HU

See you guys next year dont struggle no more

2018 COMP307 Exam at top. 2017 COMP307 Exam at bottom.

Suggested answers. Comment questions, edits or errors, or add additional answers. **Please don't directly *change* any written content**, just for our sanity. If unanswered, add.

Answers from my meeting with Yi:

- Do we calculate the β values for *all* nodes prior to updating the weights in the back propagation process? **A:** you calculate the beta values using the old weights, not the new ones, yes.
- If the answer to Q3.b.ii (2018 exam) is actually no change, just to make sure. **A:** this is correct, however you would still have to show working that this is the case.ct, however you would still
  - Kaan Demir
  - 42 mins
- Does anyone here think they will examine us on 3-node chains? I don't see it in any of the past exams.
  - Kaan Demir
  - 42 mins
- Does anyone here think they will examine us on 3-node chains? I don't see it in any of the past exams.
- Naive bayes: N ce, and is probably better practice to do so, but you're not going to get marked down for not doing so.Naive bayes: when a new instance doesn't require the table to be initialised with +1 values, should we still do so? A: it wouldn't change the result for a non-zero case instance, and is probably better practice to do so, but you're not going to get marked down for not doing so.Naive bayes: when a new instance doesn't require the table to be initialised with +1 values, should we still do so? A: it wouldn't change the result for a non-zero case instance, and is probably better practice to do so, but you're not going to get marked down for not doing so.when a new instance doesn't require the table to be initialised with +1 values, should we still do so? **A:** it wouldn't change the result for a non-zero case instance, and is probably better practice to do so, but you're not going to get marked down for not doing so.
- In a given bayesian network, should we assume non-discussed nodes are present or not (in reference to Q6.b.iii)? **A:** unless stated (e.g. as 'conditionally indepeeting with Yi:
  -
- Do we calculate the β values for all nodeendent' or as 'given'), you should assume a node is not given.
  - Given there seems to be a lot of confusion: how to build a n
- on-optimal ordering bayesian network. **A:** The answer to 2018's Q6.f is correct.

- When to use enumeration and when to simply deduce a joint probability from the CPTs? (reference to 2017: Q 6.e). **A:** when a hidden node separates the query node(s) and evidence node(s), you should use enumeration.
- Will a formula sheet be provided? **A:** yes, anything that's needed will be provided.

1. Search
    a. Iterative deepening:
        i. Increase the maximum allowable depth of a depth-limited search algorithm by one, from zero, until a suitable solution has been found or a maximum depth has been reached.
        ii. As breadth-first search is an exhaustive search, significantly more memory will be used, despite the repeated search of the lower layers (the breadth-first search will use more memory? Yes, because it has to track all the nodes).
        iii. When the search space is large and the solution depth is unknown.
    b. Greedy (best first) search and A* Search:
        i. Neither.
        ii. Greedy (best first)
        iii. A* Search
    c. Hill Climbing, A*, gradient descent, genetic beam search:
        i. A* is an informed graph search method which iteratively constructs a path to the goal. Hill climbing, is a state space search method that aims to step from the current state to a neighbour state in the direction of higher fitness. In contrast, only stores one state and its evaluation, whereas A* stores all nodes it has visited, so is much more expensive.
        ii. Both easily get stuck in local optima.
        iii. A mechanism to detect when an optima has been reached (e.g. no fitness change has been detected for X evaluations), at which point the state is randomly reinitialised and the process restarted (to some stopping point).
        iv. Hill climbing operates in the discrete search space, gradient descent in the continuous. the best and w
        v. Gradient descent is a local search in the sense that it moves a single state through a search space in a controlled manner towards a known objective (a local optima). Beam search is global in the sense that it has a population of individuals that are first randomly initialised through the search space, then operated on to perform the search. These operators are not in a controlled manner, however, and minor genetic changes can result in major fitness changes, exploring large portions of the state space. All individuals in a beam search are nonetheless solutions. In contrast, in gradient descent, there is (usually) only a single solution.
2. Machine learning basics
    a. Supervised vs unsupervised learning

i. Supervised: instances in the training set are labelled, Unsupervised: instances in the training set are unlabelled
ii. Supervised: 1. facial recognition at border security, 2. sorting robots on a production line, 3. regression: mapping f(x) = y. Unsupervised: 1. autonomous galaxy detection (star clustering), 2. Finding valuable data from a crowded database (feature selection).

b. Paradigms
   i. Induction: Decision trees
   ii. Connectionist: Neural networks
   iii. Genetic/evolutionary learning: Genetic algorithms/programs
   iv. Statistical learning: Naive Bayes

c. K-fold cross validation

   Separate the dataset into K subsets. Repeat the training process K times, using each subset as the test set exactly once, with the remaining sets as the training set. The results across the K runs can then be averaged (or otherwise combined) to provide a single estimation.

d. K-means clustering assumptions:Gradient Descent assumes you optimize a continuous function and can compute it's gradient in a given state.

