DS102 - Final Exam Practice Problems

Tuesday, Dec 10, 2019

1. For each of the following, answer true or false. **Circle T for true and F for false**. You don't need to justify your answer.

   (a) (1 point) (T / F) A classification tree with unlimited depth can perfectly represent any training dataset with 100% accuracy.

   Solution: False. Imagine the (x,y) pairs: (0,1), (0,0). The classification tree has to choose.

   (b) (1 point) (T / F) Dynamic programming is a technique that caches intermediate answers to sub-parts of a larger problem. It is most useful when there is structural redundancy in the problem subparts.

   Solution: True.

   (c) (1 point) (T / F) Sensitivity = 1 - false discovery proportion.

   Solution: False, check definitions.

   (d) (1 point) (T / F) Suppose we have samples (X_i, Y_i), where $Y_i \in \{0, 1\}$ and $X_i | Y_i \sim N(\mu_{Y_i}, 1)$. It is reasonable to use the expectation-maximization (EM) algorithm to learn μ_0, μ_1.

   Solution: False. If we know which sample belongs to which Gaussian there is no point in using EM.

   (e) (1 point) (T / F) The UCB algorithm has linear regret.

   Solution: False, it has sublinear regret.

   (f) (1 point) (T / F) In control theory we can sometimes approximate a nonlinear system with a linear system and still get good performance.

   Solution: True, the cartpole problem from lecture is one such example.

   (g) (1 point) (T / F) The optimal Q-function is a mapping from states to values. It can be interpreted as the expected discounted reward given that we start at the input state and that we use an optimal policy.

   Solution: False, the Q-function is a function of state-action pairs.

(h) (1 point) (T / F) The deeper the decision tree, the lower the error on the test set.

> **Solution:** False, the tree could overfit and performance could degrade.

(i) (1 point) (T / F) You can achieve sublinear regret with an explore-then-exploit strategy if you know the smallest gap between the expected reward of the optimal arm and the expected reward of sub-optimal arms.

> **Solution:** True, this was a homework problem.

(j) (1 point) (T / F) The Hoeffding bound can be used to compute a confidence interval for a bounded random variable.

> **Solution:** True, this is what we have used the Hoeffding bound to do.

2. Suppose your GSIs want to conduct a survey about whether the midterm exam was difficult or not. To protect the students' privacy, they deploy a randomized response strategy.

   A student's true opinion is denoted by T, and takes values in $\{0, 1\}$ (0 being "midterm was not difficult", and 1 being "midterm was difficult"). Denote the student's response by A.

   (a) Suppose the two possible responses are 1 ("midterm was difficult"), and 0 ("midterm was not difficult"), i.e. $A = \{0, 1\}$. The students should respond truthfully with probability $p := \mathbb{P}(A = 0 \mid T = 0) = \mathbb{P}(A = 1 \mid T = 1)$, and lie with probability $1 - p$. Suppose we have to make the responses ϵ-differentially private, so we need:

   $$p \leq e^{\epsilon}(1 - p).$$

   We can easily ensure differential privacy by making the responses completely random (by setting $p = 1 - p = 0.5$). However, we want the results to be informative, so we want to set p such that the signal-to-noise ratio (SNR) $\frac{p}{1-p}$ is maximized. What is the value of p that maximizes $\frac{p}{1-p}$ subject to the privacy constraint, as a function of ϵ? What is the value of $\frac{p}{1-p}$?

   Solution: By the privacy constraint, we know $p \leq e^{\epsilon}(1 - p)$. Therefore, the maximum possible value of $\frac{p}{1-p}$ is e^{ϵ}. By setting $p = e^{\epsilon}(1-p)$, we can now solve for p, because we know:

   $$e^{\epsilon}(1 - p) + 1 - p = 1.$$

   We get $p = \frac{e^{\epsilon}}{1+e^{\epsilon}}$.

