; 0, 1.2) \\ R &\sim \mathcal{IG}(R; 0.4, 250) \\ x_k|x_{k-1}, A, R &\sim \mathcal{N}(x_k; Ax_{k-1}, R) \\ x_0 &= 1 \qquad\qquad x_1 = -6 \end{aligned}$$

1. Draw the directed graphical model and the factor graph
2. Write the expression for the full joint distribution and assign terms to the individual factors on the factor graph
3. Derive the full conditional distributions $p(A|R, x_0, x_1)$ and $p(R|A, x_0, x_1)$
4. Derive the joint distribution $p(A, R, x_0 = 1, x_1 = -6)$ and create a contour plot.

Part II

Consider the following model discussed in detail during the lectures.

$$\begin{aligned} A &\sim \mathcal{N}(A; 0, P) \\ R &\sim \mathcal{IG}(R; \nu, \nu/\beta) \\ x_k|x_{k-1}, A, R &\sim \mathcal{N}(x_k; Ax_{k-1}, R) \end{aligned}$$

where $\mathcal{N}$ is a Gaussian and

$$\mathcal{IG}(R; a, b) = \exp\left(-(a+1)\log R - \frac{b}{R} - \log\Gamma(a) + a\log b\right)$$

Caution: (This definition is different from the definition of $\mathcal{IG}$ given in some of the earlier lectures.)
We are given the hyperparameters $\theta = (\nu, \beta, P)$

$$\begin{aligned} \nu &= 0.4 \qquad \beta = 100 \qquad P = 1.2 \\ x_0 &= 1 \qquad x_1 = -6 \end{aligned}$$

1. Derive and implement an EM algorithm to find the MAP estimate

$$R^* = \operatorname*{argmax}_R p(R|x_0, x_1, \theta)$$

2. Derive and implement an EM algorithm to find the MAP estimate

$$A^* = \operatorname*{argmax}_A p(A|x_0, x_1, \theta)$$

3. Derive and implement an ICM (Iterative conditional modes) algorithm to find

$$(R^*, A^*) = \underset{A,R}{\operatorname{argmax}}\, p(A, R|x_0, x_1, \theta)$$

4. In the lectures, we have shown that the unnormalised posterior is

$$\begin{aligned}
\phi &= p(A, R, x_1 = \hat{x}_1|x_0 = \hat{x}_0, \theta) = \mathcal{N}(x_1; Ax_0, R)\mathcal{N}(A; 0, P)\mathcal{IG}(R; \nu, \nu/\beta) \\
&\propto \exp\left(-\frac{1}{2}\frac{x_1^2}{R} + x_0 x_1 \frac{A}{R} - \frac{1}{2}\frac{x_0^2 A^2}{R} - \frac{1}{2}\log 2\pi R\right) \\
&\quad \exp\left(-\frac{1}{2}\frac{A^2}{P} - \frac{1}{2}\log|2\pi P|\right) \\
&\quad \exp\left(-(\nu+1)\log R - \frac{\nu}{\beta}\frac{1}{R} - \log\Gamma(\nu) + \nu\log(\nu/\beta)\right)
\end{aligned}$$

We know also that the marginal log-likelihood

$$\log Z = \log p(x_1 = \hat{x}_1|x_0 = \hat{x}_0, \theta)$$

is lower bounded by

$$\mathcal{B}_{VB} = \langle \log\phi \rangle_Q + H[Q]$$

where

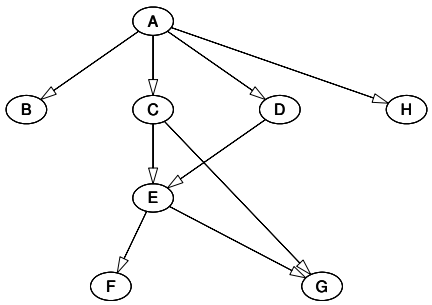
$$\begin{aligned}
Q &= q(A)q(R) \\
q(A) &= \mathcal{N}(A; m, \Sigma) \\
q(R) &= \mathcal{IG}(R; a, b)
\end{aligned}$$

Extend the VB algorithm given in the slides so that you compute this bound at every iteration and plot the bound $\mathcal{B}$ as a function of iterations. You should observe that the VB fixed point **monotonically** increases this lower bound. Restart your algorithm several times and compare the largest bound you find with the bound you find with importance sampling.

Return to List of exercises. Return to List of exercises.

Q43**: Directed Graphical Models

Consider the following directed graph G



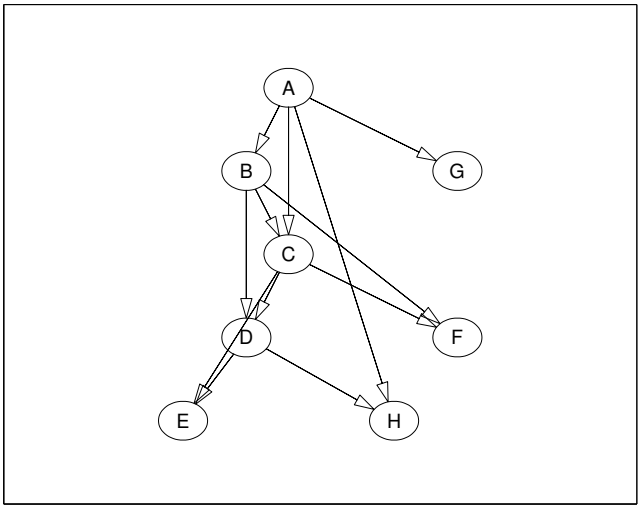
1. Find a topological ordering of the variables,
2. Write down the implied factorisation of the probability distribution that respects the conditional independence structure implied by G,
3. Draw the associated factor graph,
4. Suppose, each variable has two states. How many free parameters does each conditional probability table have?
5. Draw an equivalent undirected graphical model.

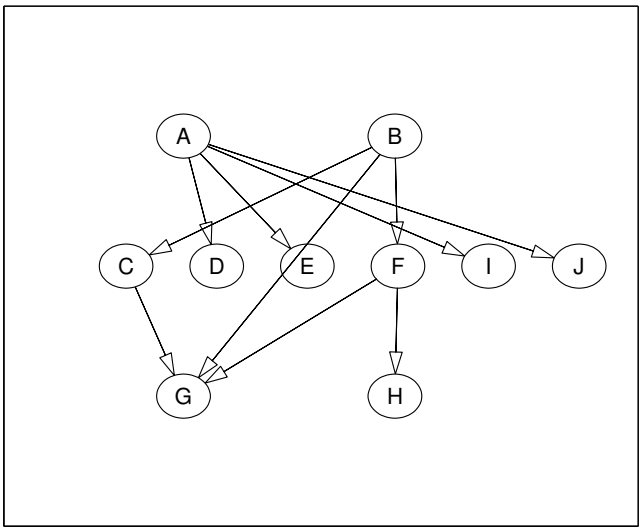
Return to List of exercises. Return to List of exercises.

Q44*: Some Basic Graph Operations

The adjacency matrix of a graph with N nodes is a $N \times N$ matrix with entries 0 or 1, where $a_{i,j} = 0$ denotes a missing directed edge from i to j. Here, we represents an undirected edge when $a_{i,j} = a_{j,i}$.

1. Find a topological ordering for the following graphs:





2. Write a program for topological sort with the following specification:

```
% TOPOSORT A Topological ordering of nodes in a directed graph
%
%   [SEQ] = TOPOSORT(ADJ)
%
% Inputs :
%     ADJ : Adjacency Matrix.
```

```
%    ADJ(i,j)==1 ==> there exists a directed edge
%    from i to j
%
% Outputs :
%     SEQ : A topological ordered sequence of nodes.
%    empty matrix if graph contains cycles.
%
% Usage Example :
%        N=5;
%        [l,u] = lu(rand(N));
%        adj = ~diag(ones(1,N)) & u>0.5;
%        seq = toposort(adj);
```

3. Assuming the graphs encode a Bayesian network with discrete random variables, write a program that counts the number of free parameters.

```
% COUNT_BNET   Counts the number of free parameters given a graph
% compatible with a graph
%
% [CNT] = COUNT_BNET(ADJ, SIZES)
%
% Inputs :
%   ADJ : N_by N Adjacency Matrix.
%     ADJ(i,j)==1 ==> there exists a directed edge
%     from x_i to x_j
% SIZES : 1 by N Array. SIZES(i) gives
%               the number of states of random variable x_i
%
% Outputs :
%   CNT : 1 by N Array of number of free parameters
%             for each probability table $p(x_i| parents(x_i))$
%
% Usage Example :
%     N=5;
%     [l,u] = lu(rand(N));
%     adj = ~diag(ones(1,N)) & u>0.5;
%      sizes = [2 2 3 2 5];
%     cnt = count_bnet(adj, sizes);
```

You may find the following matlab package useful for visualisation of your graphs: `http://www-sigproc.eng.cam.ac.uk/~atc27/matlab/layout.html`

Return to List of exercises. Return to List of exercises.

Q45**: Transmission of Strings

Suppose we have an alphabet over two symbols a and b. Each word is surrounded by a delimiter symbol c. We know that in the language the probability of the current symbol depends only on the previous symbol that is transmitted. In the transmission, some characters may be corrupted by noise and confused by the others. For example, if the true symbol that was transmitted was an a it could be detected as b or c, similarly for other symbols.

1. Define the random variables
2. Propose a probability model for this scenario
3. Express the following queries as Bayesian inference problems *For example, for the dice example finding the outcome of a dice λ given the sum D requires calculation of $p(\lambda|D)$. If we require the most likely outcome, we calculate* $\arg\max_\lambda p(\lambda|D)$. *Those are the inference problems.*
   a) The most likely string given the observations so far
   b) The probability of the most likely string given the observations so far
   c) The most likely true next symbol given observations so far
   d) The probability of the next observation given observations so far
   e) Most likely observation at time $t + 5$ given observations until time t
   f) The probability that exactly two complete words have been transmitted so far
   g) The positions of most likely word boundaries

Return to List of exercises. Return to List of exercises.

Q46*: Sequential application of the Bayes Theorem

Recall problem Q7, where we have three coloured boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes. Boxes are chosen in sequence according to the following rules:

- If $t = 0$, choose a box with probabilities $p(r) = 0.2$, $p(b) = 0.2$, $p(g) = 0.6$.
- If t is odd, choose another box with equal probability, that is different from the current box.
- If t is even, choose another box with equal probability, that is different from the current box and choose a fruit with replacement.

