CCS Problem Set 2
Due: 9 May @ 11.55pm

Your writeups should include text and figures. Many problems can be solved either analytically or with code: unless specified otherwise, you may do whichever one you wish (and need not do both!). If you code something up, you should always include the (commented) code in your write-up in addition to reporting the solution. We have written the problem set assuming that you are coding in R; if you would like to use a different language, please contact Wai Keen first. Writeups should be submitted as pdfs; LaTeX is preferable but not obligatory. Questions labeled as thought questions may be attempted even if you couldn't your code in other parts to work.

In this problem set, you have suddenly found yourself in the world of Harry Potter. You are attending Hogwarts, the wizarding school. (Note: you don't actually need to know anything about Harry Potter to do this problem set). Not being a wizard, you decide to use your skills from CCS to learn things about the world.

1. **Your first battle** (10 marks)

   On your first day you run into an evil wizard and get into a fight. After a short time, you determine that he only ever uses two spells: *Aguamenti* (A) and the *Bat-Bogey Hex* (B). You have also figured out the transition probabilities between each of the spells he uses. These are shown in the matrix below:

	A	B
A	0.1	0.5
B	0.9	0.5

   (a) This is a Markov chain, which means that it has the property of limited horizon. What does this mean for you, as your are fighting your battle with the evil wizard?

   (b) What is the state space of this Markov chain?

   (c) What is the stationary distribution of this Markov chain? Which spell does your opponent use most often?

   (d) Write some R code that will generate a sequence of 100 moves from this Markov chain. [Hint: you may find the `sample()` function useful.] Include your R code with your writeup. What is the number of times *Aguamenti* is used? The *Bat-Bogey Hex*? Does this agree with your calculations in part (c) above?

2. **Food at Hogwarts** (25 marks)

   One of your immediate concerns is to figure out the weird food they eat at Hogwarts. In particular, you really want to know what different types of food there are. You have noticed that many of the desserts can be classified according to two dimensions: dimension A is SWEETNESS (0=not sweet at all; 1 = very sweet) while dimension B is CONSISTENCY (0=soft and squishy; 1 = very hard). Since you haven't yet made any friends and don't know who you can trust, you decide to first see how a Rational Model of Categorisation (RMC) would classify the different desserts.

   The files called `runRMC.R` and `classificationHelpers.R`, found in the same place as this problem set, will be useful for this problem. Note that you should not use files of similar names that were provided previously along with the lectures; these are modified versions that are more helpful for this problem set.

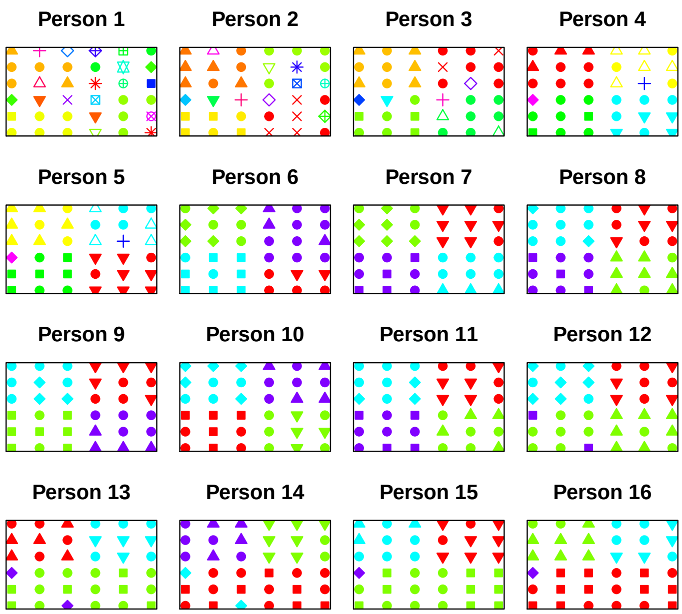
   The function `runRMC()`, found at the bottom of the `runRMC.R` file, generates a feature matrix called `features` that contains 36 desserts, which vary systematically on each of the two dimensions. On each dimension the lowest value is 0.15, the highest is 0.9, and the items are separated by values of 0.15. The code in `runRMC()` assigns half of the items to a random label chosen from a set of 36 possible labels (so most of them tend to have different labels, but this is not guaranteed). It then runs the RMC on that dataset with the CRP parameter α given by `alphaval`.