   (b) Now suppose the respondents are fine with a slightly weaker privacy guarantee, (ϵ, δ)-differential privacy, for fixed $\epsilon, \delta > 0$. We will add two more possible responses - "midterm was *really* not difficult!" (0^*) and "midterm was *really* difficult!" (1^*). If $T = 0$, then with probability δ the answer should be $A = 0^*$. Similarly, if $T = 1$, then with probability δ the answer should be $A = 1^*$. In other words, with probability δ the respondents should state their true opinion (slightly overemphasized).

   We want to guarantee (ϵ, δ)-differential privacy, so for each of the 4 responses $a \in \{0, 1, 0^*, 1^*\}$,

   $$\mathbb{P}(A = a \mid T = 0) \leq e^{\epsilon}\mathbb{P}(A = a \mid T = 1) + \delta, \;\; \mathbb{P}(A = a \mid T = 1) \leq e^{\epsilon}\mathbb{P}(A = a \mid T = 0) + \delta.$$

   What constraint do we have on the value of $\mathbb{P}(A = 0^* \mid T = 1)$ and $\mathbb{P}(A = 1^* \mid T = 0)$?

   Solution: These two probabilities must be 0. For example:

   $$\delta = \mathbb{P}(A = 1^* \mid T = 1) \leq e^{\epsilon}\mathbb{P}(A = 1^* \mid T = 0) + \delta,$$

   and since $e^{\epsilon} > 0$, we must set $\mathbb{P}(A = 1^* \mid T = 0) = 0$.

(c) Let $p = \mathbb{P}(A = 0 \mid T = 0) = \mathbb{P}(A = 1 \mid T = 1)$. For true opinion $T = a \in \{0, 1\}$, truthful responses are now both a and a^*. Suppose we still want to maximize the SNR (ratio of probabilities of truthful responses and false responses):

$$\frac{p + \delta}{1 - (p + \delta)}.$$

What is the value of p that maximizes this ratio, as a function of ϵ and δ? Show that the SNR for such p is equal to:

$$\frac{p + \delta}{1 - (p + \delta)} = \frac{2\delta + e^\epsilon(1 - \delta)}{1 - 2\delta}.$$

Solution: The SNR is increasing in p, so its best to set p to be as large as possible. On the other hand, we must have $e^\epsilon(1 - (p + \delta)) + \delta - p \geq 0$. Since $e^\epsilon(1 - (p + \delta)) + \delta - p$ is decreasing in p, by setting it exactly equal to 0 we find the maximum possible admissible p. Solving for p gives:

$$p = \frac{e^\epsilon(1 - \delta) + \delta}{1 + e^\epsilon}, \quad 1 - (p + \delta) = \frac{1 - 2\delta}{1 + e^\epsilon}.$$

Plugging these expressions into the SNR expression gives:

$$\frac{2\delta + e^\epsilon(1 - \delta)}{1 - 2\delta}.$$

(d) Is the SNR of part (c) greater than the SNR of part (a), or less? If you don't think you got the correct answer in part (a), you can get partial credit by taking a guess and giving a correct intuitive justification.

Solution: The SNR of part (c) is greater, which makes sense because the target privacy constraint is weaker.
More formally, the function $\frac{2\delta+e^\epsilon(1-\delta)}{1-2\delta}$ is increasing in δ (which can be seen by taking the derivative, which is equal to $\frac{2+e^\epsilon}{(1-2\delta)^2} > 0$).

3. (a) Let T be an observed test statistic. Recall that the p-value is defined as:

$$P = \mathbb{P}(T_0 \geq T \mid T),$$

where T_0 is an imaginary test statistic drawn from the null distribution. Prove that, if T is also drawn from the null distribution, P is uniform on $[0,1]$. You can assume that the CDF of null test statistics $F(\cdot)$ is invertible.

Solution: Let F denote the CDF of T_0. For all $u \in [0,1]$:

$$\begin{aligned}
\mathbb{P}(P \leq u) &= \mathbb{P}(\mathbb{P}(T_0 \geq T \mid T) \leq u) \\
&= \mathbb{P}(1 - F(T) \leq u) \\
&= \mathbb{P}(F(T) \geq 1 - u) \\
&= \mathbb{P}(T \geq F^{-1}(1-u)) \\
&= 1 - F(F^{-1}(1-u)) \\
&= u,
\end{aligned}$$

which corresponds to the CDF of the uniform distribution.