   We need to assume we have prior or expert knowledge on the number of clusters (i.e. we know k), that the clusters are of similar size, the clusters are somewhat spherical in shape, and we have a suitable distance measure for use in the given problem.

e. K nearest neighbour
   i. Because the three features operate on very different scales. For example, F1 ranges from -0.72 to 0.94, whereas F3 ranges from 12 to 947 - a factor of 1000. This makes F3 dominate the distance measure, effectively being the only one that matters.
   ii. Kate could normalise the distance calculated (i.e. F1x - F1y) on the range across all instances for each feature. I.e. change (F1x - F1y) to ((F1x-F1y)/range(F1)). X

f. Decision trees

   TEMP impurity (1/4 * 3/4 * 4/10) + (1/2 * 1/2 * 2/10) + (3/4 * 1/4 * 4/10) = y:0.2

   HUMIDITY impurity: (4/6 * 2/6 * 6/10) + (1/4 * 3/4 * 4/10) = 0.2083
   WINDY impurity: (3/5 * 2/5 * 5/10) + (3/5 * 2/5 * 5/10) = 0.24
   TEMP has lowest impurity, so we select that.

3. Neural networks
   a. 1. A validation set to initialise early stopping, 2. K-fold cross validation to falsely 'enlarge' the dataset size, 3. Decrease the number of weights to train by decreasing the number of hidden nodes. 4…?
   b. Feed forward neural network calculations
      i. Output of node 6 calc: [**Verified x8**]
      O1 = I1 = 0

O2 = I2 = 0
I3 = O1 * w13 + O2 * w23 + b3 = 0*1.2 + 0*-2.6 + 3.1 = 3.1
O3 = sigmoid(3.1) = 0.95689
I4 = O1 * w14 + O2 * w24 + b4 = 0*-2.3 + 0*0.3 + 0.9 = 0.9
O4 = sigmoid(0.9) = 0.71095
I5 = O1 * w15 + O2 * w25 + b5 = 0*3.1 + 0*-3.6 + -1.3 = -1.3
O5 = sigmoid(-1.3) = 0.21417
I6 = O3 * w36 + O4 * w46 + O5 * w56 + b6
= 0.95689 * 2.2 + 0.71095 * -1.7 + 0.21417 * 2.5 + 1.8 = 3.23197
O6 = sigmoid(3.23197) = 0.96201

ii. Back prop calc: [**Verified by Yi**]

- B6 = 0 - 0.96201 = -0.96201
- B4 = w46 * O6 * (1 - O6) * B6 =
  -1.7 * 0.96201 * (1-0.96201) * -0.96201 = 0.05977
- Δw14 = η * O1 * O4 * (1 - O4) * B4 =
  0.2 * 0 * 0.71095 * (1 - 0.71095) * 0.05977 = 0
- ∴ w14' = w14
  O6 = sigmoid(3.23197) = 0.96201

  Back prop calc: [Verified by Yi]

  B6 = 0 - 0.96201 = -0.96201

  B4 = w46 * O6 * (1 - O6) * B6 =

  -1.7 * 0.96201 * (1-0.96201) * -0.96201 = 0.05977

  Δw14 = η * O1 * O4 * (1 - O4) * B4 =
  INCORRECT: ◂ why are we having this with grey colour in some questions? Like why does it say Original answer is incorrect? Does that mean it has been changed and this grey color does indicate wrong answer? Just somewhat it takes spaces and bothering.
  This original calculation is incorrect, due to using w46' to calculate B4. We should instead use the original w46 to calculate B4, then calculate Δw14 from there, as above.
- Bit of a trick question - see the final line.
- Because we only want the weight for w14, we only need the error value (B) for node 4 (B4).
- First, we need B6, then we have to calculate the change in weight of w46.
- B6 = difference between the goal (0) and the received (0.96201, from i. above) output values for the output node 6, so = -0.96201

// note: there is a comment suggesting that we calculate the error values (B4 in this case) before we calculate and update the weight change.

