December 2020 Exam Feedback – Introduction to Machine Learning

Q1
(a) (i) Most students scored full marks here as long as the answer is correct and they either showed their workings or explained their answer intuitively.
(ii) This question required an understanding of entropy/information gain as a measure of the "purity" of a dataset. While many students were able to answer this question well, many also struggled. The calculations are relatively straightforward as in the lectures once the student figures out the numbers to plug in. A lack of workings may cost the student one or two marks. A few students stopped at computing entropies; these would usually be valid, but unfortunately the question specifically asked for information gain, so some marks may be lost if only entropies were calculated.
(iii) Marks were awarded based on whether students chose the attribute with the maximum information gain from their answer in (ii), rather than whether the attribute itself is correct. Minimum entropy was also accepted.

(b) (i) The ideal answer would require students to select the model which minimizes the model's BIC or negative log likelihood (or maximizes the (log) likelihood). BIC would have been better in practice as the two models have a different number of components, but the examiner accepted both. The ideal base for log would have been log_e, but log_10 and log_2 were also accepted.
Common mistakes:

1. Computing \sum_k^K log(pi_k * \sum_i^M N(x_i | theta_k)), rather than \sum_i^M log(\sum_k^K pi_k * N(x_i | theta_k)). Note where the summation across the 5 instances occur. You should perform the weighted sum across components for *one* instance, and then sum across all M instances.
2. Summing probabilities rather than multiplying them (you should either sum the LOG probabilities, or multiply probabilities)
3. Computing component responsibilities/using EM: these do not demonstrate model quality

Some students explained with diagrams - although none were convincing enough to score full marks for the question. Some students lost some marks for calculation errors - how much is deducted depends on the severity of the errors and how much workings were provided: many students who showed their workings were largely saved.

(ii) The examiner was lenient and accepted most reasonable answers. A model answer would include a discussion on the model's expressiveness/complexity and on overfitting. A discussion about bias-variance trade-off was also a common answer and was accepted assuming the student explained it correctly (does increasing K increase or decrease the bias? Some students got this wrong). It would have been ideal to be concrete about what it means by low/high bias/variance in the context of a GMM rather than in general, but the examiner accepted these for this exam. A discussion about the effects on the NLL/BIC was also generally accepted.

Q2

(a) Calculation of the forward pass was mostly performed correctly. Some students picked the value from the wrong output neuron when asked about the output probability. Also the columns and rows were sometimes mixed up when it came to arranging the weights into a matrix so that it can be used for calculations, which resulted in incorrect values.
The backpropagation part was more of a challenge. In order to complete it in reasonable time, it was important to perform the calculations in an efficient way. For example, using the joint derivative of cross-entropy and softmax given in the lectures, instead of starting to differentiate them by hand. Also, reusing the values from the first part whenever possible, such as for calculating the derivative of tanh. There were some common errors regarding not following the question - calculating updates for the weights instead of the input, or using MSE as the loss function instead of cross-entropy as was specified in the question. I awarded as many points as

possible for any partially completed answers. Any errors or omissions resulted in deductions but they did not affect points given for downstream calculations.

(b) Mostly solved correctly with a few common errors. When designing a binary classifier then multiple different output types were allowed. However, one combination that cannot work is a single output neuron with softmax activation - if you do the calculation, you'll see that this neuron would always give 1 as output.
Also, there seemed to be some confusion about input layer sizes. An input layer of size x means the network can take x real-valued inputs. If you want to have two 256-dimensional vectors as input, you need to have an input layer of size 512.

Q3

(a) Most students scored full marks here, as long as the answer contained the main reasons for using a Quality-Diversity algorithm.

(b) The vast majority of students scored full marks here as well. However, a common mistake was to search for a complex answer that was not using the information provided by the simulator. Another common mistake was to mix elements of the behavioural descriptor (the size/weight of the parcels) in the fitness function. MAP-Elites will naturally only compare solutions from the same type of solution (i.e., same parcel types). Therefore, including this in the fitness is not necessary.
Finally, two points needed to be addressed and were missing in some answers: 1) what to do when the delivery time was "-1" and 2) how to ensure that the collection can only contain up to 400 solutions.

(c) The challenge of this question was that we had to consider both discrete and continuous parameters in the genotype. While the ideal answer should have considered a genotype composed of floats, I also accepted solutions discretising the width and length parameters, or a mixture of both floats and bitstrings, as long as the overall answer was consistent.
The most common errors came from either missing elements in the answer (e.g., the function to develop the genotype into the phenotype) and inappropriate mutation operators or configurations. For instance, a gaussian mutation operator cannot be used on a bit-string. Also, when using a genotype with floats, which is then discretised for the discrete values in the phenotype, it is important to ensure that the mutation is always effective. This is not the case with a standard gaussian mutation, which if the sampled value is too small might not be enough to make a parameter move to the next range. To avoid this, a more advanced mutation operator leveraging the specificities of the problem could have been proposed.

(d) The difficult aspect of this question was to recognise that this was an unsupervised learning problem: Clustering a dataset into 400 clusters. Most of the students who recognise this also used the appropriate algorithm to be used in this case.