2010 PART IIA 3F3
SIGNAL AND PATTERN PROCESSING
PROFESSOR S D GODSILL

ENGINEERING TRIPOS PART IIA

2010

Module 3F3

SIGNAL AND PATTERN PROCESSING - WORKED SOLUTIONS

*Answer not more than **three** questions.*

All questions carry the same number of marks.

*The **approximate** number of marks allocated to each part of a question is indicated in the right margin.*

There are no attachments.

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS
Engineering Data Book
CUED approved calculator allowed

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 (a) Define the power spectrum for a discrete time random process and give an interpretation for the power spectrum in terms of signal power. Detail any conditions that should be satisfied for a random process to have a power spectrum. [20%]

Solution:

For a wide-sense stationary random process $\{X_n\}$, the power spectrum is defined as the discrete-time Fourier transform (DTFT) of the discrete autocorrelation function:

$$\boxed{\mathcal{S}_X(e^{j\Omega}) = \sum_{m=-\infty}^{\infty} r_{XX}[m]\, e^{-jm\Omega}} \tag{1}$$

Power spectrum for a random process

where $\Omega = \omega T$ is used for convenience.

The condition is that the process is wide-sense stationary.

(b) If two wide-sense stationary random processes $\{A_n\}$ and $\{B_n\}$ are independent, show that the power spectrum of their product $X_n = A_n B_n$ can be expressed as a convolution:

$$\frac{1}{2\pi}\int_{-\pi}^{+\pi} S_A(e^{j\Omega})S_B(e^{j(\theta-\Omega)})\,d\Omega$$

where S_A and S_B are the power spectral densities for the two independent processes. [30%]

Solution:

First find autocorrelation function:

$$\begin{aligned} E[X_n X_{n+l}] &= E[A_n B_n A_{n+l} B_{n+l}] \\ &= E[A_n A_{n+l}]E[B_n B_{n+l}] \text{ [since } A \text{ and } B \text{ are independent]} \\ &= r_{AA}[l]\, r_{BB}[l] \end{aligned}$$

Now, take DTFT of this to get power spectrum (from first principles, but could quote

(cont.

the result):

$$
\begin{aligned}
S_Y(e^{j\theta}) &= \sum_{l=-\infty}^{+\infty} r_{AA}[l] r_{BB}[l] e^{-jl\theta} \\
&= \sum_{l=-\infty}^{+\infty} \frac{1}{2\pi} \int_{-\pi}^{+\pi} S_A(e^{j\Omega}) e^{+jl\Omega} d\Omega r_{BB}[l] e^{-jl\theta} \text{ (subst. with inverse DTFT for } r_AA) \\
&= \frac{1}{2\pi} \int_{-\pi}^{+\pi} S_A(e^{j\Omega}) \sum_{l=-\infty}^{+\infty} r_{BB}[l] e^{-jl(\theta-\Omega)} d\Omega \\
&= \frac{1}{2\pi} \int_{-\pi}^{+\pi} S_A(e^{j\Omega}) S_B(e^{j(\theta-\Omega)}) d\Omega
\end{aligned}
$$

as required.

(c) A music synthesiser generates a sound by random amplitude modulation of a tone as follows:

$$X_n = (1 + A_n)\cos(n\omega_0 T + \phi)$$

where A_n is a zero mean, wide-sense stationary random process and ω_0 is a constant frequency.

(i) $\phi = 0$

Solution:

$$E[X_n] = E[(1 + A_n)]\cos(n\omega_0 T + \phi) = \cos(n\omega_0 T + \phi)$$

which depends on n, hence not Wide-sense stationary.

(ii) ϕ is a random variable, independent of $\{A_n\}$ and uniformly distributed between $-\pi$ and $+\pi$

[30%]

Solution:

$$E[X_n] = E[(1+A_n)]E[\cos(n\omega_0 T+\phi)] = E[\cos(n\omega_0 T+\phi)] \text{ (since both terms are independent)}$$

Now:

$$E[\cos(n\omega_0 T + \phi)] = E[\cos(n\omega_0 T)\cos(\phi)] - E\sin((n\omega_0)T)\sin(\phi)] = 0$$

$$\text{(since } E[\sin(\phi)] = E[\cos(\phi)] = 0\text{)}$$

(TURN OVER for continuation of Question 1

Hence mean is constant.

