2011 Part IIA 3F3 Signal and pattern processing Prof S Godsil

1 Examiner's comment:

A fairly popular question, but not well answered in general. A significant part was taken from the lecture notes, but clearly not learned by many candidates.

(a) The DFT of a sequence $\{x_n\}$ is $\{X_p\}$, and that of a second sequence $\{y_n\}$ is $\{Y_p\}$, for $n = 0, 1, ..., N-1$ and $p = 0, 1, ..., N-1$. We obtain a third DFT, $Z_p = X_p Y_p$, by multiplying X_p and Y_p. Show that the inverse DFT of $\{Z_p\}$ can be expressed in the following form:

$$z_n = \sum_{m=0}^{N-1} y_m x_{\text{mod}(n-m,N)}$$

where $\text{mod}(P,N)$ denotes the number P represented in modulo N arithmetic, e.g. $\text{mod}(3,10) = \text{mod}(13,10) = 3$, etc. [40%]

Answer: This was covered in the lecture course.

To see why this is so [note this is worked through with Y=HX instead of Z=YX]

$$Y_m = H_m X_m, \text{ where } H_m = \sum_{n=0}^{N-1} h_n e^{\frac{-j2\pi nm}{N}}, \quad X_m = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi nm}{N}}$$

Thus,

$$Y_m = \sum_{n_1=0}^{N-1} h_{n_1} e^{\frac{-j2\pi n_1 m}{N}} \sum_{n_2=0}^{N-1} x_{n_2} e^{\frac{-j2\pi n_2 m}{N}}$$

Taking inverse DFTs:

$$\begin{aligned}
y_p &= \frac{1}{N} \sum_{m=0}^{N-1} \left\{ \sum_{n_1=0}^{N-1} h_{n_1} e^{\frac{-j2\pi n_1 m}{N}} \sum_{n_2=0}^{N-1} x_{n_2} e^{\frac{-j2\pi n_2 m}{N}} \right\} e^{+\frac{jmp2\pi}{N}} \\
&= \frac{1}{N} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} h_{n_1} x_{n_2} \sum_{m=0}^{N-1} e^{\frac{-j2\pi(n_1+n_2-p)m}{N}} \\
&= \frac{1}{N} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} h_{n_1} x_{n_2} \times \begin{cases} N, & \text{mod}(n_1+n_2-p,N) = 0 \\ 0, & \text{otherwise} \end{cases} \\
&= \frac{N}{N} \sum_{n_1=0}^{N-1} h_{n_1} x_{\text{mod}(p-n_1,N)} = \sum_{n=0}^{N-1} h_n x_{\text{mod}(p-n,N)}
\end{aligned}$$

(b) Now, assume that the sequence $\{y_n\}$ contains non-zero values only in its first M elements, i.e.

$$y_n = y_0, y_1, \cdots, y_{M-1}, 0, 0, \ldots, 0.$$

Explain why, in this case, the inverse DFT of $\{Z_p\}$ is equal to the standard discrete time convolution of $\{x_n\}$ with $\{y_n\}$, but that this only applies for time indexes $n = M, \ldots, N-1$, i.e. that:

$$z_n = \sum_{m=0}^{M-1} y_m x_{n-m}, \quad n = M, \ldots, N-1.$$

[20%]

Answer:

Consider filtering a sequence x with a filter h having order M [once again, we are using Y, H and X instead of Z, Y and X]. The required convolution is:

$y_n = \sum_{m=0}^{M} h_m x_{n-m}$

Now, for a data length N>M. We notice that for M-1<n<N,

mod(n-m,N)=n-m

In other words, the result of cyclic convolution is the same as that of standard convolution:

$y_n = \sum_{m=0}^{M} h_m x_{n-m} = \sum_{m=0}^{M} h_m x_{\mathrm{mod}(n-m,N)}, \quad M-1 < n < N$

as required.

