University of Glasgow | School of Computing Science

DayOfWeek DayOfMonth Month 2018
XX.XX am/pm – XX.XX am/pm

(Duration: 1.5 hours)

SPECIMEN EXAMINATION FOR

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

Data Fundamentals (H)

Answer 2 of 3 Questions

This examination paper is worth a total of 50 marks

For examinations of at least 2 hours duration, no candidate shall be allowed to leave the examination room within the first hour or the last half-hour of the examination

INSTRUCTIONS TO INVIGILATORS

Please collect all exam question papers and return to School together with exam answer scripts

This a *specimen* paper. It has not been through the vetting process and errors may still be present.

ANSWERS ARE HIGHLIGHTED.

1.

(a) Given a hypothesis *H* and some observed data *D*, Explain how Bayes' rule relates prior belief, posterior belief, likelihood, and evidence, giving appropriate equations.

[3]

Writing prior as P(H), evidence as P(D), likelihood as P(D|H), posterior as P(H|D), [#1] then Bayes Rule gives us:

P(H|D) = P(D|H)P(H) / P(D) [#2]

[Unit 8: Probability and random variables]

(b) In the fabrication of semiconductors, an industrial manufacturer has a process that produces silicon wafers. Sometimes, the wafers come out defective, and have to be discarded.

(i) Every wafer has a production cost *c*. Every correct wafer can be sold for *v;* defective wafers cannot be sold and bring in no money. Assuming the probability of a defective wafer is written *P(D=1),* where D is a random variable that may take on the values 0 and 1, write down the expression for the expected **profit** of manufacturing a single wafer.

[6]

Expectation is: sum over every state (product of (probability of states, value of states))
E[f(X)] = sum P(X=x)f(x) [#2 for correct concept of expectation]

The profit of a correct wafer is v-c. The profit of an incorrect wafer is -c. [#1 for correct idea of profit and loss]

E[f(D)] = (correct probability)*(correct profit) + (defect probability)*(defect product)
E[f(D)] = (1-P(D=1))(v-c) + P(D=1)(-c)
E[f(D)] = (1-P(D=1))(v-c) - P(D=1) c [#3]

Working is not required.

[Unit 9: Sampling and inference]

(ii) The defect rate in one production line is 1:100000. The fabrication company learns of a new device that can predict whether the wafer will be defective after only a few seconds with a reliability of 99%. This will allow the fabricator to pull the wafer and abort the production process before the more expensive lithography process begins.

Explain how Bayes' rule would help the fabricator decide if this device is a worthwhile investment and give your recommendation based on these figures. You do **not** have to provide an exact calculation but you should give approximate figures. State any assumptions you make.

[6]

Writing T=0 to mean test is negative, and T=1 to mean test is negative.

Assuming that 99% reliability means that: [note: this is an **assumption** and must be stated; other assumptions are valid answers]
P(T=1|D=1) = 0.99
P(T=0|D=0) = 0.99
then
P(T=1|D=0) = 0.01
P(T=0|D=1) = 0.01
[#1]

The probability that a wafer is defective given that the test is positive is:

P(D=1|T=1) = P(T=1|D=1)P(D=1) / P(T=1) [#1 for Bayes' Rule]

P(D=1|T=1) = 0.99 * 0.00001 / [P(T=1|D=1)P(D=1) + P(T=1|D=0)P(D=0)]

~= 0.00001 / [0.00001 + 0.01] [note: 0.99 has been rounded off to 1]

~= 0.00001 / 0.01 [rounding 0.00001 off to 0.0]

~= 0.001 [#2 for any answer within an order of magnitude]

For every thousand wafers pulled when the test was positive, just one would be expected to actually be faulty [#1]. Unless the cost of pulling the wafer is extremely small, this test is nearly useless given the rarity of defects. [#1]

[Unit 8: Probability and random variables]

(c) The production system is to be optimized to minimize the number of defects.

(i) The fabrication process has adjustable temperature, dopant bias and slice thickness. The temperature cannot exceed 900C. State the objective function, parameters and constraints in this optimization problem.

[3]

The objective function is the number of defects, [#1] as a function of the parameters temperature, bias and thickness [#1]. The constraints are the requirement that the temperature be less than 900C. [#1]
[Unit 6: Introduction to optimization]

(ii) The fabricator is considering optimizing the defect rate using **simulated annealing**. Outline how the simulated annealing metaheuristic works, and describe in what situations in might be a better choice than random local search as applied in stochastic hillclimbing.