Test autocorrelation function:

$$\begin{aligned} E[X_n X_{n+1}] &= E[(1+A_n)\cos(n\omega_0 T+\phi)(1+A_{n+l})\cos((n+l)\omega_0 T+\phi)] \\ &= E[(1+A_n)(1+A_{n+l})]E[\cos(n\omega_0 T+\phi)\cos((n+l)\omega_0 T+\phi)] \\ &= 0.5(1+r_{AA}[l])(E[\cos((2n+l)\omega_0 T+2\phi)]+E[\cos(l\omega_0 T)] \\ &= 0.5(1+r_{AA}[l])\cos(l\omega_0 T) \end{aligned}$$

which does not depend on n. Also $r_{XX}[l] < \infty$ as long as $r_{AA}[0] < \infty$ which it is, as it is WSS itself, so the whole process is WSS.

(d) Describe and sketch the power spectrum of the process $\{X_n\}$ with phase as in part (b)(ii) when

(i) $A_n = \cos(n\omega T/10+\theta)$, and θ is a random variable, independent of ϕ, and uniformly distributed between $-\pi$ and $+\pi$.

(ii) $\{A_n\}$ is a second order autoregressive process having poles at $0.9\exp(\pm i\omega_0 T/10)$.

[20%]

Solution:

The power spectrum is obtained from the result in part a). In this case we must convolve the power spectrum of the random-phase cosine (which is a train of delta functions centered on $\pm\omega_0 \pm n2\pi/T$), with the power spectrum of $1+A_n$, which is a delta function at $\omega = 0$ plus the power spectrum of $\{A_n\}$. Hence we get the following:

(i) Here $r_{AA}[l] = 0.5\cos(l\omega_0 T/10)$ and hence the power spectrum of $1+A_n$ is

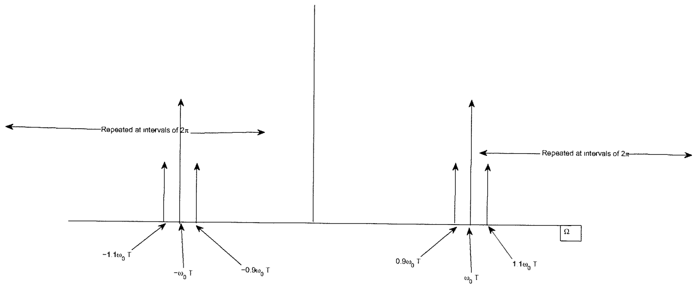
$$\delta(\Omega)+0.5\delta(\Omega\pm\omega_0 T/10)$$

This, once convolved with the power spectrum of the random phase cosine, leads to a power spectrum:

(ii) Here the power spectrum of A is the power spectrum of the AR process, having a peak close to the pole frequency. Otherwise unchanged, leading to:

[You may use the result that the DTFT of $\cos(\omega_0 nT)$ is $0.5\pi\delta(\omega-\omega_0)$ for values of ω between 0 and π/T.]

(cont.



2 (a) The Discrete Fourier Transform (DFT) is typically implemented using a Fast Fourier Transform (FFT) algorithm. Assuming that the number of data points N is even, split the summation in the basic DFT equation into two parts: one for even n and one for odd n, and then show that the DFT values X_p and $X_{p+N/2}$ may be expressed as

$$X_p = A_p + W^p B_p \quad X_{p+N/2} = A_p - W^p B_p$$

Derive A_p, B_p and W in the above expression, and thus define the FFT "butterfly" structure. [30%]

Answer. Let us take the basic DFT equation:

$$X_p = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}np},\ p = 0, \ldots, N-1$$

and split the summation into two parts: one for even n and one for odd n

$$X_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)p}$$

$$X_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}np} + e^{-j\frac{2\pi}{N}p} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}np}$$

$$X_p = A_p + W^p B_p$$

$$A_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}np}$$

$$B_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}np}$$

(TURN OVER for continuation of Question 2

$$W = e^{-j\frac{2\pi}{N}p}$$

Now take the same equation and split the summation into two parts again: one for even n and one for odd n

$$X_p = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)p} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)p}$$

and evaluate at frequencies $p+N/2$ one for even n and one for odd n

$$X_{p+N/2} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N}(2n)(p+N/2)} + \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N}(2n+1)(p+N/2)}$$

$$X_{p+N/2} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}(n)(p+N/2)} + e^{-j\frac{2\pi}{N}(p+N/2)} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}(n)(p+N/2)}$$