(c) (i) Describe how the results in (a) and (b) above can be used to implement FIR filtering with $M = 100$ filter taps on a long sequence of data whose length is $N - M$, where $N = 2^Q > M$, and Q is an integer. Assume that all data prior to time $n = 0$ are zero-valued. [15%]

Answer:

This is essentially the overlap-save method that filters a long sequence of data x in chunks of length N-M, as follows:

Say, h_n is the impulse response of the FIR filter, and is of length M. Choose a much longer blocklength N, append N-(M) zeros to make the vector y and compute its DFT Y via the DFT. Note that Y only needs to be calculated once. Then compute the FFT of the data, with M zeros appended at the start. This is X. Then, multiply Z=YX elementwise, take inverse DFT. delete first M elements and these are the correctly fiiltered version of $\{x_n\}$, as shown by the result of part b).

(TURN OVER for continuation of Question 1

(ii) Determine the computational load of such a scheme, in terms of real additions and multiplications, in the case where the DFTs are implemented using an appropriate fast algorithm. [10%]

Answer: Direct method using full DFT (this not required in the solution): DFT of zero-padded x takes N^2 complex multiplies = $4N^2$ real multiplies and N^2 complex additions = $2N^2$ real additions. Elementwise product YX takes N complex multiplies = $4N$ real multiplies. Inverse DFT takes the same time as forward DFT.

Total is $8N^2 + 4N$ multiplies and $8N^2$ additions.

An appropriate fast algorithm is the FFT. This will take $4(N/2)\log_2 N$ real multiplications plus $6(N/2)\log_2 N$ real additions, instead of the $4N^2$ multiplies and $2N^2$ additions of the direct DFT. Thus,

Total is $4N\log_2 N + 4N$ multiplies plus $6N\log_2 N$ additions.

(iii) Determine the length of data N below which the fast DFT-based algorithm will achieve a reduction in total operation count (real multiplications plus additions) compared with a direct time-doman implementation of the $M = 100$ filter. [15%]

You may assume that the DFT of the filter's impulse response has been pre-computed and that N is much greater than M.

Answer:

The direct time-domain implementation requires $(N-M)M$ real multiplies and $(N-M)(M-1)$ additions, a total of approximately $2NM$ real operations for large N.

Comparing (ii) with the time-domain implementation, we see that (for large $N >> M$ so that we can neglect the $4N$ terms and $N - M \approx N$) the total number of operations becomes equal once:

$$10N\log_2 N = 2NM$$

i.e. $5Q = M$ and $Q = 20$ - hence at data lengths of $N < 2^{20}$ we gain an improvement compared with the direct time-domain implementation.

Examiner's comment: very few candidates worked through to a reasonable answer in this last part.

2 Examiner's comment:

A very popular question and well answered by most.

(a) Explain the terms IIR and FIR in digital filtering. How is the filter's impulse response calculated from the filter coefficients in each case? Describe, with diagrams, how to implement an IIR filter using Direct Form I and Direct Form II structures, indicating any advantages or disadvantages of the two approaches. [30%]

Answer: This is a very standard piece of bookwork from the lecture notes. The folliowing answer is much more extensive than would be required under exam conditions.

The general form for the digital filter is

$$\begin{aligned} y_n &= -a_1 y_{n-1} - a_2 y_{n-2} - \ldots - a_N y_{n-N} + b_0 x_n + \ldots + b_M x_{n-M} \\ &= -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k} \end{aligned}$$

where the coefficients $\{a_k\}$ and $\{b_k\}$ are real.

The larger of M or N is known as the *order* of the filter.

By applying the z-transform, one can obtain the transfer function of the system:

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \\ &= b_0 \frac{\prod_{k=1}^{M} \left(1 - z_k z^{-1}\right)}{\prod_{k=1}^{N} \left(1 - p_k z^{-1}\right)} \end{aligned}$$

thus

$$\begin{aligned} H(\omega) &= \frac{\sum_{k=0}^{M} b_k e^{-jk\omega}}{1 + \sum_{k=1}^{N} a_k e^{-jk\omega}} \\ &= b_0 \frac{\prod_{k=1}^{M} \left(1 - z_k e^{-j\omega}\right)}{\prod_{k=1}^{N} \left(1 - p_k e^{-j\omega}\right)} \end{aligned}$$

where the $\{z_k\}$ and $\{p_k\}$ may be complex-valued. The complex-valued poles and zeros occur in complex-conjugate pairs.

