tqlf3yadu

April 29, 2024

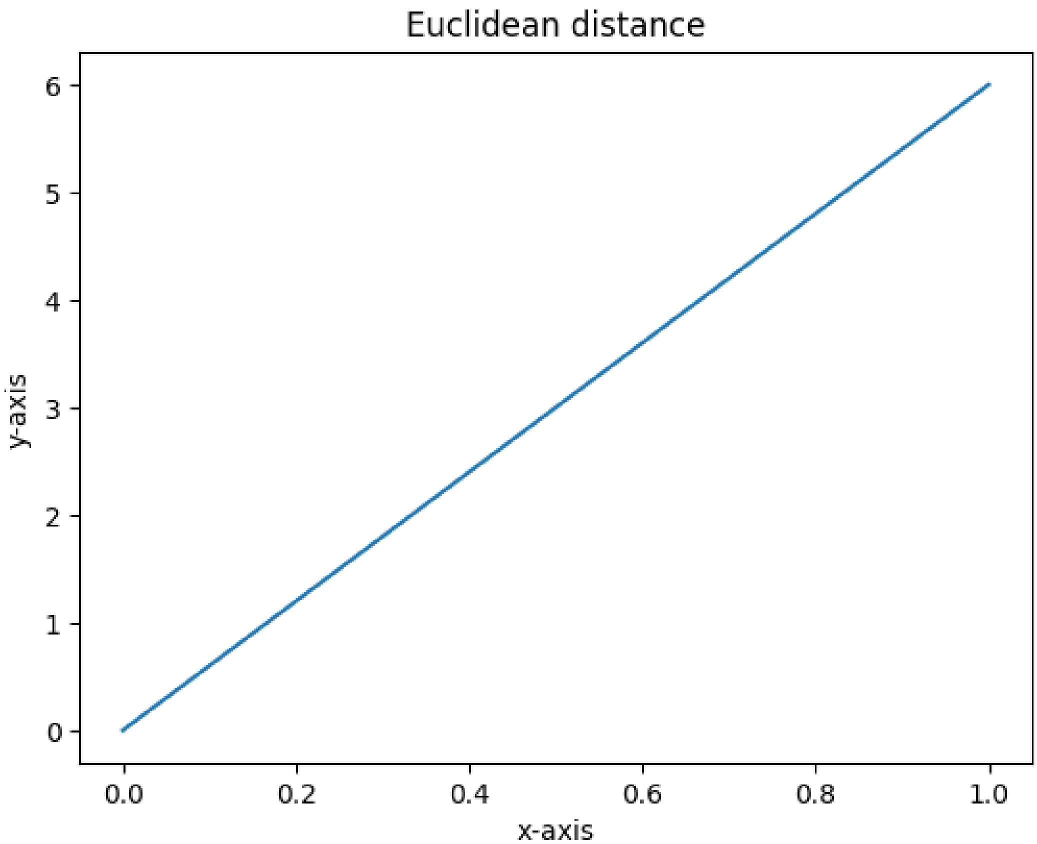
EUCLIDEAN DISTANCE

Euclidean distance represent the shortest distance between two points into the vector space.

The formula for calculating the distance is: Let suppose two points P(p1, p2, p3), Q(q1, q2, q3) The euclidean distance between these two points is thus represented as: d = ((q1 - p1)^2 + (q2 - p2)^2 + (q3 - p3)$^{2)}$(0.5)

```
[ ]: # importing the library
from scipy.spatial import distance
import numpy as np
import matplotlib.pyplot as plt

# defining the points
point_1 = np.random.randint(0,10,size=2)
point_2 = np.random.randint(0,10,size=2)

#calculating the euclidean distance between the given two points
euclidean_distance = distance.euclidean(point_1,point_2)
print('Euclidean Distance b/w', point_1, 'and', point_2, 'is: ',␣
 ↪euclidean_distance)

# plotting the points
plt.plot(point_1,point_2)
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Euclidean distance")

plt.show()
```

```
Euclidean Distance b/w [0 1] and [0 6] is:  5.0
```