1. Choose the appropriate random variables, write down the generative model and draw the associated directed graphical model.
2. Draw a state transition diagram (for the boxes only).
3. Define the conditional probability tables given the rules above.
4. Write a program to find numerically the probability of selecting the red (blue, gree) box at a given t. Plot the probabilities as a function of t for $t = 1, 3, \ldots, 50$.
5. If we observe that the first selected fruit is a lime and the second fruit is an orange, what is the probability that the current box is red (blue, green)?
6. Write a program to compute the probability of the next fruit given the fruits observed so far.

Return to List of exercises. Return to List of exercises.

Q47**: Beta Function

In this exercise, we prove that the beta distribution, given by

$$\mathcal{B}(w; a, b) \equiv \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} w^{a-1}(1-w)^{b-1}$$

is correctly normalized. This is equivalent to showing that

$$\int_0^1 w^{a-1}(1-w)^{b-1} dw = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{2}$$

1. Show that (2) is true. *Consider the hint in Bishop, problem 2.5, pp128*
2. Using (2), show that

$$\begin{aligned} \langle w \rangle &= \frac{a}{a+b} \\ \langle w^2 \rangle - \langle w \rangle^2 &= \frac{ab}{(a+b)^2(a+b+1)} \\ w^* &= \arg\max_w \mathcal{B}(w; a, b) = \frac{a-1}{a+b-2} \qquad a, b > 1 \end{aligned}$$

Return to List of exercises. Return to List of exercises.

Q48**: Inverting the Arrow in a Gaussian Network

Given a factorisation of the form $p(y|x)p(x)$ where

$$\begin{aligned} x &\sim \mathcal{N}(x; \mu, \Sigma) \\ y|x &\sim \mathcal{N}(y; Cx, R) \end{aligned}$$

Express this distribution in form of $p(x|y)p(y)$.

Return to List of exercises. Return to List of exercises.

Q49**: The Nasty Lecturer

Every week k, a class of students have to write a quiz, if the random variable $r_k = 1$. The model for the quizes is as follows:

$$\pi|a \sim \mathcal{B}(a, 2) \tag{3}$$
$$r_k|\pi \sim \mathcal{BE}(r_k; \pi) \tag{4}$$

Here, the $\mathcal{B}$ and $\mathcal{BE}$ are Beta and Bernoulli distributions respectively. Suppose, we have observed the values of $r_1, r_2, \ldots, r_n$. We let $r_{1:k}$ detote $r_1, r_2, \ldots, r_k$.

1. Draw the directed graphical model for the generative model. Include parameter a also as a random variable.
2. Suppose $a = \sqrt{3}/2$. Compute the probability that there will be a quiz at week $k = n + 1$. Draw also the factor graph for this problem.
3. Suppose a is unknown. Find the log-likelihood function for a, $\log p(r_{1:k}|a)$.
4. Assume that $p(a)$ is uniform on $[0.1, 5]$. Write a program to compute and plot the posterior density $p(a|r_{1:k})$ numerically. Make sure that the density is normalised. Plot the posterior densities $p(a|r_{1:k})$ for $k = 1 \ldots 5$ for the following observation sequence $r = [10011]$. For example for $k = 2$, you plot $p(a|r_1 = 1, r_2 = 0)$ and ignore r_3, r_4 and r_5. For each k, compute the mean and variance of the posterior distribution.
5. Consider $p(r_{k+1}|r_{1:k}, a)$ and $p(r_{k+1}|r_{1:k})$. Are those quantities different from each other?

Return to List of exercises. Return to List of exercises.

Q50**: The Nastier Lecturer

Repeat question for the model in Q49.

$$\begin{aligned} \pi_k|a &\sim \mathcal{B}(a,2) \\ r_k|\pi_k &\sim \mathcal{BE}(r_k;\pi_k) \end{aligned}$$

and comment, how this model is different from the one given in Eq.(3) and Eq.(4).

Return to List of exercises. Return to List of exercises.

Q51****: Self Localization

A robot is moving across a circular coridor. We assume that the possible positions of the robot is a discrete set with N locations. The initial position of the robot is unknown and assumed to be uniformly distributed. At each step k, the robot stays where it is with probability ϵ, or moves to the next point in counterclock direction with probability $1-\epsilon$. At each step k, the robot can observe its true position with probability w. With probability $1-w$, the position sensor fails and gives a measurement that is independent from the true position (uniformly distributed).

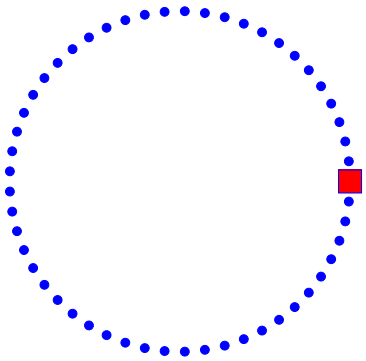


Figure 7: Robot (Square) moving in a circular corridor. Small circles denote the possible N locations.

1. Choose the appropriate random variables, define their domains, write down the generative model and draw the associated directed graphical model.
2. Define the conditional probability tables given the verbal description above.
3. Specify the following verbal statements in terms of posterior quantities using mathematical notation. *for example "the distribution of the robots location two time step later given its current position at time k" should be answered as* $p(s_{k+2}|s_k)$
   - Distribution of the robots current position given the observations so far,
   - Distribution of the robots next position given the observations so far,
   - Distribution of the robots next sensor reading given the observations so far,
   - Distribution of the robots initial position given observations so far,
   - Marginal Distributions of the robots positions at the past given observations so far,
   - Most likely current position of the robot given the observations so far,
   - Most likely trajectory taken by the robot from the start until now given the observations so far,
4. Implement a program that simulates this scenario; i.e., generates realisations from the movements of the robot and the associated sensor readings. *You can use the* `randgen` *function you wrote earlier. Simulate a scenario for* $k = 1 \dots 100$ *with* $N = 50, \epsilon = 0.3, w = 0.8$
5. (Optional) Implement a program that computes the posterior quantities in 3, given the sensor readings.

6. (**The kidnap**) Assume now that at each step the robot can be kidnapped with probability κ. If the robot is kidnapped its new position is independent from its previous position and is uniformly distributed. Repeat 4 and 5 for this new model with $\kappa = 0.1$. *Can you reuse your code?*

Return to List of exercises. Return to List of exercises.

Q52****: Self Localization on Prime Numbers

Part I

A robot is moving across a circular corridor. We assume that the possible positions of the robot are elements of a discrete set with N locations, numbered as $i = 1, \ldots, N$. The exact initial position of the robot is unknown but it is known to be located on one of the non-prime locations. At each step k, the robot stays where it is with probability ϵ, or moves to the next point in counterclock direction with probability $1 - \epsilon$.

At each step k, the robot can observe the color of the tile it is on, independently from previous readings. The corridor is designed such that the tiles with prime numbered locations $i = 2, 3, 5, 7, 11, \ldots$ are white, others are blue. Due to the noise present at the visual sensor, with probability δ a white (blue) tile is observed as blue (white).

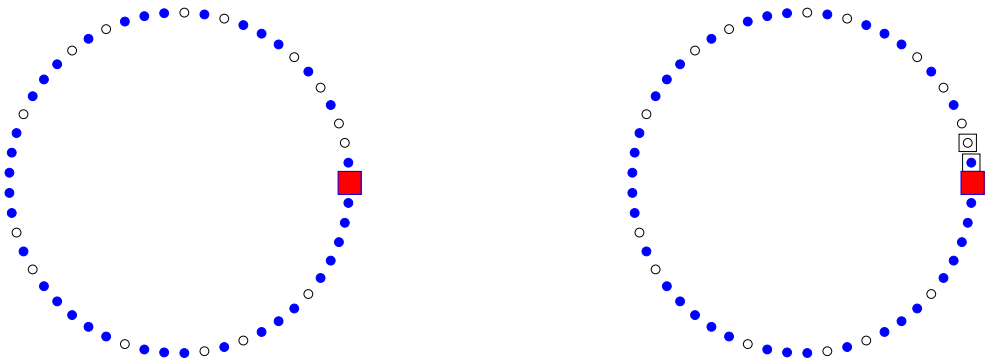


Figure 8: (Left) Robot (Square) moving in a circular corridor. Small circles denote the possible N tiles. Prime numbered tiles are white, others are blue. The robot can sense the color of the tile it is on. (Right) The sensor can sense the color of two tiles in front.

1. Define the conditional probability tables given the verbal description above. Write a program that generates one given N, δ and *epsilon. Matlab has a function called* `isprime`.
2. Implement a program that simulates this scenario; i.e., generates realisations from the movements of the robot and the associated sensor readings. *You can use the* `randgen` *function you wrote earlier. Simulate a scenario for* $k = 1, 2, \ldots K$ *with sufficiently large* K *for* $N = 50, \epsilon = 0.3, \delta = 0.9$
3. (Optional) Implement a program that computes, for each time step, the posterior distribution over the robots position, given the sensor readings so far.
4. How many time steps are needed from the start on average until the location is known with 90% certainty when $\delta = 0.99$, provided the model is correct.
5. Suppose we modify the sensor such that it can sense the color of two tiles in front (in counterclock direction), independent from each other and independent from previous readings. Define the appropriate random variables.

Part II

A robot is moving across a circular corridor. We assume that the possible positions of the robot are elements of a discrete set with N locations, numbered as $i = 1, \ldots, N$. The exact initial position of the robot is unknown but it is known to be located on one of the non-prime locations. At each step k, the robot stays where it is with probability ϵ, moves to the next point in counterclock direction with probability $(1-\epsilon)/2$ or moves to the next point in clock direction with probability $(1-\epsilon)/2$.

At each step k, the robot can observe the color of the tile it is on, independently from previous readings. The corridor is designed such that the tiles with prime numbered locations $i = 2, 3, 5, 7, 11, \ldots$ are white, others are blue. Due to the noise present at the visual sensor, the true color is observed only with probability δ, with probability $1-\delta$ a white (blue) tile is observed as blue (white).

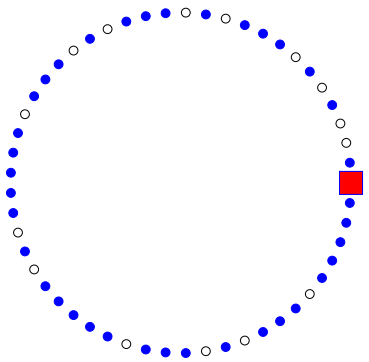


Figure 9: Robot (Square) moving in a circular corridor. Small circles denote the possible N tiles. Prime numbered tiles are white, others are blue. The robot can sense the color of the tile it is on.

