CSE 512: Homework 5 Due Nov. 19

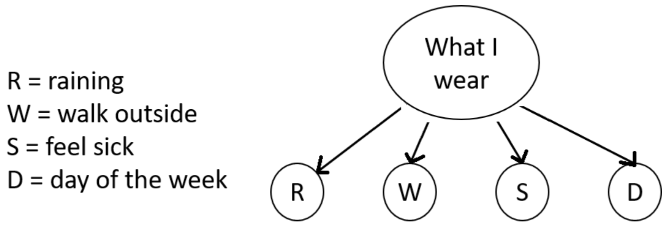
1. **Directed graphical models and probability inference** I have 4 tops: a red sweater, a blue T-shirt, a green hoodie, and a white tank top. I need your help to decide what to wear.

   (a) **(1 pts)** I decide what to wear based on four factors:

   - if it's raining
   - if I want to take a walk outside
   - if I feel sick
   - the day of the week it is

   Using the Naive Bayes assumption, draw a graphical model that indicates how I will make a decision of what to wear each day.

   Ans.

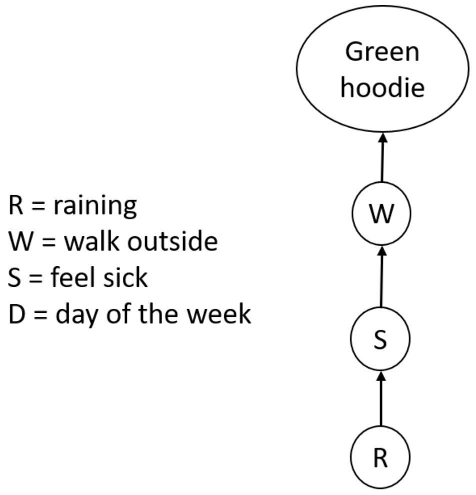


   (b) **(1 pts)** To be more specific,

   - I only wear the green hoodie when I walk outside, regardless of all other factors.
   - If I feel sick, I will walk outside 10% of the time. If I feel well, I will walk outside 60% of the time.
   - When it rains, I feel sick 70% of the time; otherwise, I feel sick 15% of the time.

   Draw a corresponding graphical model for determining whether I wear a green hoodie. Given that it is raining, infer the probability that I am wearing a green hoodie.

   Ans.



$$\begin{aligned}
\mathbf{Pr}(\text{sick} = S|\text{rain} = 1) &= \begin{cases} 0.7 & \text{if } S = 1 \\ 0.3 & \text{if } S = 0 \end{cases} \\
\mathbf{Pr}(\text{walk} = W|\text{rain} = 1) &= \mathbf{Pr}(\text{walk} = W|\text{sick} = 1)\mathbf{Pr}(\text{sick} = 1|\text{rain} = 1) \\
&\quad + \mathbf{Pr}(\text{walk} = W|\text{sick} = 0)\mathbf{Pr}(\text{sick} = 0|\text{rain} = 1) \\
&= \begin{cases} 0.1 \cdot 0.7 + 0.6 \cdot 0.3 = 0.25 & \text{if } W = 1 \\ 0.9 \cdot 0.7 + 0.4 \cdot 0.3 = 0.75 & \text{if } W = 0 \end{cases} \\
\mathbf{Pr}(\text{green hoodie} = G|\text{rain} = 1) &= \mathbf{Pr}(\text{green hoodie} = G|walk = 1)\mathbf{Pr}(\text{walk} = 1|\text{rain} = 1) \\
&\quad + \mathbf{Pr}(\text{green hoodie} = G|\text{walk} = 0)\mathbf{Pr}(\text{walk} = 0|\text{rain} = 1) \\
&= \begin{cases} 1 \cdot 0.25 + 0 \cdot 0.75 & \text{if } G = 0 \\ 0 \cdot 0.25 + 1 \cdot 0.75 & \text{if } G = 1 \end{cases}
\end{aligned}$$

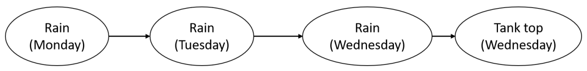
So, the probability that I will wear a green hoodie is 25%.