[3]

Simulated annealing accepts "bad" steps at the start of the optimization [#1], with decreasing probability as the optimization progresses, given by a "temperature schedule" [#1]. This provides resistance to local minima [#1] which will trap stochastic hillclimbing.
[Unit 6: Introduction to optimization]

(iii) In the context of testing optimization algorithms, what relevant diagnostic could be visualised to evaluate how well the optimization is going?

[2]

The convergence [#1] of the optimization could be indicated by looking at a plot of the objective function value against iteration [#1]
[Unit 7: Numerical nonlinear optimization]

(iv) After discussion with the plant engineers, it is decided that the process is to be optimized to minimize defects *and* keep the temperature of the baking process as small as possible. Describe how this relates to the concept of Pareto optimality and discuss why a Pareto optimization approach could be superior to a convex sum approach to multi-objective optimisation.

[3]

A Pareto optimal solution is a solution in which improving any sub-objective function makes at least one other sub-objective function worse [#1]. Pareto optimality does not depend on the absolute scaling of sub-objective functions [#1], and so weightings do not have to be estimated [#1]
[Unit 7: Numerical nonlinear optimization]

2. You are building a suite of scientific software which has to eventually be GPU accelerated. This means that the code has to be vectorised.

(a) In writing software for a vectorised architecture, what programming pattern should be **avoided** in order to take advantage of hardware vectorisation support?

[2]

Explicit iteration (or explicit looping) should be avoided [#2] to allow operations to be parallelized automatically.

[Unit 2: Numerical issues and tensors]

(b) In NumPy, write a *vectorised* function which computes and returns the result of the equation below. **x** and **y** are two 1D vectors with the same length N.

$$z = \sum_{i=0}^{N-1} \mathbf{x}_i^{\frac{i}{k}} \mathbf{y}_i^{\frac{i}{k}}$$

The function should have the signature:

```
def compute(x,y,k):
    ...
    return z
```

[6]

```
def compute(x,y,k):
    exponent = np.arange(len(x))/k
    z = np.sum(x**exponent * y**exponent)
    return z
```

[#1] for arange
[#1] for length of k
[#1] for divide by k
[#1] for use of sum
[#1] for exponentiation
[#1] for product of x and y

NO MARKS if an explicit iteration (`for/while`) is used.
Partial marks if arange is used but for for computation or vice versa

[Unit 1: Arrays]
[Unit 2: Numerical issues and tensors]

(c) The software will do computations on arrays of IEEE754 float64 numbers.

(i) If the exponent of a float64 has bit pattern 11111111111 (all ones) what three distinct values could this represent, and how are they distinguished?

[3]

- +inf, if sign is 0 and mantissa is all zero [#1]
- -inf if sign is 1 and mantissa is all zero [#1]
- NaN if mantissa is nonzero [#1]

[Unit 2: Numerical issues and tensors]

(ii) Explain the role of the implied leading 1 in the mantissa, and state why this is a useful approach in binary floating point representations, but not in decimal.

[3]

The mantissa represents a binary number of the form 1.xxxx [#1]. Every nonzero binary number must have a leading 1 [#1], which can always be shifted into the right position using the value of the exponent. However, a decimal number could have a leading digit 1-9 [#1] and an implied leading digit would not be feasible.
[Unit 2: Numerical issues and tensors]

(iii) You are asked to perform a floating point calculation of the form $((a-b) + c) / d$. Give three different sources of floating point inaccuracy that might occur in this computation, and properties of a,b,c and d that would trigger these problems. Assume that a,b,c,d are all nonzero finite IEEE754 float64s.

[3]

If a and b are very similar in value, the subtraction is likely to introduce error (cancellation error); [#1]
If (a-b) and c are very different in magnitude, roundoff error is very likely; [#1]
if d is very small, division magnification is very likely. [#1]
[Unit 2: Numerical issues and tensors]

(d) You are asked to help optimise some vectorised GPU code, which uses a matrix multiplication and inversion to compute the equation:

$$\mathbf{y} = A^{-1}\mathbf{x}$$

At the moment, the implementation uses the SVD. However, you have learned that A is in fact a **square diagonal** matrix. Suggest a much faster approach, and write vectorised NumPy code that computes the value of **y** from A and **x**. The diagonal of a matrix A is returned by `np.diag(A)`