IIR - infinite impulse response - the filter's digital impulse response never decays to zero. These filters have both poles and zeros and hence both a and b coefficients in the above. To get the impulse response, take inverse z-transforms of the filter's z-domain transfer function.

(TURN OVER for continuation of Question 2

FIR - finite impulse response - the filter's digital impulse response has only a finite number of non-zero terms, and hence only b coefficients ($N = 0$ above). These filters have only zeros (except for 'trivial' poles at the origin). The impulse response in this case equals the filter coefficients.

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.

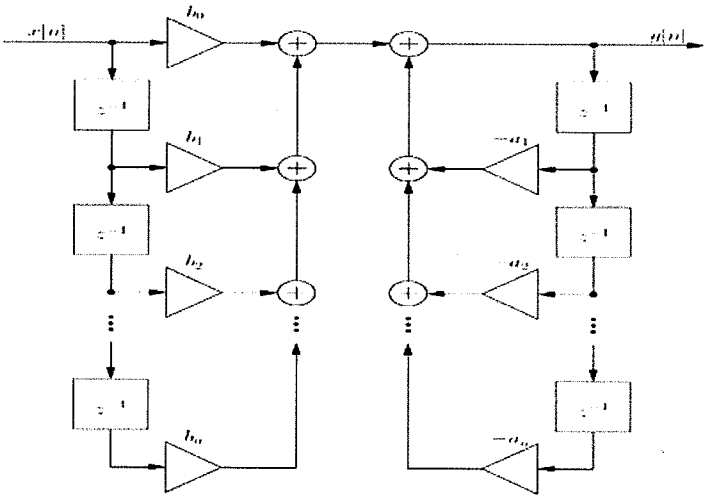


Fig. 1

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

(cont.

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.

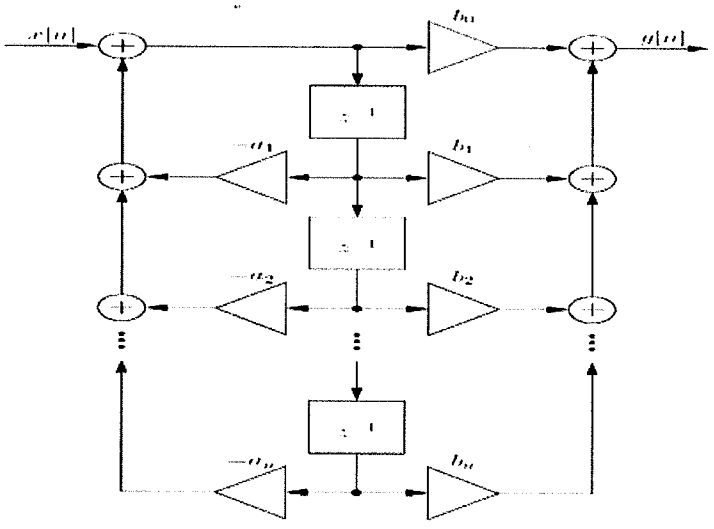


Fig. 2

(b) It is proposed to convert an analogue prototype low-pass filter using the

(TURN OVER for continuation of Question 2

substitution formula

$$s \to \frac{s^2 + \omega_1\omega_2}{s(\omega_2 - \omega_1)},$$

where s is the standard Laplace variable. ω_2 and ω_1 are constants satisfying $\omega_2 > \omega_1$.