- $\Delta w46 = \eta * O4 * O6 * (1 - O6) * B6 =$ $0.2 * 0.71095 * 0.96201 * (1 - 0.96201) * -0.96201 = -0.004999$
- $w46'$ (new) = w46 (old) + $\Delta w46 = -1.7 + -0.004999 = -1.704999$
- Then we can calculate B4, because we have updated all of its output weights (there's only one link from node 4 to node 6):
- INCORRECT: $B4 = w46' * O6 * (1 - O6) * B6 =$ $-1.704999 * 0.96201 * (1-0.96201) * -0.96201 = 0.05994$
- $\Delta w14 = \eta * O1 * O4 * (1 - O4) * B4 =$ $0.2 * 0 * 0.71095 * (1 - 0.71095) * 0.05994 = 0$
- $\therefore$ $w14'$ = w14 (as there is no change).

4. Evolutionary computation and learning
    a. Tournament selection: randomly select k individuals from the population to form a tournament of size k (typically ~7 in genetic programming). From this, the most fit, by some objective function, individual is selected as the 'winner'.
    b. Genetic operators (What am i reading o_0)
        i. Single point crossover: randomly choose a single index point in the parents' genome at which the remaining data in the genome is switched. For example, if index = 2 (start 0): AAAAA + BBBBB → AABBB & BBAAA
        ii. Uniform crossover: randomly choose a random number of non-descending index points, which form the start and end of crossover points. For example, if indexes = 1, 1, 3, 3: AAAAA + BBBBB → ABABA + BABAB.
        Both forms of crossover can be done in the form of a crossover mask, where the single point crossover example above would be represented as: 11000 and the uniform crossover would be represented as: 10101.
    c. Genetic programming: symbolic regression
        i. 1. The division operator is not protected, so when the denominator is 0 a NaN is returned, which GP cannot handle. 2. There are no boolean valued terminals in the terminal-set, nor returned by any function that the IF function can utilise, so whenever it is used in a tree, it will return an error.
        ii. 1. Modify the standard division operator to protected division %, which returns 1 if the denominator is 0. 2. Modify the IF operator to work in the same value space as the other terminals: return b if a > 0 and c if a <= 0.
    d. Genetic programming: binary classification
        i. X1, X2, X3, X4, X5, integers[1-9]
        ii. *, %, +, -, IF (structure as in 4.c.ii)
        iii. If the evolved program returns a value over a given threshold T (say, 0), assign the class to a specific class (say, O), otherwise, assign it to the

other class (X). Genetic programming should be capable of being indifferent to the value of T.

iv. 1. Define multiple values of T, where each interval defined between T_i, T_j assigns the instance to a different class. 2. Instead of the evolved program simply returning a floating point number, modify it to return a vector based object of floating point numbers, increasing the possible dimensionality (the number of classes). 3. [**Unsure**: Could you break this down into every possible binary classification problem, then deduce which is most likely?]. 4. [Classify a training set using supervised learning and classify the remaining with k-nearest neighbour?].

5. Reasoning under uncertainty
   a.
      i. P(A,B) = P(A)P(B|A) = 0.4 * 0.25 = 0.1
      ii. P(B|-A) = can't calc? **[uncertain]**
      iii. P(-B|A) = P(A,-B)/P(A) = 0.15 / 0.25 = 0.6
         - P(A,-B) got via sum rule
         - Alternatively, much more simply:
           P(-B|A) = 1 - P(B|A) via normalisation = 1 - 0.4 = 0.6
      iv. P(B) = can't calc.
         Original answer: *INCORRECT* ◂ why are we having this with grey colour in some questions? Like why does it say Original answer is incorrect? Does that mean it has been changed and this grey color does indicate wrong answer? Just somewhat it takes spaces and bothering.
         (P(A)P(B|A))/P(A,B) = (0.25 * 0.4) / 0.1 = 1.0
         - via (incorrect) Bayes' rule modification
         - 100% chance of B seem9s like a bit of a trick question.
   b. A ⊥ B?: No, A is not independent of B as P(A,B) != P(A)P(B), i.e. 0.3 != 0.3 * 0.7. If two events are independent, then their joint probability is directly proportional to their singular probability. I.e. two flips of a coin do not affect each other, yet the chance of landing two heads in succession is less than a head and a tails.
   c. Three boolean variables:
      i. P(A, B|C) = P(A|C)P(B|C) = 0.5 * 0.2 = 0.1
      ii. P(A|B,C) = P(A |C) = 0.5 (how?) :)
   d. Joint probability tables
      i. 0.05 + 0.25 + 0.2 + 0.1 = 0.6
      ii. 0.1 + 0.15 = 0.25
      iii. 0.15 / 0.5 = 0.3
         [**should show working in the form like P(A=1, B=0) etc etc etc**]
6. Bayesian Networks
   a. Naive Bayes assumes that all features are conditionally independent.
   b. Bayes belief network
      i. True
      ii. True, D is unknown (common effect)