(c) **(1 pts)** The probability that I wear a tank top, independently of all the other clothes, is 75% if it's raining and 25% if it's not raining.

- Today is Monday and it is raining.
- The probability that it will rain, given that the previous day rained, is 70%. The probability that it will rain, given that the previous day did not rain, is 10%.

Draw a graphical model predicting whether I will wear a tank top on Wednesday, and calculate this probability.

Ans.



Denote M, T, W the random variables for the event of raining on Monday, Tuesday, and Wednesday. Denote U for the event that I wear the white tank top.

$$\begin{aligned}
\mathbf{Pr}(T|M = 1) &= \begin{cases} 0.7 & \text{if } T = 1 \\ 0.3 & \text{if } T = 0 \end{cases} \\
\mathbf{Pr}(W|M = 1) &= \mathbf{Pr}(W|T = 1)\mathbf{Pr}(T = 1|M = 1) \\
&\quad + \mathbf{Pr}(W|T = 0)\mathbf{Pr}(T = 0|M = 1) \\
&= \begin{cases} 0.7 \cdot 0.7 + 0.1 \cdot 0.3 = 0.52 & \text{if } W = 1 \\ 0.3 \cdot 0.7 + 0.9 \cdot 0.3 = 0.48 & \text{if } W = 0 \end{cases} \\
\mathbf{Pr}(U|M = 1) &= \mathbf{Pr}(U|W = 1)\mathbf{Pr}(W = 1|M = 1) + \mathbf{Pr}(U|W = 0)\mathbf{Pr}(W = 0|M = 1) \\
&= \begin{cases} 0.75 \cdot 0.52 + 0.25 \cdot 0.48 = 0.51 & \text{if } U = 0 \\ 0.25 \cdot 0.52 + 0.75 \cdot 0.48 = 0.49 & \text{if } U = 1 \end{cases}
\end{aligned}$$

So, the probability that I will wear a white tank top is 51%.

2. **Clustering(3 pts)**

- Open the python notebook `mnist_dimred_clustering.ipynb`, and load the MNIST data by running the first cell. Don't change the way I've formatted it, or the checksums won't work.
- Fill in the function for determining the Euclidean distance between any sample point and the entire training data. From the print function, you should get a value of `160239119987.1912`.
- Using this function, code up a K-Means method, using an initialization of

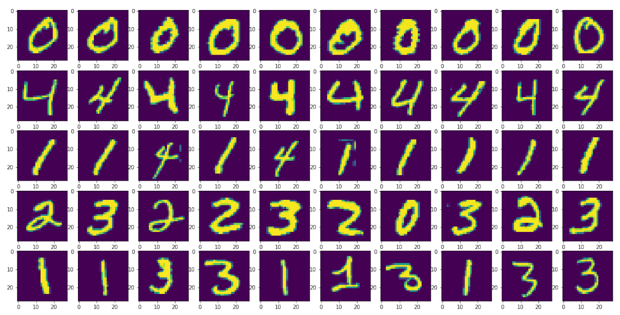
```
mu = X[:K,:]
```

where $K = 10$ is the number of clusters. Run this for 10 iterations.

- Define the *purity* of a set $\mathcal{S}$ as the following fraction:

$$\text{class\_purity}(\mathcal{S}) = \max_i \frac{|\{y_j = i, j \in \mathcal{S}\}|}{|\mathcal{S}|}$$

that is, it is the size of the largest same-label subset of $\mathcal{S}$ divided by the total size of $\mathcal{S}$. Report the purity of your clustering method after running k-means *on only the first 25 datapoints*, up to 3 digits after the decimal. Plot also the clustering result.



