IE531: Algorithms for Data Analytics
Midterm Exam
Spring, 2018

Instructions

1. **A (correct) answer that is not accompanied by an appropriate justification will not be given any credit. Make sure you answer all questions.**
2. **Try to finish on time.**

Instructions

1. (*25 points*) (**Rank (and Determinant) of a Matrix**) Consider the matrix **A** shown below, where $x \in \mathcal{R}$ is a real-valued variable.

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 5 & 7 & 0 \\ 2 & 4 & 6 & 8 & 0 \\ 0 & 0 & 0 & 0 & x \end{pmatrix}$$

(a) (*2 points*) What is the value of x that makes $Rank(\mathbf{A})$ as small as possible?

$x = 0$. Catch me after class, if you need help with this.

(b) (*2 points*) What is the smallest possible value for $Rank(\mathbf{A})$?

$Rank(\mathbf{A}) \geq 2$. Catch me after class, if you need help with this.

(c) (*2 points*) What is the value of x that makes $Rank(\mathbf{A})$ as large as possible?

$x \neq 0$. Catch me after class, if you need help with this.

(d) (*2 points*) What is the largest possible value for $Rank(\mathbf{A})$?

$Rank(\mathbf{A}) \leq 3$. Catch me after class, if you need help with this.

(e) (*2 points*) What is the value of x that ensures there is a solution to the equation $\mathbf{Ax} = \mathbf{y}$, where $\mathbf{y} = (1\ 1\ 0)^T$?

Has a solution for any value of x, as $Rank(\mathbf{A}) \geq 2$, and $\mathbf{y}$ has a zero for the third-entry.

(f) (*2 points*) What is the value of x that ensures there is a solution to the equation $\mathbf{Ax} = \mathbf{y}$, where $\mathbf{y} = (1\ 0\ 1)^T$?

Has a solution for any value of $x \neq 0$, as $Rank(\mathbf{A}) = 3$ when $x \neq 0$.

(g) (*2 points*) Suppose $\mathbf{X}$ and $\mathbf{Y}$ are arbitrary matrices, is it possible for $rank(\mathbf{X}+\mathbf{Y}) > rank(\mathbf{X}) + rank(\mathbf{Y})$?

No. You can never increase the number of linearly independent columns by the process of addition (why?).

(h) (*2 points*) Suppose $\mathbf{z}$ is a $n \times 1$ column. What are the number of rows and number of columns of $\mathbf{zz}^T$? What is $rank(\mathbf{zz}^T)$?

$\mathbf{zz}^T$ is a $n \times n$ matrix, and $rank(\mathbf{zz}^T) = 1$ (Why? Because, each column of $\mathbf{zz}^T$ is a multiple of the first-column).

(i) (*2 points*) If $\mathbf{X}$ is an $n \times n$, square matrix with $rank(\mathbf{X}) < n$, what is $det(\mathbf{X})$?

$det(\mathbf{X}) = 0$. By definition, all most.

(j) (*7 points*) If $\mathbf{X}$ is an $n \times n$, square matrix, where the (i, j)-th entry, $\mathbf{X}_{i,j} = i + j$, what is $rank(\mathbf{X})$?

$rank(\mathbf{X}) = 2$. To see this, note

$$\mathbf{X} = \begin{pmatrix} 2 & 3 & 4 & \cdots & n+1 \\ 3 & 4 & 5 & \cdots & n+2 \\ 4 & 5 & 6 & \cdots & n+3 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ n+1 & n+2 & n+3 & \cdots & 2n \end{pmatrix}.$$

Since $rank(\mathbf{X})$ does not change with fundamental row- and column-operations, we can subtract the i-th column from the $(i+1)$-th column, where $1 \leq i \leq n-1$, to get the matrix

$$\mathbf{Y} = \begin{pmatrix} 2 & 1 & 1 & \cdots & 1 \\ 3 & 1 & 1 & \cdots & 1 \\ 4 & 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ n+1 & 1 & 1 & \cdots & 1 \\ & \underbrace{}_{\mathbf{X}_{\bullet,2}-\mathbf{X}_{\bullet,1}} & \underbrace{}_{\mathbf{X}_{\bullet,3}-\mathbf{X}_{\bullet,2}} & \cdots & \underbrace{}_{\mathbf{X}_{\bullet,n}-\mathbf{X}_{\bullet,n-1}} \end{pmatrix}.$$

It follows that $rank(\mathbf{X}) = rank(\mathbf{Y}) = 2$ when $n \geq 2$. If $n = 1$, then $rank(\mathbf{X}) = 1$, obviously.

To receive full-credit for this problem, make sure you give me an explanation for your answers.

