Objectives

- Dimensionality - Code Examples
- Distance Explanation
- Introduction to K-NN

---

1 Review

$$\vec{x} \in \mathbb{R}^p$$

Break down:

- $\vec{x}$ represents a vector.
- $\in$ represents an element belonging to a particular set.
- $\mathbb{R}$ represents the set of all real numbers.
- p represents the dimension of the vector space.
- **Meaning**: $\vec{x}$ is a vector with all elements being real numbers in p-dimensional space.
- **Terminology**: p can have other names such as Feature Space, and Factors.

Example: Column vector: $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$

$$d(x, y)$$

Distance metric: Measures the distance (dissimilarity) between two points x and y.

$$d(x, y) = ||x - y||$$

Euclidean distance: This is a specific type of distance metric. The straight line distance between two points in an Euclidean space.

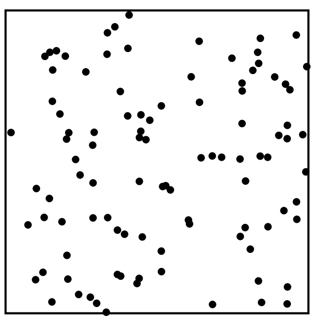
$$p >> 1$$

Meaning: This is p much greater than one. If this happens then vector x has a high number of dimensions.
Problem: High dimensionality can cause issues like data sparsity and overfitting. Similarly, there is the "Curse of Dimensionality".

2 Lecture

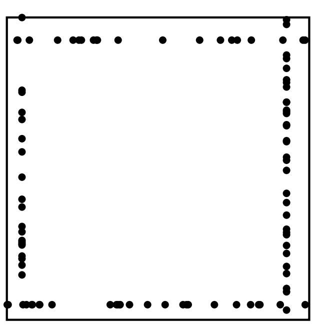
2.1 Dimensionality

Two dimensional vector space:



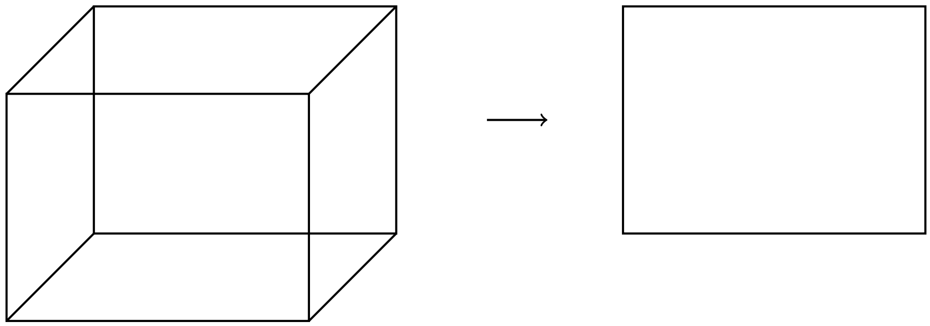
Curse of Dimensionality: In a feature space with p dimensions, $[0, 1]^p$, full of randomly distributed points, as p approaches infinity, the average distance from any given point to the closest edge decreases.

Example: High dimensionality (p value) in a two dimensional space.



Dimensionality Reduction: reduce the number of dimensions in a dataset while retaining as much of the relevant information as possible.

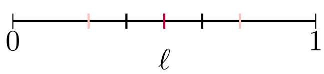
Example: From three dimensions to two dimensions.



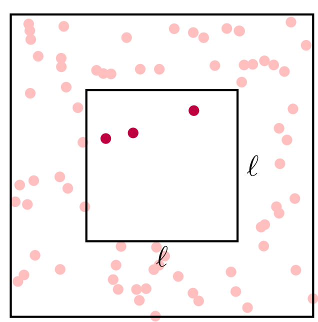
What is the size of the box ℓ that always has k number of dots in it?
Breakdown:

- k = fixed number $< n$ (red dots)
- n = number of samples (pink dots)
- p = dimensions
- Terminology: If p is $\geqslant 4$ it's a **hyper-cube**

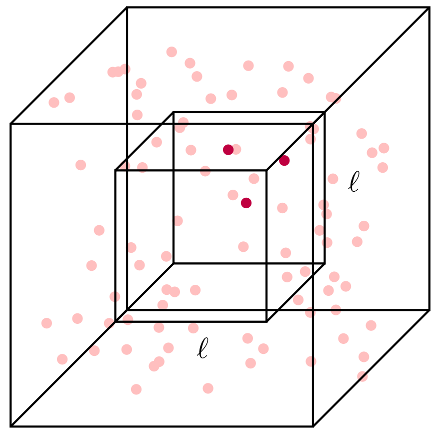
Example: $k = 1$, $p = 1$, $n = 3$



Example: $k = 3$, $p = 2$, $n = 100$



Example: $k = 3$, $p = 3$, $n = 100$



What are the volumes of the boxes?