[4]

The inverse of a diagonal matrix is just the reciprocal of its entries [#1]. And the product of a diagonal matrix with a vector is the elementwise product of the diagonal with that vector. [#1]

Therefore a more efficient solution would be:

```
y = 1.0/np.diag(A) * x
```

[#2]

[Unit 5: Computational linear algebra]

(e) Part of the vectorised software being developed will involve probabilistic calculations; in particular the calculating of the likelihood of multiple simultaneous system failures, each of which is very rare. After calculation, a computed probability has to be shown on an alphanumeric display so an operator can make a judgement as to whether to take action. Recommend appropriate choices for computation and display of these probabilities, justifying your choice.

[4]

Storing and computing the probabilities in "ordinary" form will be likely to be subject to underflow [#1] Log-probabilities [#1] could be used to do computations with much less risk of underflow. In particular, multiplying together many likelihood terms can be done by summing log-probabilities [#1]. Display might be best converted to odds for user understanding [#1] *(log-odds would also be an acceptable answer here, though good luck teaching your average user about log-odds!)*
[Unit 8: Probability and random variables]

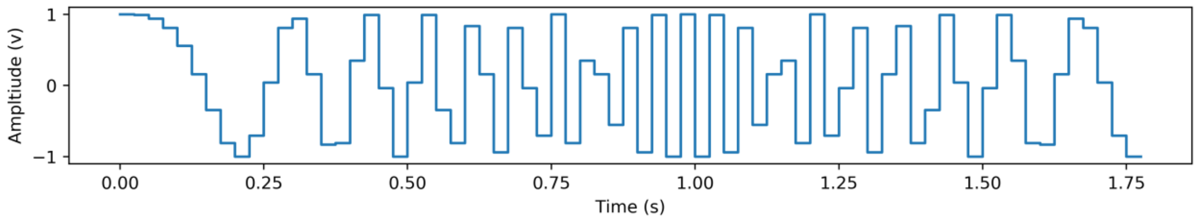
3. (a) Define the Nyquist limit, specifying both its definition and its relation to the phenomenon of aliasing.

[3]

The Nyquist limit a frequency equal to half the sampling rate $f_n = f_s / 2$ [#1]. A signal which changes more frequently than the Nyquist rate (has oscillatory components above Nyquist) will be subject to aliasing [#1], where high-frequency components will be "aliased" to apparent low frequency components [#1].

[Unit 10: Time series and signals]

(b) You are involved in building a test system for a low-bandwidth seismic monitoring device. You are presented with the following graph, showing a plot of a regularly sampled signal from the device. This shows the measurement of a test signal which continuously increased in frequency, starting 0Hz and increasing by 20Hz every second. Estimate the sampling rate of the seismic device, and justify your reasoning.



[5]

We could estimate the sampling frequency by looking for aliasing [#1]. We know the signal is increasing in frequency, but the plot looks like the frequency starts to *decrease* at about 1.0 seconds, [#1] this would indicate the Nyquist rate is ~20Hz [#1] and thus the sampling frequency is ~40Hz [#2]. Any answer between 20Hz and 80Hz would be acceptable for full credit as long as the reasoning is present.

[Unit 10: Time series and signals]

(c) You are asked to produce a figure showing an ultrawideband seismic device's sensitivity across a range of frequencies. The device has three different settings: A, B, C, which influence the sensitivity response. For each setting, the response at each test frequency is measured 100 times to provide an estimate the uncertainty.

The caption for the figure is:

"Figure 1 shows the sensitivity of the prototype UWB seismic device as the input frequency, in Hz, is varied from 0.001Hz to 10kHz. The sensitivity in the settings A, B, and C are shown. Measured sensitivity ranges from 0 to 500 V/mm s^{-2}"

Sketch an outline of an appropriate figure, using *layering*. This figure will be reproduced in black and white. You do not have any knowledge of the data, so you may assume any reasonable form for the graph. Ensure all details are present.