1. Implement a program that simulates this scenario; i.e., generates realisations from the movements of the robot and the associated sensor readings. *You can use the* `randgen` *function you wrote earlier. Simulate a scenario for* $k = 1, 2, \ldots K$ *with sufficiently large* K *for* $N = 50, \epsilon = 0.3, \delta = 0.9$
2. (**Filter**) Implement a program that computes, for each time step, the posterior distribution over the robots position, given the sensor readings so far.
3. (**Fixed lag smoother**) Implement a program that computes, for each time step k, the posterior distribution over the robots past L positions $p(x_l|y_{1:k})$, $k - L + 1 \leq l \leq k$.
4. (**Viterbi Path**) Implement a program that computes the most likely state trajectory, given all the observations.
5. (**Interpolation**) Implement a program that computes the smoothed state estimates, given observations $y_{1:L}$ and $y_{K-L+1:K}$ for any L such that $1 < L < K/2$.
6. For all the tasks above run your program with two different ϵ and δ settings, and create figures similar to the ones shown in the lecture slides.
7. Comment on self localisation performance. In particular comment how ϵ and δ effect it. Discuss if the prime numbers are special in some respect. Could we get the same performance with coloring, say, odd numbers ?
8. (**Parameter Estimation**) The goal of this exercise is to see if model parameters can be estimated from data as well. Let us denote the true parameters by $\theta_{\text{true}} = (\epsilon, \delta)$. Generate

data from a model with $K = 500$ and $N = 50, \epsilon = 0.3, \delta = 0.9$. Compute the evidence $\mathcal{L}(\theta) = p(y_{1:K}|\theta)$ where θ is varied on a sufficiently dense grid on the unit square $[0 \quad 1]^2$. Generate a contour plot of $\mathcal{L}(\theta)$ and compare its peak with θ_{true}.

9. (Optional) Develop a numerical method for finding $\theta^* = \arg\max_\theta \mathcal{L}(\theta)$ without the exhaustive search.

Return to List of exercises. Return to List of exercises.

Q53***: Adding Gaussian Random Variables

An important property of Gaussian random variables is that the sum is also Gaussian distributed.

$$\begin{aligned} x_1 &\sim p_1(x_1) = \mathcal{N}(x_1; \mu_1, P_1) \\ x_2 &\sim p_2(x_2) = \mathcal{N}(x_2; \mu_2, P_2) \\ y &= x_1 + x_2 \end{aligned}$$

1. Using the Jacobian formula, show that

$$p(y) = \int p_1(x) p_2(y - x) dx$$

   (That is the convolution of p_1 and p_2).

2. A moment generating function is defined as

$$M_x(t) = \langle \exp(tx) \rangle$$

   Show that when two pdf's are convolved, the resulting pdf has a moment generating function that is the product of the individual generating functions.

3. Derive the moment generating function for a Gaussian random variable with density $\mathcal{N}(x; \mu, P)$.

4. Using the above results, show that y has a Gaussian distribution. Find the mean and variance of y.

Return to List of exercises. Return to List of exercises.

Q54***: Adding Poisson Random Variables

An important property of Poisson random variables is that the sum is also Poisson distributed.

$$\begin{aligned} x_1 &\sim p_1(x_1) = \mathcal{PO}(x_1; \mu_1) \\ x_2 &\sim p_2(x_2) = \mathcal{PO}(x_2; \mu_2) \\ y &= x_1 + x_2 \end{aligned}$$

1. Show that

$$p(y) = \sum_x p_1(x) p_2(y - x)$$

   (That is the convolution of p_1 and p_2).

2. A probability generating function of a nonnegative discrete random variable is defined as

$$G_x(z) = \sum_{t=0}^{\infty} p(x = t) z^t$$

   Show that when two pdf's are convolved, the resulting pdf has a probability generating function that is the product of the individual generating functions.

3. Derive the probability generating function for a Poisson random variable with density $\mathcal{PO}(x; \mu)$.

4. Using the above results, show that y has a Poisson distribution. Find the mean and variance of y.

5. Find the posterior distribution $p(x_1, x_2|y)$.

6. Find the posterior marginal $p(x_1|y)$.

Return to List of exercises. Return to List of exercises.

Q55*: Woodbury Formula

A very useful result from linear algebra is the *Woodbury* matrix inversion formula, also known as the Matrix inversion lemma, given by

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1}+DA^{-1}B\right)^{-1}DA^{-1}$$

1. Verify the correctness of this formula by multiplying both sides by $(A+BCD)$.
2. Using the matrix inversion lemma, verify the following where I denotes identity matrices (not necessarily the same size)

$$\begin{aligned}(A-BCD)^{-1} &= A^{-1} + A^{-1}B\left(C^{-1}-DA^{-1}B\right)^{-1}DA^{-1}\\ \left(I+B^\top DB\right)^{-1} &= I - B^\top\left(D^{-1}+BB^\top\right)^{-1}B\\ \left(A^{-1}+B^\top B\right)^{-1} &= A - AB^\top\left(I+BAB^\top\right)^{-1}BA\\ \left(A-C^{-1}\right)^{-1} &= A^{-1} + A^{-1}\left(C-A^{-1}\right)^{-1}A^{-1}\end{aligned}$$

3. Woodbury formula is particularly useful for reducing the computational load and improving stability in matrix computations. Assume D is a $N\times N$ **diagonal** matrix and v is a N vector. Assuming that inversion of a matrix is $O(N^3)$, estimate approximately the computational requirement for a direct evaluation of

$$G \;=\; (D+vv^\top)^{-1}$$

4. The inverse of a diagonal matrix is easy to compute in $O(N)$. Using the Woodbury formula, rewrite G to exploit this fact. Estimate the computational requirement and compare to the naive method.
5. (Optional) Implement both equations in matlab and verify your result. Compare the execution time for both implementations for $N = 100, 1000, 10000$.
6. (Very optional – but very useful) Write a Matlab program with the following specification:

```
%    MATRIX_INV_LEMMA   Prints the tex string for the matrix lemma
%
%    [str1 str2] = matrix_inv_lemma(A_11, A_12, A_22, A_21, <property, valu
%
%
%    Inputs :
%     A_11, A_12, A_22, A_21 : tex strings
%
%              alpha : '+' or '-', Default = '+'
%              invert : true or false
%
%    Outputs :
%              if invert is false
%             str1 = (A_11 + \alpha A_12 A_22^{-1} A_21 )^{-1}
```

```
%             str2 = A_11^{-1} - alpha A_11^{-1}  A_12 (A_{22}
%                      + alpha A_21 A_11^{-1} A_{12})^{-1} A_21 A_11^{-1}
%              if invert = true
%             str1 = (A_11^{-1} + \alpha A_12 A_22 A_21 )^{-1}
%             str2 = A_11 - alpha A_11  A_12 (A_{22}^{-1}
%                       + alpha A_21 A_11 A_{12})^{-1} A_21 A_11
%
%   Usage Example :
%       matrix_inv_lemma('D', 'C^\top', 'R', 'C', 'invert', true, 'alpha',
%     ans =
%     \left( D^{-1} - C^\top R^{-1} C \right)^{-1} =
%     D + D C^\top \left(R - C D C^\top \right)^{-1} C D
%
%
%   matrix_inv_lemma('D', 'C^\top', 'R', 'C', 'invert', false, 'alpha', '+'
%     ans =
%       \left( D + C^\top R C \right)^{-1} =
%      D^{-1} - D^{-1} C^\top \left(R^{-1} + C D^{-1} C^\top \right)^{-1} C
%
```

Return to List of exercises. Return to List of exercises.

Q56***: Gaussian Process Regression

The goal of this exercise is to test your understanding of manipulations associated with multivariate Gaussians. You may also find it helpful to read Bishop 6.4.

In Bayesian machine learning, a frequent problem that pops up is the regression problem where we are given a pairs of inputs $x_i \in \mathbb{R}^N$ and associated noisy outputs $y_i \in \mathbb{R}$. We assume the following model

$$y_i \quad \sim \quad \mathcal{N}(y_i; f(x_i), R)$$

The interesting thing about a Gaussian process is that the function f is not specified in close form, but we assume that the function values

$$f_i \quad = \quad f(x_i)$$

are jointly Gaussian distributed as

$$\begin{pmatrix} f_1 \\ \vdots \\ f_L \end{pmatrix} \quad = \quad f_{1:L} \sim \mathcal{N}(f_{1:L}; 0, \Sigma(x_{1:L}))$$

Here, we define the entries of the covariance matrix $\Sigma(x_{1:L})$ as

$$\Sigma_{i,j} \quad = \quad K(x_i, x_j)$$

for $i, j \in \{1, \ldots, N\}$. Here, K is a given *covariance function*. Now, if we wish to predict the value of $\hat{f}$ for a new $\hat{x}$, we simply form the following joint distribution:

$$\begin{pmatrix} f_1 \\ \vdots \\ f_L \\ \hat{f} \end{pmatrix} \quad \sim \quad \mathcal{N}((f_{1:L}, \hat{f}); 0, \Sigma(x_{1:L}, \hat{x}))$$

Here, $\Sigma(x_{1:L}, \hat{x})$ is a $L+1$ by $L+1$ covariance matrix with entries

$$\Sigma_{i,L+1} = \Sigma_{L+1,i} \quad = \quad K(x_i, \hat{x}) = K(\hat{x}, x_i)$$