iii. False, B and D are unknown (indirect cause)
iv. False, once D is known they become dependent, and E is unknown (common effect) **[Verified]**
v. True, B is known (indirect cause)
vi. False, oFalse, once G is known they become dependent, and D is unknown (common effect)False, once G is known they become dependent, and D is unknown (common effect)False, once G is known they become dependent, and D is unknown (common effect)nce G is known they become dependent, and D is unknown (common effect)

c. Naive Bayes Classifier
   i. Score(approve):
   P(apr)P(J=f|apr)P(D=h|apr)P(F=s|apr) = 5/10 * 1/5 * 3/5 * 3/5 = 0.036
   Score(reject):
   P(rej)P(J=f|rej)P(D=h|rej)P(F=s|rej) = 5/10 * 3/5 * 1/5 * 1/5 = 0.012
   Score(approve) > Score(reject), therefore we approve.
   ii. Score(approve):
   P(apr)P(J=t|apr)P(D=h|apr)P(F=ch|apr) = 6/12 * 5/7 *4/7 * 1/8 = 0.0255
   Score(reject):
   P(rej)P(J=t|rej)P(D=h|r=ej)P(F=ch|rej) = 6/12 * 3/7 * 2/7 * 3/8 = 0.0230
   Score(approve) > Score(reject), therefore we approve.

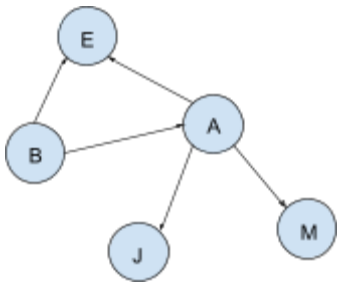
d. P(B)P(E)P(A|B,E)P(J|A)P(M|A) **[Is this what they want?]** I wouldnt say this is wrong because it did not give us any conditions.

e. Variable elimination via join and sum out operations [Yi confirmed method]:
   P(b,j,m) = P(b)P(E)P(A|b,E)P(j|A)P(m|A)
   Sum over all A & E - the hidden variables we don't care about.
   P(b,j,m) =
   P(b)P(e)P(a|b,e)P(j|a)P(m|a) + P(b)P(-e)P(a|b,-e)P(j|a)P(m|a) +
   P(b)P(e)P(-a|b,e)P(j|-a)P(m|-a) + P(b)P(-e)P(-a|b,-e)P(j|-a)P(m|-a)
   = 0.01*0.02*0.95*0.90*0.7 + 0.01*0.98*0.94*0.9*0.7 +
   0.01*0.02*0.05*0.05*0.01 + 0.01*0.98*0.06*0.05*0.01
   = 0.0059

f. Building a Bayesian network **[Correct]**:
   i. Add B.
   ii. Add A.
   P(A|B) = P(A)? I.e. is Alarm $\perp$ Burglary? Given knowledge of A & B, we know it's not, so B $\rightarrow$ A.
   iii. Add J.
   P(J|A,B) = P(J)? I.e. is John Calling $\perp$ Alarm & Burglary? No.
   P(J|A,B) = P(J|A)? Yes, due to indirect cause, so only link A $\rightarrow$ J
   iv. Add E.
   P(E|A,B,J) = P(E)? No
   P(E|A,B,J) = P(E|A)? No
   P(E|A,B,J) = P(E|B)? No

…
P(E|A,B,J) = P(E|A,B)? (we know earthquakes are only impacted by alarms and burglaries) Yes, so two links B → E & A → E

v. Add M.
P(M|A,B,J,E) = P(M)? No
P(M|A,B,J,E) = P(M|A)? Yes. As A is already known, the common cause in the true graph causes J & M to be independent, thus no J → M link.