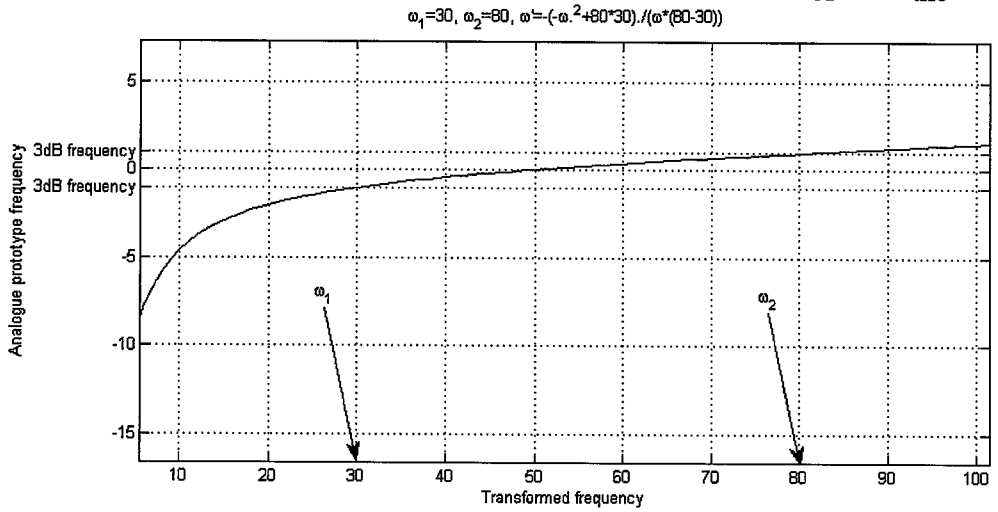
Describe the effect of this transformation on the filter's frequency response, explaining with the aid of sketches. Consider in particular the frequencies 0, ω_1, ω_2 and ∞ rad.s^{-1} in the transformed filter. [20%]

Answer: this is a low-pass to bandpass transformation with lower and upper cut-offs at ω_1 and ω_2, respectively. Taking $s = j\omega$ we get the equivalent frequency substitution:

$$\omega \to \frac{\omega^2 - \omega_1\omega_2}{\omega(\omega_2 - \omega_1)}$$

In particular, taking $\omega = 0$, we find that this corresponds to $\omega' = \infty$ in the original prototype filter, and so does $\omega = \infty$. Hence frequencies 0 and ∞ are heavily attentuated in the transformed filter. ω_1 corresponds to $\omega' = -1$ in the original filter and and ω_2 to $\omega' = +1$, which means that these two values define corner (typically 3dB) frequencies in the new filter. Finally, the frequency $\omega' = 0$ corresponds to $\omega = \sqrt{\omega_1\omega_2}$ (the 'geometric mean') in the bandpass design, a frequency that lies in the passband between ω_1 and ω_2, hence the bandpass region is between ω_1 and ω_2.

A sketch of this would be of the form:



(c) A low-pass analogue filter with -3dB frequency of 1 rad.s^{-1} has the following transfer function:

$$H(s) = \frac{1}{s+1}.$$

(cont.

In a digital audio system it is desired to reduce the effects of bass 'boom' up to 200Hz and percussion noise above 8kHz. The sampling frequency of the system is 44.1 kHz.

Starting with the analogue prototype filter above, design a bandpass digital IIR filter according to these criteria, assuming that -3dB attenuation will be adequate at the edges of the filter's pass-band. [30%]

Answer:

First warp the cut-off frequencies from normalised digital 'Ω' to analogue 'ω':

$$\Omega_1 = 200 \times 2\pi/44100 \rightarrow \omega_1 = \tan(\Omega_1/2) = 0.0142 rad.s^{-1}$$

$$\Omega_2 = 8000 \times 2\pi/44100 \rightarrow \omega_2 = \tan(\Omega_2/2) = 0.6408 rad.s^{-1}$$