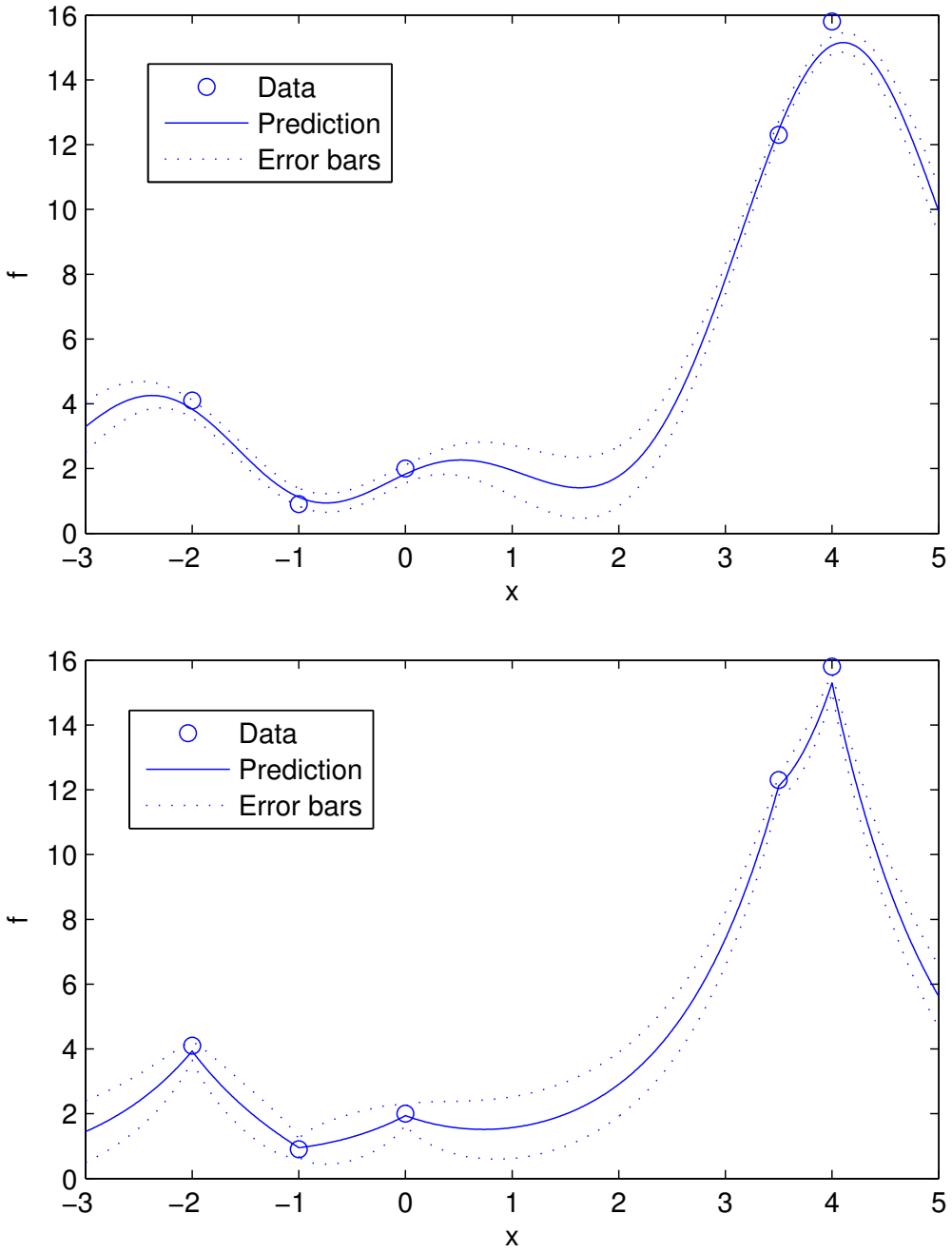


Figure 10: Gaussian Process Regression. Result obtained with a Bell shaped K_1 (Top) and Laplacian (Bottom) covariance function.

Popular choices of covariance functions to generate smooth regression functions include a Bell shaped one

$$K_1(x_i, x_j) = \exp\left(-\frac{1}{2}\|x_i - x_j\|^2\right)$$

and a Laplacian

$$K_2(x_i, x_j) = \exp\left(-\frac{1}{2}\|x_i - x_j\|\right)$$

1. Derive the expressions to compute the predictive density

$$p(\hat{y}|y_{1:L}, x_{1:L}, \hat{x})$$

2. Write a program to compute the mean and covariance of $p(\hat{y}|y_{1:L}, x_{1:L}, \hat{x})$ to generate figures like Figure 10 for the following data:

```
x = [-2 -1 0 3.5 4]';
y = [4.1 0.9 2 12.3 15.8]';
```

   Try different covariance functions and observation noise covariances R and comment on the nature of the approximation.

3. Suppose we are using a covariance function parameterised by

$$K_\beta(x_i, x_j) \quad = \quad \exp\left(-\frac{1}{\beta}\|x_i - x_j\|^2\right)$$

   Find the optimum regularisation parameter $\beta^*(R)$ as a function of observation noise variance via maximisation of the marginal likelihood, i.e.

$$\beta^* \quad = \quad p(y_{1:N}|x_{1:N}, \beta, R)$$

   Generate a plot of $b^*(R)$ for $R = 0.01, 0.02, \ldots, 1$ for the dataset given in 2.

Return to List of exercises. Return to List of exercises.

Q57*: Partitioned Inverse Equations

We partition a matrix Z as

$$Z = \begin{pmatrix} A & B \\ D & C \end{pmatrix}$$

and define the Schur complements of Z with respect to this partitioning as

$$\begin{aligned} M &= (A - BC^{-1}D) \\ N &= (C - DA^{-1}B) \end{aligned}$$

Verify the following

1.

$$Z^{-1} = \begin{pmatrix} M^{-1} & -M^{-1}BC^{-1} \\ -C^{-1}DM^{-1} & C^{-1} + C^{-1}DM^{-1}BC^{-1} \end{pmatrix}$$

2.

$$Z^{-1} = \begin{pmatrix} A^{-1} + A^{-1}BN^{-1}DA^{-1} & -A^{-1}BN^{-1} \\ -N^{-1}DA^{-1} & N^{-1} \end{pmatrix}$$

Return to List of exercises. Return to List of exercises.

Q58***: Multivariate Gaussian Distribution

A convenient property of the multivariate Gaussian distribution is that its marginals and conditionals are Gaussians, hence can be expressed by a mean and a covariance parameter. In this exercise, we will investigate these important properties.

Consider the joint distribution over the variable

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

where the joint distribution is Gaussian $p(x) = \mathcal{N}(x; \mu, \Sigma)$ where

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$
$$\Sigma = \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_2 \end{pmatrix}$$

1. (Conditionals) Find the following
   a) $p(x_1|x_2)$
   b) $p(x_2|x_1)$
2. (Marginals) Find
   a) $p(x_1)$
   b) $p(x_2)$

   Using the partitioned inverse equations, you need to rearrange

$$p(x_1, x_2) \propto \exp\left(-\frac{1}{2}\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^\top \begin{pmatrix} \Sigma_1 & \Sigma_{12} \\ \Sigma_{12}^\top & \Sigma_2 \end{pmatrix}^{-1} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}\right)$$

   bring the expression in form of $p(x_1)p(x_2|x_1)$ (or $p(x_2)p(x_1|x_2)$) where the marginal and conditional can be easily identified. See also Bishop, section 2.3.

Return to List of exercises. Return to List of exercises.

Q59**: Prediction and Update Equations

Consider the following model:

$$\begin{aligned} x_0 &\sim \mathcal{N}(x_0; \mu_0, \Sigma_0) \\ x_1|x_0 &\sim \mathcal{N}(x_1; Ax_0, Q) \\ y_1|x_1 &\sim \mathcal{N}(y_1; Cx_1, R) \end{aligned}$$

1. Draw the graphical model
2. Find the following quantities and express them as Gaussian Distributions in moment parametrisation
   a) $p(x_1)$
   b) $p(x_0, x_1)$
   c) $p(y_1)$
   d) $p(x_1|y_1)$
   e) $p(x_0, x_1|y_1)$

Return to List of exercises. Return to List of exercises.

Q60*: Multiplication of Gaussian Kernels

Express

$$G(x) \quad = \quad K_1(x)K_2(x)$$

in form $e^{\alpha}\mathcal{N}(x;\mu,\Sigma)$ where for $i = 1, 2$

1. $K_i = \mathcal{N}(x; 0, 1)$
2. $K_i = \mathcal{N}(x; 0, P_i)$
3. $K_i = \mathcal{N}(x; \mu_i, 1)$
4. $K_i = \mathcal{N}(x; \mu_i, P_i)$

Return to List of exercises. Return to List of exercises.

Q61**: Log-partition Function and Its Derivatives

We call a probability distribution with density $p(x;\theta)$ on a set $\mathcal{X}^n \subset \mathbb{R}^n$ an exponential family[1] if it has the following functional form

$$\begin{aligned} p(x;\theta) &= \exp(\theta^\top \phi(x) - A(\theta)) \\ \int_{\mathcal{X}^n} dx\; p(x;\theta) &= \exp(-A(\theta)) \int_{\mathcal{X}^n} dx\; \exp(\theta^\top \phi(x)) = 1 \\ A(\theta) &= \log \int_{\mathcal{X}^n} dx\; \exp(\theta^\top \phi(x)) \end{aligned}$$

- The elements of the vector θ are known as *exponential* or *canonical* parameters.
- The functions $\phi(x)$ are the *sufficient statistics*.
- The function $A(\theta)$ is known as the *log partition function* or the *cumulant generating function* and ensures that the distribution normalizes to one. The log-partition function is defined through an integral. Hence we have to ensure that this integral exists (i.e. is finite). We define the set of valid parameters as $\Theta \equiv \{\theta | A(\theta) < \infty\}$

We already know that many well-known distributions belong to an exponential family. The derivatives of $A(\theta)$ provide the cumulants of the distribution[2].

1. Show that

$$\frac{\partial}{\partial \theta} A(\theta) = \langle \phi(x) \rangle_{p(x;\theta)}$$

   i.e., the derivative of the log-partition function gives the expected sufficient statistics

2. Show that the Hessian (the matrix of second derivatives) is given as

$$\frac{\partial^2}{\partial \theta \partial \theta^\top} A(\theta) = \left\langle \phi(x)\phi(x)^\top \right\rangle - \langle \phi(x) \rangle \langle \phi(x) \rangle^\top$$

   This latter equation, by Jensen's inequality shows that the Hessian is always positive definite, hence $A(\theta)$ is convex.

3. Express the following distributions as an exponential family, identify the log partition function $A(\theta)$ and the canonical parameters θ, and by calculating the derivatives of $A(\theta)$, find the expected sufficient statistics
   a) Gaussian $\mathcal{N}(x;\mu,\Sigma)$
   b) Gamma $\mathcal{G}(x;a,b)$
   c) Beta $\mathcal{B}(x;\alpha,\beta)$
   d) Bernoulli $\mathcal{BE}(x;\pi)$

[1] Note that in the lecture we gave a slightly more general definition of an exponential family including a scaling function $h(x)$.

[2] The mean is the first cummulant, the covariance is the second cumulant e.t.c. The term "cumulant" come from the fact that when two independent random variables are added, their cumulants are added (accumulated), too. Note the subtle difference between a moment generating function which is defined as $\langle \exp(\theta x) \rangle_{p(x)}$.

e) Poisson $\mathcal{P}(x;\lambda)$

Return to List of exercises. Return to List of exercises.

