K-Nearest Neighbors

KNN using python.

Iván Andrés Trujilllo Abella

trujilloiv@javeriana.edu.co

Predictive consumer behavior

...

Identify patterns in consumer behavior allow us to take better decisions, capitalize opportunities with improved strategies and business processes.

...

- Churn prediction
- From normal to premium account
- Buy warranty
- ...

Customer Data

Income	Location(alt and long)	New client	Kind of product
10	(10,23)	0	IT
20	(22,41)	1	Construction
45	(33,12)	1	IT
⋮	⋮	⋮	⋮
...	...	...	...

Table: Customer Data

Notation

- f total number of features
- N number of observations or clients in dataset.
- x_i the i variable from the f total variables.

Mean of the feature j

$$\bar{x_j} = \frac{1}{N} \sum_{i=1}^{N} x_{ji} \tag{1}$$

Euclidean distance

Usually in some Machine Learning(ML) models, the measures of distance are associated with similarity. There are some ways of measure distances given the **nature** of the variable.

Definition in $\mathbf{R}^2$

The distance (d) among the two **patterns** or points $A = (x_a, y_a), B = (x_b, y_b)$ is:

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \tag{2}$$

is $\mathbf{R}^n$ easily extensible.

Python implementation

```
def euclidean_dist(A,B):
  C = np.array(A) - np.array(B)
  return np.sqrt(C @ C)
```

Euclidean distance

The distance as a approximation to similarity.

$$d_{ij} = \sqrt{\sum (x_{if} - x_{jf})^2} \tag{3}$$

where f indicate the feature of the individuals ij

	A	**B**	**C**	**D**
A	0	d_{AB}	d_{AC}	d_{AD}
B	d_{BA}	0	d_{BC}	d_{BD}
C	d_{AC}	d_{BA}	0	d_{CD}
D	d_{AD}	d_{BD}	d_{DC}	0

Table: Euclidean distance matrix

note the symmetry $d_{AB} = d_{BA} = \sqrt{(11-10)^2 + (0.7-0.5)^2}$.

Distance

Manhatan distance

$$d(A, B) = \sum_{i}^{f} |A_{fi} - B_{fi}| \tag{4}$$

where f is the number of features.

Distance among binaries variables

for instance a variable as financial risk: yes or not. we can map yest to 1 and not to 0.
Hamming distance allow us to measure the distance among two binaries variables; $d(0,0) = 0$, $d(0,1) = 1$, $d(1,1) = 0$.

Hamming distance

```
def hamming(A,B):
  counter = 0
  for i in range(len(A)):
    if A[i] != B[i]:
      counter+=1
  return counter
```

Vector hamming distance

for instance $(1, 0, 0, 0, 1)$ where 1 is the presence of the characteristic and 0 the absence, we can calculate the distance as the number of different bits, for instance:

$$d((1, 0, 1), (1, 1, 1)) = 1 \tag{5}$$

We can also normalize the vector, dividing for the lentgh of the array given that the before arrays have 3 bits then the distance will be $\frac{1}{3}$.

Association coefficients

		B	
		Feature	**Not feature**
A	**Feature**	a	b
	Not feature	c	d

$S_{(ij)} = \frac{a+d}{a+b+c+d}$ take in mind that two objects could be similar by lacking feature the following could be tackle this problem $J_{(ij)} = \frac{a}{a+b+c}$. Notice that the both are numbers between zero and one, the first indicate not similarity a

Categorical variables

for instance the risk could be; low, middle and high and the firm could befinancial, agriculture and IT, and we need only count the number of time in which the variables are not equal, dividied by the number of categorical variables.

Considerations

Greater number of instances in each categorical variable increase the probability that are different.

Ponder distance in categorical variables

The variable X have k possible instances (values), and Y have m, therefore the distance if the values are different will be:

$$\frac{1}{m}(1) + \frac{1}{k}(1) \tag{6}$$

Ordinal variables

variables as time in terms; "recent", "new", "normal" and old. we can try this variables mapping to $[0, 1]$. After used, euclidean distance.