7. Planning and Scheduling
    a. Classic planning solutions, are typically in the form of a sequence of actions to achieve a goal state from an initial state, whereas in scheduling a solution is, unsurprisingly, a minimal makespan schedule. More importantly: scheduling tasks are made of subcomponent operations, which take a specified length of time, so are assigned a specific start and end time upon assignment and completion, respectively.
    b. Conventional forward state-search enumerates all possible paths of the state space tree, which, for dynamic job shop scheduling is a phenomenally large space - to the point of infeasibility. Dispatching rules are effective as these evaluate the current state and select a ‘good enough’ path through the state space tree in real time, far more efficiently than traditional tree search methods.
    c. PDDL representations:
        i. Invalid. -At(Left) violates the functionless requirement.
        ii. Valid
        iii. Invalid. Both terms take variables.
        iv. Invalid. Plane(Tom) would return a variable, so it violates the functionless requirement.
    d. Mike’s cakes
        i. Action(Eat(Cake)
            Precond: Have(Cake)
            Effect: Eaten(Cake) ^ -Have(Cake))
        ii. Eaten(Cake)
        Note: no specification of -Have(Cake) because it would violate the functionless requirement and given it is not defined, it is false by default due to the closed world assumption.
    e. Job shop scheduling.

Poorly articulated, here. Is it the same as in lectures where engines have to be mounted first? Presumably the machines and jobs are available at t=0? If so:

i. - Process(AddEngine1, EngineHoist, 0)
   - Process(AddEngine2, EngineHoist, 0)
ii. 0
iii. Infinite (∞)

f. Vehicle routing problem.
   i. Infeasible.
      1. Neither route starts or ends at the depot
      2. R1 should be split into two different routes around the return to the depot.
      3. R2 serves a total of 5 demand, exceeding capacity.
      4. R1 and R2 share task 1 when it should only be served once
   ii. Infeasible
      1. Neither route starts or ends at the depot
      2. R1 should be split into two different routes around the return to the depot.
      3. R2 serves a total of 5 demand, exceeding capacity.
      4. R1 and R2 share both tasks 1 and 6 when each should only be served once.
   iii. Infeasible
      1. R1 serves a total of 5 demand, exceeding capacity.
   iv. Infeasible
      1. R1 and R2 repeat task 6 when it should only be served once.
      2. Neither route serves task 1.

8. Other topics
   a. Neural network deep learning architectures:
      i. Types of architectures: Convolutional neural networks, Generative adversarial network, autoencoder, **others…?**
      ii. Facial recognition, self driving cars, native language processing & translation, **others…?**
   b. Non-neural network deep learning architectures:
      i. Logistic regression, deep belief networks, restricted Boltzmann machines, deep forest, protocol component analysis, genetic programming based deep learning, **others…?**
   c. 5Vs of big data:
      i. Variety: the different types (forms, sources) of data.
      ii. Velocity: the frequency at which data is being produced or moved.
      iii. Veracity: the quality, trustworthiness, availability of the data.
      iv. Volume: the amount of data you have (Gb, Tb, etc).
      v. Variability: inconsistency in the dataset.
      vi. Value: the types variables take on (integer, boolean, etc).

d. SVMs attempt to step a dataset that is currently not linearly separable to a higher dimension in the hopes that a linear model in said dimension will be able to do so. The algorithm doesn't directly perform this dimensionality increase, but instead optimises an objective function to maximise the distance between instances on the boundary between the two (or more) classes (called the SVM).
e. Somewhere like the UN where people of many ethnicities need near-perfect, real-time translation from any language to their own. A voice controlled assistant on your smartphone for use while driving.

2017 Exam:

1. Search
    a. 1,2,3,4,5,6,7,8,9,10,11,12,13
    Or 1 2 3 1 2 4 5 1 3 6 7 1 2 4 8 9 … etc very long but 1 mark so the above answer
    b. Reminder: iterative deepening is depth firsGradient Descent assumes you optimize a continuous function and can compute it's gradient in a given state.t search.
    d=0: 1
    d=1: 1,2,3
    d=2: 1, 2,4,5,3,6,7
    d=3: 1,2,4,8,9,5,10,11,3,6,12,13,7
    c. A → C, A → B, A → C → D, A → C → D → E
    Final solution: A → C → D → E
    d. Greedy best first search will choose the next node with the lowest estimated cost to the goal state from node n, irrespective of the cost to get to node n from the initial state. A* on the other hand, does consider this factor, making it more well rounded as an evaluation metric.
    e. Beam vs gradient descent
        i. Genetic algorithms
        ii. Hill climbing, simulated annealing
        iii. Gradient descent is a local search in the sense that it moves a single state through a search space in a controlled manner towards a known objective (a local optima). Beam search is global in the sense that it has a population of individuals that are first randomly initialised through the search space, then operated on to perform the search. These operators are not in a controlled manner, however, and minor genetic changes can result in major fitness changes, exploring large portions of the state space. All individuals in a beam search are nonetheless solutions. *[same answer as in the 2018 exam]*
2. Machine learning basics