(b) Suppose that we have one null p-value P_1, and another non-null p-value P_2, which is independent from P_1 and distributed according to:

$$P_2 = \begin{cases} 0, & \text{with probability } 1 - \delta \\ 1, & \text{with probability } \delta. \end{cases}$$

Suppose that our decision rule is simply to proclaim a discovery if $P_i \leq \alpha$, $i \in \{1, 2\}$. What is the probability of making at least one false discovery (i.e. FWER)? What is the FDR? For which values of δ is the FDR less than or equal to α? What is the expected sensitivity (i.e. power)?

Solution: The FWER is equal to the probability of rejecting P_1, which is α by uniformity of nulls.

The FDR is equal to:

$$\text{FDR} = \alpha(1-\delta) \cdot \frac{1}{2} + \alpha\delta \cdot 1 = \frac{\alpha}{2} + \frac{\alpha\delta}{2}.$$

This is less than or equal to α for all $\delta \in [0,1]$.

The power is equal to the probability of discovering P_2, which is $1 - \delta$.

(c) How does the decision rule from part (b) applied to P_1, P_2 differ from applying the Benjamini-Hochberg method to P_1, P_2? Try to be as precise as possible. For example, you can identify events when the set of discoveries of the two methods would be the same, and events when this is not the case. Also make sure to say what the set of discoveries on each of these events is.

Solution: When $P_2 = 0$ and $P_1 \leq \alpha$, both P_1, P_2 get discovered by both methods. When $P_2 = 0$ and $P_1 > \alpha$, only P_2 gets discovered by both methods. When $P_1 = 1$ and $P_2 \leq \frac{\alpha}{2}$, only P_2 is discovered by both methods, however if P_2 is between $\alpha/2$ and α, BH doesn't discover it, while the procedure from part (a) does. When $P_1 = 1$ and $P_2 > \alpha$, neither procedure makes any discoveries.

4. A historian has asked for your help in designing an experiment to assess the average age homes in two cities. From historical records, you know that in the first city, all homes are at most a years old, and in the second city, homes are similarly at most b years old. The historian will query homes in a city, with each query returning the age of a single home of the historian's choice. The historian would like to know how many homes to query in each city, given only the research budget to query n total homes.

   (a) Assuming the historian spends n_1 i.i.d. trials to sample the first population and n_2 to sample the second, use the Hoeffding inequality to derive confidence intervals for the mean of each population with confidence level $1 - \alpha$.
   Recall that the Hoeffding inequalities state that if Z is the average of independent bounded random variables Z_i where Z_i is bounded in $[a, b]$, and $E(Z) = \mu$, then

$$P(Z - \mu \geq t) \;\leq\; \exp\big(-\frac{2nt^2}{(b-a)^2}\big)$$

   and

$$P(Z - \mu \leq -t) \;\leq\; \exp\big(-\frac{2nt^2}{(b-a)^2}\big)\ .$$

   Solution: Letting $\bar{\mu}_1$ denote the sample average of samples taken from population 1, the confidence interval for the first population will be $\bar{\mu}_1 + -t$, where δ satisfies $P(Z - \mu \geq t) \leq \alpha/2$ and $P(Z - \mu \leq -t) \leq \alpha/2$. Inverting the upper bound for these quantities we find

$$\exp\left(-\frac{2n_1t^2}{(a-0)^2}\right) = \alpha/2$$

$$t = \sqrt{\frac{a^2}{2n_1}\log(2/\alpha)}$$

   So that the confidence intervals are

$$\left(\bar{\mu}_1 - \sqrt{\frac{a^2}{2n_1}\log(2/\alpha)}, \bar{\mu}_1 + \sqrt{\frac{a^2}{2n_1}\log(2/\alpha)}\right)$$

$$\left(\bar{\mu}_2 - \sqrt{\frac{b^2}{2n_2}\log(2/\alpha)}, \bar{\mu}_2 + \sqrt{\frac{b^2}{2n_2}\log(2/\alpha)}\right)$$