Ans.
purity = 0.713

- Now, put the above section in a nice function, as we will use it again and again to evaluate future embeddings.
- At this point, we should also de-mean the data, as it will improve the embedding performances:

```
X = X - np.outer(np.ones(X.shape[0]),np.mean(X,axis=0)).
```

  – Using `np.linalg.svd`, implement PCA, and reduce the data dimension to $d = 10$, 100, and 500. Re-run K-Means to get new class memberships. (You may want to enact the option `full_matrices = False` when computing the SVD.)
    Ans.
    * For 10 dimensions, purity = 0.713
    * For 100 dimensions, purity = 0.713
    * For 500 dimensions, purity = 0.713

  – Use random hashing (as promoted by the JL lemma) and reduce the feature dimension to 10,100,500 dimensions, and cluster with MNIST.
    Ans. Answers may vary, but not by that much.
    * For 10 dimensions, purity = 0.430
    * For 100 dimensions, purity = 0.747
    * For 500 dimensions, purity = 0.700

  – Use Isomap, LLE, or spectral embeddings to reduce dimension to 10,100,500, and cluster with MNIST. You only need to do one of the three, but you do need to code it up from scratch. When constructing the nearest neighbor graph, use Euclidean distance, and pick a threshold that seems reasonable to you. (Can be distance thresholding or nearest number of neighbors.) For isomap, you may find `scipy.sparse.csgraph.shortest_path` useful, if you can create the appropriate CSR formatted adjacency matrix. For all other hyperparameters, make a choice that seems reasonable to you.
    Ans. Answers may vary, and even trends might differ based on hyperparameter choices. As long as the comments in the next parts are consistent with the results, we will accept the answer. For me, these were the results I was able to obtain.

Isomap

* For 10 dimensions, purity = 0.277
* For 100 dimensions, purity = 0.278
* For 500 dimensions, purity = 0.276

LLE

* For 10 dimensions, purity = 0.824
* For 100 dimensions, purity = 0.530
* For 500 dimensions, purity = 0.652

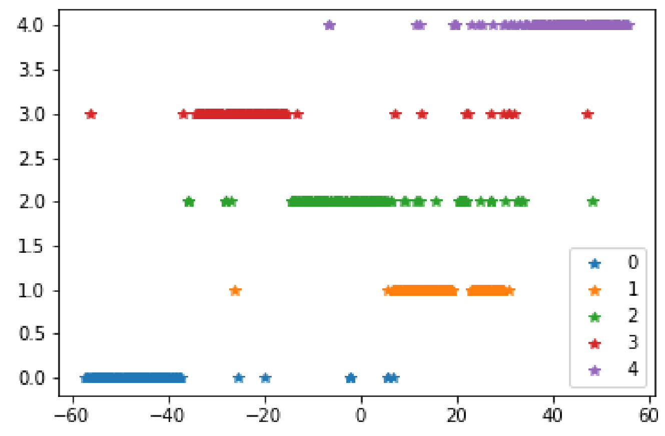
Spectral embedding

* For 10 dimensions, purity = 0.630
* For 100 dimensions, purity = 0.718
* For 500 dimensions, purity = 0.744

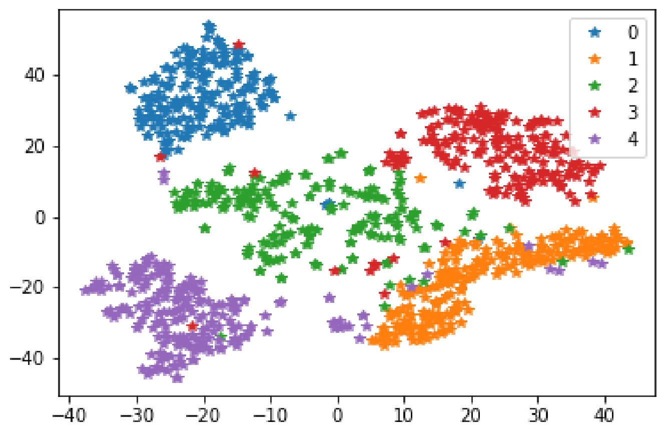
– Use sklearn's t-SNE to to reduce dimension to 1,2,3, and cluster with MNIST. In addition, plot something that shows the clustering effect, either in 1,2, or 3-D space. Keep most default hypermarameter settings to make your answers comparable.
Ans.
Answers may vary, but not by much.