Q62**: Gibbs Sampler For One Sample Source Separation

In this exercise you will implement a Gibbs sampler for a toy model described in the lecture.

$$\begin{aligned} s_1 &\sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \\ s_2 &\sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \\ x|s_1, s_2 &\sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R) \end{aligned}$$

1. Derive an expression for the exact posterior $p(s_1, s_2|x)$ when P_i, μ_i and R are known parameters for $i = 1, 2$.
2. Derive the full conditionals $p(s_1|s_2, x)$ and $p(s_2|s_1, x)$
3. Implement a Gibbs sampler. We will use the following parameters: $\mu_1 = 3$, $\mu_2 = 5$, $P_1, P_2 = 0.5$ and $R = 0.3$. Monitor the convergence of ergodic averages to the exact mean and covariance. How many iterations does it take until the posterior mean and covariance are correct 5% if $s^{(0)} = (\mu_1, \mu_2)$?.
4. Repeat the previous experiment with $R = 0.005$. How many iterations does it take until convergence if $s^{(0)} = (\mu_1, \mu_2)$? This should illustrate the fact that when two variables are strongly correlated the Gibbs sampler moves very slowly.

Return to List of exercises. Return to List of exercises.

Q63***: Sampling For Gaussians

Given the model

$$\begin{aligned} x_0 &\sim \mathcal{N}(x_0; 0, \Sigma) \\ x_1|x_0 &\sim \mathcal{N}(x_1; Ax_0, Q) \end{aligned}$$

where

$$\mathcal{N}(x; \mu, \Sigma) \equiv |2\pi\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

is the multivariate Gaussian distribution, A, Q and Σ are known matrices.

1. Find the joint distribution $p(x_0, x_1)$ and express it as a multivariate Gaussian using block matrices.
2. Using the block matrix inverse equations, find a factorisation of $p(x_0, x_1)$ as $p(x_1)p(x_0|x_1)$ and express the factors as Gaussian distributions.
3. Assume now that we observe y where

   $$y|x_1 \sim \mathcal{N}(y; Cx_1, R)$$

   Find an expression for $p(x_0, x_1|y)$.
4. Derive the full conditionals $p(x_0|x_1, y)$ and $p(x_1|x_0, y)$
5. Implement a Gibbs sampler to sample from $p(x_0, x_1|y)$.
6. Implement a rejection sampler to sample from $p(x_0, x_1|y)$ using $Mp(x_0, x_1)$ as the proposal, where M is a suitable positive number. Compute the rejection ratio.
7. Implement an importance sampler to sample from $p(x_0, x_1|y)$ using $p(x_0, x_1)$ as the proposal.

Return to List of exercises. Return to List of exercises.

Q64***: Transition Kernels

x_1 and x_2 are two discrete random taking values in the discrete set $\mathcal{X}$. Suppose we have the joint distribution given as a table $\phi_{a,b}$ where $\pi(x) = p(x_1 = a, x_2 = b) = \frac{1}{Z}\phi_{a,b}$.

Suppose we implement a Metropolis algorithm to sample from this target distribution with the following proposal technique: Given the current configuration $x^{(n)} = (x_1^{(n)}, x_2^{(n)})$, for each n, we choose an index $i^{(n)} \in \{1, 2\}$ randomly with probability 0.5 and choose $x_{i^{(n)}}$ according to a uniform distribution.

1. Implement a Metropolis algorithm using the above proposal mechanism
2. Write down the state transition diagram of the proposal distribution and indicate the state transition probabilities,
3. Write a program to compute the acceptance probability and the transition Kernel T_M of this Metropolis algorithm. Verify the results of your program for $\phi_{-1,-1} = \phi_{1,1} = 3$, $\phi_{-1,1} = \phi_{1,-1} = 0.2$. *The transition kernel will be a* 4×4 *matrix. If you are programming in Matlab, you may consider using* $4 - D$ *arrays and the function* `reshape`.
4. (Optional) Verify numerically if detailed balance condition is satisfied by this particular Metropolis algorithm (i.e., if $T_M(x|x')\pi(x') = T_M(x'|x)\pi(x)$).
5. Implement a deterministic scan Gibbs sampler (that is we sample alternatingly from the full conditional distributions $p(x_1|x_2)$ and $p(x_2|x_1)$).
6. Write a program to compute the Gibbs transition Kernel T_G. Verify the results of your program for $\phi_{-1,-1} = \phi_{1,1} = 3$, $\phi_{-1,1} = \phi_{1,-1} = 0.2$.
7. Using eigenvalue decomposition of T_G and T_M, estimate which transition kernel will converge faster to the stationary distribution.

Return to List of exercises. Return to List of exercises.

Q65*: Factorization of Probability Tables

Consider the following probability table

$p(x_1, x_2)$	$x_2 = 0$	$x_2 = 1$
$x_1 = 0$	1/10	3/10
$x_1 = 1$	2/10	4/10

1. Factorise this table as
   a) $p(x_1)p(x_2|x_1)$
   b) $p(x_2)p(x_1|x_2)$
2. Suppose we would like to enforce a new prior $p^*(x_1)$ for this probability table such that

$$p^*(x_1) = \sum_{x_2} p^*(x_1, x_2)$$

   We let

$$p^*(x_1, x_2) = \frac{p^*(x_1)}{p(x_1)} p(x_1, x_2)$$

   This operation can be interpreted as multiplying-in a new prior. Suppose $p^*(x_1) = [0.5 \;\; 0.5]$. Find the new table $p^*(x_1, x_2)$.

Return to List of exercises. Return to List of exercises.

Q66**: Gibbs sampler for the AR model

In this problem, you will develop a Gibbs sampler for an AR model:

$$\begin{aligned} A &\sim \mathcal{N}(A; 0, 1.2) \\ R &\sim \mathcal{IG}(R; 0.4, 250) \\ x_k|x_{k-1}, A, R &\sim \mathcal{N}(x_k; Ax_{k-1}, R) \\ x_0 &= 1 \qquad\qquad x_1 = -6 \end{aligned}$$

$$\begin{aligned} \mathcal{N}(x; m, r) &= \exp\{-\frac{1}{2}(x^2 + m^2 - 2xm)/r - \frac{1}{2}\log(2\pi r)\} \\ \mathcal{IG}(r; a, b) &= \exp\left(-(a+1)\log r - \frac{1}{br} - \log\Gamma(a) - a\log b\right) \end{aligned}$$

1. Write the expression for the full joint distribution and assign terms to the individual factors on the factor graph

$$\begin{aligned} p(A, R|x_0, x_1) &\propto p(x_1|x_0, A, R)p(A)p(R) \\ &= \mathcal{N}(x_1; Ax_0, R)\mathcal{N}(A; 0, P)\mathcal{IG}(R; \nu, \beta/\nu) \\ &\propto \exp\left(-\frac{1}{2}\frac{x_1^2}{R} + x_0x_1\frac{A}{R} - \frac{1}{2}\frac{x_0^2A^2}{R} - \frac{1}{2}\log 2\pi R\right) \\ &\quad \exp\left(-\frac{1}{2}\frac{A^2}{P}\right)\exp\left(-(\nu+1)\log R - \frac{\nu}{\beta}\frac{1}{R}\right) \end{aligned}$$

2. Derive the full conditional distributions $p(A|R, x_0, x_1)$ and $p(R|A, x_0, x_1)$ *The result should be*

$$\begin{aligned} p(A|R, x_0, x_1) &= \mathcal{N}(A; \mu_A, \Sigma_A) \\ \Sigma_A &= \left(\frac{x_0^2}{R} + \frac{1}{P}\right)^{-1} \\ \mu_A &= \Sigma_A\frac{x_0x_1}{R} \\ p(R|A, x_0, x_1) &= \mathcal{IG}\left(R; \nu + \frac{1}{2}, \left(\frac{1}{2}(x_1 - Ax_0)^2 + \frac{\nu}{\beta}\right)^{-1}\right) \end{aligned}$$

3. Implement the Gibbs sampler and plot the results

   The Matlab code is given below. To generate an inverse gamma random variable, we used $1/(b$ `gamrnd`$(a))$

```
beta_nu = 250; nu = 0.4; P = 1.2; x_0 = 1; x_1 = -6; T = 10000;
R = zeros(1, T); A = zeros(1, T);
A(1) = -6; R(1) = 0.00001;

for t=2:T,
    Sig = 1/(x_0^2/R(t-1) + 1/P);
    mu = Sig*x_0*x_1/R(t-1);
    A(t) = sqrt(Sig)*randn + mu;

    b = 0.5*(x_1 - A(t)*x_0).^2 + 1/beta_nu;
    R(t) = 1/(gamrnd(nu+0.5, 1/b));
end;
```

4. Implement the simulated annealing and iterative improvement to find the mode of the posterior $p(A, R|\cdot)$. *The mode of an inverse gamma distribution is at* $r = 1/((a+1)b)$.

Return to List of exercises. Return to List of exercises.

Q67*: Sampling from Multivariate Gaussians

Consider a Gaussian random vector $y \in \mathbb{R}^N$. We wish to generate samples from this distribution using independent Gaussians.

1. Let $z \in \mathbb{R}^N$

$$z \sim \mathcal{N}(z; 0, I_N)$$

Show that, if

$$y = Wz + \mu$$

where $\Sigma = W^\top W$, then

$$y \sim \mathcal{N}(y; \mu, \Sigma)$$

2. Write a program that generates samples from $\mathcal{N}(y; \mu, \Sigma)$.
3. Suppose we wish to set some elements of the vector y indexed by α are set to known values given by $\tilde{y}_\alpha$, i.e.,

$$y = \begin{pmatrix} \tilde{y}_\alpha \\ y_{-\alpha} \end{pmatrix}$$

For example, if $N = 4$ and $\alpha = \{1, 4\}$, then $-\alpha = \{2, 3\}$. Derive the conditional distribution of $y_{-\alpha}$ and write a program, that given μ and Σ samples y such that $y_\alpha = \tilde{y}_\alpha$.

Return to List of exercises. Return to List of exercises.

Q68**: Resampling

In the literature, there are several resampling methods. The most popular are

- Multinomial Resampling
- Systematic Resampling
- Residual Resampling

In this exercise, we will implement and investigate these techniques.

