The questions were done in separate pdf's (as they would be in a real exam). They're merged into one for submission on kudos. Unfortunately this has messed up the page counting.

At this point, the main area which I feel needs work is answering past questions since I'm finding a recurring theme of "I'm confused as to what question is asking me to do" rather than "I don't know how to answer this question" if that makes sense. However, since MLRWD is fairly new and doesn't have many past questions I'm not too sure how to address this.

The first question went well. I finished with enough time to review and refine. The main "I'm not sure exactly what they want" was the diagram. I ended up drawing a Moore Machine.

I found the second question very awkward. There were quite a few parts on which I was confused as to what the question wanted. Looking back I understand them now (except the first part of (a)(i) which still looks bizarre).

Harry Langford
hjel2@cam.ac.uk

(a) Let A be the transition matrix. a_{ij} is the probability that the next state will be state j given that the current state is state i.

A is a 5×5 matrix (since there are three hidden states and a start and end state).

$$A = \begin{pmatrix} a_{SS} & a_{SF} & a_{SL_1} & a_{SL_2} & a_{SE} \\ a_{FS} & a_{FF} & a_{FL_1} & a_{FL_2} & a_{FE} \\ a_{L_1S} & a_{L_1F} & a_{L_1L_1} & a_{L_1L_2} & a_{L_1E} \\ a_{L_2S} & a_{L_2F} & a_{L_2L_1} & a_{L_1L_2} & a_{L_2E} \\ a_{ES} & a_{EF} & a_{EL_1} & a_{EL_2} & a_{EE} \end{pmatrix}$$

We can estimate a_{ij} by the formula:

$$a_{ij} = \frac{Count(i \to j)}{Count(i)}$$

The sum of each row in the transiiton matrix is 1 since every state must transition to some state.

By definition, since S is the start state and only occurs at the start of a sequence, the probability of a state transitioning to $S = 0$ for any state. Note also that the probability of E transitioning into any other state is also zero (the HMM would terminate when the state became E but for the purposes of this question I will mark the probability of E transitioning to E as 1 and any other state as 0).

Let B be the emission matrix. b_{ij} is the probability that hidden state i emits symbol j.

B is a 5×8 matrix – since there are 5 hidden states (including the two special states for start and end) and 8 symbols in the emission alphabet (one for each dice roll and two special ones [k_s and k_e] for start and end).

$$B = \begin{pmatrix} b_{Sk_s} & b_{S1} & b_{S2} & b_{S3} & b_{S4} & b_{S5} & b_{S6} & b_{Sk_e} \\ b_{Fk_s} & b_{F1} & b_{F2} & b_{F3} & b_{F4} & b_{F5} & b_{F6} & b_{Fk_e} \\ b_{L_1k_s} & b_{L_11} & b_{L_12} & b_{L_1L_3} & b_{L_14} & b_{L_15} & b_{L_16} & b_{L_1k_e} \\ b_{L_2k_s} & b_{L_21} & b_{L_22} & b_{L_13} & b_{L_24} & b_{L_25} & b_{L_26} * b_{L_2k_e} & \\ b_{Ek_s} & b_{E1} & b_{E2} & b_{E3} & b_{E4} & b_{E5} & b_{E6} & b_{Ek_e} \end{pmatrix}$$

We can estimate b_{ij} by the formula:

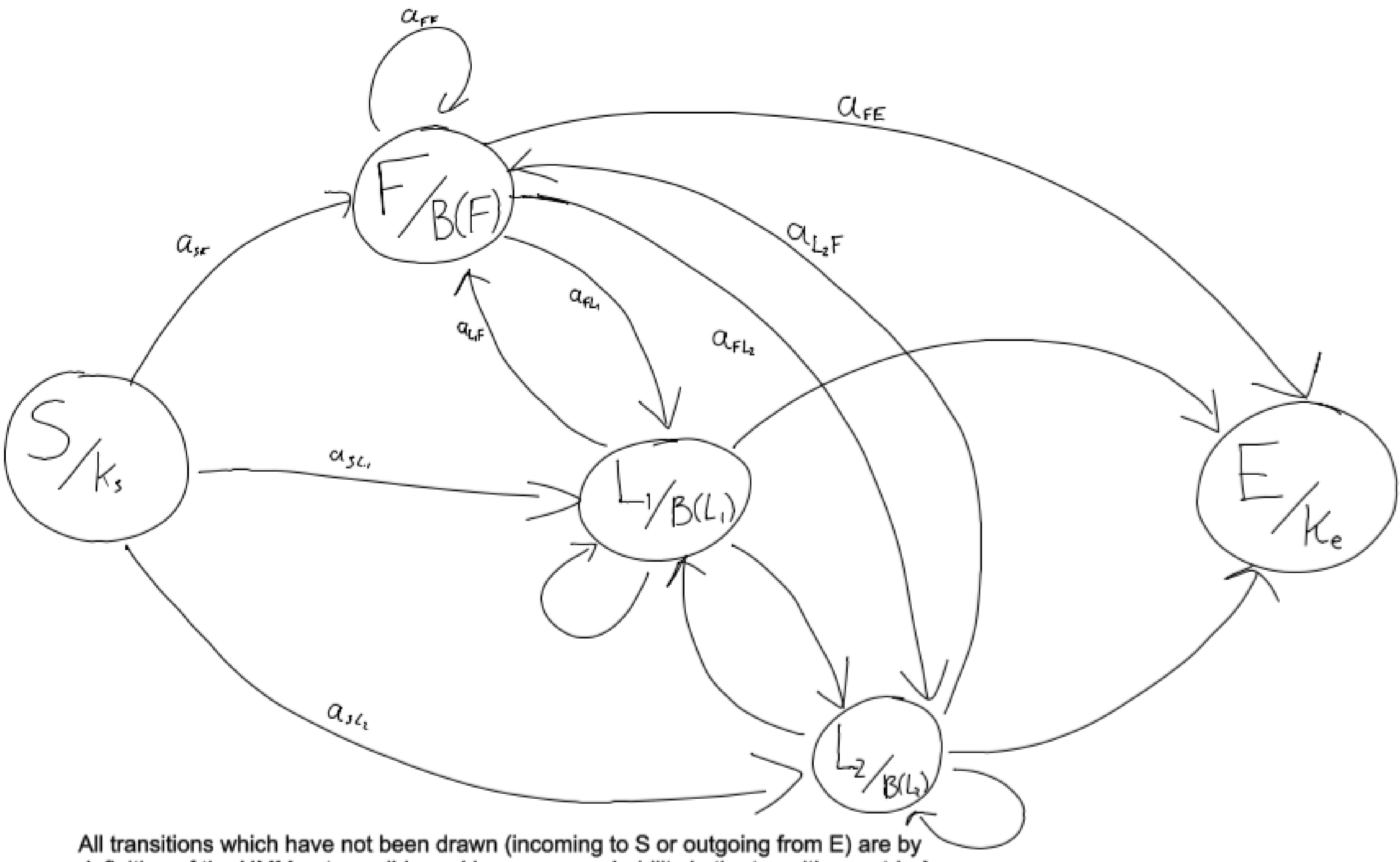
$$b_{ij} = \frac{Count(i, j)}{Count(i)}$$

Note that since by definition, k_s is the state emitted by the start state; b_{Si} for $i \neq k_s$ will be 0 and $b_{Sk_s} = 1$. The same statement holds for the end state E and the end symbol k_e.

The sum of each row in the emission matrix sums to one – since every state is guaranteed to emit some symbol.

(b)

Need to add emissions as well.
"different types of probability involved".



All transitions which have not been drawn (incoming to S or outgoing from E) are by definition of the HMM not possible and have zero probability in the transition matrix A

(c) (a) The probability a_{FL_2} is equal to zero.

Setting this probability to zero means that it is impossible for the HMM to model a switch directly from F to L_2. The HMM would now model this. (if no smoothing used)

(b) For all hidden states s, $s \neq F \iff a_{sE} = 0$.

IE the probability of transitioning from any state which is not F to the end state E is zero.

This model will ensure that every sequence ends on F – as the croupier does.

(c) $a_{L_2L_2} = 0$ × MORE than twice : twice is allowed.

IE the probability of transitioning from L_2 to L_2 is zero. So the HMM cannot model it happening and the HMM now models the new behavior appropriately.

We would have to maintain the invariant that the outgoing probabilities of L_2 sum to 1.

(d) This can't be modelled by the HMM †. By definition of the HMM, the next state is only dependent on the previous state – and not on the emissions. So the emission of the previous state can have no impact on the next state. So the HMM cannot model this behavior.

