1 Hyperball vs Hypercube

1.1 Volume of hyperball:

for n = 2k

$$V_{2k}(R) = \frac{\pi^{2k}}{k!} R^{2k} \tag{1}$$

for n = 2k + 1

$$V_{2k+1}(R) = \frac{2(k!)(4\pi)^k}{(2k+1)!} R^{2k+1} \tag{2}$$

1.2 Volume of hypercube:

for hyper cube with edge length of 'a':

$$V_n(a) = a^n \tag{3}$$

1.3 Answer

a = 2.0 and R = 1.0 :

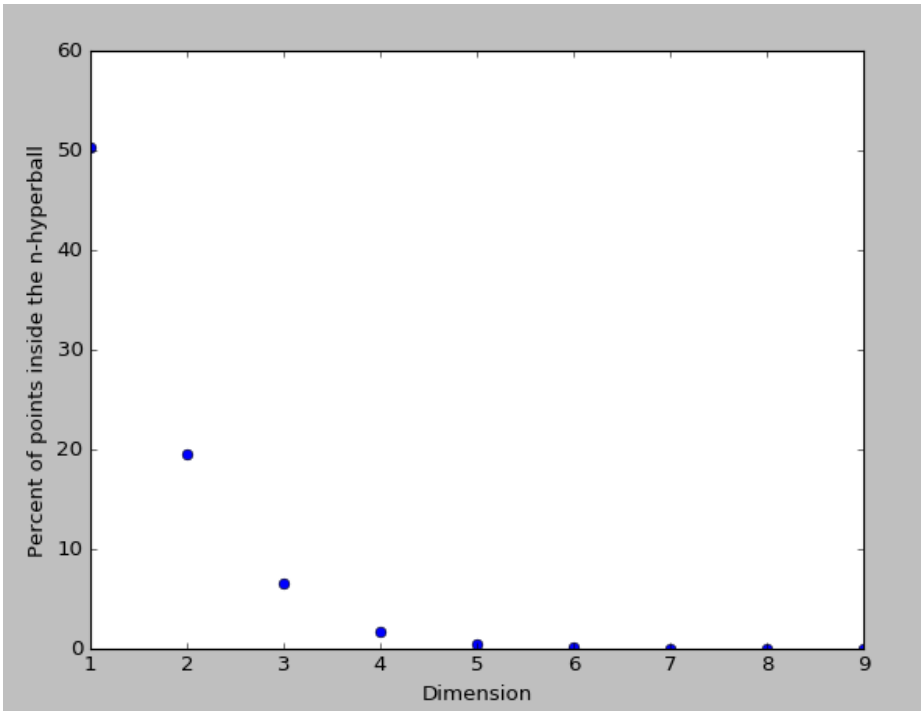
1.3.1 n = 2k

$$\frac{Hyperball \quad volume}{Hypercube \quad Volume} * 100\% = \frac{\pi^{2k}}{k! 2^{2k}} * 100\% \tag{4}$$

1.3.2 n = 2k + 1

$$\frac{Hyperball \quad volume}{Hypercube \quad Volume} * 100\% = \frac{k! \pi^{2k}}{(2k+1)!} * 100\% \tag{5}$$

1.4 Experimental results: Percentage of points in hypercube



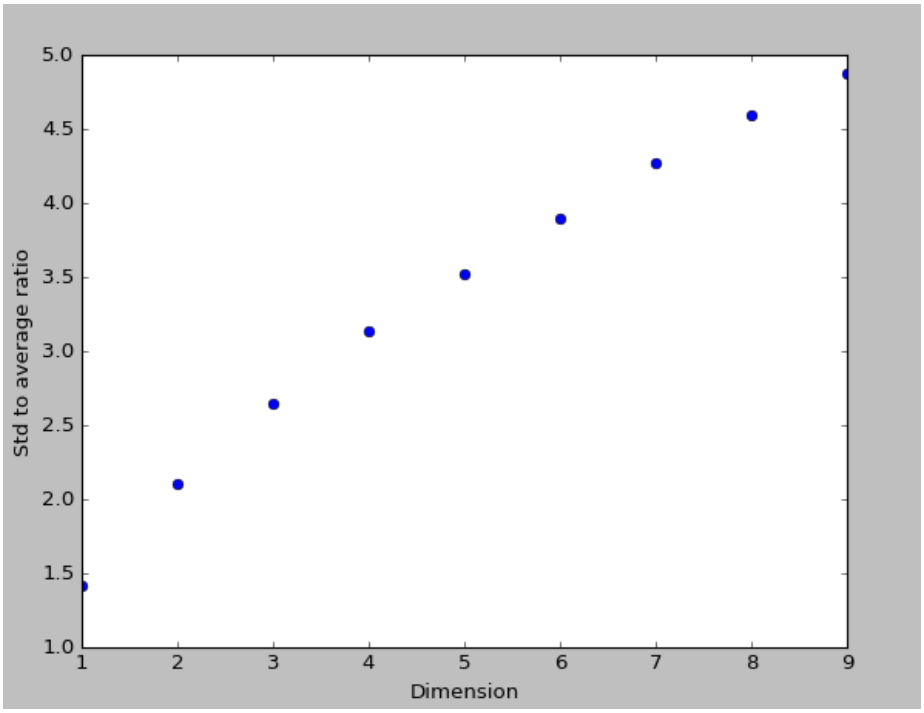
1.4.1 Discussion

The plot shows that the percentage of point that fall into the hypercube decreases exponentially. Which yields two main conclusions: human intuition isn't a good tool in high dimension hyperspace and secondly the number of point in a randomly choses neighbourhood drops dramatically.

2 CoD and k-NN

The k-nearest neighbours rule requires the neighbours to be a reasonably good representative of the neighbourhood of the query point (they need to be relatively close to the query point). Because the volume in hyperspace increases exponentially with the number of dimensions, data points density drops significantly, therefore the points are less likely to be in the near neighbourhood of the query point. In effect k-NN doesn't perform well.

2.0.1 Experimental results: Ratio of std to average value



2.0.2 Discussion

From the plot above one could conclude that in high dimension space the ratio of std to average distance grows very slowly, which means that an euclidean distance is no loger a viable tool for comparing datapoints in high dimension hyperspace.