* For 1 dimension, purity = 0.758
* For 2 dimensions, purity = 0.942
* For 3 dimensions, purity = 0.849

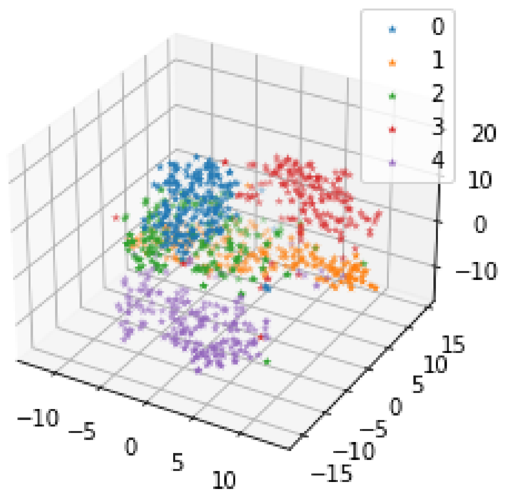
Here are some fun visualizations



1D



2D



3D

- For these three situations, return the purity for each method and dimension size.
- Comment on the effect of the different dimensionality reduction schemes in the MNIST clustering task. What are the tradeoffs, in terms of performance and computational complexity?

  Ans. It's actually kind of amazing how much you can push down the dimensionality and still get good kmeans performance, which is to say that clustering structure is mostly preserved! In my implementation, the two methods that didn't do so well were random hashing and Isomap, but the rest were able to achieve near state-of-the art performance at 10 or 2 dimensions (T-SNE) even!

  Graders, accept any answer that sounds true / consistent with given data.

3. **Hidden Markov Model spellchecker (4 pts)** In this exercise we will make a spell-checker using a HMM. To do this, download `alice_nlp_release.ipynb` and follow the instructions.

   - Read through the first two blocks to get an idea of what the task is. The idea is to go through the corrupted corpus, identify words which have probably been corrupted, and correct them probabilistically.
   - In the 4th box, fill in the functions to construct the word probabilities (weighted frequencies in uncorrupted corpus) and transition matrix (which gives Pr(word | prev word)). If done correctly, the lines printed out should read

     ```
     prob. of "alice" 0.014548615047424706
     prob. of "queen" 0.002569625514869818
     prob. of "chapter" 0.0009069266523069947
     ```

     with smoothing

     ```
     prob. of "the alice" 0.00025406504065040653
     prob. of "the queen" 0.016514227642276422
     prob. of "the chapter" 0.012957317073170731
     ```

     no smoothing

     ```
     prob. of "the alice" 0.0
     prob. of "the queen" 0.03968253968253968
     prob. of "the chapter" 0.0
     prob. of "the hatter" 0.031135531135531136
     ```

   - In the 5th box, fill in the function for computing the emission probability. The first 10 words closest to Alice should be

     ```
     ['abide', 'alice', 'above', 'voice', 'alive', 'twice', 'thick', 'dance', 'stick', 'prize']
     ```

   - Construct and run your Hidden Markov Model spell checker using the functions computed for the prior probabilities, emission probabilities, and transition probabilities. List some words whose spelling was corrected correctly, and some examples where the spell-correcter did not work as expected. Report the recovery rate of the "fixed" corpus.

     Ans. (4 pts) Recovery rate of corrupted corpus: 0.759.

     Recovery rate of fixed corpus, with smoothed transition matrix: 0.187. (Yikes! There was a bug in my code originally. This is not what I had hoped the recovery rate would be, but it turned out that smoothing was a terrible idea, since most of the values in the transition matrix should have been 0.)

     Recovery rate of fixed corpus without smoothing: 0.924.

     (Note: this is the only part of the problem that needs to be scored.)

Challenge!

1. **Correlated mixture of sequential experts.**

   I want to buy a yacht, but I'm not sure if it's a good idea given the economy. So, I decide to question m consultants. Each consultant has more-or-less the same qualifications, and they come in one at a time.

   The first consultant comes in my office. I ask, "Should I buy a yacht?" She says yes with probability p.

   On her way out the building, she meets the second consultant. They chat briefly, and she leaves, he comes, and the process repeats. Each time, I ask the consultant if I should buy a yacht, and receive "yes" with probability p; each time, the consultant chats briefly with the next consultant. *However*, the answers now are *not* i.i.d., but rather each expert's answer is correlated with the answer of experts he/she chatted with in the lobby.