Simplify terms as follows

$$e^{-j\frac{2\pi}{N/2}n(p+N/2)} = e^{-j\frac{2\pi}{N/2}np} e^{-j\frac{2\pi N/2}{N/2}n} = e^{-j\frac{2\pi}{N/2}np} e^{-j2\pi n} = e^{-j\frac{2\pi}{N/2}np}$$

$$e^{-j\frac{2\pi}{N}(p+N/2)} = e^{-j\frac{2\pi}{N}p} e^{-j\frac{2\pi N/2}{N}} = e^{-j\frac{2\pi}{N}p} e^{-j\pi} = -e^{-j\frac{2\pi}{N}p}$$

Hence,

$$X_{p+N/2} = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} e^{-j\frac{2\pi}{N/2}np} - e^{-j\frac{2\pi}{N}p} \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} e^{-j\frac{2\pi}{N/2}np}$$

$$X_{p+N/2} = A_p - W^p B_p$$

This defines the FFT butterfly structure.

(b) If N is a power of 2 the process above can be repeated several times resulting in a radix-2 FFT algorithm. Determine the number of such stages required, the number of "butterfly" computations per stage and the total number of complex multiplications if $N = 64$. [20%]

Answer If $\frac{N}{2}$ is even the same process can be carried out on each of the $\frac{N}{2}$-point DFTs to further reduce computations. It can be seen that if $N = 2^M$ this process can be repeated M times resulting in M stages of the algorithm, each stage using $N/2$ "butterflies". Each butterfly is defined by

$$X_p = A_p + W^p B_p$$

(cont.

$$X_{p+N/2} = A_p - W^p B_p$$

where W^p are pre-computed and stored. Thus, one "butterfly" requires 1 complex multiplication and the radix-2 N-point FFT requires $(N/2) log_2 N$ complex multiplications which for $N = 64$ results in 192 complex multiplications.

(c) A wide sense stationary discrete time random process $\{X_n\}$ has autocorrelation function

$$r_{XX}[0] = 1, , r_{XX}[\pm 1] = 0.5, 0, \text{otherwise}.$$

The process is filtered through a first order noisy channel to give:

$$y_n = x_n - 0.2x_{n-1} + v_n$$

where v_n is white noise having variance 0.4.

(i) Determine the autocorrelation function of the noisy filtered process $\{Y_n\}$ and the cross-correlation function between $\{X_n\}$ and $\{Y_n\}$. [20%]

.

Solution:

$$\begin{aligned} r_{XY}[m] &= E[X_n Y_{n+m}] \\ &= E[X_n(X_{n+m} - 0.2X_{n+m-1} + V_{n+m})] \\ &= r_{XX}[m] - 0.2r_{XX}[m-1] \text{ since } X \text{ is uncorrelated with } V \end{aligned}$$

Hence, from given values of r_{XX} we have

$$r_{XY}[m] = \begin{cases} 0.5, & m = -1 \\ 0.9, & m = 0 \\ 0.3, & m = 1 \\ -0.1, & m = 2 \\ 0, & \text{otherwise} \end{cases}$$

Now, for r_{YY}:

$$\begin{aligned} r_{YY}[m] = E[y_n y_{n+m}] &= E[(x_n - 0.2x_{n-1} + v_n)y_{n+m}] \\ &= r_{XY}[m] - 0.2r_{XY}[m+1] + r_{VV}[m] \text{ since } V \text{ is uncorrelated with } X \end{aligned}$$

(TURN OVER for continuation of Question 2

Hence from values of R_{XY} and $r_{VV}[m] = 0.4\delta[m]$:

$$r_{YY}[m] = \begin{cases} -0.1, & m = \pm 2 \\ 0.32, & m = \pm 1 \\ 1.24, & m = 0 \\ 0, & \text{otherwise} \end{cases}$$