2. (*25 points*) (**Generating Constrained, High-Dimensional RVs**) The *d-dimensional Unit-Cube* is defined as

$$\mathbf{C} = \{\mathbf{x} \in \mathcal{R}^d \mid 0 \leq \mathbf{x}_i \leq 1, i = 1, 2, \ldots, d\}$$

That is, $\mathbf{C}$ is a collection of d-dimensional vectors where each entry/component of the vector is in the closed-interval $[0, 1]$.

(a) (*10 points*) How would you generate points Uniformly at Random in $\mathbf{C}$? I am looking to see some pseudo-code, along with an informal justification for why your code will pick any point in $\mathbf{C}$ with equal probability.

You make d-many calls to get_uniform(). That is, x_i = get_uniform(), $i = 1, 2, \ldots, d$, and output $\mathbf{x} = (x_1\ x_2\ \cdots x_d)^T$. There is an equal-chance of any point in $\mathbf{C}$ being picked. See figure 2 for pseudo-code.

(b) (*15 points*) How would you generate points Uniformly at Random on the $((d-1)$-dimensional) *surface* of $\mathbf{C}$? I am looking to see some pseudo-code, along with an informal justification for why your code will pick any point on the $(d-1$-dimensional) surface of $\mathbf{C}$ with equal probability.

You pick any one of the d-many dimensions at random. If you picked $i \in \{1, 2, \ldots, d\}$ in this step, you let $x_i = 0$ with probability $\frac{1}{2}$, and $x_i = 1$ with probability $\frac{1}{2}$. That is, you freeze the value of x_i at either 0 or 1 with equal probability. You then make $d - 1$-calls to get_uniform() to populate the rest of the entries of the vector. This will give you uniformly distributed RV from any of the $(d-1)$-dimensional faces of $\mathbf{C}$. See figure 3 for pseudo-code. For a 3-D illustration, see figure 1.

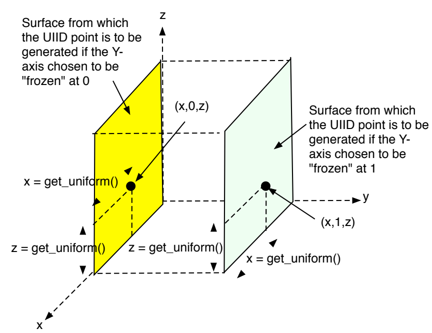

Figure 1: 3D Illustration of the process of generating UIID RVs on the $(d - 1)$-dimensional surface of the Unit Cube.

I have uploaded some C++ code, along with a Flipped-Classroom Video on Compass that implements the pseudo-code of figures 2 and 3.

Some Observations (Start)

If you understand what is meant by the term "($(d-1)$-dimensional) *surface of* $\mathbf{C}$," you can skip this factoid. You may want to think about solving problem 2b by trying your hand for the special-cases of $d = 1, 2, 3, 4$ first (this in itself will get you no points!), and then generalize it to arbitrary values of d.

A *1-dimensional Unit-Cube* is the Unit-Interval [0, 1]. There are 2-many 0-dimensional surfaces (i.e. points) on this Unit-cube, and they are the points 0 and 1.

A *2-dimensional Unit-Cube* is a square with unit-length sides. There are 4-many 1-dimensional surfaces (i.e. lines) on this Unit-cube, and they are the four lines that define the square.

A *3-dimensional Unit-Cube* is a cube with unit-length sides. There are 6-many 2-dimensional surfaces (i.e. areas) on this Unit-cube, and they are the six square-areas that define the cube.

There are 8-many 3-dimensional surfaces (i.e. volumes) in a *4-dimensional Unit-Cube*.

Some Observations (End)

RVs over the d-dimensional Unit-Cube (dimension d)

```
for i ∈ {1, 2, ..., d} do
    x_i = get_uniform().
end for
return Vector x
```

Figure 2: Pseudo-code for problem 2a.

RVs over surface d-dimensional Unit-Cube (dimension d)

```
x = get_uniform().
for i ∈ {1, 2, ..., d} do
    if (i-1)/d ≤ x < i/d then
        chosen-coordinate = i.
    end if
end for
y = get_uniform()
if y ≤ 1/2 then
    x_i = 0
else
    x_i = 1
end if
for i ∈ {1, 2, ..., d} − {chosen-coordinate} do
    x_i = get_uniform().
end for
return Vector x
```

Figure 3: Pseudo-code for problem 2b.