   (b) The historian would like to design their sample collection so as to minimize the width of the largest confidence interval. Prove that the resulting confidence intervals have the same width (assume that you can sample at non-integer n_1, n_2). *Hint: consider a proof by contradiction.*

Solution: Suppose the optimal allocation is such that the widths are not the same, and without loss of generality the width of set 1 is larger. Since the confidence widths are decreasing and continuous in n_1 and n_2, we can find an ϵ so that $n_1 - \epsilon$ and $n_2 + \epsilon$ each have width smaller than the original width for set 1.

(c) In order to achieve this design (that minimizes the maximum confidence interval width), what fraction of the historian's n observations should be spent on set 1?

Solution: To set the intervals equal, we would have:

$$\sqrt{\frac{a^2}{2n_1}\log(2/\alpha)} = \sqrt{\frac{b^2}{2n_2}\log(2/\alpha)}$$

$$n_2 = \frac{b^2}{a^2}n_1$$

Since $n_1 + n_2 = n$, we have that

$$n_1\left(\frac{a^2}{a^2} + \frac{b^2}{a^2}\right) = n$$

$$n_1 = \frac{a^2}{a^2+b^2}n$$

The experiment design should spend $\frac{a^2}{a^2+b^2}$ fraction of the total observations on set 1.

Drinks	Yes	1.3
Coffee?	No	2.1

Table 1: Average number of academic clubs attended for coffee drinkers and non coffee drinkers.

		Is Undergrad?	
		Yes	No
Drinks	Yes	2.7	1.1
Coffee?	No	2.3	0.9

Table 2: Average number of academic clubs attended per subgroup.

5. You'd like to test the following causal hypothesis: drinking coffee causes students to join more academic clubs. We'll consider two ways of testing this hypothesis.

   (a) First, suppose you have access to a database that someone else collected, which has responses to a survey that asked undergraduate and graduate students (1) whether they drink coffee, (2) what type of degree they were pursuing (undergraduate/graduate), and (3) the numbers of academic clubs they are a part of. The summary of the results is in Tables 1 and 2.

   The average number of clubs for coffee drinkers in Table 2 is higher than that of non coffee drinkers for both subgroups, yet the results in Table 1 indicate that coffee drinkers join more clubs, on average.

      i. Give one possible reason for this discrepancy.

      Solution: If there are many more graduate students in the survey than undergraduates, this may happen.

      ii. Is reporting the differences in Table 2 sufficient to answer the original question? Why or why not?

      Solution: It is not; there may always be other sources of variation in an observational study that we haven't accounted for.

   (b) Being unsatisfied with your analysis from the provided dataset, you decide to collect your own data via a randomized control trial (RCT). You have a random sample of 200 Berkeley students: 100 of whom are undergraduates (drawn randomly with each student drawn equal chance from the undergraduate population) and 100 of whom are graduate students (also drawn such that each graduate student at Berkeley has equal chance of appearing in the sample).

   To half of your sample of 200 students, you will assign treatment (drink coffee), and to the other half, you will assign control (drink no coffee). For the sake of the problem, ignore any ethical qualms about denying students their coffee. You decide to stratify your sample, so that you pick 50 graduate students and 50 undergraduate students at random from your samples to form the control group, and the rest form the treatment group.

i. Assume that we choose the 50 undergraduate students in the control group by first enumerating all $\binom{100}{50}$ ways of dividing the students, and picking one uniformly at random (and similarly for the graduate students). Let $y_i(0)$ denote the potential outcome of individual i under control (the number of clubs they would join if they don't drink coffee), and $y_i(1)$ the potential outcome under treatment (the number of clubs they would join if they drink coffee). Show that for each group, the expectation of the within group treatment effect estimate is the actual treatment effect for that group. That is, show that

$$\mathbb{E}\left[\frac{1}{50}\sum_{i\in\text{treatment}} y_i(1) - \frac{1}{50}\sum_{j\in\text{control}} y_j(0)\right] = \tau$$

where for any individual i in your subgroup, $\mathbb{E}\left[y_i(1) - y_i(0)\right] = \tau$.