Then, using these warped values, perform the low-pass to bandpass transformation from part b):

$$\begin{aligned} H(s) &\rightarrow \frac{1}{\frac{s^2+\omega_1\omega_2}{s(\omega_2-\omega_1)}+1} \\ &= \frac{1}{\frac{s^2+\omega_1\omega_2}{s(\omega_2-\omega_1)}+1} \\ &= \frac{s(\omega_2-\omega_1)}{s^2+\omega_1\omega_2+s(\omega_2-\omega_1)} \end{aligned}$$

Now, apply biliinear transform,

$$s = \psi(z) = \frac{1-z^{-1}}{1+z^{-1}};$$

$$\begin{aligned} H(z) &= H(s)|_{s=\frac{1-z^{-1}}{1+z^{-1}}} \\ &= \frac{(\omega_2-\omega_1)(1-z^{-2})}{1+\omega_2-\omega_1+\omega_2\omega_1+(2\omega_1\omega_2-2)z^{-1}+(1+\omega_1\omega_2-(\omega_2-\omega_1)z^{-2})} \\ &= \frac{0.383(1-z^{-2})}{1-1.211z^{-1}+0.234z^{-2}} \end{aligned}$$

where we have maintained the unity in the numerator and denominator for implementation purposes.

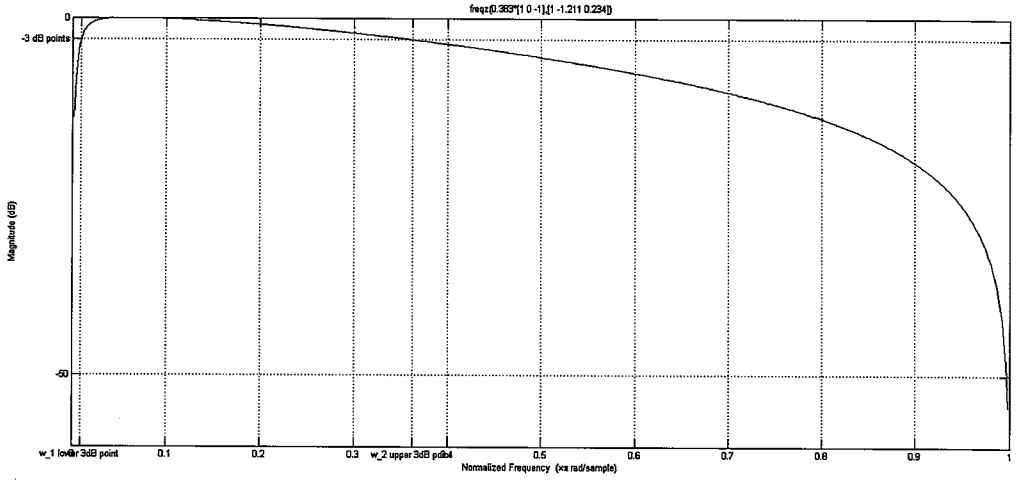
(TURN OVER for continuation of Question 2

Determine the poles and zeros of the resulting IIR filter and hence explain the shape of the frequency magnitude response in the digital domain. [20%]

The zeros are at ± 1 by inspection. The poles are at

$$z = 0.970, 0.2143$$

Hence not resonant poles (compared say with the examples sheet case). However, it does achieve the required frequency response:



The zeros at ± 1 cause the deep troughs at normalised frequencies of 0 and π. The poles serve to adjust the magnitude response so it is passband in the correct region - note that both poles are on the positive real axis, hence they bias the passband to be in the frequency range 0 to π.

3 Examiner's comment:

This question was very well answered by many - there was good fluency in manipulation of expectations and calculation of optimal filters.

(a) In a data measurement system, some of the data points x_n are found to be heavily corrupted with noise. It is decided to 'clean' the data by performing an interpolation of an anomolous data point at time index n_0 according to the formula

$$\hat{x}_{n_0} = ax_{n_0-1} + bx_{n_0+1}$$

where x_{n_0-1} and x_{n_0+1} are the data points immediately before and after the corrupted data point at n_0, and there are two constants a and b to be determined.