(ii) Determine the coefficients of a second order Wiener filter for extraction of $\{X_n\}$ from measurements $\{Y_n\}$. Compare its expected mean-squared error with the expected mean-squared error between $\{Y_n\}$ and $\{X_n\}$. [30%]

Solution:

Set up the Wiener-Hopf equations as in lecture notes with $P = 2$:

$$\mathbf{R}_Y \mathbf{h} = \mathbf{r}_{YD}$$

where:

$$\mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \end{bmatrix} \qquad \mathbf{r}_{YX} = \begin{bmatrix} r_{YX}[0] \\ r_{YX}[1] \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.5 \end{bmatrix}$$

[since $r_{XY}[m] = r_{YX}[-m]$] and and

$$\mathbf{R}_Y = \begin{bmatrix} r_{YY}[0] & r_{YY}[1] \\ r_{YY}[1] & r_{YY}[0] \end{bmatrix} = \begin{bmatrix} 1.24 & 0.32 \\ 0.32 & 1.24 \end{bmatrix}$$

The coefficient vector can be found by matrix inversion:

$$\mathbf{h} = \mathbf{R}_Y{}^{-1} \mathbf{r}_{YX} = [0.67 \ 0.23]^T \qquad (2)$$

For the optimal solution, lecture notes give the Expected squared error as:

$$J_{\text{min}} = E[\varepsilon_n x_n]$$

$$= E[(d_n - \sum_{p=0}^{P-1} h_p y_{n-p}) x_n]$$

$$= r_{XX}[0] - \sum_{p=0}^{P-1} h_p r_{YX}[p]$$

$$= r_{XX}[0] - \mathbf{r}_{YXd}^T \mathbf{h} = r_{XX}[0] - \mathbf{r}_{YX}^T \mathbf{R}_Y{}^{-1} \mathbf{r}_{YX} = 1 - 0.67 \times 0.9 - 0.23 \times 0.5 = 0.28$$

(cont.

Compare this with the expected squared error between Y and X:

$$\begin{aligned} E[(Y_n - X_n)^2] &= E[(X_n - 0.2X_{n-1} + v_n - X_n)^2] \\ &= 0.4E[X_{n-1}]^2 + E[v_n^2] \\ &= 0.4 \times 1 + 0.4 \\ &= 0.8 \end{aligned}$$

i.e. we get a significant improvement from using the Wiener filter as compared with just taking the measured Y.

3 (a) Describe the direct form I structure for implementation of infinite impulse response (IIR) filters. Explain why implementing a digital filter in direct form I is satisfactory in Matlab where double precision floating-point is used but is not usually a good idea in fixed point implementation. What are other alternative realization structures of IIR filters and their potential advantages? Describe the direct form II structure and explain why it may be preferable to the direct form I implementation. [30%]

Answer: The transfer function of the IIR filter is

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

The direct form I implementation is straightforward and is illustrated in Figure 1.

In fixed point or VLSI implementation, direct form is not usually a good idea because one has often severe speed and power consumption constraints.

If **speed** is the main concern, then if multiplications take longer than additions, we aim to reduce the number of multiplications; otherwise to reduce the total operation count. The area of a fixed-point parallel multiplier is proportional to the product of the coefficient and data wordlengths, making wordlength reduction advantageous.

Hence much work has gone into structures which allow reductions in

- the number of multipliers; or
- the total operation count (multipliers, adders and perhaps delays); or
- data or coefficient wordlengths

(TURN OVER for continuation of Question 3

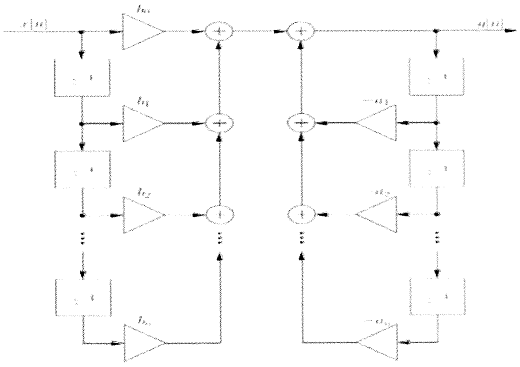


Fig. 1

If **power consumption** is the concern, then reducing total operation count and wordlength are desirable. Since general multiplication takes much more power than addition, we try to reduce the number of multiplications, or to replace general multiplications by, for example, binary shifts.