Solution:

$$\mathbb{E}\left[\frac{1}{50}\sum_{i\in\text{treatment}} y_i(1) - \frac{1}{50}\sum_{j\in\text{control}} y_j(0)\right]$$
$$= \mathbb{E}\left[\frac{1}{50}\sum_{i=1}^{100} \mathbb{I}[i \text{ assigned treatment }] \cdot y_i(1) - \mathbb{I}[i \text{ assigned control }] \cdot y_i(0)\right]$$

Since treatment is independent of the outcomes and by linearity of expectation,

$$= \frac{1}{50}\sum_{i=1}^{100} \mathbb{P}[i \text{ assigned treatment }] \cdot \mathbb{E}[y_i(1)] - \mathbb{P}[i \text{ assigned control }] \cdot \mathbb{E}[y_i(0)]$$

Since each individual has a 1/2 probability of being in the control or treatment group, this is

$$\begin{aligned}
&= \frac{1}{50}\sum_{i=1}^{100} 1/2 \cdot \mathbb{E}[y_i(1)] - 1/2 \cdot \mathbb{E}[y_i(0)] \\
&= \frac{1}{100}\sum_{i=1}^{100} \mathbb{E}[y_i(1)] - \mathbb{E}[y_i(0)] \\
&= \mathbb{E}[y_i(1) - y_i(0)] = \tau
\end{aligned}$$

ii. Alternatively, we could have combined the graduate students and undergraduate students together into a sample of 200, then divided into control and treatment group by picking one of the $\binom{200}{100}$ such groupings uniformly at random, and use as our estimate of the treatment effects :

$$\mathbb{E}\left[\frac{1}{100}\sum_{i\in\text{treatment}} y_i(1) - \frac{1}{100}\sum_{j\in\text{control}} y_j(0)\right]$$

What are the conditions on the treatment effects for each group $\tau_{\text{undergraduate}}$, τ_{graduate} for which this would be a valid experiment to test our hypothesis?

Solution: When $\tau_{\text{undergraduate}} = \tau_{\text{graduate}}$.

iii. Would we ever prefer the second approach (group everyone together, then assign treatment) to the first approach (first stratify by graduate/undergradute, then assign treatment)? Why or why not (answer in ≤ 2 sentences).

Solution: We would always want to stratify. In the case that there is in fact no difference in treatment effects, each of the individual samples will still in expectation give $\tau_{\text{undergraduate}} = \tau_{\text{graduate}}$. Further, but not expected in a typical solution: if we suspect $\tau_{\text{undergraduate}} = \tau_{\text{graduate}}$, we can concatenate the treatment and control group individuals after stratification and compute estimator for the common treatment effect.

6. (a) Describe the UCB algorithm in words, you don't need to include any specific formulas.

Solution: The UCB algorithm can be described as follows

1. For each arm compute the upper confidence bound of the estimated mean of the reward for each arm
2. Find the arm with the maximum upper confidence bound
3. Pull that arm and observe a reward
4. Update the upper confidence bound to incorporate the arm that was just pulled.

(b) What should you set the original upper confidence bound of each arm in UCB to? Why?

Solution: You should set the initial upper confidence bound of each arm to be infinity so as to guarantee each arm will be pulled at least once.

(c) An alternative bandit algorithm simply uses the empirical mean of the rewards of each arm. The algorithm will pick the arm with the highest empirical mean. Explain why UCB is preferred to this algorithm.

Solution: UCB is able to tradeoff exploration and exploitation since arms that have been underexplored will have very wide confidence intervals. On the other hand the empirical mean algorithm has no concept of exploration and will instead always exploit by picking the best estimate it has causing a potentially good arm to be unexplored.

