Homework 3

Computer Science Theory for the Information Age

致远 12 级 ACM 班　　刘爽　　5112409048

June 11, 2013

1. (a) What is the surface area of a unit cube in d-dimensions?

   Solution:

   The d-dimensional cube has $2d$ faces, and each face has area 1, so the surface area is $2d$.

   (b) Is the surface area of a unit cube concentrated close to the equator, define here as the hyperplane $\{x : \sum\limits_{i=1}^{d} x_i = \frac{d}{2}\}$, as is the case with the sphere?

   Solution:

   Yes.
   Suppose we are uniformly generating points on the surface of the cube. Then to prove the surface area is concentrated close to the equator, we only need to prove that the distance between the random generated point the the hyperplane is almost always near zero.
   To show this, let p be a random generated point, having coodinates x_1 to x_d, X be a random variable, $X = \sum_{i=1}^{d} xi$. And d be the distance between l and the hyperplane representing the equator. Then we have $X = \sqrt{d}l$. So what we really need to prove is the random variable X, which takes value from 0 to d, is almost always near $\frac{d}{2}$.
   First we notice that, due to the symmetrical properties, $E(X) = \frac{d}{2}$. Then we need to calculate $Var(x)$. But the d coordinates are not independent, so we

should do a little bit transformation. Every time we generate a point. We can consider the generation in such a way. First we choose one dimension, let its coodinate to be 0 or 1, with equal probability. Then let the rest $d-1$ coodinates take value uniformlly from 0 to 1. At last we change the dimension considered first to the first coodinate, this doesn't change the value of X. In this way, the d coodinates are independent, and

$$\begin{aligned} Var(x) &= Var(x_1) + \sum_{i=2}^{d} x_i \\ &= \frac{1}{4} + (d-1)\int_0^1 (x-\frac{1}{2})^2 dx \\ &= \frac{1}{4} + (d-1)\frac{1}{12} \\ &= \frac{d}{12} + \frac{1}{6} \end{aligned}$$

Then according to the Chebyshev's Inequality, we have

$$p(|X - E(X)| \geq \epsilon) \leq \frac{Var(X)}{\epsilon^2}$$

this means

$$p(|X - \frac{d}{2}| \geq \epsilon) \leq \frac{\frac{d}{12} + \frac{1}{6}}{\epsilon^2} \leq \frac{d}{4\epsilon^2}$$

The distance between the equator and the farthest point from it is $\frac{\sqrt{d}}{2}$. Now as the dimension goes higher and higher, we consider a slice near equator of length $c\frac{d}{2}$, where c can be a quite small number. This slice takes a fixed proportion of the height of the north pole from the equator, where the north pole is the farthest point from the equator we have mentioned before. Since a point's distance from equator equals $\frac{X-E(X)}{\sqrt{d}}$, we should let ϵ be $c\frac{\sqrt{d}}{2}$ times $\sqrt{d}$, which is $\frac{d}{2}$, then the equation becomes

$$p(|X - \frac{d}{2}| \geq c\frac{d}{2}) \leq \frac{1}{c^2 d}$$

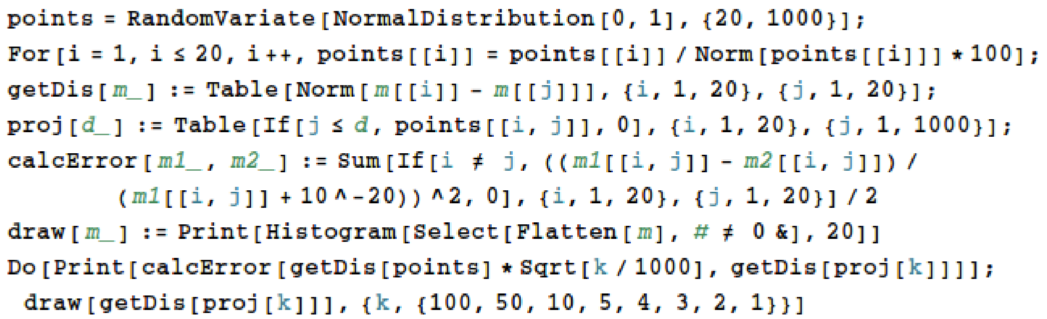
This means, for very large d, the surface area is always near the equator.

