SDM_the_search_for_powers_of_2_everywhere

April 17, 2018

1 Sparse Distributed Memory: How many subspaces?

In his original analysis, Kanerva uses... ...1,000 dimensions ...1,000,000 circles ...with radius=451 bits so that a random bistring will be stored in aproximately 1,000 hard locations.

A first observation here is that powers of 10 are not well suited, either for the mathematical analysis of the space, or for computer science. Let us play a little with numbers here, starting from Kanerva's parameters. Our goal will be to find how many circles we should use and the radius of a circle. Can we find some optimum set of parameters?

In [3]:
```
n = 1000 # number of dimensions
print ('number of dimensions=',n)
```

```
number of dimensions= 1000
```

In [4]:
```
def mu(n):
    return n/2

import math

def sigma(n):
    return math.sqrt(n)/2

def percentage_of_n(n):
    return (100*(6*sigma(n))) / n
```

In [5]:
```
def analysis(n):
    print ('************ ANALYSIS FOR n=',n)
    print ('orthoghonal distance=',mu(n))
    print ('standard deviation=',sigma(n))
    print ('the space offers ', str(n//sigma(n)), ' standard deviations.')
    print ('3 sigma=', 3.09*sigma(n))
    print ('the distance from a pole to the equator=', math.sqrt(n), 'standard deviation
    print ('the percentage of n (in relation to the range R Bits) is', percentage_of_n(n
    print ('Because 3 standard deviations contain ~1/1000 of a normal distribution, ' \
          'approximately one in a thousand items will be found in a circle of', \
          mu(n)-3.09*sigma(n),' radius')
    print ('.............................................\n\n\n\n')
```

```
In [6]: for n in [100, 1000, 10000, 256, 1024]:
            analysis(n)
```

```
************ ANALYSIS FOR n= 100
orthoghonal distance= 50.0
standard deviation= 5.0
the space offers  20.0  standard deviations.
3 sigma= 15.45
the distance from a pole to the equator= 10.0 standard deviations
the percentage of n (in relation to the range R Bits) is 30.0
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
...



************ ANALYSIS FOR n= 1000
orthoghonal distance= 500.0
standard deviation= 15.811388300841896
the space offers  63.0  standard deviations.
3 sigma= 48.857189849601454
the distance from a pole to the equator= 31.622776601683793 standard deviations
the percentage of n (in relation to the range R Bits) is 9.486832980505138
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
...



************ ANALYSIS FOR n= 10000
orthoghonal distance= 5000.0
standard deviation= 50.0
the space offers  200.0  standard deviations.
3 sigma= 154.5
the distance from a pole to the equator= 100.0 standard deviations
the percentage of n (in relation to the range R Bits) is 3.0
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
...



************ ANALYSIS FOR n= 256
orthoghonal distance= 128.0
standard deviation= 8.0
the space offers  32.0  standard deviations.
3 sigma= 24.72
the distance from a pole to the equator= 16.0 standard deviations
```

```
the percentage of n (in relation to the range R Bits) is 18.75
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
...



************ ANALYSIS FOR n= 1024
orthoghonal distance= 512.0
standard deviation= 16.0
the space offers  64.0  standard deviations.
3 sigma= 49.44
the distance from a pole to the equator= 32.0 standard deviations
the percentage of n (in relation to the range R Bits) is 9.375
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
...
```

```
In [86]: print_analysis(1024)
```

```
************ ANALYSIS FOR n= 1024
orthoghonal distance= 512.0
standard deviation= 16.0
3 sigma= 48.0
the distance from a pole to the equator= 32.0 standard deviations
the percentage of n (in relation to the range R Bits) is 9.375
Because 3 standard deviations contain ~1/1000 of a normal distribution, approximately one in a t
################################################################
```

Let us focus on the word *aproximately*. The estimation is given by usual tables with metrics of the normal curve.

The *unit* of analysis provided is that of the standard deviation.

Our intention is to make the bit the unit of analysis, and estimate the statistics of the system in a more precise form.

```
In [ ]:
```