(d) Consider a setting where we have two arms, assume we are using the algorithm from Part c and that we have properly initialized the empirical means. Describe a scenario in which the arm with the highest mean will only get pulled once.

Solution: Consider the setting where arm 1 is the optimal arm where its rewards are distributed according to $\mathcal{N}(1,1)$, arm 2 had rewards distributed according to $\mathcal{N}(0.5,1)$. We first pull each arm once and get very unlucky. We observe a rewards of -2 for arm 1 and a reward of 2 for arm 2. On subsequent pulls we keep pulling arm 2 and the empirical mean of arm 2's rewards never drop below -1 until the algorithm terminates.

7. After graduating, you go and find a job at WebFlix, a new streaming website. WebFlix has released a new TV show and noticed that changing the thumbnail for the TV show can increase the number of people that click on and watch the show. Your boss has asked you to find the image that gets the most clicks out of a set of K images. Since you can see when a user looks at an image and whether or not they click on it, you decide to use a Bandit algorithm.

   You model each image $i = 1, ..., K$ as an arm and assume that each image has a reward distribution which is $Bernouilli(p_a)$ which describes how likely the user is to click on the image. Therefore, with probability p_a the user clicks on the TV show, and you receive a reward of 1. You decide to use Thompson Sampling because that way you can make use of prior information that you have collected.

   (a) Given that the rewards are Bernouilli, what of the following distributions is the natural choice for the prior distribution over the means of the arms?

$$p_a \sim Gamma(\alpha, \beta) : P(x|\alpha, \beta) \propto x^{\alpha-1}e^{-\beta x} \quad : \quad x \in [0, \infty)$$
$$p_a \sim Beta(\alpha, \beta) : P(x|\alpha, \beta) \propto x^{\alpha-1}(1-x)^{\beta-1} \quad : \quad x \in [0, 1]$$
$$p_a \sim Normal(\alpha, \beta) : P(x|\alpha, \beta) \propto e^{-\frac{(x-\alpha)^2}{2\beta^2}} \quad : \quad x \in (-\infty, \infty)$$
$$p_a \sim Exponential(\alpha) : P(x|\alpha) \propto \alpha e^{-\alpha x} \quad : \quad x \in [0, \infty)$$
$$p_a \sim Uniform(0, 1) : P(x) \propto 1 \quad : \quad x \in [0, 1]$$

   Solution: Beta Distribution is the natural choice because they are conjugate priors. Note that uniform distribution would not work because they would not allow you to use prior information.

   (b) Given your choice of prior in the previous part, give an expression for the posterior distribution over the parameter of an arm after having seen one sample:

$$Pr(p_a|r_a)$$

   You do not need to compute the normalizing constant.

   Solution: Using a Beta prior with parameters α and β the update rule is:

$$Pr(p_a|r_a) = Beta(p_a; \alpha + r_a, \beta + 1 - r_a)$$

   (c) Given your choice of prior and the update rule, you have asked WebFlix's software engineers to code up a function to choose the image. The software engineers have not taken DS102 though, and need some help. They have some unfinished code and a package that lets you sample from any of the distributions in part *a* by calling $distribution(\alpha, \beta)$. Therefore if your posterior is (for example) Normal with mean α and variance β, you could call $Normal(\alpha, \beta)$ to generate one sample from the distribution.