3. (*25 points*) Consider the following SVD of a matrix $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where

$$\underbrace{\begin{pmatrix} 4.7511 & 1.9078 & 0.9344 & 3.2316 \\ 0.1722 & 3.8276 & 2.4488 & 3.5468 \\ 2.1937 & 3.9760 & 2.2279 & 3.7734 \end{pmatrix}}_{=\mathbf{A}} = \underbrace{\begin{pmatrix} -0.5493 & -0.7744 & -0.3139 \\ -0.5379 & 0.6152 & -0.5763 \\ -0.6394 & 0.1477 & 0.7545 \end{pmatrix}}_{=\mathbf{U}} \underbrace{\begin{pmatrix} 9.8247 & 0 & 0 \\ 0 & 3.7411 & 0 \\ 0 & 0 & 0.2958 \end{pmatrix}}_{=\mathbf{D}}$$

$$\underbrace{\begin{pmatrix} -0.4179 & -0.5750 & -0.3313 & -0.6205 \\ -0.8685 & 0.3915 & 0.2973 & 0.0633 \\ 0.2178 & 0.6594 & -0.0800 & -0.7151 \end{pmatrix}}_{=\mathbf{V}^T}$$

(a) (*5 points*) What is the best-possible rank 1 approximation to $\mathbf{A}$? (**PS**: No need to carry out the multiplication; just give me the form/structure)

$$\mathbf{A}_1 = 9.8247 \times \begin{pmatrix} -0.5493 \\ -0.5379 \\ -0.6394 \end{pmatrix} \times \begin{pmatrix} -0.4179 & -0.5750 & -0.3313 & -0.6205 \end{pmatrix}$$

$$\left(= \mathbf{U} \times \begin{pmatrix} 9.8247 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \times \mathbf{V}^T \right)$$

(b) (*5 points*) What is the best-possible rank 2 approximation to $\mathbf{A}$? (**PS**: No need to carry out the multiplication; just give me the form/structure)

$$\underbrace{\mathbf{A}_1}_{\text{of prob 2b}} + 3.7411 \times \begin{pmatrix} -0.7744 \\ 0.6152 \\ 0.1477 \end{pmatrix} \times \begin{pmatrix} -0.8685 & 0.3915 & 0.2973 & 0.0633 \end{pmatrix}$$

$$\left(= \mathbf{U} \times \begin{pmatrix} 9.8247 & 0 & 0 \\ 0 & 3.7411 & 0 \\ 0 & 0 & 0 \end{pmatrix} \times \mathbf{V}^T \right)$$

(c) (*5 points*) What is the best-possible rank 3 approximation to $\mathbf{A}$? (**PS**: No need to carry out the multiplication; just give me the form/structure)

$\mathbf{A}$ is itself its best possible rank-3 approximation.

(d) (*10 points*) What is the best-possible rank 4 approximation to $\mathbf{A}$? (**PS**: No need to carry out the multiplication; just give me the form/structure)

If the "approximation matrix" $\mathbf{B}$ should be a 3×4 matrix, then there can be **no B** with rank 4 (why?).

If $\mathbf{B}$ does not <u>*have*</u> to be a 3×4 matrix, but it <u>*has*</u> to be of rank 4, such that the $\|\mathbf{B}\|_F \approx \|\mathbf{A}\|_F$, then $\mathbf{B}$ is not unique. To see this, note that for any $0 < \epsilon$,

$$\mathbf{B}(\epsilon) = \begin{pmatrix} 4.7511 & 1.9078 & 0.9344 & 3.2316 \\ 0.1722 & 3.8276 & 2.4488 & 3.5468 \\ 2.1937 & 3.9760 & 2.2279 & 3.7734 \\ 0 & 0 & 0 & \epsilon \end{pmatrix}, \mathit{rank}(\mathbf{B}(\epsilon)) = 4, \text{ and } \|\mathbf{B}(\epsilon)\|_F = \|\mathbf{A}\|_F + O(\epsilon^2)$$

Therefore, by making ϵ smaller-and-smaller $\|\mathbf{B}(\epsilon)\|$ will get closer-and-closer to $\|\mathbf{A}\|_F$. But, ϵ cannot be equal to zero (because, if it did, the $rank(\mathbf{B}(\epsilon)) \neq 4$).

Either way, there is no unique best-possible rank 4 approximation to **A**. This was the subject of the Flipped-Classroom Video with the title *Best-Rank-K-Approximation Problem: The algorithm and some subtle theoretical-issues* on Compass for Lesson 3.

While we are at it, assuming you caught the non-uniqueness argument presented above, you may think the same issue will come to haunt us when we are looking at rank 3, rank 2 or rank 1 approximations of **A**. It turns out that all the lower-rank approximation **B** are unique if the singular values of **A** are unique (i.e. there are no repeated singular-values).

4. (*25 points*) **MCMC-MH**: We need to generate identically-distributed samples $X \sim f(x)$ according to some (complicated) PDF $f(\bullet)$ that can be computed upto a proportionality constant.

