Monte Carlo Numerical Estimation of Hypersphere Volume

Created using Maple 14.01

Jake Bobowski

> restart;
with(stats):
with(plots):
with(Statistics):
with(StringTools):

FormatTime("%m-%d-%Y, %H:%M");

"03-21-2013, 21:57"

In all Monte Carlo simulations it is necessary to generate random or pseudo-random numbers. The following statement will generate a random number drawn from a uniform distribution between 0 and 1.

> X := x->stats[random, uniform][0, 1](1):
X();

0.1931398164

This tutorial will attempt to numerically determine the volume of a sphere of arbitrary dimension using the Hit & Miss method. This is equivalent to evaluating a d-dimensional integral. In d-dimensions, a hypersphere is defined by \( x_1^2 + x_2^2 + x_3^2 + \ldots + x_d^2 < R^2 \).

To find the volume of a sphere we'll generate random numbers over some "square" volume and count how many land inside the sphere. The probability of the randomly generated point landing within the sphere is \( p = \frac{V_{\text{sphere}}}{V_{\text{square}}} \). If we use spheres of radius 1, then the volume of the sphere that contains the sphere is \( 2^d \) where \( d \) is the number of dimensions that we're working with. The Monte Carlo simulation is used to determine \( p = \frac{Z_n}{n} \) where \( n \) is the number of trials and \( Z_n \) is the number of "Hits". Collecting all of these results we have \( p = \frac{Z_n}{n} = \frac{V_{\text{sphere}}}{2^d} \) or \( V_{\text{sphere}} = 2^d \left( \frac{Z_n}{n} \right) \).

We'll start with the trivial example of a 1-D sphere. In 1-D, \( x_1^2 < R^2 \) simply translates to \(-R < x_1 < +R\). Throughout this tutorial will restrict ourself to the positive quadrant. That's we'll generate random values of \( x_1 \) between \([0, R]\) (where \( R = 1 \) in this tutorial). The 1-D sphere is trivial because ALL of the randomly generated numbers will fall within the sphere. The volume of the 1-D sphere is just the length of the line. \( V_{1D} = 2R \).

> d := 1:
n := 100000:
Zn := 0:
for i from 1 to n do:
x1 := X();
R := sqrt(x1^2):
if R <= 1 then;
Zn := Zn + 1:
end if;
end do:

V1D := evalf\left( \frac{2^d \cdot Zn}{n} \right);
evalf\left( \frac{V1D}{2} \right);
Next a 2-D sphere (a circle). In 2-D, $x_1^2 + x_2^2 < R^2$. Now we'll have to generate random values for $x_1$ and $x_2$ both being between $[0, R]$ (where $R = 1$ in this tutorial). The volume of the 2-D sphere is expected to be the area of a circle. $V_{2D} = \pi R^2$. One of the virtues of the Monte Carlo method is that going to higher dimensions requires only VERY minor changes to our simulation code. Note that the example of the area of a circle is often used as a Monte Carlo calculation of the numerical value of $\pi$.

```
> d := 2 :
n := 100000 :
Zn := 0 :
for i from 1 to n do:
x1 := X( ) :
x2 := X( ) :
R := sqrt(x1^2 + x2^2) :
if R <= 1 then;
    Zn := Zn + 1 :
end if;
end do:
V2D := evalf\(\frac{2^d \cdot Zn}{n}\);
```

```
evalf\(\frac{V2D}{Pi}\);
```

```
V2D := 3.139000000
0.9991747325
```

A 3-D sphere is a ball. Here, $x_1^2 + x_2^2 + x_3^2 < R^2$. Now we'll have to generate random values for $x_1$, $x_2$, and $x_3$ all being between $[0, R]$ (where $R = 1$ in this tutorial). The expected volume of the 3-D sphere is $(4/3)\pi R^3$.

```
> d := 3 :
n := 100000 :
Zn := 0 :
for i from 1 to n do:
x1 := X( ) :
x2 := X( ) :
x3 := X( ) :
R := sqrt(x1^2 + x2^2 + x3^2) :
if R <= 1 then;
    Zn := Zn + 1 :
end if;
end do:
V3D := evalf\(\frac{2^d \cdot Zn}{n}\);
```

```
evalf\(\frac{V3D}{\frac{4}{3} \cdot Pi}\);
```

```
V3D := 4.193760000
1.001186451
```
Now we're onto something more interesting. We cannot picture a 4-D sphere, but mathematically the
definition is obvious: \( x_1^2 + x_2^2 + x_3^2 + x_4^2 < R^2 \). Now we'll have to generate random values for
\( x_1, x_2, x_3, \) and \( x_4 \) all being between \([0, R]\) (where \( R = 1 \) in this tutorial). So that we can compare our
Monte Carlo simulation to expected results, we'll note that a 4-D sphere has a volume of \((1/2)\pi^2 R^4\).

```plaintext
> d := 4:
n := 100000:
Zn := 0:
for i from 1 to n do:
x1 := X():
x2 := X():
x3 := X():
x4 := X():
R := sqrt(x1^2 + x2^2 + x3^2 + x4^2):
  if R <= 1 then;
    Zn := Zn + 1:
  end if;
end do:
V4D := evalf(2^d * Zn/n);
evalf((V4D / (1/2) * pi^2))
```

\[ V4D := 4.904960000 \]

\[ 0.9939527054 \]