The alternative structures for IIR filters implementation include parallel, cascade and feedback implementations. In general, in fixed-point implementation alternative structures may offer the following advantages: decreased number of multiplications or overall computational load, reduced sensitivity of the response to coefficients imprecision (coefficient quantisation), reduced quantisation noise among others.

The direct form II structure is obtain if one looks at a transfer function as a cascade of $H_1(z)$ and $H_2(z)$ where ($N = M$ for convenience in this example)

$$H_1(z) = \sum_{k=0}^{N} b_k z^{-k}, \; H_2(z) = \frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}.$$

$H_1(z)$ can be realized with a parallel structure and $H_2(z)$ with a feedback structure (where the $\sum_{k=1}^{N} a_k z^{-k}$ part in $H_2(z)$ can be realized with another parallel structure). Putting all together, we obtain the block diagram displayed in Figure 2. Direct form II is preferable to Direct form I as it requires a smaller number of memory locations.

(b) The Goertzel algorithm is often used for detection and measurement of single sinusoidal tones since it computes a single Discrete Fourier Transform (DFT) component. The Goertzel algorithm is implemented as a second-order IIR filter with two real feedback

(cont.

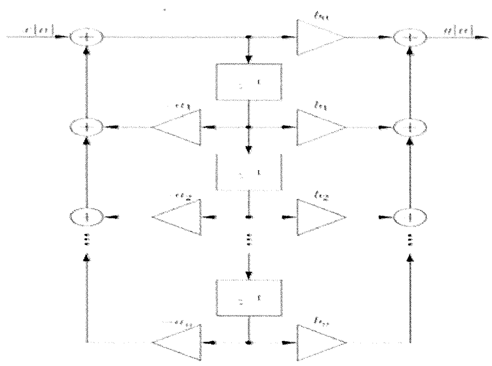


Fig. 2

coefficients and a single complex feedforward coefficient. The transfer function of the Goertzel filter is

$$H(z) = \frac{1 + b_1 z^{-1}}{1 + a_1 z^{-1} + z^{-2}}$$

where the filter coefficients for the mth bin of an N-point DFT are

$$b_1 = -e^{-j\frac{2\pi}{N}m} \quad a_1 = -2\cos(2\pi m/N)$$

The filter is to be implemented in direct form II. Determine the coefficients of the filter for $m = 15$ and $N = 64$ and sketch the implementation. [30%]

Answer: The coefficients can be calculated as follows

$$b_1 = -e^{-j\frac{2\pi}{N}m} = -e^{-j\frac{2\pi}{64}15} = -e^{-j1.473} = -0.098 + j0.995$$

$$a_1 = -2\cos(2\pi m/N)\text{=-2cos}(2\pi 15/64)\text{=-0.196}$$

Direct form II implementation of the filter is illustrated in Figure 3.

(c) Show that the Goertzel filter in part (b) is equivalent to a first order complex all-pole filter having a single pole at $z = e^{-j\frac{2\pi}{N}m}$. Hence show that, assuming the input signal is zero prior to $n = 0$, the Goertzel algorithm delivers the expected DFT coefficient (up to a simple complex scale factor) after N data points have been passed through the filter. [40%]

(TURN OVER for continuation of Question 3

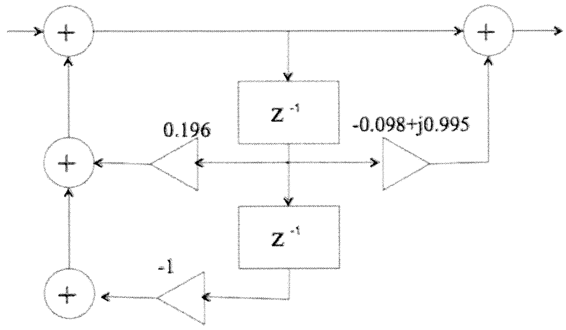


Fig. 3

[Hint for part (c): consider the poles and zeros of the filter, which will lead to a simplification of the transfer function.]