Exercise

Calculate the distance among A, B:

Id	Location	Rating	New client
A	(23,45)	Good	1
B	(56,34)	Normal	0
C	(31,44)	Bad	1

Table: Data: Assigment car

An possible solution to map Rating.

Good $\rightarrow$ 1, Normal $\rightarrow$ 0.5, bad $\rightarrow$ 0.

KNN (1967)

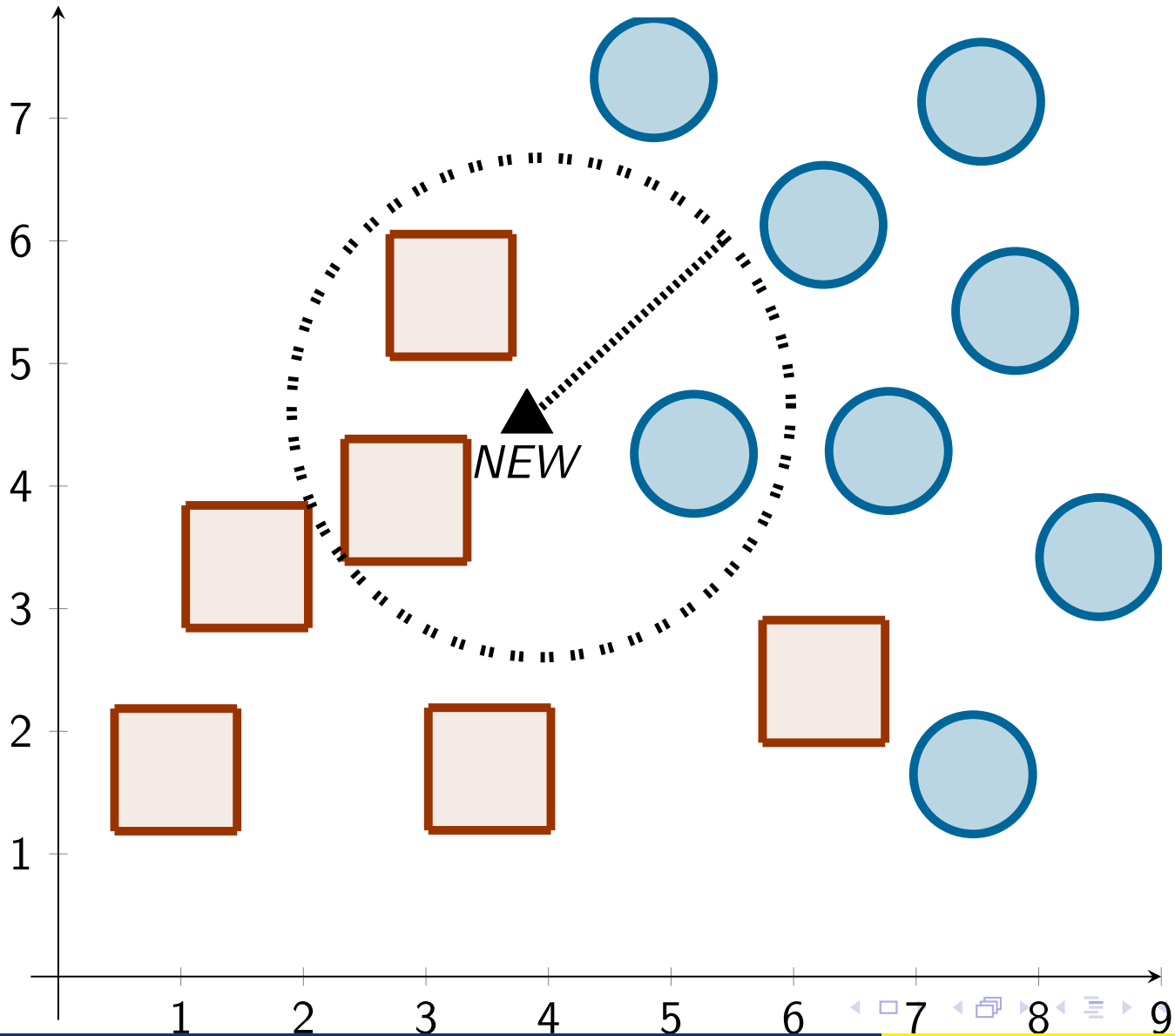
Think a binary classification problem.