We'll do two more examples. First, a 5-D sphere: \( x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 < R^2 \). Now we'll
have to generate random values for \( x_1, x_2, x_3, x_4, \) and \( x_5 \) all being between \([0, R]\) (where \( R = 1 \) in this
tutorial). So that we can compare our Monte Carlo simulation to expected results, we'll note that a 5-D
sphere has a volume of \((8/15)\pi^2 R^5\).

```plaintext
> d := 5:
n := 100000:
Zn := 0:
for i from 1 to n do:
x1 := X():
x2 := X():
x3 := X():
x4 := X():
x5 := X():
R := sqrt(x1^2 + x2^2 + x3^2 + x4^2 + x5^2):
  if R <= 1 then;
    Zn := Zn + 1:
  end if;
end do:
V5D := evalf(2^d * Zn/n);
evalf((V5D / (8/15) * pi^2))
```

\[ V5D := 5.226560000 \]
By now you should have noticed how easy it is to extend the number of dimensions. To prove a point, our last example will be used to find the volume of a 9-D sphere: 
\[ x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2 + x_9^2 < R^2. \]
Now we'll have to generate random values for \( x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, \) and \( x_9 \) all being between \([0, R]\) (where \( R = 1 \) in this tutorial). Keep in mind that we are now effectively evaluating a 9-D integral with a tricky restriction on the integration limits. So that we can compare our Monte Carlo simulation to expected results, we'll note that a 9-D sphere has a volume of \((32/945) \pi^4 R^9\).

```plaintext
> d := 9;
    n := 100000:
    Zn := 0:
    for i from 1 to n do:
        x1 := X( ):
        x2 := X( ):
        x3 := X( ):
        x4 := X( ):
        x5 := X( ):
        x6 := X( ):
        x7 := X( ):
        x8 := X( ):
        x9 := X( ):
        R := sqrt(x1^2 + x2^2 + x3^2 + x4^2 + x5^2 + x6^2 + x7^2 + x8^2 + x9^2):
        if R <= 1 then;
            Zn := Zn + 1:
        end if;
    end do:
    V9D := evalf(2^d*Zn/n):
    evalf(V9D/32/945/\pi^4);
```

```
V9D := 3.374080000
1.022910684
```

Here is a table from Wikipedia that shows the volume and surface areas of \( n \)-dimensional spheres from \( n = 1 \) to \( n = 9 \).
<table>
<thead>
<tr>
<th>Number of Dimensions</th>
<th>VOLUME ((V_n))</th>
<th>SURFACE AREA (\left(S_{n-1}\right))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n = 1)</td>
<td>(2R)</td>
<td>(2\pi)</td>
</tr>
<tr>
<td>(n = 2)</td>
<td>(\pi R^2)</td>
<td>(2\pi R)</td>
</tr>
<tr>
<td>(n = 3)</td>
<td>(\frac{4}{3} \pi R^3)</td>
<td>(4\pi R^2)</td>
</tr>
<tr>
<td>(n = 4)</td>
<td>(\frac{1}{2} \pi^2 R^4)</td>
<td>(2\pi^2 R^3)</td>
</tr>
<tr>
<td>(n = 5)</td>
<td>(\frac{8}{15} \pi^3 R^5)</td>
<td>(8\pi^3 R^4)</td>
</tr>
<tr>
<td>(n = 6)</td>
<td>(\frac{1}{6} \pi^3 R^6)</td>
<td>(\pi^3 R^5)</td>
</tr>
<tr>
<td>(n = 7)</td>
<td>(\frac{16}{105} \pi^4 R^7)</td>
<td>(16\pi^4 R^6)</td>
</tr>
<tr>
<td>(n = 8)</td>
<td>(\frac{1}{24} \pi^4 R^8)</td>
<td>(\pi^4 R^7)</td>
</tr>
<tr>
<td>(n = 9)</td>
<td>(\frac{32}{945} \pi^4 R^9)</td>
<td>(32\pi^4 R^8)</td>
</tr>
</tbody>
</table>

"N SpheresVolumeAndSurfaceArea" by Joseph Lindenberg - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - [http://commons.wikimedia.org/wiki/File:N_SpheresVolumeAndSurfaceArea.png#/media/File:N_SpheresVolumeAndSurfaceArea.png](http://commons.wikimedia.org/wiki/File:N_SpheresVolumeAndSurfaceArea.png#/media/File:N_SpheresVolumeAndSurfaceArea.png)