   At the end of the day, I have met with m consultants. I will make a decision whether to buy a yacht based on majority rule. We will now calculate the probability that I will buy a yacht.

   (a) We model the answer of each consultant as $Y_i = 1$ if the ith consultant recommended "yes", and $Y_i = -1$ otherwise. Show that if distribution

   $$\mathbf{Pr}(Y_1 = 1) = p, \qquad \mathbf{Pr}(Y_i = 1 | Y_{i-1} = 1) = c + p - cp, \qquad \mathbf{Pr}(Y_i = -1 | Y_{i-1} = -1) = cp - p + 1$$

   then $\mathbf{Pr}(Y_i = 1) = p$ for all i.

   Ans. It suffices to show that $\mathbf{Pr}(Y_2 = 1) = p$, since everything else will happen recursively. Using Law of Total Probability,

   $$\begin{aligned}
   \mathbf{Pr}(Y_2 = 1) &= \mathbf{Pr}(Y_2 = 1|Y_1 = 1)\mathbf{Pr}(Y_1 = 1) + \mathbf{Pr}(Y_2 = 1|Y_1 = -1)\mathbf{Pr}(Y_1 = -1) \\
   &= \mathbf{Pr}(Y_2 = 1|Y_1 = 1)\mathbf{Pr}(Y_1 = 1) + (1 - \mathbf{Pr}(Y_2 = -1|Y_1 = -1))(1 - \mathbf{Pr}(Y_1 = 1)) \\
   &= (c + p - cp)p + (1 - (cp - p + 1))(1 - p) \\
   &= p.
   \end{aligned}$$

   (b) The Pearson correlation coefficient between a random variable U and V can be expressed as

   $$\mathbf{corr}(U, V) = \frac{\mathbb{E}[UV] - \mathbb{E}[U]\mathbb{E}[V]}{\sqrt{\mathbf{var}(U)\mathbf{var}(V)}}.$$

   Show that the correlations between each pair of sequential experts $\mathbf{corr}(Y_i, Y_{i-1}) = c$,
   Hint: Go ahead and use a symbolic calculator, like WolframAlpha, to simplify messy expressions.
   Ans. Taking each piece one at a time,

   $$\begin{aligned}
   \mathbb{E}[Y_i Y_{i-1}] &= \mathbf{Pr}(Y_i = 1, Y_{i-1} = 1) + \mathbf{Pr}(Y_i = -1, Y_{i-1} = -1) - \mathbf{Pr}(Y_i = 1, Y_{i-1} = -1) - \mathbf{Pr}(Y_i = -1, Y_{i-1} = 1) \\
   &= \mathbf{Pr}(Y_i = 1|Y_{i-1} = 1)\mathbf{Pr}(Y_{i-1} = 1) + \mathbf{Pr}(Y_i = -1|Y_{i-1} = -1)\mathbf{Pr}(Y_{i-1} = -1) \\
   &\qquad - \mathbf{Pr}(Y_i = 1|Y_{i-1} = -1)\mathbf{Pr}(Y_{i-1} = -1) - \mathbf{Pr}(Y_i = -1|Y_{i-1} = 1)\mathbf{Pr}(Y_{i-1} = 1) \\
   &= (c + p - cp)p + (cp - p + 1)(1 - p) - (1 - (cp - p + 1))(1 - p) - (1 - (c + p - cp))p \\
   &= 1 + 4p(c(1 - p) + p - 1)
   \end{aligned}$$

   Then

   $$\begin{aligned}
   \mathbb{E}[Y_i] &= p - (1 - p) = 2p - 1 \\
   \mathbf{var}(Y_i) &= \mathbb{E}[\underbrace{Y_i^2}_{=1}] - (\mathbb{E}[Y_i])^2 = 1 - (2p - 1)^2
   \end{aligned}$$

   Putting it all together,

   $$\mathbf{corr}(Y_i, Y_{i-1}) = \frac{1 + 4p(c(1 - p) + p - 1) - (2p - 1)^2}{1 - (2p - 1)^2} = c.$$