   (a) Your first step is to investigate how changing the prior on α changes the classification of desserts. Set $\alpha = 0.1$ and run it four times. Show the resulting four plots and include your R code with your writeup. Now run it four times for $\alpha = 10$ and show those plots. Based on these plots, what does α do?

   (b) You now want to investigate how you would expect the desserts to be classified after the concepts have evolved for a while. Write R code that will run an iterated learning chain of length 9, where the first "person" in the chain is the same as in part (a); that is, it is the RMC with half of the items given a random label. Give each subsequent person in the chain a random subset of half of the labels learned during that run. Run one chain with $\alpha = 0.1$ and one with $\alpha = 10$. Show each of the "people" in each of the chains in a figure, and include your R code with your writeup.

(c) Are the results of the chains in part (b) different for different α? How do these results compare with the prediction that iterated learning chains should converge to $p(h)$? [Hint: Remember that our α value reflects the prior probability, since it is the parameter for the Chinese Restaurant Process prior over the number of clusters].

(d) **Thought question.** You finally make 16 friends and decide to run an iterated learning experiment with them. As with a standard iterated learning experiment, each person's labels serve as the next person's input. For the first person you just make up random labels for 18 of the random dessert items in your space, as the code in the original `runRMC()` file did. The results are shown below: different labels are shown with different colours. What does this experiment indicate about your friends' prior beliefs about how many categories there are (i.e., their true α values)? Consider the iterated learning experiment presented in lecture in which world structure was varied. Are your results consistent or inconsistent with the CONTROL condition of that experiment? Why or why not?



(e) Of course, in real life the set of desserts isn't uniformly distributed around SWEETNESS×CONSISTENCY space as in your fake data so far. You therefore decide to carefully observe the desserts that are served at Hogwarts over the next 36 days. You carefully noted the SWEETNESS and CONSISTENCY features of each dessert but forgot to write down the names of the desserts. The results of this observation can be found in `desserts.RData`. Run one iterated learning chain of length 9 for $\alpha = 0.1$ and one for $\alpha = 10$ on this data, where the initial labels are assigned as before (i.e., half of the items are assigned to a random label chosen from a set of 36 possible ones). How do your results compare to part (b)? What do they reveal about how the structure of events in the world interacts with prior biases?

(f) **Thought question.** Suppose there was a tight bottleneck between each learner in the chain, such that each one saw only a handful of data points (say, five desserts with five different labels; each learner might see different desserts, although in all cases the desserts would be selected from the set of possible desserts). Would you still expect there to be a strong effect of different α over time? Would you expect there to be a strong effect of the structure of the world – the "true" underlying distribution of desserts? Why or why not?

3. **Kidnapping in the wizarding world** (35 marks)

You come up with a devious plan that involves kidnapping Draco Malfoy (the son of one of the evil wizards) and holding him hostage. Unfortunately, you haven't actually been able to kidnap him; he's too good at evading your clutches. However, you are very observant and you have written down many of the sentences Draco has said in the past. These are found in the `draco.txt` corpus. Your goal is to fake a ransom note as if it were written by Draco by using the information you gather from this corpus.

(a) Your first step is to calculate the unigram and bigram statistics from `draco.txt`. Run `getbigramtallies()` on the corpus (this function is found in the `ngramfunctions.R` file). Report the unigram and bigram plots created using the `plotngrams()` function. Do these have the same power law structure as the examples from lecture? Would you expect them to? Why or why not?

(b) Report all of the unigrams and bigrams said three or more times. Are these qualitatively like the more frequent unigrams and bigrams in the corpora from lecture? What do they reveal about Draco, if anything?

(c) The file `harry.txt` contains many of the sentences spoken by Harry Potter. Use the `calculateoverlap()` function, found in the `ngramfunctions.R` file, to report on how many unigrams and bigrams are shared between their files. Do these numbers surprise you?

(d) Another, possibly more principled or revealing way of calculating overlap between corpora, is to compare only the high-frequency n-grams. Consider, therefore, the unigrams from each corpus that occur more than two times. Report which ones are common to *both* Harry and Draco and how many there are. Report also on which ones were only spoken by Harry or only spoken by Draco. What does this reflect about the nature of the differences and similarities between the two? (Again, you don't need to know anything about Harry Potter to answer these questions; use the information in the corpora to answer).