a. Supervised vs unsupervised *[same as the 2018 exam]*
    i. Supervised: instances in the training set are labelled, Unsupervised: instances in the training set are unlabelled
    ii. Supervised: 1. facial recognition at border security, 2. sorting robots on a production line, 3. regression: mapping f(x) = y. 0Unsupervised: 1. autonomous galaxy detection (star clustering), 2. Finding valuable data from a crowded database (feature selection).
b. Supervised machine learning
    i. Training set: to provide a fitness evaluation mechanism to direct learning during the training process.
    Validation set: to detect overfitting during the training process on which the individuals are evaluated every X epochs, an--d the data is not provided back to the training system. A sign of overfitting is once the classification accuracy on the validation set starts to increase, at which point early stopping criteria (or other recourse) is actioned.
    Test set: an unseen set of instances to evaluate the individual(s) after the training process.
    ii. K-fold cross validation is not an instance set, it is an experiment setting. You split the entire dataset into k subsets, and in k different experiment runs, you use each of the k subsets as the test set exactly once and the remaining sets as the training set. The results from the k runs can then be averaged or otherwise combined to receive a single value.
c. K-means & nearest neighbour
    i. K-nearest neighbour is for classification, K-means clustering is for clustering.
    ii. K-nearest neighbour is usually used for numerical data, as there is a requirement of a distance metric.
    iii. If you're using the same random seed, and you randomly generate the same starting centroids, it should. **[confirm?]**
d. XOR problem
    i. No. An individual perceptron cannot solve XOR problems. A multi-layer perceptron, however, can.
    ii. It is only able to solve linearly separable problems.
e. Mushroom impurity: 4/5 * 1/5 * 5/10 + 1/5 * 4/5 * 5/10 = 0.16
Vegetarian impurity: 4/6 * 2/6 * 6/10 + 1/4 * 3/4 * 4/10 = 0.208
Size impurity: 1/3 * 2/3 * 3/10 + 2/3 * 1/3 * 3/10 + 2/4 * 2/4 * 4/10 = 0.233
Mushroom has lowest impurity, so we select that.

3. Neural Networks
    a. Feed forward neural network
        i. NN node 5 output: [**Verified: x3**]
        O1 = I1 = 0
        O2 = I2 = 0
        I3 = O1 * w13 + O2 * w23 + b3 = 0*1 + 0*2 + -1.9 = -1.9

O3 = sigmoid(-1.9) = 0.13011
I4 = O1 * w14 + O2 * w24 + b4 = 0*-1 + 0*1.5 + 2.4 = 2.4
O4 = sigmoid(2.4) = 0.91683
I5 = O3 * w35 + O4 * w45 + b5 = 0.13011*1 + 0.91683*2 + 0.5 = 2.46377
O5 = sigmoid(2.46377) = 0.92156

I6 = O3*w36 + O4*w46 + b6 = 0.13011*1.5 + 0.91683*2.6 + -6 = -3.42108
O6 = sigmoid(-3.42108) = 0.03164

ii. Back prop: **[Verify]**
B5 = d0 - O5 = 0 - 0.92156 = -0.92156
Δw35 = η * O3 * O5 * B5 =
0.15 * 0.13011 * 0.92156 * (1 - 0.92156) * -0.92156 = -0.001300
w35' = w35 + Δw35 = 1 + -0.00129 = 0.99870

Added by Anon: correct logic, corrected calculation:
B5 =d5-O5
=0.0-0.922
=-0.922
Change in W3->5 = n*O3*O5*(1-O5)*B5
= 0.15*0.13*0.922*(1-0.922)*(-0.922)
= -0.00129
W35' =W35 + change in W3->5
=1.0+(-0.00129)
=0.99871

$B_j = \sum_k W_{j \to k} * O_k * (1 - O_k) * B_k$

b. Seems to be overfitting. Possible solutions: 1. reduce the number of hidden nodes, 2. Introduce a validation set and an early stopping time, 3. Introduce k-fold cross validation, 4. decrease the number of epochs. Planning and Scheduling 3 -- Dynamic Scheduling