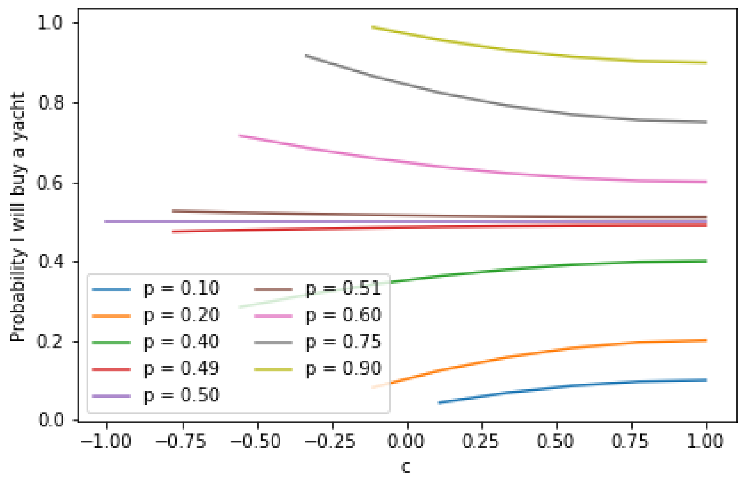
(c) For $m = 3$, what is the probability that I will buy a yacht, in terms of $P_{11} = \mathbf{Pr}(Y_i = 1|Y_{i-1} = 1)$, $P_{00} = \mathbf{Pr}(Y_i = -1|Y_{i-1} = -1)$ and p?

Ans. For $m = 3$, I will buy a yacht if any two or all 3 advisers advise for it. That is,

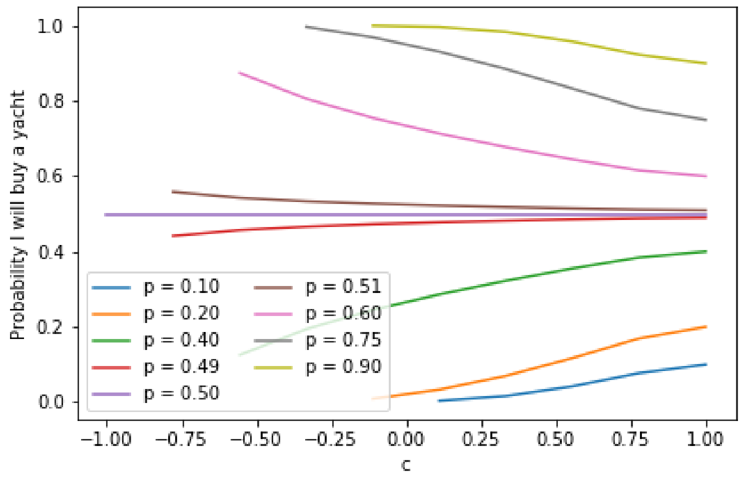
$$\begin{aligned}
\mathbf{Pr}(\text{yacht is bought}) &= \mathbf{Pr}(Y_1 = 1, Y_2 = 1) + \mathbf{Pr}(Y_1 = -1, Y_2 = 1, Y_3 = 1) + \mathbf{Pr}(Y_1 = 1, Y_2 = -1, Y_3 = 1) \\
&= \mathbf{Pr}(Y_2 = 1|Y_1 = 1)\mathbf{Pr}(Y_1 = 1) + \mathbf{Pr}(Y_3 = 1|Y_2 = 1)\mathbf{Pr}(Y_2 = 1|Y_1 = -1)\mathbf{Pr}(Y_1 = -1) \\
&\quad + \mathbf{Pr}(Y_3 = 1|Y_2 = -1)\mathbf{Pr}(Y_2 = -1|Y_1 = 1)\mathbf{Pr}(Y_1 = 1) \\
&= P_{11}p + P_{11}(1 - P_{00})(1 - p) + (1 - P_{00})(1 - P_{11})p
\end{aligned}$$

(d) **Code.** Compute exactly the probability that I will buy a yacht, for $m = 10$, in Python or MATLAB. Generate a plot that shows the probability that I will buy a yacht, sweeping $c \in [-1, 1]$. (Note that there are cases where p and c are an infeasible pair, and can be detected when probabilities are not in the range (0,1)–these cases should not be plotted.) Do this for several interesting values of p. Comment on how the decision changes as a function of c, p, and m.