Solution:

There is a pole that cancels with the zero of the filter leading to:

$$H(z) = \frac{1}{1 - e^{+j\frac{2\pi}{N}m} z^{-1}}$$

Consider this simplified first-order all-pole version of the Goertzel filter with input data x_n:

$$y_n = x_n + e^{+j\frac{2\pi}{N}m} y_{n-1}$$

Now, with $y_n = x_n = 0$ for $n < 0$:

$$\begin{aligned}
y_0 &= x_0 \\
y_1 &= x_1 + e^{+j\frac{2\pi}{N}m} x_0 \\
y_2 &= x_2 + e^{+j\frac{2\pi}{N}m}(x_1 + e^{+j\frac{2\pi}{N}m} x_0) \\
&\dots \\
y_{N-1} &= \sum_{n=0}^{N-1} x_n e^{+j\frac{2\pi}{N}m(N-1-n)} = e^{+j\frac{2\pi}{N}m(N-1)} \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}mn} \\
&= e^{+j\frac{2\pi}{N}m(N-1)} DFT(y_{0:N-1})
\end{aligned}$$

by inspection (could be proved by induction, but not necessary). Hence shown.

4 Consider a binary classification problem with scalar real-valued observations x, and class labels $y \in \{0, 1\}$. Assume that $p(x|y=0)$ is a Gaussian distribution with mean 0 and variance 2, and $p(x|y=1)$ is a Gaussian distribution with mean 1 and variance 2. Furthermore, assume that $p(y=0) = p(y=1) = 1/2$.

(a) Compute the probability that given an observation $x=2$, its corresponding class label is $y=1$. [30%]

(b) Derive the general expression for $p(y=0|x)$ as a function of x, and discuss how this relates to logistic classification. [40%]

Now assume that you fit a maximum likelihood Gaussian distribution $p(x|y=0)$ with mean μ_0 and variance σ_0^2 to the observed data with label $y=0$, and similarly you fit a separate maximum likelihood Gaussian distribution $p(x|y=1)$ with mean μ_1 and variance σ_1^2 to the observed data with label $y=1$.

(c) Describe several ways in which the above procedure differs from maximum likelihood logistic classification, paying particular attention to the role of the variances and likelihood that is being optimised. [30%]

ANSWERS

(a) By Bayes rule:

$$
\begin{aligned}
p(y=1|x=2) &= \frac{p(x=2|y=1)p(y=1)}{p(x=2|y=1)p(y=1)+p(x=2|y=0)p(y=0)} \\
&= \frac{\frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}\cdot\frac{1}{2}}{\frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}\cdot\frac{1}{2}+\frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-0)^2}{2^2}\}\cdot\frac{1}{2}} \\
&= \frac{\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}}{\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}+\exp\{-\frac{1}{2}\frac{(2-0)^2}{2^2}\}} \\
&= \frac{\exp\{-\frac{1}{8}\}}{\exp\{-\frac{1}{8}\}+\exp\{-\frac{1}{2}\}} \\
&= \frac{1}{1+e^{-3/8}}
\end{aligned}
$$

(TURN OVER for continuation of Question 4

(b) Again by Bayes rule:

$$
\begin{aligned}
p(y=0|x) &= \frac{\exp\{-\frac{1}{2}\frac{x^2}{2^2}\}}{\exp\{-\frac{1}{2}\frac{x^2}{2^2}\}+\exp\{-\frac{1}{2}\frac{(x-1)^2}{2^2}\}} \\
&= \frac{1}{1+\exp\{\frac{1}{8}x^2-\frac{1}{8}x^2+\frac{x}{4}-\frac{1}{8}\}} \\
&= \frac{1}{1+\exp\{\frac{x}{4}-\frac{1}{8}\}}
\end{aligned}
$$

This is a logistic function, exactly as in logistic classification. So the classification probabilities are equivalent to that of logistic classication.

(cont.

(c) There are a number of ways in which these two procedures differ:

(i) they optimise different likelihoods. Logistic classification maximises $\prod_n p(y_n|x_n)$ whereas the above procedure maximises

$$\prod_{n:y_n=1} p(x_n|y_n = 1) \prod_{n:y_n=0} p(x_n|y_n = 0)$$

(ii) If the variances are not equal for the two classes then the boundary has two locations, not one, and the probabilities are no longer given by the logistic function.

(iii) If the data are perfectly separable, then the logistic function for logistic classification tends to a step function, whereas this is not the case for the above procedure.

END OF PAPER