MANHATTEN DISTANCE

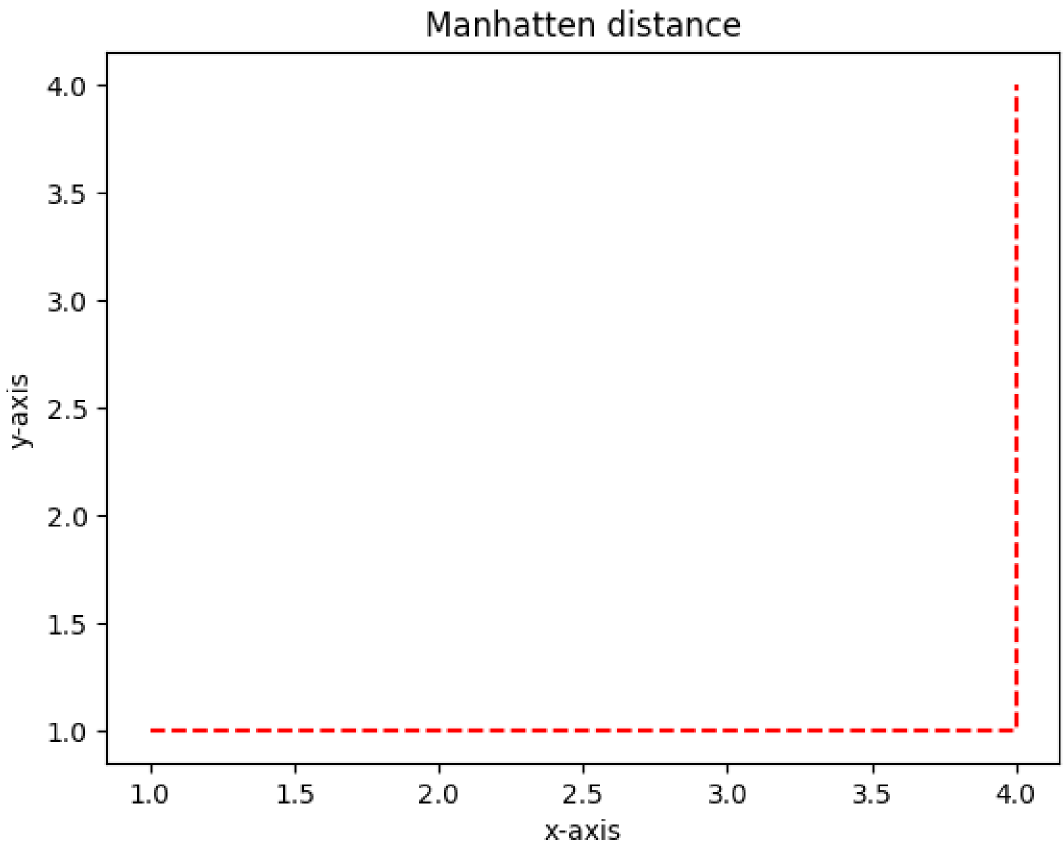
Manhatten Distance is also known as city block distance

which is given in terms of the coordinates as:

d= |(p1 - q1)| + |(p2 - q2)| + |(p3- q3)|

[ ]:
```
manhatten_distance = distance.cityblock(point_1, point_2)
print('Manhatten Distance b/w', point_1, 'and', point_2, 'is: ',␣
 ↪manhatten_distance)

# plotting the points
# plotting the points with Manhattan distance
plt.plot([1, 4, 4], [1, 1, 4], "r--")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Manhatten distance")

plt.show()
```