Ans. Coding this up requires 10 nested for loops, but still I was able to produce a fairly smooth plot waiting only maybe 1 minute total. What's interesting here is to see how, when c is low, the experts have the ability to "gravitate" toward 1 if $p > 0.5$ and 0 otherwise; but as c approaches 1, the "gravitation speed" seems to reduce, converging to the original probability p. With 10 vs 3 experts, it seems like the "separation" lasts a bit longer (for larger values of c), but still exhibits a similar trend. This is consistent with what we would expect: more experts is better (concentrates on a more definitive choice), but higher correlation leads to less useful information.



3 experts

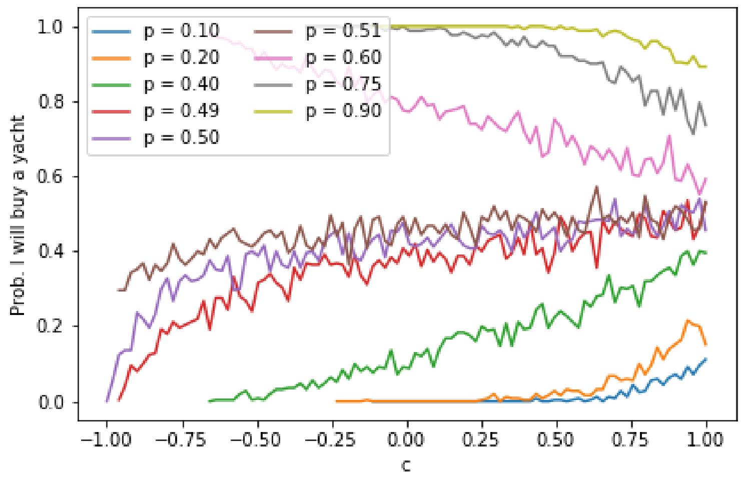


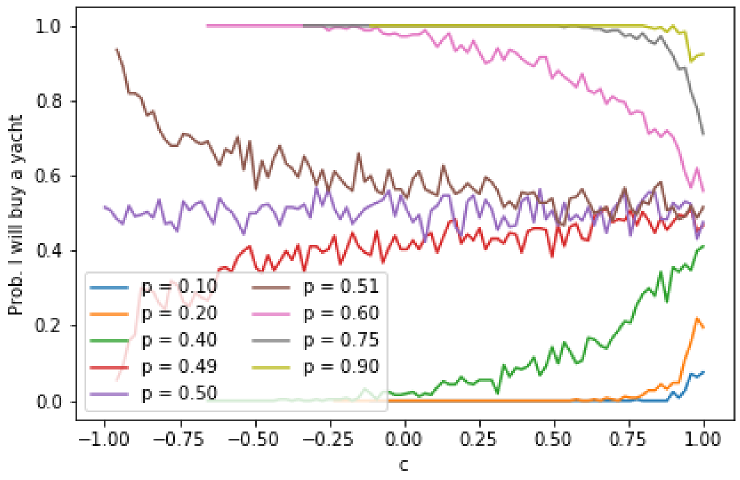
10 experts

(e) Simulate the sequential advisers, and give a numerical estimate of what my decision will be if $m = 25$ and $m = 100$. Generate a similar plot, using these numerical estimates of **Pr**(I will buy a yacht). Use your own discretion to decide how many trials you need, and what values of c and p are useful.

Ans. Here are plots I generated, averaging over 250 trials. It's pretty messy, but it's clear that the trends are similar!

At this point we can also notice an interesting separation between the initial seed probabilities p: for p closer to 1 or 0, more experts allow for a more concentrated decision, even as c is close to 1. But for p closer to 0.5, more experts seem to cause more uncertainty, concentrating to 0.5 as m increases!




Prob. I will buy a yacht
c
p = 0.10
p = 0.20
p = 0.40
p = 0.49
p = 0.50
p = 0.51
p = 0.60
p = 0.75
p = 0.90