†with the exception of the case where one dice d always rolls a 6 – in this case $a_{dd} = 0$.

Harry Langford
hjel2@cam.ac.uk

(a) (a) We're using a Naïve Bayes classifier. So the approach we're using is a smoothed probabilistic generative multiclass classifier.

white?

The features I would use are whether each pixel is black or white. Given that the majority of all pixels will be black, I will model the presence of black pixels. The features are the colour of each of the individual pixels: so there are $150 \times 200 = 30000$ features. With this many features and any reasonably-sized training dataset, we will have a very high uncertainty on the probabilities of each feature given a class and each feature will be very uninformative.

The probability of an image i being in a class c

$$\log P(c|i) = \log P(c) + \sum_{w_i \in i} \lg P(w_i|c)$$

Note that we are using log-probabilities for numerical stability (else all probabilities would tend to zero with an unusably high uncertainty rapidly).

Unfortunately, the probabilities are not independent – so the Naïve Bayes assumption is broken since the probability of one pixel being black massively increases the probability of one of it's neighbours being black.

(b) The training data is the set of pixels from each of the hand-drawn images. It was collected by human annotators who annotated each picture according to what it represented.

The models parameters were estiamted according to the following formula:

$$P(c) = \frac{Count(c)}{\sum_{c_i \in C} Count(c_i)}$$

$$P(w_i|c) = \frac{Count(w_i, c) + 1}{Count_{w_i \in V}(w_i) + |V|}$$

I'm not sure what smoothing achieves here.

Where i is the image, C is the set of classes (house, tent, boat and igloo) and V is the set of all pixels.

(b) We should granulate the images – reducing the resolution by merging several adjacent pixels into one pixel (this pixel being black if 3 or more constituent pixels are black). This will greatly reduce the uncertainty on the data. Currently the probability of any individual pixel being black is so low that it's got a very high uncertainty and fairly informative. If we were to merge 4 adjacent pixels and turn the images into a 75×100 pixel images (now only having 7500 features rather than 300000 we would be able to get much better predictions). Further improvements could involve a higher degree of merging however we would need to experimentally test to find out the optimal degree of merging and what the feature should be (ie I suggest the resultant pixel should be black if 3 or more of the subpixels are black – but maybe 4 or more or 2 or more would be more appropriate).

This should improve results since it massively decreases the uncertainty on probabilities. Good idea

(c) (a) Door or no Door, Sea or no Sea, lots of Curved lines or few Curved lines (what constitutes lots of curved lines would require testing), Roof or no Roof.

Igloos and tent's don't have doors, only boats and houses can have the sea (although houses are far less likely to have the sea, while tents can not have the sea at all), houses and tents are less likely than igloos or boats to have curved lines while boats and igloos are likely to have more curved lines and only houses and boats will have rooves.

I think these features would be discriminative and would be possible to predict automatically.

These features would allow us to discriminate between the classes fairly well.

- Door or no Door

  Houses and boats have doors so this feature would be discriminative. This would also be relatively easy to predict automatically since doors have a fairly uniform shape (square with a circle inside it).

- Sea or no Sea

  Almost all drawings of boats would have the sea while very few of any other category would have the sea. This feature would be very good at discriminating boats and very prevelant while still being moderatelly easy to predict automatically.

- Lots of curved lines or few curved lines.

  Igloos have a very high proportion of curved lines while the other classes have significantly fewer curved lines. This feature would be very discriminative for igloos – although it would require testing to find out an appropriate boundary for this. This would also be a prevelant feature.

Good features.

- Roof or no roof

  Houses and boats have rooves while the other classes don't usually. This would be discriminative and easy to make automatically due to the amount of shading on rooves.

(b) We could incorporate the features described above as features in our pixel classifier. Our classifier would now have 7504 features – four of which are high level. → would this make a difference? 4 extra features is not a lot (if they are equally weighted).

The high level features would have a very low uncertainty compared to the pixels – for example tents cannot be on the sea so anything on the sea would not be a tent. Our Naïve Bayes classifier is (should be) smoothed and so would move probability mass from higher probability features to this impossible feature. We could solve this by decreasing the smoothing for the high level features to account for this (so they'd still be smoothed but significantly less so – for example add-one smoothing for the pixel values and add-0.001 smoothing for the high-level features).