By considering the above equation to be a digital filter which runs over sussessive values of n_0 in the signal, giving its output at time $n_0 + 1$, i.e. with a unit time delay in order to make the system causal, determine the frequency response of the interpolation function. In the case $a = b = 0.5$, sketch the frequency response of the filter, paying attention to the DC gain and any 3dB gain points. Is the filter linear phase or not? [30%]

Answer:

Impulse response is ... b, 0, a... Hence frequency response is the DTFT of this:

$$H(e^{j\Omega}) = b\exp(j\Omega) + a\exp(-j\Omega)$$

With $a = b = 0.5$,

$$H(e^{j\Omega}) = \cos(\Omega)$$

DC gain is unity, 3dB points at $\Omega = \pi/4, 3\pi/4$. The filter is linear phase since it has an all-real frequency response (hence constant zero delay).

(b) Assume now that $\{x_n\}$ is a wide-sense stationary random process with autocorrelation function $r_{XX}[n]$. Determine the optimal coefficients a and b that will minimize the mean-squared error in the interpolator, i.e. to minimize

$$\varepsilon_{n_0} = E[(x_{n_0} - \hat{x}_{n_0})^2]$$

[30%]

Answer:

(TURN OVER for continuation of Question 3

$$\begin{aligned}\varepsilon_{n_0} &= E[(x_{n_0} - \hat{x}_{n_0})^2] \\ &= E[(x_{n_0} - (ax_{n_0-1} + bx_{n_0+1}))^2] \\ &= r[0](1 + a^2 + b^2) + r[1](-2a + -2b) + 2abr[2]\end{aligned}$$

To minimize this (dropping the '$_{XX}$' for convenience): :

$$\frac{\partial}{\partial a}\varepsilon = 2ar[0] - 2r[1] + 2br[2] = 0$$

$$\frac{\partial}{\partial b}\varepsilon = 2br[0] - 2r[1] + 2ar[2] = 0$$

Therefore,

$$r[2]/r[0](-2r[1] + 2br[2]) - 2br[0] + 2r[1] = 0,\ b(-2r[0] + 2r[2]^2/r[0]) = 2r[1](r[2]/r[0] - 1),$$

and

$$b = \frac{r[1](r[2] - r[0])}{(r[2]^2 - r[0]^2)} = \frac{r[1]}{r[2] + r[0]}$$

and, by similar reasoning (or by symmetry):

$$a = \frac{r[1]}{r[2] + r[0]}$$

(c) If $r_{XX}[n] = (-0.9)^{|n|}$, determine the mean-squared error corresponding to the optimal interpolator just derived. [20%]

Answer:

With this $r[n]$,

$$a = -0.9/(1 + 0.81) = -0.4972 = b$$

Now evaluate the mean-squared error:

$$\begin{aligned}&r[0](1 + a^2 + b^2) + r[1](-2a + -2b) + 2abr[2] \\ &= 1 + 0.4972^2 + 0.4972^2 + (-0.9) * 4 * 0.4947 + 2 * 0.4947^2 * 0.81 = 0.11\end{aligned}$$

[could also use formula in lecture notes for error at optimal coefficients]

(cont.

(d) Compare the mean-squared error in part (c) with that of the simple interpolator in part (a), having $a = b = 0.5$. Comment on your result. [20%]

Answer:

With $a = b = 0.5$ we have the following mean-squared error:

$$r[0](1+a^2+b^2)+r[1](-2a+-2b)+2abr[2]$$
$$= 1+0.5^2+0.5^2+(-0.9)*-2+2*0.25*0.81 = 3.7$$

We notice that the simple interpolator gives a much higher error than the optimal filter. This is because the data are negatively correlated at lag 1 ($r_{XX}[1] = -0.9$) and hence it is inappropriate to do a simple linear interpolation between the two adjacent data points if we wish to accurately estimate the missing data point. The optimal filter on the other hand makes the best possible interpolator in the mean-squared error sense, and notice that the coefficients are almost negative those of the simple interpolator - we are essentially just flipping the sign of the simple linear interpolator to get the optimal filter, since the correlation coefficient is 'almost' -1 (actually -0.9).