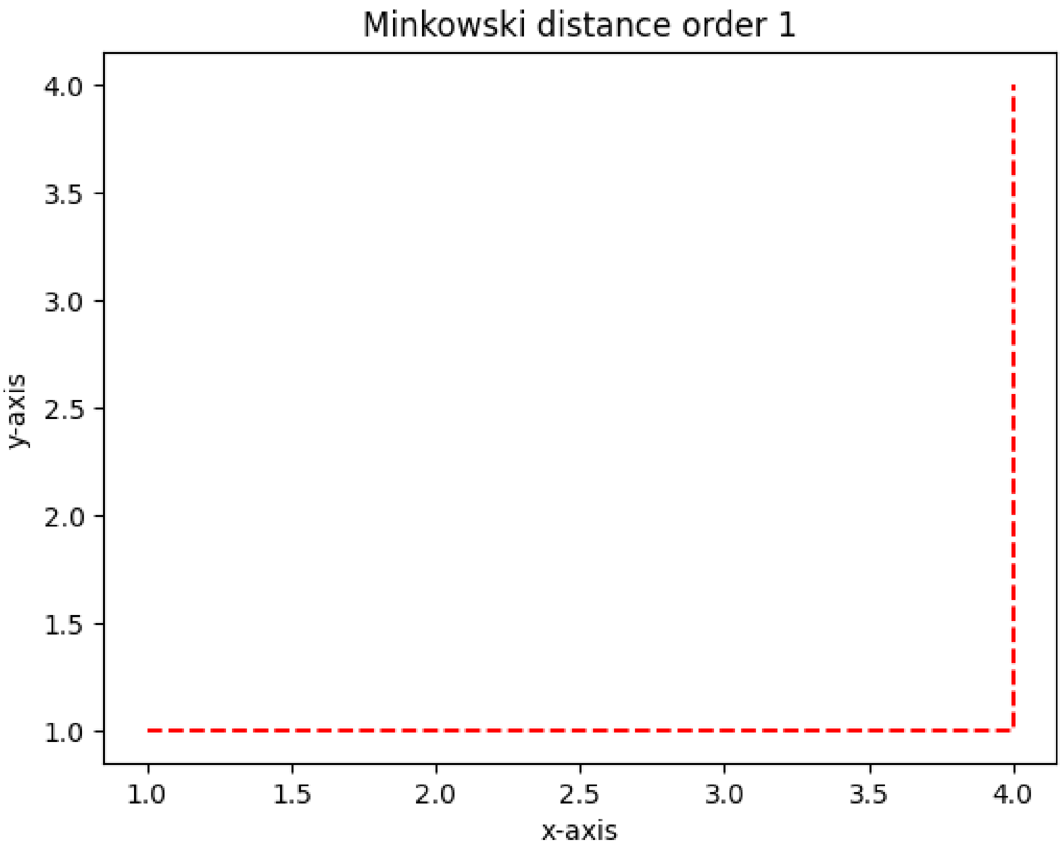
```
Manhatten Distance b/w [0 1] and [0 6] is:  5
```



MINKOWSKI DISTANCE ORDER 1:

[ ]:
```
minkowski_distance_order_1 = distance.minkowski(point_1, point_2, p=1)
print('Minkowski Distance of order 1:',minkowski_distance_order_1, '\nManhattan␣
↪Distance: ',manhatten_distance)

# plotting the points
# plotting the points with minkowski distance
plt.plot([1, 4, 4], [1, 1, 4], "r--")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Minkowski distance order 1")

plt.show()
```

```
Minkowski Distance of order 1: 6.0
Manhattan Distance:  6
```



MINKOWSKI DISTANCE ORDER 2:

[ ]:
```
minkowski_distance_order_2 = distance.minkowski(point_1, point_2, p=2)
print('Minkowski Distance of order 2:',minkowski_distance_order_2, '\nEuclidean␣
 ↪Distance: ',euclidean_distance)

# plotting the points
# plotting the points with minkowski distance
plt.plot((1,4),(1,4))
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Minkowski distance order 2")

plt.show()
```

```
Minkowski Distance of order 2: 4.242640687119285
Euclidean Distance:  4.242640687119285
```

[ ]:

COSINE DISTANCE/SIMILARITY

[ ]:
```
# import required libraries
import numpy as np
from numpy.linalg import norm

# define two lists or array
A = np.array([2,5,0,7,2,0,0,3])
B = np.array([1,2,0,2,2,0,2,4])

print("A:", A)
print("B:", B)

# compute cosine similarity
cosine = np.dot(A,B)/(norm(A)*norm(B))
print("Cosine Similarity:", cosine)
```

```
A: [2 1 2 3 2 9]
```

```
B: [3 4 2 4 5 5]
Cosine Similarity: 0.8188504723485274
```