2. Generate 20 points uniformly at random on a 1,000-dimensional sphere of radius 100. Calculate the distance between each pair of points. Then project the data onto subspaces of dimension $k = \{100, 50, 10, 5, 4, 3, 2, 1\}$ and calculate the sum of

squared error between $\frac{k}{d}$ times the original distances and the new pair wise distances for each of the above values of k.

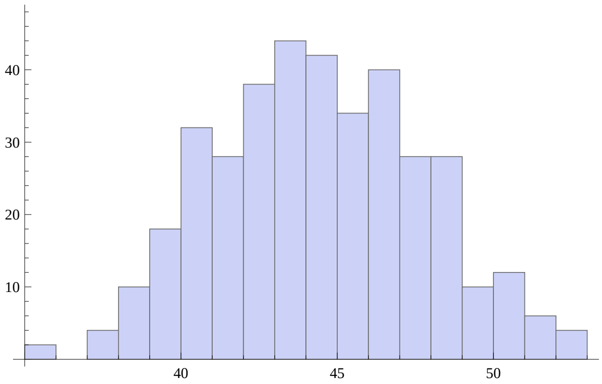
Solution:

Since what really matters is the relative error, so I calculate the sum of squared relative errors. A mathematica code is shown as below.

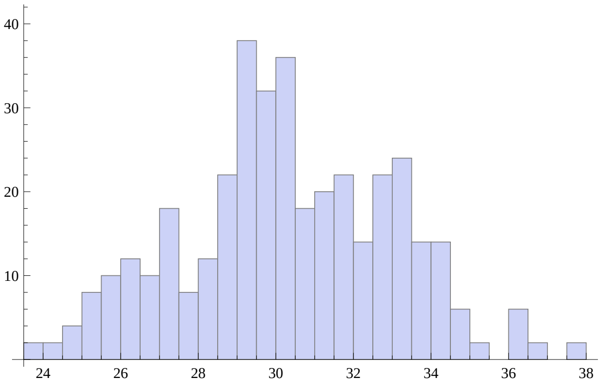


```
points = RandomVariate[NormalDistribution[0, 1], {20, 1000}];
For[i = 1, i ≤ 20, i++, points[[i]] = points[[i]] / Norm[points[[i]]] * 100];
getDis[m_] := Table[Norm[m[[i]] - m[[j]]], {i, 1, 20}, {j, 1, 20}];
proj[d_] := Table[If[j ≤ d, points[[i, j]], 0], {i, 1, 20}, {j, 1, 1000}];
calcError[m1_, m2_] := Sum[If[i ≠ j, ((m1[[i, j]] - m2[[i, j]]) /
        (m1[[i, j]] + 10^-20))^2, 0], {i, 1, 20}, {j, 1, 20}] / 2
draw[m_] := Print[Histogram[Select[Flatten[m], # ≠ 0 &], 20]]
Do[Print[calcError[getDis[points] * Sqrt[k / 1000], getDis[proj[k]]]];
 draw[getDis[proj[k]]], {k, {100, 50, 10, 5, 4, 3, 2, 1}}]
```