4 Examiner's comment:

Very unpopular, and poorly answered. As is often the case with this part of the course, this quesion was fairly easy marks for those who had learned the material.

Consider a binary classification problem with scalar real-valued observations x, and class labels $y \in \{0, 1\}$. Assume a model with parameters θ where

$$p(y=1|x,\theta) = \frac{1}{1+e^{-\theta x+\frac{1}{2}}}$$

(a) Describe the online learning rule for learning the parameter θ assuming the learning algorithm receives one data point at a time. [40%]

(b) Consider a data set $\mathscr{D}$ consisting of three data points: $(x_1 = 0,\ y_1 = 1)$, $(x_2 = -1,\ y_2 = 0)$, and $(x_3 = 1,\ y_3 = 1)$. Compute the likelihood for the parameters θ given this data set $\mathscr{D}$. [30%]

(c) Characterise the solution(s) to the maximum likelihood estimate of θ in part (b) above. Discuss properties of these solution(s), indicating any problems with the result and possible ways of resolving those problems. [30%]

Solution:

(a) This material was basically covered in lecture 3, slide 8. The idea is to write down the likelihood function for each data point and move θ a small amount in the direction of this gradient.

$$\begin{aligned}
P(y|x,\theta) &= \sigma(\theta x-0.5)^y(1-\sigma(\theta x-0.5))^{(1-y)} \\
\ln P(y|x,\theta) &= y\ln\sigma(\theta x-0.5)+(1-y)\ln(1-\sigma(\theta x-0.5)) \\
\text{use: } \frac{\partial \ln\sigma(z)}{\partial z} &= \sigma(-z) \\
\text{let: } z &= \theta x-0.5 \\
\frac{\partial \ln P(y|x,\theta)}{\partial \theta} &= yx\sigma(-z)-(1-y)x\sigma(z) \\
&= yx-yx\sigma(z)-x\sigma(z)+yx\sigma(z) \\
&= (y-\sigma(z))x
\end{aligned}$$

The online learning rule changes the parameters θ at each step t by a small step η in the

(cont.

direction given by the maximum likelihood step above:

$$\theta_{t+1} \leftarrow \theta_t + \eta(y_t - \sigma(\theta_t x_t - 1/2))x_t$$

(b) Multiplying the probabilities of each observation we get that the likelihood is:

$$\left(\frac{1}{1+e^{1/2}}\right)\left(\frac{e^{\theta+1/2}}{1+e^{\theta+1/2}}\right)\left(\frac{1}{1+e^{-\theta+1/2}}\right)$$

or equivalently:

$$\left(\frac{1}{1+e^{1/2}}\right)\left(\frac{1}{1+e^{-\theta-1/2}}\right)\left(\frac{1}{1+e^{-\theta+1/2}}\right)$$

(c) To maximise the above likelihood we write down the log likelihood and drop additive constants getting for the first expression above:

$$L = \theta - \ln(1+e^{\theta+1/2}) - \ln(1+e^{-\theta+1/2})$$

for the second expression:

$$L = -\ln(1+e^{-\theta-1/2}) - \ln(1+e^{-\theta+1/2})$$

The second expression in the solution to (b) is easier to analyse. It's clear that the likelihood increases monotonically as θ increases. This means that the maximum likelihood occurs at the limit of $\theta \to \infty$. Looking at the configuaration of the data, this solution makes sense since the data is perfectly separable and increasing θ simply increases the slope of the logistic function.

The solution is problematic because of the basis of three data points the model will make absolutely certain predictions for new points. Two ways of resolving this problem are either to add a penalty term or prior to the log likelihood function, or to do Bayesian learning of the parameter θ (see Lecture 3:Classification, Slide 10).

END OF PAPER