4. Evolutionary computation
   a. Crossover: combine the genetic material of two individuals (selected by some fitness measure) to form two (hopefully) good quality 'children'.
   Mutation: randomly change a subset of the genome of an individual.
   Elitism: pass the fittest k individuals through to the next generation by default.
   b. 1. Initialise the population (typically at random)
   2. If the stopping criteria are not met:
      - Evaluate the population by some fitness measure f(.)
      - By the fitness measure, select x individuals from the population and perform genetic operators to form the next generation.
   c. Genetic Programming for symbolic regression
      i. {x,randDouble{-1,1}}
      ii. {+,-,*,%,^,IF}, where IF{a,b,c} returns b if a > 0 and c if a <= 0 (example)

iii. $1/k \Sigma_k |y' - y|/|y|$, average absolute difference between individual output and goal output, normalised onNeed to assume the number of clusters 80/12080/120 the absolute value of the goal output (to avoid any single instance dominating the rest). [**Thoughts?**]

iv. The objective of standard statistical regression is often to find the optimal coefficients of an existing regression equation. The goal of GP is not only to find the optimal coefficients, but also the optimal regression equation (i.e. the model/structure). Further, it does this on multiple models simultaneously (the population of individuals), unlike many traditional statistical regression models which focus on a single model. [**verify**]

5. Reasoning under uncertainty
    a. 0.55
    b. P(-B|A) = 0.75
    c. If A $\perp$ B then P(A|B) = P(A), $\therefore$ P(A) = 0.35
    d. Boolean variables A,B,C
        i. If A $\perp$ B | C, then P(A,B|C) = P(A|C)P(B|C), so P(A,B|C) = 0.2 * 0.4 = 0.08
        ii. Yes. If A is independent of B, it is independent of B in all circumstances, including when C is present.
        iii. No. If A is independent of B when C is present, it may not be independent of B when C is *not* present.
    e. Joint probabilities
        i. 0.06 + 0.15 = 0.21\
        ii. P(x1,x2,x3) = P(x2,x1,x3) = P(x2|x1,x3)P(x1,x3)
            P(x2,x1,x3) = 0.15 (from graph)
            P(x1,x3) = 0.21 (from above)
            P(x2|x1,x3) = P(x2,x1,x3) / P(x1,x3) = 0.15 / 0.21 = 0.714

            Secondary method:
            P(x2=1|x1=0,X3=0) = P(X1=0,X2=1,X3=0)/P(X1=0,X3=0)
            = 0.15(from table)/0.21(from above)
            = 0.714

Why arent we using bayes rule here? P(Y|X1,X2..)

6. Bayesian Networks
    a. Naive Bayes Classification
        i. Assumption: that the variables are conditionally independent of each other.
        ii. Because of the conditional independence assumption, we do not require the analysis of as much data. [**Verify**]
    b. The network/structure (in the form of a directed acyclic graph) and the conditional probability tables for each node.
    c. True/false questions:
        i. True: indirect cause.

ii. False: once a common effect becomes known, the joint causes must compete to determine which is the primary cause.
iii. True: indirect cause.
iv. True: common cause. Once a common cause is known, the joint effects cannot infer knowledge about each other.

d. P(H)P(S)P(C|H,S)P(X|C)P(D|C) [**is this what they want?**]

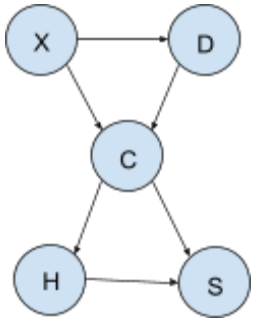
e. P(H,-S,C) = P(C,H,-S) = P(C|H,-S)P(H,-S) = P(C|H,-S)P(H)P(-S) 0
= 0.02 * 0.1 * 0.7 = 0.0014

Incorrect: variable elimination via join and sum out operations:
P(h,s,c) = P(h)P(S)P(C|h,S)P(x|C)P(d|C)
Sum over all C & S - the hidden variables we don't care about.
P(h,s,c) =
P(h)P(s)P(c|h,s)P(x|c)P(d|c) + P(h)P(-s)P(c|h,-s)P(x|c)P(d|c) +
P(h)P(s)P(-c|x,d)P(x|-c)P(d|-c) + P(h)P(-s)P(-c|h,-s)P(x|-c)P(d|-c)
= 0.1 * 0.3 * 0.05 * 0.9 * 0.65 + 0.1 * 0.7 * 0.02 * 0.9 * 0.65 +
0.1 * 0.3 * 0.95 * 0.2 * 0.3 + 0.1 * 0.7 * 0.98 * 0.2 * 0.3
= 0.007523

f. Bayes nets
i. Because adding arcs from all parents provides little useful information with regards to relationships between nodes. [**Confirm?**]
- I put that this is to minimise the number of arcs in the network and thus reduce computational complexity and improve readability.