1. Multinomial resampling is equivalent to sampling histograms with N bins where each bin has probability $\tilde{w}^{(i)}$ for $i = 1 \dots N$ and $\sum_i \tilde{w}^{(i)} = 1$. The algorithm is as follows:

   - $u_k \sim \mathcal{U}(u_k; 0, 1)$ (uniform in $[0, 1]$) for $k = 1 \dots N$
   - Define intervals

   $$I_i \quad = \quad \left(\sum_{j=1}^{i-1} \tilde{w}^{(j)}, \sum_{j=1}^{i} \tilde{w}^{(j)} \right]$$

   - Count number of u that fall into the interval I_i

   $$N_i \quad = \quad \sum_k [u_k \in I_i]$$

   This procedure is equivalent to sampling from a Multinomial distribution

   $$N_{1:N} \quad \sim \quad \mathcal{M}(N_{1:N}; \tilde{w}^{(i)}, N)$$

   and is shown in Fig 11. Implement a program for multinomial resampling. *It is possible to do*

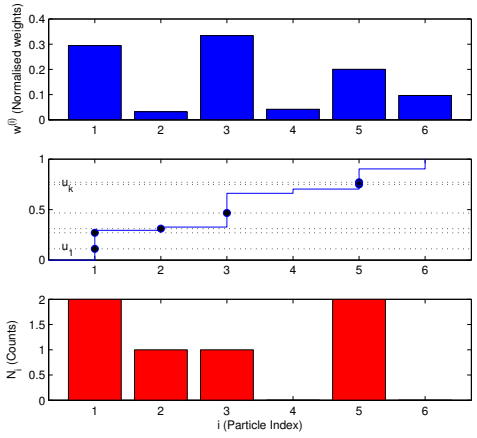


Figure 11: Sampling from a Multinomial

   this in two lines of Matlab code using the `histc` *function.*

2. Systematic resampling is very similar to multinomial sampling but the u are chosen as follows:

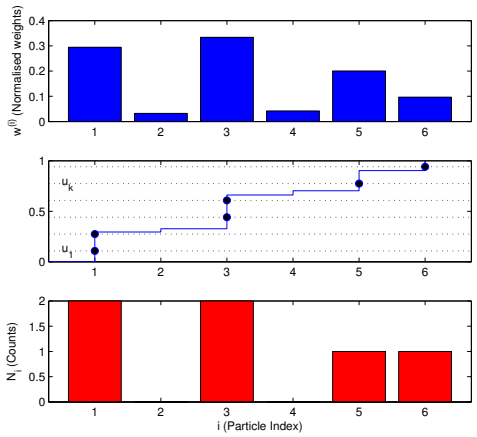


Figure 12: Systematic Resampling

- $u_1 \sim \mathcal{U}(u_1; 0, 1/N)$ (uniform in $[0, 1/N]$)
- $u_k = u_1 + (k-1)/N$ for $k = 2 \ldots N$

and is shown in Fig 12. Implement a program for systematic resampling.

3. Residual resampling works as follows

   - Every bin is assigned first a fixed number of off-springs proportional to its weight

   $$\tilde{N}_i = \lfloor N\tilde{w}^{(i)} \rfloor$$

   - Calculate the "residual" weights and normalise

   $$\begin{aligned} \bar{W}^{(i)} &= \tilde{w}^{(i)} - \tilde{N}_i/N \\ \bar{w}^{(i)} &= \bar{W}^{(i)} / \sum_{i'} \bar{W}^{(i')} \end{aligned}$$

   - Sample from the residual $\bar{N} = N - \sum_i \tilde{N}_i$

   $$\bar{N}_{1:N} \sim \mathcal{M}(\bar{N}_{1:N}; \bar{w}^{(i)}, \bar{N})$$

   An illustration is shown in Fig 13. Implement a program for residual resampling.

4. For each method, given $N = 10$ and $w^{(i)} \propto i/N$ for $i = 1 \ldots N$ estimate by simulation the mean and the variance of the Monte Carlo error

   $$N_i - w^{(i)} N$$

   Conclude which resampling method is a better choice. *The means should be very close to zero as these methods are unbiased. The variances for multinomial resampling are around* $0.18, 0.35, 0.51, 0.66, 0.83, 0.97, 1.11, 1.25, 1.37, 1.48$ *and for systematic resampling* $0.15, 0.23, 0.25, 0.20,$

Return to List of exercises. Return to List of exercises.

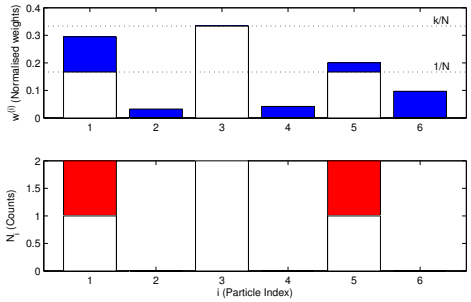


Figure 13: Residual Resampling

Q69***: Kalman Filter, Particle Filter

Consider the following linear dynamical model for $t = 1, 2, \ldots,$

$$\begin{aligned} s_t &= \begin{pmatrix} 1 & \Delta \\ 0 & 1 \end{pmatrix} s_{t-1} + w_t \\ y_t &= \begin{pmatrix} 1 & 0 \end{pmatrix} s_t + \epsilon_t \end{aligned}$$

Here, w_t and ϵ_t are independent and

$$\begin{aligned} s_0 &\sim \mathcal{N}(s_0; 0, P) \\ w_t &\sim \mathcal{N}(w_t; 0, Q) \\ \epsilon_t &\sim \mathcal{N}(\epsilon_t; 0, R) \end{aligned}$$

1. Draw the graphical model for this process.
2. Write a program that would generate $y_{1:T}$ and s_t given T, P, Q, R and Δ. How do the typical trajectories look like for $P = 1$, $Q = 0.01$, $R = 0.1$, $\Delta = 1/2$? What happens when Δ is changed?
3. Derive an algorithm for computing the mean and covariance matrix of $p(s_t|y_{1:t})$ for each t recursively. *This is the Kalman filtering recursion*
4. Write a Matlab program to visualise the exact filtering density. You can plot equal probability ellipses as shown in the lectures.
5. Implement a sequential Monte Carlo algorithm to estimate the mean and the variance of $p(s_t|y_{1:t})$ using the state transition distribution as the proposal.

Return to List of exercises. Return to List of exercises.

Q70**: Hidden Markov Model

Implement the correction smoother to compute $p(x_t|y_{1:T})$ for the following HMM with $y_t \in 0, 1$ and $x_t \in a, b, c$:

$$p(y_{1:T}, x_{0:T}) = p(x_0) \prod_{t=1}^{T} p(y_t|x_t)p(x_t|x_{t-1})$$

Note that this setup is slightly different from the model discussed in the class that due to x_0. Use the following parameters:

$$\begin{aligned} p(x_t = i|x_{t-1} = j) &= A_{i,j} \\ A &= \begin{pmatrix} 0.9 & 0 & 0.3 \\ 0.1 & 0.8 & 0 \\ 0 & 0.2 & 0.7 \end{pmatrix} \\ p(y_t = k|x_t = i) &= B_{k,i} \\ B &= \begin{pmatrix} 0.99 & 0.6 & 0.01 \\ 0.01 & 0.4 & 0.99 \end{pmatrix} \\ p(x_0) &\sim \mathcal{U}[\imath, \infty] \end{aligned}$$

Verify your results by comparing with the forward-backward algorithm on the following observation sequence:

$$y_{1:12} = [0001m110mm01]$$

where m means missing data. The results for both algorithms must be identical.

Return to List of exercises. Return to List of exercises.

Q71***: Variational Bayes for Changepoint Model

Consider the following changepoint model:

$$\begin{aligned}
\lambda_0 &\sim \mathcal{G}(\lambda_0; a, b_0) \\
r_t &\sim \mathcal{BE}(r_t; \pi_1) \qquad\qquad \pi_0 = 1 - \pi_1 \\
\lambda_t | r_t, \lambda_{t-1} &\sim \delta(\lambda_t - \lambda_{t-1})^{[r_t=0]} \mathcal{G}(\lambda_t; a, B)^{[r_t=1]} \\
x_t | \lambda_t &\sim \mathcal{PO}(x_t; \lambda_t)
\end{aligned}$$

Here, $r_t \in D_r = \{0, 1\}$ and $\lambda_t \in \mathbb{R}^+$. The symbols $\mathcal{G}$, $\mathcal{BE}$ and $\mathcal{PO}$ denote the gamma, Bernoulli and the Poisson distribution respectively

$$\begin{aligned}
\mathcal{G}(\lambda; a, b) &= \exp\left((a-1)\log\lambda - b\lambda - \log\Gamma(a) + a\log b\right) \\
\mathcal{BE}(r; \pi) &= \exp\left(r\log\pi + (1-r)\log(1-\pi)\right) \\
\mathcal{PO}(x; \lambda) &= \exp\left(x\log\lambda - \lambda - \log\Gamma(x+1)\right)
\end{aligned}$$

In the class, we have assumed that the parameters a, b_0 and B are known. In practice we don't know these.

- Assume that a and b_0 are known but

$$B \sim \mathcal{G}(B; 1, 10)$$

1. Draw the associated graphical model
2. Derive and implement an EM algorithm to find $B^* = \arg\max_B p(r_{1:T}, \lambda_{1:T}, B | x_{1:T})$
3. Derive a variational Bayes algorithm that approximates $p(r_{1:T}, \lambda_{1:T}, B | x_{1:T})$ by $q(r_{1:T}, \lambda_{1:T})q(B)$. Choose a gamma density for $q(B)$.

   Note that the algorithms will be very similar where EM learns a single parameter whereas VB estimates a distribution on B. Test your algorithms on the coal mining disaster dataset given as

   4 5 4 1 0 4 3 4 0 6 3 3 4 0 2 6 3 3 5 4 5 3 1 4 4 1 5 5 3 4 2 5 2 2 3 4 2 1 3 2 2 1 1 1 1 3 0 0 1 0 1 1 0 0 3 1 0 3 2 2 0 1 1 1 0 1 0 1 0 0 0 2 1 0 0 0 1 1 0 2 3 3 1 1 2 1 1 1 1 2 4 2 0 0 0 1 4 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1

   Show clearly the posterior probability of the changepoints.

Return to List of exercises. Return to List of exercises.

Q72***: Kalman Filtering and Smoothing

In this question, we will investigate the Kalman filter and the Kalman smoother for interpolation of signals.

Suppose we are given a noisy signal $y_{0:T-1} \equiv (y_0, y_1, \ldots, y_{T-1})$. Furthermore, suppose that some sample values are missing at time indices $t \in \mathcal{I}$, where $\mathcal{I}$ is a known set.