[8]

Figure must have:

Figure should be layered, not faceted [#1]

Must have a legend [#1]

Must have visibly different line styles for layers (dashes or markers, but **not** colour) [#1]

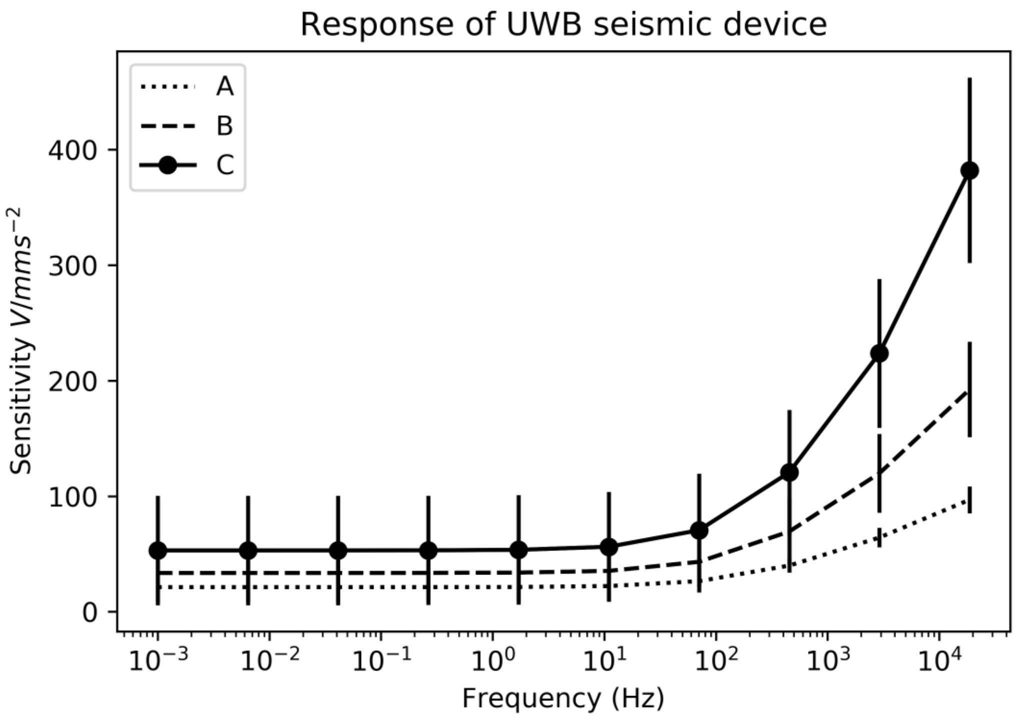
Figure must have a title [#1]

Axes must be labeled [#1] with appropriate units [#1]

Frequency must be on x axis, sensitivity on y axis [#1]

Must have some representation of uncertainty [#1] like error bars, ribbon plot, or Box plot.

Bonus point [+1] if x scale is logarithmic (only applicable if total score < 8)



Example solution above (obviously the shape of the curves does not matter) Many other forms are acceptable for full credit as long all of the listed details above are present.

[Unit 3: Scientific visualisation]

(d) You are asked to detect portions of a seismic signal where the signal appears to "repeat" itself; that is there a short portion of activity which is very similar to one that has been sensed before. These repetitions are not regularly spaced, but could occur anywhere in the signal. Describe in high-level terms how you would build an algorithm to detect these repetitions, using the following ideas: *sliding window; the infinity norm; argmin.*

Do not write code. Explain, briefly, in pseudo-code or words how you would go about this, referring explicitly to the named terms.

[6]

This could be achieved by:

- converting the signal into a series of fixed length vectors using a *sliding window* [#2]
- subtracting each "new" vector from all previous vectors and computing the *infinity norm* of the result [#2]

- finding the minimum norm vector (most similar portion) using *argmin* and deciding if it is close enough to consider to be a repetition [#2]

[Unit 10: Time series and signals]

[Unit 4: Vector spaces and matrices]

(e) (i) If the *norm* of a vector captures a notion of "length", what geometric notion does the *inner product* capture?

Inner product captures the notion of "angle" between vectors. [#1] or any similar wording

[Unit 4: Vector spaces and matrices]

[1]

(ii) Inner-product based measures of similarity are often used in problems where high-dimensional vectors are compared. For example, a seismic measurement of a suspected nuclear blast might be represented as a 10,000 dimensional vector which could be compared against a catalogue of other (equally sized) vectors of known blasts. Give an argument as to why the standard Euclidean norm of the difference of two vectors $||\mathbf{x}-\mathbf{y}||_2$ might **not** be used as a measure of similarity in this application.

[2]

In very high dimensions, the Euclidean norm can be misleading [#1] as all distances are likely to be very similar [#1].

[Unit 4: Vector spaces and matrices]