Machine Learning
Blatt 8

Markus Vieth David Klopp Christian Stricker

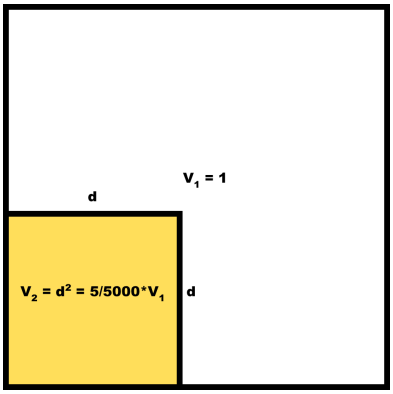
30. Juni 2016

Markus Vieth, David Klopp, Christian Stricker

Curse of Dimensionality

The nearest neighbor method breaks in high-dimensional spaces, because the "neighborhood" becomes very large for the Euclidean distance. This is called the curse of dimensionality. Suppose we have 5000 points uniformly distributed in the n-dimensional unit hypercube $C_n := [0,1]^n$ and we want to apply the 5-nearest neighbor algorithm. Suppose our query point is at the origin $(0,\dots,0)$, so, on average, we need to search 5/5000 of the hypercube's volume to capture the 5 nearest points.

1. Assume$n = 2, i.e. C_n$ is the unit square. What is the side length d of the square $[0,d]^2$ that on average captures the five nearest points to the origin?



$$d^2 = \frac{5}{5000}$$
$$\Leftrightarrow d = \sqrt{\frac{5}{5000}} = \sqrt{\frac{1}{1000}} = \sqrt{\frac{1}{10 \cdot 100}} = \frac{1}{\sqrt{10} \cdot 10} = \frac{\sqrt{10}}{100}$$

2. What is the side length d of the n-dimensional hypercube $[0,d]^n$ that on average captures the five nearest points to the origin?

$$d^n = \frac{5}{5000}$$
$$\Leftrightarrow d = \sqrt[n]{\frac{1}{1000}} = 1000^{-\frac{1}{n}}$$

3. For which number of dimensions n do we need a hypercube $[0,d]^n$ whose side length is larger than half the side length of C_n (i.e. 0.5) to capture the five nearest points?

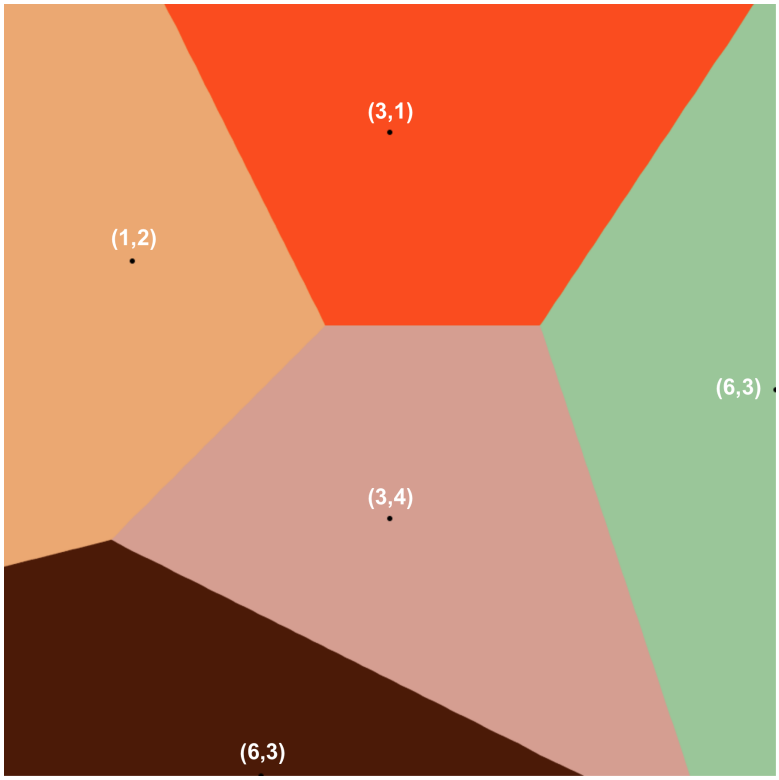
$$d \geq 0.5$$
$$\Leftrightarrow d^n \geq 0.5^n = d^n \geq 2^{-n}$$
$$\Leftrightarrow \frac{1}{1000} \geq 2^{-n}$$
$$\Leftrightarrow \log_2\left(\frac{1}{1000}\right) \geq \log_2\left(2^{-n}\right)$$
$$\Leftrightarrow \log_2\left(1000^{-1}\right) \geq -n \cdot \log_2 2$$
$$\Leftrightarrow \log_2\left(1000\right) \leq n$$
$$\Leftrightarrow n \geq 10$$

Markus Vieth, David Klopp, Christian Stricker

Voronoi Diagram

A two-dimensional data set contains the five points (1,2), (3,1), (6,3), (3,4), and (2,6).

1. Draw the Voronoi diagram for this data set.




```
1  import numpy as np
2  from scipy.spatial import Voronoi, voronoi_plot_2d
3  import matplotlib.pyplot as plt

5  points = np.array([[1, 4], [3, 5], [6, 3], [3, 2], [2, 0]])
6  vor = Voronoi(points)
7  voronoi_plot_2d(vor)
8  plt.show()
```

2. Why is it impractical to store the Voronoi diagram in order to speed up queries for k-nearest neighbor? (Two sentences)

Die Zuordnung eines Punktes zu einer bestimmten Nearest Neighbour Suchregion kann in einem Voronoi Diagram nicht effizient realisiert werden. Es resultiert also kein wirklicher Zeitgewinn in der Speicherung von Voronoi Diagrammen.