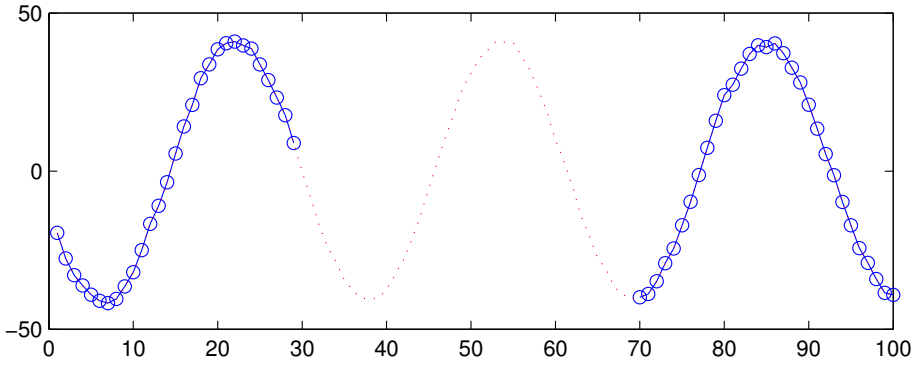
We will use the following linear dynamical system

$$\begin{aligned} x_0 &\sim \mathcal{N}(0, S) \\ x_t|x_{t-1} &\sim \mathcal{N}(x_t; Ax_{t-1}, Q) \\ y_t|x_t &\sim \mathcal{N}(Cx_t, R) \end{aligned}$$

Here, $S = 1000I_2$, $Q = 0.001I_2$ and $R = 0.5$. The transition model

$$\begin{aligned} C &= \sqrt{2}\begin{pmatrix} 1 & 0 \end{pmatrix} \\ A &= \exp(-\rho)\begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix} \end{aligned}$$

1. Write a program to generate data from this LDS given ω and ρ. By investigating different values and the associated realisations, explain the model.
2. Assume that y_t are missing for $t \in \{L \ldots T-L-1\} = \mathcal{I}$. Use the Kalman filter and smoother to estimate $p(y_t|y_{-\mathcal{I}})$ for $t \in \mathcal{I}$. Plot your results (the mean and the variance as 3σ error bars) for $T = 100$ and $L = 3, 10$ and 30.



3. In the above, we have assumed that the parameters are known. This is often not the case in practice. Devise and implement a method based on the Kalman filter to estimate the ML solution for ω and ρ. *Consider computing the loglikelihood for a grid of ρ and ω.*
4. Assume we are given the following sequence $y_{1:12} = [-300, -330, m, m, m, m, m, m, m, m, 250, 280]$. Find the mean and the variance of $p(y_t|y_{1:2}, y_{11:12}, \omega^*, \rho^*)$ for $t = 2 \ldots 10$.

Return to List of exercises. Return to List of exercises.

Q73***: Clustering with Missing Values

Suppose we are given the following dataset in the left column, where `NaN`'s denote missing values. In this question, you will only use the dataset with missing values; the complete dataset is for comparison only. Here, each row is a point on R^2.

Dataset		Without Missing	
NaN	-0.3742	1.2762	-0.3742
NaN	3.1832	-0.0999	3.1832
-0.6887	NaN	-0.6887	2.6277
2.2469	NaN	2.2469	-0.1871
-0.8456	NaN	-0.8456	1.7727
NaN	0.8245	-0.0694	0.8245
0.1537	NaN	0.1537	1.3422
-0.0754	NaN	-0.0754	2.2690
-0.7237	NaN	-0.7237	1.7694
NaN	0.5313	0.2801	0.5313
NaN	3.9396	0.8288	3.9396
2.7675	NaN	2.7675	-0.2526
NaN	1.1712	0.3429	1.1712
NaN	2.7526	1.5023	2.7526
0.6299	NaN	0.6299	1.2993
2.2560	NaN	2.2560	-0.4466
1.7509	NaN	1.7509	-0.4560
-0.8206	NaN	-0.8206	1.2730
NaN	0.6409	-0.1679	0.6409
NaN	-0.0942	2.4861	-0.0942
0.5593	NaN	0.5593	2.2157
NaN	-1.1865	2.3954	-1.1865
-0.8162	NaN	-0.8162	1.6424
0.0877	NaN	0.0877	2.9030
-0.2320	NaN	-0.2320	3.0092
2.3803	NaN	2.3803	0.9393
2.1016	NaN	2.1016	-0.5255
-1.1851	NaN	-1.1851	2.1244
NaN	1.7659	0.6548	1.7659
NaN	1.7531	-0.4326	1.7531
-0.8773	NaN	-0.8773	1.6056
-0.1446	NaN	-0.1446	2.0367
NaN	2.4891	-0.0510	2.4891
0.0875	NaN	0.0875	3.4512
NaN	3.4436	0.2496	3.4436
-0.5785	NaN	-0.5785	2.4212
-0.2082	NaN	-0.2082	2.4801
NaN	0.2512	0.4751	0.2512
NaN	-1.1528	0.9384	-1.1528
NaN	3.1762	0.3215	3.1762

1. Suppose we believe that data comes from a mixture of Gaussians with $K = 2$ components, with different mean vectors μ_1 and μ_2 and a diagonal covariance matrix $\Sigma = \mathbf{diag}(s_x, s_y)$ that is shared by both components. Develop a Bayesian model for this scenario, define appropriate priors and conditional distributions and sketch a graphical model.
2. Write the mathematical expression for the joint density
3. Develop an EM algorithm to estimate the MAP values of μ_1, μ_2 and Σ given the dataset.
4. Develop an algorithm to predict the missing values and the posterior probability of component labels (for $K = 2$) for each data point. Compute the mean of your prediction and the error bars (as given by the standard deviation of the predictive distribution for the missing values). Comment on the quality of the predictions.
5. Sketch a variational Bayes algorithm, along Bayesian model selection principles, to estimate the most likely number of components, when K is unknown.

Return to List of exercises. Return to List of exercises.

Q74***: Changepoint

Suppose we observe the following dataset $x_t \in 0, 1$ for $t = 1 \ldots 50$

0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1

1. Suppose we know that the data comes from the following model

$$\begin{aligned} \pi_1 &\sim \mathcal{B}(1,1) \\ \pi_2 &\sim \mathcal{B}(1,1) \\ x_t &\sim \begin{cases} \mathcal{BE}(x_t; \pi_1), & t <= n \\ \mathcal{BE}(x_t; \pi_2), & t > n \end{cases} \end{aligned}$$

2. Derive, compute and plot the posterior probability of $p(n|x_{1:50})$ given that n is *a-priori* uniform.

Return to List of exercises. Return to List of exercises.

Q75**: Factorizing Gaussians

Given the model

$$\begin{aligned} x_0 &\sim \mathcal{N}(x_0; 0, \Sigma) \\ x_1|x_0 &\sim \mathcal{N}(x_1; Ax_0, Q) \end{aligned}$$

where

$$\mathcal{N}(x; \mu, \Sigma) \equiv |2\pi\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

is the multivariate Gaussian distribution, A, Q and Σ are known matrices.

1. Find the joint distribution $p(x_0, x_1)$ and express it as a multivariate Gaussian.
2. Find a factorisation of $p(x_0, x_1)$ as $p(x_1)p(x_0|x_1)$ and express the factors as Gaussian distributions.

Return to List of exercises. Return to List of exercises.

Q76***: Metropolis and Gibbs

x_1 and x_2 are two discrete random variables taking values in $\{-1, 1\}$. Suppose we have the joint distribution $p(x_1 = a, x_2 = b) = \pi_{a,b}$. We further have $g = \pi_{-1,1} = \pi_{1,-1} > \pi_{1,1} = \pi_{-1,-1}$.

Suppose we implement a Metropolis algorithm to sample from this target distribution with the following proposal technique: Given the current configuration $x^{(n)} = (x_1^{(n)}, x_2^{(n)})$, for each n, we choose an index $i^{(n)} \in \{1, 2\}$ randomly with probability 0.5 and flip the sign of $x_{i^{(n)}}$.

1. Write down the state transition diagram of the proposal distribution and indicate the state transition probabilities,
2. Find an expression for the acceptance probability as a function of g,
3. Write the pseudocode for the Metropolis sampler,
4. Write down the state transition diagram of the transition Kernel T_M of this Metropolis algorithm and indicate the transition probabilities,
5. Verify if detailed balance condition is satisfied by this particular Metropolis algorithm (i.e., if $T_M(x|x')\pi(x') = T_M(x'|x)\pi(x)$) for all values of g.
6. Suppose we also implement a deterministic scan Gibbs sampler (that is we sample alternatingly from the full conditional distributions $p(x_1|x_2)$ and $p(x_2|x_1)$). Write down the pseudocode.
7. Write down an expression for the Gibbs transition Kernel T_G in terms of g.
8. Verify detailed balance is satisfied the Gibbs transition Kernel T_G for all values of g.

Return to List of exercises. Return to List of exercises.

Q77**: Variational Methods

Consider the following probability tables

$p(x_1, x_2)$	$x_2 = 0$	$x_2 = 1$
$x_1 = 0$	$\pi_{0,0}$	$\pi_{0,1}$
$x_1 = 1$	$\pi_{1,0}$	$\pi_{1,1}$

$q(x_1, x_2)$	$x_2 = 0$	$x_2 = 1$
$x_1 = 0$	$(1 - q_1)(1 - q_2)$	$(1 - q_1)q_2$
$x_1 = 1$	$q_1(1 - q_2)$	$q_1 q_2$

By minimising $KL(q||p)$, show that the solution satisfies the relation

$$\begin{aligned} q(x_1) &\propto \exp\{\langle \log p \rangle_{q(x_2)}\} \\ q(x_2) &\propto \exp\{\langle \log p \rangle_{q(x_1)}\} \end{aligned}$$

Return to List of exercises. Return to List of exercises.

Q78**: Matrix Inversion Lemma

Consider the following model hierarchical model where $k = 1 \dots K$

$$\begin{aligned} s_k &\sim \mathcal{N}(s_k; \lambda_k, 1) \\ s &= (s_1, \dots, s_K)^\top \\ x &\sim \mathcal{N}(x; Cs, vI) \end{aligned}$$

Here C is a $N \times K$ matrix with mutually orthogonal rows, v is a scalar and I is a $N \times N$ identity matrix. Using the matrix inversion lemma:

$$\left(I + C^\top V^{-1} C\right)^{-1} = I - C^\top \left(V + CC^\top\right)^{-1} C$$

Find

1. The posterior $p(s|x)$
2. Show that when $v \to 0$ we have

$$\langle s|x \rangle = \lambda + C^\top (x - C\lambda)$$

   Interpret this solution geometrically by giving an example.

Return to List of exercises. Return to List of exercises.

Q79***: Interpolation via EM

Consider a Markov chain,

$$p(x_1)p(x_2|x_1)p(x_3|x_2)$$

where the transition and initial state probabilities are given by $p(x_t = i|x_{t-1} = j) = A_{i,j}$ and $p(x_1 = i) = \pi_i$. Assume that $t = 1, 2, 3$ only and π are known.