ii. Building a bayesian network [**Verify x 2**]
- Add X
- Add D
P(D|X) = P(D)?
No, with unknown C, X & D are dependent (common cause), so add a link D → X
- Add C
P(C|D,X) = P(C)? No.
P(C|D,X) = P(C|D)? No.
P(C|D,X) = P(C|X)? No.
Therefore must be two links: X → C and D → C
- Add H
P(H|C,D,X) = P(H)? No.
…
P(H|C,D,X) = P(H|C)? Yes, [**indirect cause**], so link H → C
- Add S
P(S|H,C,D,X) = P(S)? Not independent
P(S|H,C,D,X) = P(S)? No. (because of the reason below)
...
P(S|H,C,D,X) = P(S|H,C)?

Yes, as H & S are the common causes of C, and C is known, H & S are dependent. So add two links: H → S and C → S.



g. Inference in bayesian networks
    i. **C**ancer and **H**igh Pollution
    ii. Because you have to enumerate across all of the hidden variables. To avoid this, you can join probability tables and sum out the hidden variables until you have the joint probability you wish to solve. [**Confirm**]

7. Planning and scheduling
    a. Definitions and applications
        i. Planning: aims to find a sequence of legal actions to achieve a goal state from an initial state. Applications include: route finding from A → B, optimising a sequence of minimal length routes (e.g. the TSP, or the standard VRP), game AI strategies, **others…?**
        ii. Scheduling: additionally considers time, optimising a minimal ordering of job operations, each of which take a specific length of time to complete on a specific machine. Applications include: cloud computing resource allocation, dynamic job shop scheduling, rostering, **others…?**
    b. **Don't believe we've discussed STRIPS or ADL?**
    c. Progression algorithms explore the state-space from the initial state towards the goal state, avoiding loops to states it has visited before, while regression algorithms do the inverse, making steps back to the initial state wherever possible.

I'm going to answer the rest of this as if it's referencing PDDL state representations:

d. Valid/Invalid state representations:
    i. Invalid. '*there*' is a variable, so it voids the functionless constraint.
    ii. Valid, although seems to be more of an action than a state.
    iii. Valid.
    iv. Invalid. Not a conjunction - uses an 'or' operator ∨.

e. Air cargo transport
    i. Fly(A320,CHC,WLG), Unload(Books,A320,CHC)
    ii. [**Verify**]
    Action1: Fly(A320,CHC,WLG)
    Plane(A320) ^ Cargo(Books) ^ Airport(WLG) ^ Airport(CHC) ^ At(A320, WLG) ^ In(Books, A320)

Action2: Unload(Books,A320,CHC)
Plane(A320) ^ Cargo(Books) ^ Airport(WLG) ^ Airport(CHC) ^ At(Books, CHC)

f. Vehicle routing problem
    i. Infeasible
    R1 does not end at the depot
    ii. Infeasible
    Node is E is not visited
    iii. Infeasible
    R2 exceeds vehicle capacity assumption: any fluents that are not mentioned are
    f
    iv. mFeasible

8. Other topics
    a. 5Vs?
        i. Volume: Tb/Gb of data.
        ii. Variability: inconsistency in the dataset.
        iii. Veracity: trustworthiness of the data.
        iv. Velocity: the speed at which the data moves or comes in.
        v. Variety: different types of data
        Per my comment on the same question in the 2018 exam, I think this is the same as 'value' on the pie chart in Meng's chart.
    b. Expert based systems, decision trees, knowledge engineering, **others…?**
    c. Supervised: multilayer regression, deep convolutional neural networks, **others…?**
    Unsupervised: restricted Boltzman machines, auto encoders, deep belief networks, **others…?**
    Applications: facial recognition at border security, autonomous vehicles, automatic feature detection & selection (e.g. in databases or star maps), **others…?**
    d. The process of finding legitimate, new, useful and understandable information/patterns in data.
    e. Real time language translation, machine-human interfaces, **others…?**

2016 Exam:

Sorry, in ED last night - not going to get through any other exams.

Get better soon qx