Help the software engineers by filling in the missing code (#TODO a. and #TODO b.). For each of the TODO's write a sentence to describe what your code is doing.

```
def get_posterior_sample(rewards,prior_alpha,prior_beta):
    """
    Return a sample from the posterior of a given arm, given your choice of prior.

    Inputs:

    rewards            - a list containing the samples received from pulling the arm so far/
                        (a list of 0's and 1's)
    prior_alpha        - the α for prior for the current arm.
    prior_beta         - the β for prior for the current arm.

    Returns:
    sample             - a sample from the posterior distribution of the arm.
    """

    sample=#TODO a.

    return sample


def TS_pull_arm(rewards,prior_alphas,prior_betas):
    """
    Implement the choice of arm for the Thompson Sampling Algorithm when the arms are bernouilli and the prior is
    the distribution you have chosen.

    Inputs:

    rewards            - a list of K lists. Each of the K lists holds the samples received from pulling each arm
                         so far.
    prior_alphas       - a list of all the α's for the arms for the type of prior you have chosen.
    prior_betas        - a list of all the β's for the arms for the type of prior you have chosen.

    Returns:
    arm                - integer representing the arm that the Thompson Sampling algorithm chooses.
    """

    posterior_samples=[]

    for arm in range(K):

        sample=get_posterior_sample(rewards[arm],prior_alphas[arm],prior_betas[arm])

        posterior_samples.append(sample)

    arm=#TODO b.

    return arm
```

Figure 1: Fill in the code above for question 7.

Solution: The lines of code are:

#TODO a. is to sample from the posterior

$$\begin{aligned}
&\alpha = prior_alpha + np.sum(rewards) \\
&\beta = prior_beta + len(rewards) - np.sum(rewards) \\
&sample = Beta(\alpha, \beta)
\end{aligned}$$

#TODO b. chooses the maximum sample.

$$arm = np.argmax(posterior\_samples)$$

8. In this question, we will look at the control of simple linear dynamical systems of the form:

$$x_{k+1} = Ax_k + Bu_k$$

Where $x_k \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$,$B \in \mathbb{R}^{d\times 1}$,$u \in \mathbb{R}$, $k \in \mathbb{N}$.

(a) In this first part we will assume that there is no control input u, so that the dynamics are given by:

$$x_{k+1} = Ax_k$$

Derive an explicit formula of x_k given only A and x_0, the initial position.

Solution:

$$\begin{aligned} x_k &= Ax_{k-1} \\ &= A(Ax_{k-2}) \\ &\vdots \\ &= A^k x_0 \end{aligned}$$

(b) Suppose that $d = 2$ and the matrix A is given by:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix},$$

and $x_0 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ plot the value of x_k for $k = 0, 1, 2, 3$.

(c) Given the setting of Part b, compute $\lim_{k\to\infty} \|x_k\|_2$. What will happen to each coordinate of x_k as $k \to \infty$?

Solution: $\|x_k\|_2 \to \infty$ with the first coordinate going to infinity and the second coordinate going to 0.

(d) Suppose that $d = 1$ with $A = 2$, $B = 4$, $x_0 = 10$. Can you chose a constant value c such that setting $u_k = c$ for all k will result in $x_k \to 0$ as $k \to \infty$. If yes, what is this value of c if no explain why not.

Solution: No, since a positive value of c will push the input toward ∞ while a negative value will push the input toward $-\infty$.

(e) Given the setting of Part d can you design a function $f(x_k) = cx_k$ such that setting $u_k = f(x_k)$ for all k will result in $x_k \to 0$ as $k \to \infty$? If yes, what is this value of c if no explain why not.

9. Fill in the code boxes in Figure 2. The function names should be the names of the algorithms each function implements, and should have the form "x_sampling." If the boxes aren't big enough write your answer to the right with an arrow to the box. (Each answer should be one line.)

```
import numpy as np
def draw_from_generating_function(...):
    # draws a sample from the generating distribution g
    ...
    return sample

def q(x):
    # pdf corresponding to the generating function at x
    ...
    return pdf

def p(x):
    # pdf corresponding to the target function at x
    ...
    return pdf

def                    (n):

    sample = []
    weights = []
    for i in range(n):

        x = draw_from_generating_distribution()

        sample.append(x)

        w =

        weights.append(w)

    return sample, weights

def g(x):
    # generating function at x
    ...
    return fxn_value

def f(x):
    # target function at x such that f(x) < g(x) for all x
    ...
    return fxn_value

def                    (n):

    sample = []

    while len(sample) < n:

        x = draw_from_generating_distribution()

        w =

        if np.random.uniform(0,1) < w :
            sample.append(x)

    return sample
```

Figure 2: Fill in the code above for question 9.