Assume that for right now, you have a present value $x^{(j)}$ for $X \sim f(x)$. You have access to a known/well-understood *Proposal Distribution* $q(\bullet)$ that is conditioned on $x^{(j)}$. You get a sample $\widehat{x} \sim q(x \mid x^{(j)})$, you set $x^{(j+1)} = \widehat{x}$ with probability

$$p(\widehat{x}, x^{(j)}) = \min\left\{1, \frac{f(\widehat{x})}{f(x^{(j)})} \times \frac{q(x^{(j)} \mid \widehat{x})}{q(\widehat{x} \mid x^{(j)})}\right\}. \quad (1)$$

The sequence of (not necessarily independent) samples $\{x^{(j)}\}_{j=1}^{\infty}$ will be distributed according $f(x)$. In many cases, we will pick *Proposal Distributions* that are symmetric, that is, $q(x^{(j)} \mid \widehat{x}) = q(\widehat{x} \mid x^{(j)})$, in which case the above expression simplifies to

$$p(\widehat{x}, x^{(j)}) = \min\left\{1, \frac{f(\widehat{x})}{f(x^{(j)})}\right\}. \quad (2)$$

(a) (*7.5 points*) Show that $q(\widehat{x} \mid x^{(j)}) = N(x^{(j)}, 1)$ (i.e. it returns r.v.'s that are Normally-distributed with *mean* $x^{(j)}$ with *variance* of 1) is a *symmetric* distribution (cf. the definition/discussion above).

$\widehat{x} \sim N(x^{(j)}, 1)$ – which means we are drawing $\widehat{x}$ from a unit-normal centered around a mean of $x^{(j)}$. The probability of getting $\widehat{x}$ under this scheme is exactly the same that of getting $x^{(j)}$ from a unit-normal centered around a mean of $\widehat{x}$. See figure 4 for a graphic illustration – this means that this specific Proposal Distribution is symmetric and we can use equation 2.

(b) (*7.5 points*) Suppose $q(\widehat{x} \mid x^{(j)}) = \mathsf{Uniform}(x^{(j)}, x^{(j)} + 10)$ (i.e. it returns r.v.'s that are Uniformly-distributed between $x^{(j)}$ and $x^{(j)} + 10$). Is this *Proposal Distribution* symmetric? Explain your answer (or, you get no points!)

From figure 5 you can see that if we got $\widehat{x}$ from the Uniform-Distribution whose minimal-value is $x^{(j)}$, then it is impossible to get $x^{(j)}$ from the Uniform-Distribution whose left-most value is $\widehat{x}$. Clearly, one of the probabilities is $\frac{1}{10}$, while the other is zero. Not symmetric.

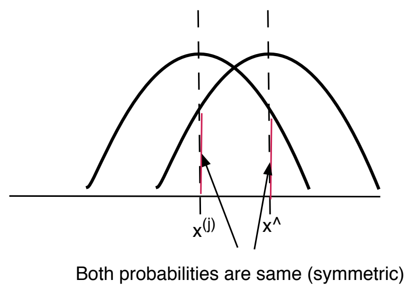

Figure 4: Graphic illustration of the symmetric nature of the Proposal Distribution of problem 4a.

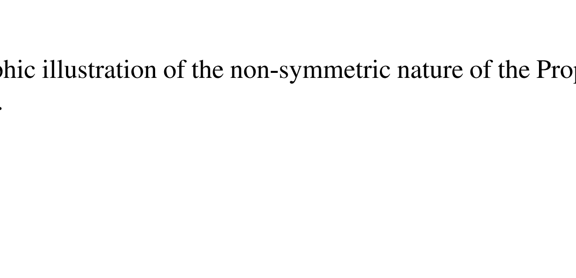

Figure 5: Graphic illustration of the non-symmetric nature of the Proposal Distribution of problem 2b.

(c) (*10 points*) As a follow-on to problem 2b, which version of the acceptance probability would you use – Equation 1? Equation 2? Explain your answer (**ps**. This is a trick-question, just saying!).

As I noted in the Post-Script (i.e. **ps**) – this is a trick-question. From the answer to problem 2b, we know the Proposal Distribution of problem 2b is not symmetric (so, clearly we cannot use equation 2). But, that does not mean you can use equation 1 either – it is obvious that $q(\widehat{x} \mid x^{(j)}) = \frac{1}{10}$ and $q(x^{(j)} \mid \widehat{x}) = 0$ (because you can never get $x^{(j)}$ from the Uniform-Distribution whose left-most value is $\widehat{x}$). This means you will always reject $\widehat{x}$. Or, the only sample you will have for eternity is $x^{(j)}$, which makes this a bad Proposal Distribution for MCMC-MH.

FYI: This was covered in great-detail from the 2:00 minute marker to the 31:00 minute marker in Lecture 17, Tuesday, March 13, 2018 on Echo360.