Main idea

Similarity is based in distance...

KNN

Ilustration



KNN

According to the before image we could think in $\vec{x_i}$ describing classifying.
X
A **Majority vote**, Note that the previous image, where the number of neighbors is three or $k = 3$ then the mode(or more frecuent class) is red, therefore **NEW** belong to that class.

Pseudocode

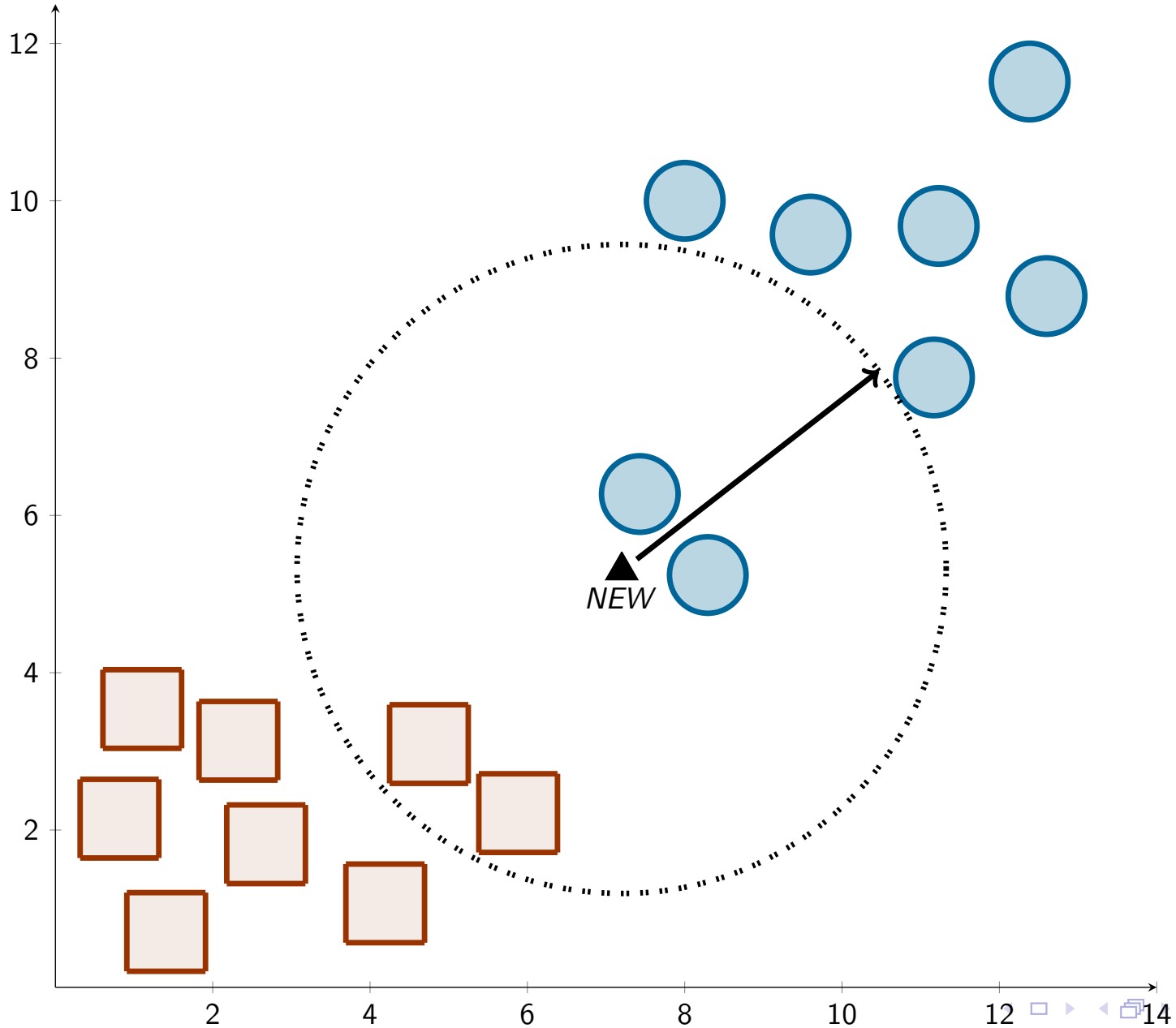
```
Choose k (integer).

Estimate the distance of all new patterns,
to the another known patterns.

Choose the k-nearest patterns (neigbours) and
take as label of new patters,
the mode of classes in the set of the k-nearest neighbours.
```