Suppose we observe N realisations from the model. However, for $n = 1 \dots N$, we observe $x_1 = \hat{x}_1^{(n)}$ and $x_3 = \hat{x}_3^{(n)}$ but all the observations for x_2. How can we estimate the transition matrix $A_{i,j}$?.

This is a typical setting for the EM algorithm where the observations are x_1 and x_3, the missing data are x_2 and the unknown parameters are π and A.

1. Draw the graphical models for this inference problem
2. Write down the complete data loglikelihood $\log p(x_{1:3}|\pi, A)$
3. Show how to compute the E-step

$$p(x_2^{(n)}|x_1^{(n)}, x_3^{(n)})$$

4. In the M step, find the energy term

$$\mathcal{F}(\theta; \theta^{(\tau)}) \quad = \quad \langle \log p(x_{1:3}|\pi, A) \rangle$$

   and in order to maximize w.r.t. $A_{s,j}$, form the Lagrangian and take the derivative and set to zero. Show that this leads to the result

$$A_{i,j} \quad = \quad \frac{\sum_n \left(\left\langle \left[i = x_2^{(n)} \right] \right\rangle \left[j = x_1^{(n)} \right] + \left[i = x_3^{(n)} \right] \left\langle \left[j = x_2^{(n)} \right] \right\rangle \right)}{\sum_s \sum_n \left(\left\langle \left[s = x_2^{(n)} \right] \right\rangle \left[j = x_1^{(n)} \right] + \left[s = x_3^{(n)} \right] \left\langle \left[j = x_2^{(n)} \right] \right\rangle \right)}$$

5. The following Matlab code generates data from the model

```
% Number of sequences
N = 100;
% Number of time slices
T = 3;
S = 2;
A_true = [0.9 0.1;0.3 0.7]';
pi_true = [0.4 0.6];
data.x = zeros(N, T);
% Generate data
for n=1:N,
  for t=1:T,
    if t==1,
      data.x(n, t) = randgen(pi_true);
    else
      data.x(n, t) = randgen(A_true(:,data.x(n, t-1)));
```

```
    end;
  end;
end;
```

The following code contains the skeleton of the EM algorithm. Fill in the M step

```
%% EM algorithm
%% We pretend that data.x(:, 2) are missing
%% We only estimate A

A = 0.5*ones(2);
p_x2 = zeros(S, N);
for ep=1:100,
% E step
for n=1:N,
  Z = A(data.x(n, 3), :)*A(:, data.x(n, 1));
  p_x2(:, n) = A(data.x(n, 3), :)'.*A(:, data.x(n, 1))./Z;
end;
% logLikelihood computation
LL = zeros(1, N);
for n=1:N,
  LL(n) = log(A(data.x(n, 3), :)*A(:, data.x(n, 1))) ...
          + log(pi_true(data.x(n, 1))) ;
end;
lk = sum(LL);

% M step
.... <Fill in>


A
lk  % the likelihood must increase monotonically!
pause
end;
```

Return to List of exercises. Return to List of exercises.

Q80**: A Probability Table

The goal of this exercise is to investigate sampling algorithms and variational algorithms on a toy example and provide warm up for the following exercises.

Suppose we are given a $N_1 \times N_2$ table L where the entry at i'th row and j'th column denotes

$$L_{i,j} \quad = \quad \log p(x_1 = i, x_2 = j) + \log Z$$

where $p(x_1, x_2)$ is the joint distribution of x_1, x_2 with $x_1 \in \{1, \ldots, N_1\}$ and $x_2 \in \{1, \ldots, N_2\}$ and Z is an unspecified positive constant.

1. Write a program to compute the "variational marginals", i.e., two distributions $q_1(x_1)$ and $q_2(x_2)$ such that

   $$KL(q_1 q_2 || p)$$

   is minimised.
2. Compare the "variational marginals" with the exact marginals $p(x_1)$ and $p(x_2)$ given L.
3. Derive the update equations of an EM algorithm to compute

   $$x_1^* = \operatorname*{argmax}_{x_1} \sum_{x_2} p(x_1, x_2)$$

Return to List of exercises. Return to List of exercises.

Q81**: A Chain

Suppose we are given a probability distribution that factors according to

$$p(x_0, \ldots, x_T) \quad = \quad \frac{1}{Z} \exp\left(\sum_{t=1}^{T} \psi(x_{t-1}, x_t)\right)$$

We know that $x_t \in \{1, \ldots, N\}$

1. Describe a procedure to compute Z and the marginals $p(x_t)$ for $t = 0, \ldots, T$ exactly. Write a program to compute $\log Z$ and $\log p(x_t)$ for $t = 0, \ldots, T$ given $\psi(x_{t-1} = i, x_t = j)$ as a $N \times N \times T$ array with `psi(i, j, t)`. *Draw the factor graph and observe the similarities to the HMM derivation given in the lectures.*
2. Derive an algorithm to compute the Viterbi path

   $$x^*_{0:T} \quad = \quad \operatorname*{argmax}_{x_{0:T}} p(x_0, \ldots, x_T)$$

   and write a program to compute it.
3. Derive a simulated annealing (SA) algorithm to sample from

   $$\frac{1}{Z_\beta} \exp\left(\beta \sum_{t=1}^{T} \psi(x_{t-1}, x_t)\right) = p_\beta(x_0, \ldots, x_T)$$

   where β is an inverse temperature variable. Design an annealing schedule $\beta \to \infty$ to compute the Viterbi path. Compare your solutions to the exact solution.
4. Derive a variational Bayes algorithm that uses a fully factorised approximating distribution

   $$Q \quad = \quad \prod_{t=0}^{T} q(x_t)$$

   Derive the update equations and implement the algorithm. Compare the variational marginals to the true marginals.
5. Derive the variational lower bound and write an algorithm to compute it. Compare to the exact $\log Z$ you have computed earlier. Show with a plot that it is strictly increasing during the iterations.
6. The variational method can also be used to compute the Viterbi path by targeting p_β with $\beta \to \infty$. Using the same schedule as the SA, compare if you find better solutions with annealed VB. The solutions can be compared according to the number of mismatches with the true trajectory and the probability that the solution achieves.

Return to List of exercises. Return to List of exercises.

Q82**: Bayesian Estimation of Gaussians

Consider the following model

$$\begin{aligned} \beta &\sim \mathcal{G}(\beta; \nu, 1) \\ \mu &\sim \mathcal{N}(\mu; 0, 1000) \\ x_i &\sim \mathcal{N}(x_i; \mu, \beta^{-1}) \end{aligned}$$

for $i = 1 \ldots N$. Suppose we are given the following dataset

$$x_{1,6} = \{-6,\ -1,\ -0.2,\ 0.1,\ 2,\ 4\} \equiv X$$

1. Derive and implement a Gibbs sampler to sample from

$$p(\mu, \beta | X, \nu = 0.1)$$

2. Derive and implement a variational Bayes algorithm to approximate $p(\mu, \beta | \nu = 0.1, X)$. Take as the approximating distribution a factorised distribution $Q = q_1 q_2$ where

$$\begin{aligned} q_1(\beta) &= \mathcal{G}(\beta; a, b) \\ q_2(\mu) &= \mathcal{N}(\mu; m, \Sigma) \end{aligned}$$

3. How would you find

$$\nu^* = \operatorname*{argmax}_{\nu} p(x_{1:N} | \nu)$$

   i.e., the ML estimate of ν? Explain.

Return to List of exercises. Return to List of exercises.

Q83****: Movie Rating

A popular model for collaborative filtering for recommender systems is based on matrix factorization. Here, a matrix X (with possibly a lot of missing entries) represents, for example, the ratings x_{ij} of users i for movie j. If we can assume that the actual X is low rank, there is hope for reconstruction of the missing elements. Consider the following model that assumes $X \approx wh^\top$, i.e., X is approximated well with a rank one matrix:

$$\begin{aligned}
x_{ij} &\sim \mathcal{PO}(x_{ij}; w_i h_j) \\
m_{ij} &= \begin{cases} 1 & \text{if } x_{ij} \text{ is observed} \\ 0 & \text{if } x_{ij} \text{ is not observed} \end{cases} \\
h_j &\sim \mathcal{G}(h_j; \alpha, \beta) = \exp((\alpha-1)\log h_j - \beta h_j - \log\Gamma(\alpha) + \alpha\log\beta) \\
w_i &\sim \mathcal{G}(w_i; \alpha, \beta) = \exp((\alpha-1)\log w_i - \beta w_i - \log\Gamma(\alpha) + \alpha\log\beta)
\end{aligned}$$

where

$$\begin{aligned}
i &= 1 \dots I \\
j &= 1 \dots J
\end{aligned}$$

and h_j is the jth movie's rating independent of users' opinion; w_i, the ith user's tendency; and x_{ij}, jth movie's rating by the user i. The matrix M of m_{ij} denotes the known (if 1) and the missing data (if 0).

Generate a synthetic data set $X = \{x_{ij}\}$ with $I = 50$ and $J = 20$. Sample the true w and h values from a gamma distribution with parameters $\alpha = 0.5$ and $\beta = 0.1$. Sample the mask matrices M_q from Bernoulli distributions with parameters $q = 0.1, 0.2, \dots, 0.9$. Our goal will be to predict for each q the same missing entries, corresponding to zero elements of $M_{0.9}$ only. Intuitively, provided that the model is correct, the more data points there are, the better our predictions should be. We will measure the accuracy of our predictions by MSE (mean square error) defined as

$$MSE(\hat{X}) = \sum_{i,j \text{ s.t. } M_{0.9}(i,j)=0} (x_{ij} - \hat{x}_{ij})^2 / \sum_{i,j \text{ s.t. } M_{0.9}(i,j)=0} 1$$

where $\hat{X}$ is the prediction generated by the model.

To generate the masks M_q consistently for all q, you can use the MATLAB code: `q = 0.1:0.1:0.9; rM = rand(I,J); for u=1:9, M = (rM < q(u)); ... end;` .

1. Derive and implement an ICM algorithm for the model to solve

   $$(w,h)^* = \arg\max_w p(x|w,h)p(w)p(h)$$

   predict x via the mean of $p(x|w^*,h^*)$.

2. Derive and implement an EM algorithm for the following inference problem

   $$w^* = \arg\max_w p(x|w)p(w)$$

   predict x via the mode of $p(x|w^*)$.

3. Derive and implement an EM algorithm for the following problem

$$h^* = \arg\max_h p(x|h)p(h)$$

predict x via the mean of $p(x|h^*)$.

Return to List of exercises. Return to List of exercises.