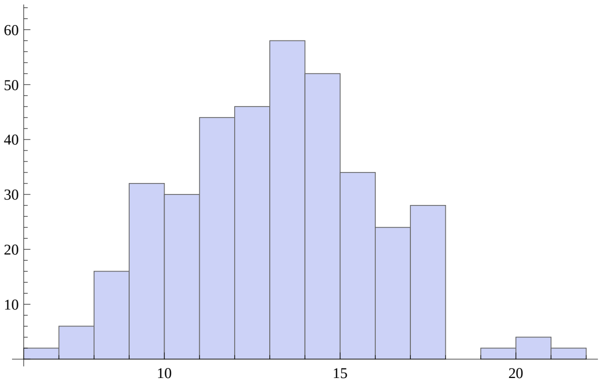
When $k = 100$, the answer is 1.04293



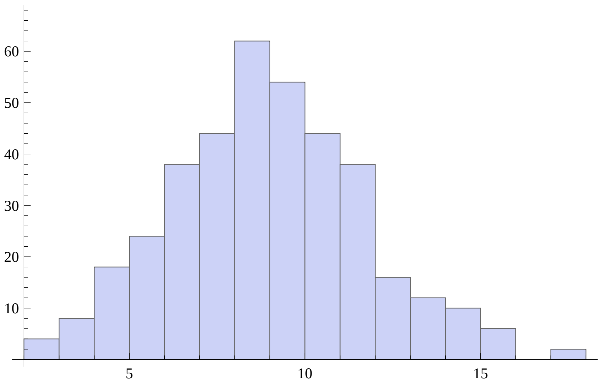
When $k = 50$, the answer is 1.71877



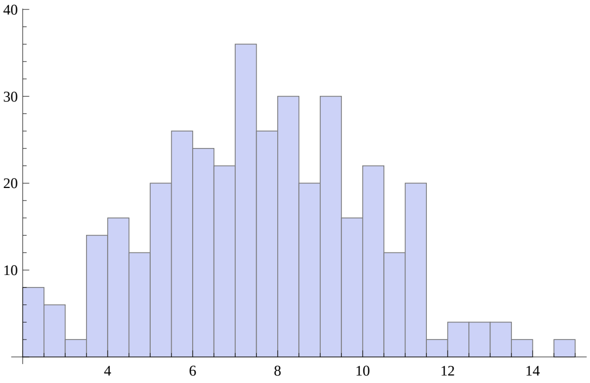
When $k = 10$, the answer is 8.20201



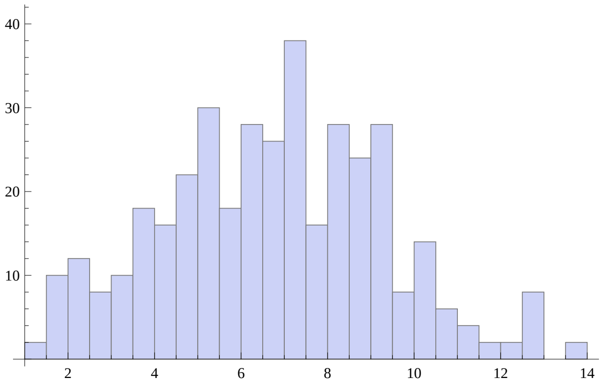
When $k = 5$, the answer is 16.6067



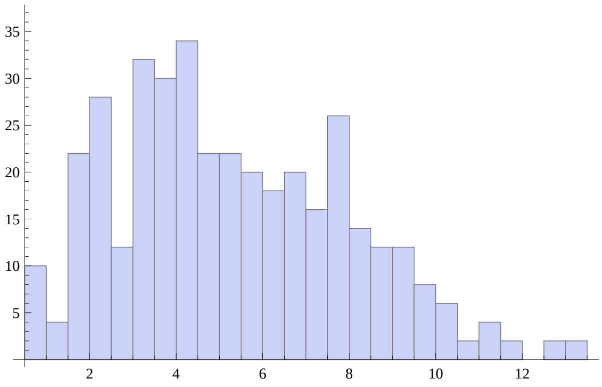
When $k = 4$, the answer is 19.3611



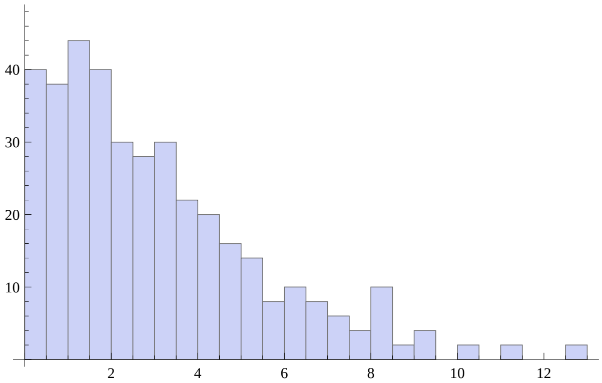
When $k = 3$, the answer is 24.5451



When $k = 2$, the answer is 39.0224



When $k = 1$, the answer is 75.2894



As we can see, when the dimension which be projected to goes lower and lower, the relative error goes higher and higher, which is as we expected.