Note that we select k therefore this a **hyperparameter**.

Weighted distance



Considerations

Indertermination, **Weighted voting** and **Weighted features** you can multiply each vote by the weight for the point i and A is:

...

$$predict(\blacktriangle) = \sum_{i=1}^{k} \left(\frac{w_i}{\sum_{j=1}^{k} w_j} \right) class(Neighbor_i) \tag{7}$$

where $w_i = \left(\frac{1}{d_{\blacktriangle,i}} \right)$ is the inverse of the distance from the new point $\blacktriangle$ to the pattern i.

Processing data

The ML could be in three set of data
Handling data

...

- Standardize numerical variables (is needed that all variables have the same importance in distance function).
- Binaries to zero and one
- Ordinales map to $[0, 1]$.

Standardize

There are different ways of standardize data for instance:

$$z_i = \frac{x_i - \bar{x}}{\sigma_x} \tag{8}$$

Workflow

- Training dataset
- Validation dataset (hyperparameter tunning)
- Testing dataset

Why it is important?

Different models of *KNN* due given the hyperparameters! how select k and a metric of distance.

Curse of dimensionality

Curse of dimenstionality

Assume that we have N points generated randomly, if we increase the number of dimension to generate those points, then $\frac{max(dist)}{min(dist)}$ diminish faster. **Run simulation (click here)**.