(e) Let's not forget your purpose in playing around with these models – you want to fake a ransom note! Write a program that will take a set of conditional bigram probabilities and use it to generate sentences where the transition probabilities between words follows those probabilities. For the first word of the corpus, choose randomly from the set of words that start sentences (i.e., that follow the end-of-corpus symbol of $), proportional to their probability at that point. [Hint: the code for this program is similar in concept to what you did in Problem 1(d)]. Generate 10 sentences from Draco, 10 sentences from Harry, and then make it so the program generates 10 sentences from one of them at random (so you don't know who it was). Based just on the sentences can you guess who is represented twice? What does this reveal about the distinctness of Draco and Harry?

(f) Write a new function called `getlaplacebigramtallies()` which takes as an argument two corpora, A and B, and then calculates the unigram and bigram tallies for corpus A based on new vocabulary inferred from corpus B. It should also take as an argument the variable `lambda`, which corresponds to the λ of Lidstone's Law discussed in lecture. A wrapper for this function can be found in `ngramfunctions.R`. Your function should calculate and return both the MLE and the smoothed tallies. [Note that much of the function will be similar to `getbigramtallies()`; you just have to modify it where appropriate]. Include your R code in your writeup.

Run this function with `draco.txt` as the first file and `harry.txt` as the second. Show the results of this run in a table formatted like the data on slide 20 of lecture 17. Compare the results you get between MLE and $\lambda = 1$ for both raw and conditional probabilities, for all bigram counts. Are these results consistent with lecture?

(g) Based on the smoothed probabilities you calculated in part (f), generate one new sentence. Is anything different from part (e)? Why or why not?

4. **Learning about Hogwarts** (10 marks)

As you spend more time in the wizarding world, you have learned more and more about the way it works. You have found that overhypothesis models are sometimes useful for assisting you in this. [Note that none of this questions in this part require any coding; they are all thought questions.]

(a) The school you are at contains many rooms. These rooms are often filled with mysterious items. The table below shows the rooms you explored one day. (These are exhaustive lists; they were very odd rooms!)

Room	Contents
A	14 silver snuffboxes, 2 exploding snap card games, 1 gobstone, and 1 hand of glory
B	21 enchanted coins, 5 remembralls, 3 foe glasses, 1 sneakoscope, 1 time turner, and 1 Quidditch ball
C	8 books and 1 quill
D	16 owl features, 3 cauldrons, 1 rusted locket, and 1 talking snake

You now enter a new room, but only have a chance to glimpse 3 large cabinets before the door swings shut. How many different kinds of items do you think there are in the room, and why? Can this learning be captured by any of the overhypothesis models presented in class in lectures 11, 12, or 13? If so, which model(s), and how? If not, why not? Can the RMC capture it? Why or why not?

(b) Hogwarts is a school with four houses; every student is placed in one of them, and lives there for the duration of their school years. You have noticed that students who belong to each of these houses have certain characteristics. For instance, every day during exam week all students in Hogwarts were tested on a test of loyalty. Each student in Hufflepuff scored very highly on the test (between 8 and 10) on every day. All of the students in the other three houses varied considerably: it wouldn't be unusual for one student to earn a 2 one day and a 9 the next. A similar pattern was seen for the test of ambition, but on that test it was the Slytherin students who uniformly did well, while the students in the other three houses varied (even within students). For the test of courage, the Gryffindors all did well every day, but the students in the other three houses varied (even within students). Finally, for the test of intelligence, the Ravenclaws always did well every day, but the students in the other houses varied (even within student). You have managed to sneak into the headmaster's office and procure a copy of the exam grades yourself; they are found in the file `exam.csv`. [Note that you don't need this file to do the exercise below; it's just there to clarify the situation if the verbal description here was confusing.]

How might the overhypothesis model in Lecture 11 capture the nature of the houses? In particular, please clearly specify: (i) what corresponds to the different kinds? (ii) within a kind, what corresponds to the categories? (iii) within a category, what corresponds to the individual data points? (iv) what are the features of the model? (v) what are reasonable values of α and β for each kind and each feature? Note that you do not need to code anything here, so the α and β values are approximate; use them to illustrate your understanding of the model. You can illustrate your β values with pictures or with numbers, whichever you prefer and can describe clearly.