Outter Box Volumes:

$$\begin{aligned} \text{For } p = 1: &\quad V_{\text{big}} = 1 \\ \text{For } p = 2: &\quad V_{\text{big}} = 1 \\ \text{For } p = 3: &\quad V_{\text{big}} = 1 \\ &\quad \dots \\ \text{For } p = p: &\quad V_{\text{big}} = 1 \end{aligned}$$

Inner Box Volumes:

$$\begin{aligned} \text{For } p = 1: &\quad V_{\text{small}} = \ell < 1 \\ \text{For } p = 2: &\quad V_{\text{small}} = \ell^2 \\ \text{For } p = 3: &\quad V_{\text{small}} = \ell^3 \\ &\quad \dots \\ \text{For } p = p: &\quad V_{\text{small}} = \ell^p \end{aligned}$$

Volume calculation:

$$\left(\frac{\ell}{1}\right)^p = \ell^p \approx \frac{k}{n}$$

k = a fixed number $< n$

Answer to the first question:
How to know ℓ size? Solve algebraically.

$$\ell \approx \left(\frac{k}{n}\right)^{\frac{1}{p}}$$

2.2 Code

Language: Julia
Platform: Jupyter notebook

Code:

```
using Distances
using LinearAlgebra
```

```
x = rand(2)
```

```
2-element Vector{Float64}:
 0.11715724827332152
 0.8703834178825096
```

```
y = rand(2)
```

```
2-element Vector{Float64}:
 0.983074530416966
 0.9654697003440244
```

```
Euclidean()(x,y)
```

```
0.8711223453840379
```

```
Minkowski(2)(x,y)
```

```
0.8711223453840379
```

```
Hamming()(x,y)
```

```
2
```

```
norm(x-y)
```

```
0.8711223453840379
```

```
L(p)=@. (k/n)^(1/p)
```

```
L (generic function with 1 method)
```

```
p=[1,2,3,10,20,100]
```

```
6-element Vector{Int64}:
   1
   2
   3
  10
  20
 100
```

```
n=1000
k=11
L.(p)
```

```
6-element Vector{Float64}:
 0.011
 0.10488088481701516
 0.22239800905693158
 0.6369997597182926
 0.7981226470400978
 0.9559032250692122
```

```
N = 500
d = 5
D = 0.0 # distance
for _=1:N
  x = rand(d)
  y = rand(d)
  D += norm(x - y)
end

println(D)
```

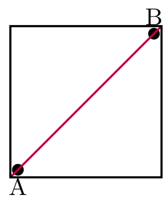
```
450.01189208407664
```

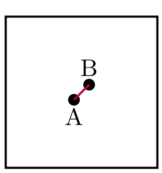
```
# Computes the minimum distance of the max norm from 0 and 1 for 100 random vectors.
for _ = 1:100
  x = rand(d)
  min(1-norm(x,Inf), norm(x,Inf))
end
```

2.3 Distance

Divergence: The distance between two points increases infinitely.
Converging: The distance between two points converges to zero.

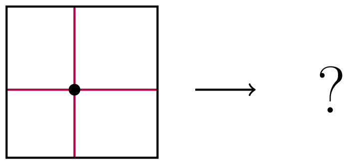
Example: The left box shows divergence. The right box shows convergence.





Question: Cosine distance is not a distance: why? Because it is not nonnegative.

Question: What's the minimum distance to an edge?



To find the minimum distance we use norms ($||x||$). There are different types of norms, like: Euclidean norms
Infinity norms
...

2.4 K-NN

Meaning: K-NN is K-Nearest Neighbor

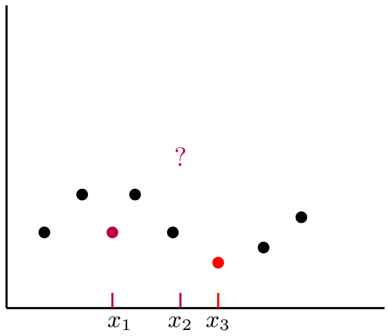
How it works:

1. Have a data point.
2. Find the distance between the point and all the data points. Euclidean metric is the most common.
3. Sort the distances
4. Select K Neighbors with the smallest distances from the point.
5. Perform the average, or mode.

Why does it work?

Because not assuming the numbers are uniform will prevent the curse of dimensionality.

Class demonstration: Don't follow the pattern



Limitations: If dimensions increase then it's not Nearest Neighbor.

K-NN is used for:

- Binary Classification
- Regression

Question: What do you do with missing data?

Example:

$$\begin{bmatrix} 1 \\ ? \\ 3 \\ 4 \\ 7 \\ ? \\ 5 \end{bmatrix}$$

Methods:

1. delete it
2. mean or median
3. K-NN (take the nearest neighbors and their average)

Example: Maine is missing temperature data. Taking the mean won't work since places like Texas and Arizona will effect the results.

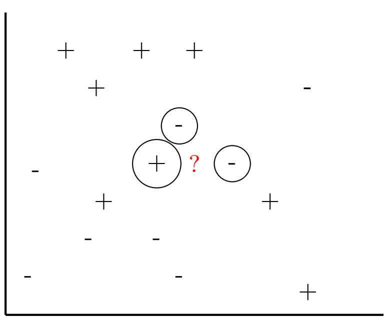
How to solve this problem?
Use K-NN. Do this by taking the temperatures of the closest states and perform the average.

K-NN setup:

$$\text{Data} = D = (x_i, y_i)^n \leqslant \mathbb{R}^p x(-1,1)$$
$$x \in \mathbb{R}^p \text{ and } y = -1 \text{ or } y = 1$$

Rule: K always needs to be an odd number for classification. This is to prevent a tie from occurring.

Example: Is ? positive or negative?
$K = 3$, $p = 2$, $n = 16$



Answer: The k = 3 closest are a positive negative and negative. Since there are two negatives we assume the ? is negative.

Order:

- Calculation of distance: $O(np)$
- Sort distances: $O(nlogn)$
- Pick k that are the smallest: $O(k)$