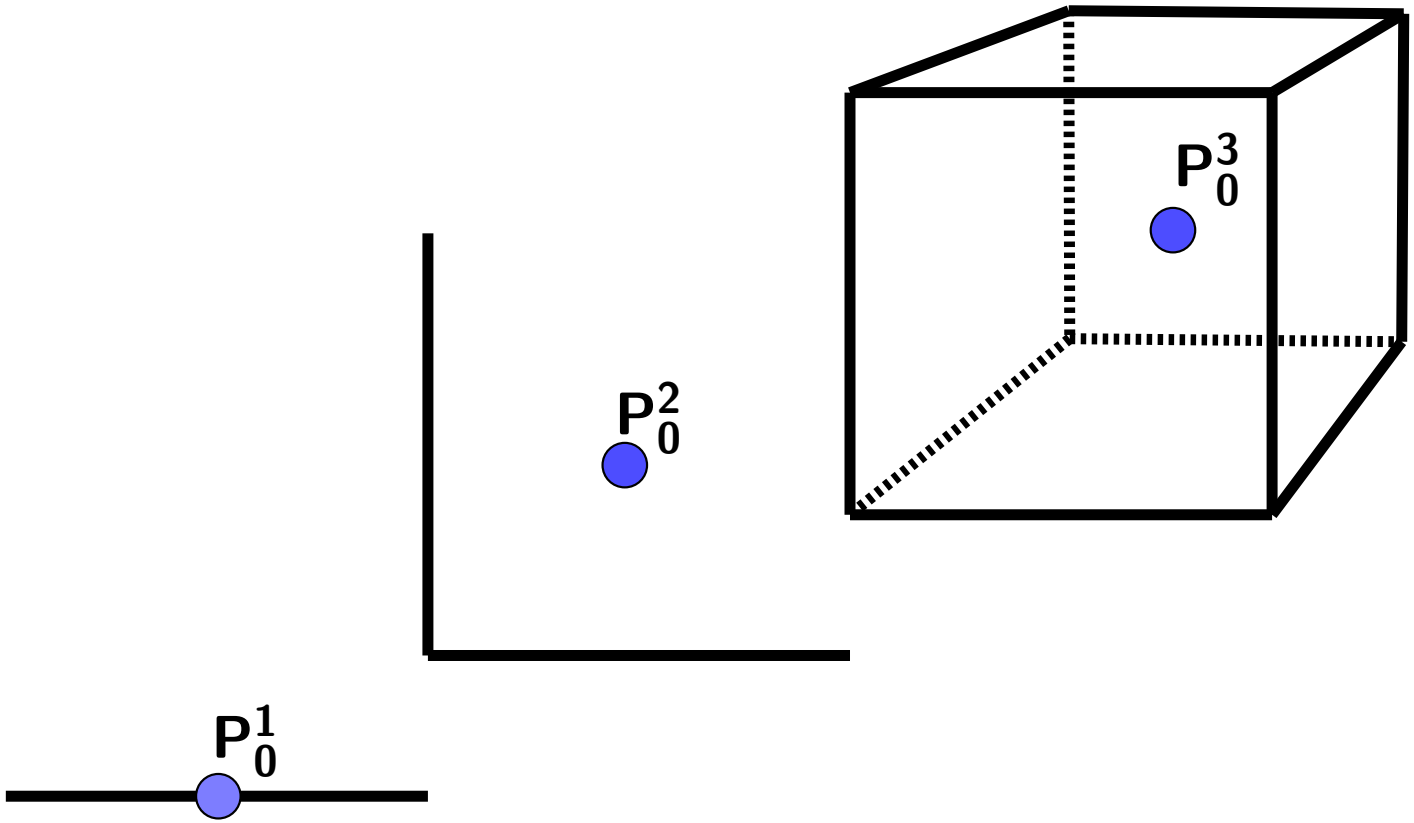
Curse of dimensionality

Curse of dimenstionality

Assume that we have N points generated randomly, if we increase the number of dimension to generate those points, then $\frac{max(dist)}{min(dist)}$ diminish faster. **Run simulation (click here)**.

