MATH 728D: Machine Learning
Project 2: High Dimensional Sampling and Ranking

---

1 Uniform Sampling of the Unit Ball

For this exercise, we will be sampling $B^m(r)$, the ball of radius $r = 1$ in dimensions $m = 2$ and $m = 20$. Let u and n represent uniform and normal random values, while $\mathbf{u}$ and $\mathbf{n}$ are m-vectors of uniform or normal random values. Then the "projection method" can randomly select a vector $\mathbf{x}$ from $B^m(r)$ by:

$$\mathbf{x} = (r * u)^{(1/m)} * \frac{\mathbf{n}}{||\mathbf{n}||_2}$$

1. Use the projection method to generate $n = 10,000$ pairs of random points p and q in $B^2(1)$. Compute their distance $d = ||p - q||_2$, and c, the absolute value of the cosine of the angle between them, $c = |\,(p \cdot q)/||p||_2/||q||_2\,|$. Report the average values of d and c; also report the arc-cosine of the average value of c, in radians and degrees. Is your average d near 0.90? Is your average angle near 45 degrees?

```
Table 1.1  m = 2  n = 10,000

  average d = ____________
  average c = ____________
  arccos (average c) = ______(radians)
                     = _____ (degrees)
```

2. Repeat the previous exercise, for $B^{20}(1)$.

```
Table 1.2  m = 20  n = 10,000

  average d = ____________
  average c = ____________
  arccos (average c) = ______(radians)
                     = _____ (degrees)
```

2 Sampling the Shrinking Ball in Higher Dimensions

For this exercise, we estimate the volume of $B^m(1)$ for $m = 1...8$, by generating n random samples using the following rejection method:

```
k = 0
for i = 1 : n
  begin loop
    u = random vector with entries in [0,1]^m
    x = 2 * u - 1.0    % random vector in [-1,+1]^m unit hypercube
    if norm(x,2) <= 1  % if x is actually in B^m(1)...
      exit loop with success
    end
    k = k + 1
  end loop
end
```

So $n + k$ is the number of points generated in the unit hypercube, while n and k count the number of those points inside or outside the unit hyperball. Clearly, the ratio $\frac{n}{n+k}$ tells us something about the volume of the hyperball relative to that of the hypercube.

1. For $m = 1$ to 8, compute $n = 10,000$ rejection samples of $B^m(1)$. Estimate the volume of $B^m(1)$ by

$$\text{Vest}(B^m(1)) \approx 2^m * \frac{n}{n+k}$$

where 2^m is the volume of the unit cube that encloses the ball. The exact volume is

$$\text{Volume}(B^m(1)) = \frac{\pi^{m/2}}{\Gamma(m/2+1)}$$

Make a table of $m, k, \text{Vest}, \text{Vest-Volume}(B^m(1))$. As m increases, how is k changing? What does this say about using a rejection method in high dimensions?

Table 2.1

m	n	k	Vest	|Vest-Volume Bm(1)|
1	10,000	_____	_____	_____
2	10,000	_____	_____	_____
3	10,000	_____	_____	_____
4	10,000	_____	_____	_____
5	10,000	_____	_____	_____
6	10,000	_____	_____	_____
7	10,000	_____	_____	_____
8	10,000	_____	_____	_____

3 Sampling $\mathbb{R}^m$ with Gaussian Weight

For this exercise, we see how to sample the entirety of m-dimensional space with a Gaussian distribution. We will only consider the simple "spherical" Gaussian, for which $\boldsymbol{\mu} = 0$ and $\Sigma = I$. A sample vector $\mathbf{x}$ can be generated in accordance with this distribution by simply generating an m-vector each of whose entries is drawn from a univariate standard normal distribution $\mathcal{N}(0, 1)$.

1. Compute $n = 10,000$ pairs of samples p and q of a spherical Gaussian distribution in dimension $m = 10$. For each pair, compute their distance $d = ||p - q||_2$, and report the average value of d. Is your value close to $\sqrt{2 * m}$?

   Table 3.1 m = 10 n = 10,000
   Average distance = _____

2. Compute $n = 10,000$ samples x of a spherical Gaussian distribution in dimension $m = 16$. Compute the norm $||x||_2$ of each sample. Make a table that reports the number of samples with norm between 0.0 and 0.5, between 0.5 and 1.0, and so on, up to the range 9.5 to 10. The norms of spherical Gaussian samples are expected to "cluster" near the value $\sqrt{m}$. How do your samples behave?

   Table 3.2 m = 16 n = 10,000

	Count
0.0 <= ||x|| < 0.5	_____
0.5 <= ||x|| < 1.0	_____
1.0 <= ||x|| < 1.5	_____
1.5 <= ||x|| < 2.0	_____
...	
9.5 <= ||x|| < 10.0	_____

4 A Version of HITS

The HITS algorithm for the nodes in a directed network computes two vectors of node ratings:

- **a**, the authority index, the degree to which a node is pointed to by reference (hub) nodes;
- **h**, the hub index, the degree to which a node points to important (authoritative) nodes;

If A is the adjacency matrix, the vectors satisfy: $a = \frac{A'*h}{||A'*h||}$ and $h = \frac{A*a}{||A*a||}$.

Here is an algorithm for the vectors a and h for a directed network of n nodes with adjacency matrix A:

```
a(1:n) = 1     %Initialize
h(1:n) = 1

do k times:    % Iterate k times
  for j = 1 to n
    a(j) = sum of h(i)'s for which node j is pointed to by node i
  end
  a = a / norm ( a )   % Use the l2-norm
  for i = 1 to n
    h(i) = sum of a(j)'s for which node i points to node j
  end
  h = h / norm ( h )   % Use the l2-norm
end
```

Write a computer program (to be turned in) that implements the HITS algorithm for a given adjacency matrix A. You can test your program on the "tiny" dataset whose adjacency matrix is

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

for which the results should be approximately:

$$a = \begin{pmatrix} 0.6572 \\ 0.3690 \\ 0.6572 \\ 0.0004 \\ 0.0000 \end{pmatrix} \quad h = \begin{pmatrix} 0.0000 \\ 0.6154 \\ 0.0002 \\ 0.7882 \\ 0.0002 \end{pmatrix}$$

For the exercise, you will need the **"large"** dataset. This is a directed graph whose $1,458 \times 1,458$ adjacency matrix A is stored on the class Blackboard site as the file *large_sparse.txt*. (There are also "medium", "small" and "tiny" files available in case you want practice.) This file contains 3,545 lines, one for each nonzero entry of A, of the form:

```
i  j  Aij       <-- means that A(i,j) = Aij
```

If you have copied the data file to your current directory, then a standard MATLAB matrix A can be created by the commands:

```
data = load ( 'large_sparse.txt' );
A = spconvert ( data );
A = full ( A );
```

1. Run your program on the "large" dataset, using 30 steps; report the first 5 entries of a and h.

Table 4.1	(1)	(2)	(3)	(4)	(5)
A vector (iteration):	_____	_____	_____	_____	_____
H vector (iteration):	_____	_____	_____	_____	_____

2. An alternative algorithm uses the SVD factorization of the adjacency matrix, $A = U * S * V'$. We can conclude that $a = V(:, 1)$ and $h = U(:, 1)$. Apply this algorithm to the "large" dataset; report the first 5 entries of a and h.

Table 4.2	(1)	(2)	(3)	(4)	(5)
A vector (SVD):	_____	_____	_____	_____	_____
H vector (SVD):	_____	_____	_____	_____	_____

5 What to Turn in:

The project should be completed and submitted by the beginning of class on April 11th. You should submit a report, roughly 3-4 pages in length, stored as a single PDF file, that includes:

1. **Team information:** The names of the team members. Teams can involve from 1 to 4 members. Each team member should participate in the project, and should be able to explain the process involved in getting the results of the exercises.
2. **Problem Description:** Briefly discuss the exercises, including problems that arose, programming issues, interesting features that you noticed, and respond to any questions posed in each exercise.
3. **Your tables:** You should turn in neatly formatted versions of Tables 1.1, 1.2, 2.1, 3.1, 3.2, 4.1, 4.2.
4. **Program:** Include the text of your program used for exercise #4, the HITS algorithm.
5. **Send report, including program #4, to:** *burkardt@mailbox.sc.edu*

Except for the program used for exercise #4, you **do not** need to turn in your programs. However you should keep copies of them available, as they may be requested for review if there is an issue with your results.