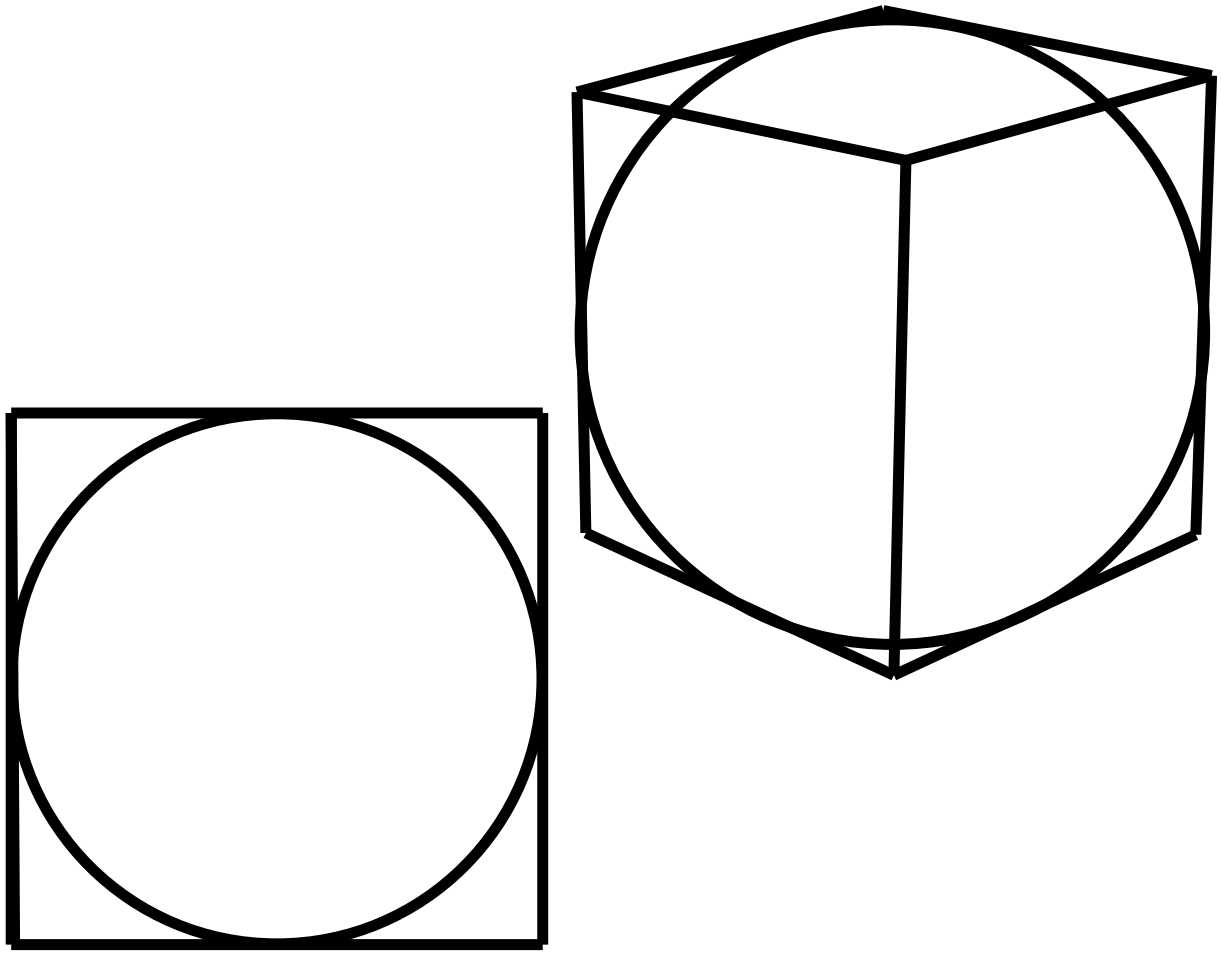
Notice, that means that the distances are more homogenous and therefore the plot tend to 1 **or** $\log(1) = 0$**.**

Curse of dimensionality



Hypershpere and hypercube

Circunscrition



Volume ratios

the area of square is l^2, the area of circle is πr^2 assuming that $l = 2r$ then:

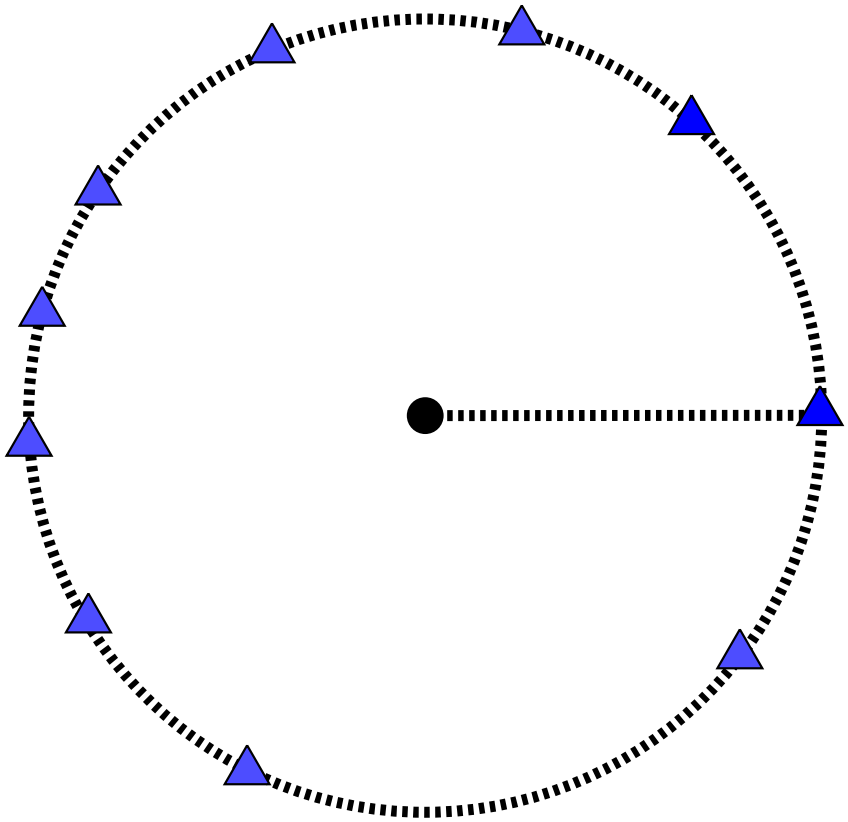
$$ratio_{\mathbb{R}^2} = \frac{\pi}{4} \tag{9}$$

The volume of a cube is l^3 and the volume a sphere id $\frac{4}{3}\pi r^3$ assuming that $l = 2r$ then:

$$ratio_{\mathbb{R}^3} = \frac{\pi}{6} \tag{10}$$

The problem is $ratio_{\mathbb{R}^n} \longrightarrow 0$ if $n \longrightarrow \infty$. Whats mean? that the almos eneterly that unit volume of n-cube lies outside of the sphere.

Equidistance



Notice, that all blue patterns are equidistance ($min = max$ **distance**) to the black unkown pattern, but what find previously in simulation?

What happen with our classifier

To diminish the ratio $min - max$ then is more indishtinguable the distance among points of different labels!

Curse of dimensionality

Curse of dimensionality

If dimension grows then the number of data points that we need to have a good performance grow exponentially.

A bit of history

Bellman(1961) coined the term **curse of dimensionality**

A lot of features

- could appear irrelevant variables?
- could appear multicolinearity (what models are affected by this)?
-

Question

how many distances we need calculate with N points?

Question

how many distances we need calculate with N points? The answer could be getted with combinatorial thinking!

Question

how many distances we need calculate with N points? The answer could be getted with combinatorial thinking!

$$(d_1, d_2)$$

Question

how many distances we need calculate with N points? The answer could be getted with combinatorial thinking!

$$(d_1, d_2)$$

the first position could be occupied by N and the second for $N - 1$, and

both could be permutate two times, therefore:

$$\frac{N * (N - 1)}{2}$$

Ml model distance dependents

- K-nearest neigbhours
- Support vector machine
- K-means

Insights

More dimensions imply more features..

Lab Work

If more datapoints are need for holding classifers performance, then how many data points are needed?.
hint: keep constant the **avarage** distance.

curse of dimensionality

Is harder predict because is less likely that the test patterns are similar to training patterns. All this open the doors to reduce dimensionality.

Analize high dimensional data

The euclidean distance among two points $d(P, Q)$ is non-decreasing, namely; suppose d as number of featutres and $f_{i,k}$ is i-th feature for k-th individual.

$$\sum_{i=1}^{d+1}(f_{i,a} - f_{i,b})^2 \geq \sum_{i=1}^{d}(f_{i,a} - f_{i,b})^2 \tag{11}$$

this imply if the values are different then, $d(A, B) \longrightarrow \infty$ if $d \longrightarrow \infty$.