

# Voronoi y Delaunay

October 8, 2020

## 1 Diagramas de Voronoi y triangulación de Delaunay

Para calcular y dibujar el diagrama de Voronoi y la triangulación de Delaunay es necesario importar de la librería `scipy.spatial` y de `matplotlib` las siguientes funciones.

```
[49]: from scipy.spatial import Delaunay, Voronoi, voronoi_plot_2d
import matplotlib.pyplot as plt
import matplotlib.colors
import matplotlib as mpl
```

Además importaremos la librería `numpy` para poder generar arrays de puntos.

```
[50]: import numpy as np
```

El siguiente comando genera aleatoriamente 10 puntos en el plano.

```
[51]: points = np.random.rand(10,2)
```

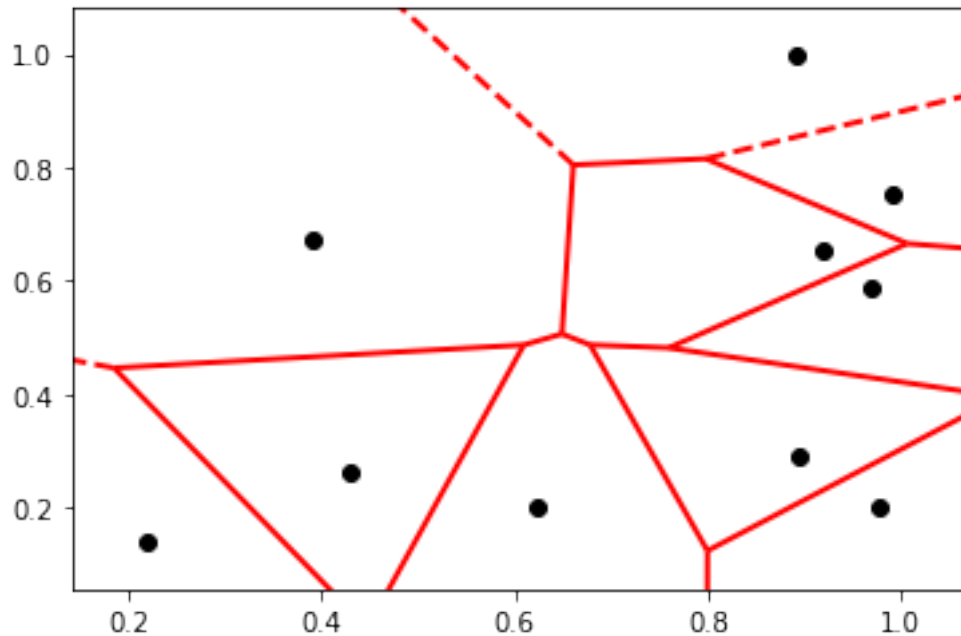
La siguiente expresión calcula el diagrama de Voronoi.

```
[52]: vor=Voronoi(points)
```

Si queremos representarlo gráficamente hacemos

```
[53]: fig = voronoi_plot_2d(vor, show_vertices=False, line_width=2, line_colors='red')
plt.plot(points[:,0], points[:,1], 'ko')
```

```
[53]: [<matplotlib.lines.Line2D at 0x7f9e05c2f9d0>]
```



Para calcular la triangulación de Delaunay asociada a los mismos puntos hacemos

```
[54]: Del=Delaunay(points)
```

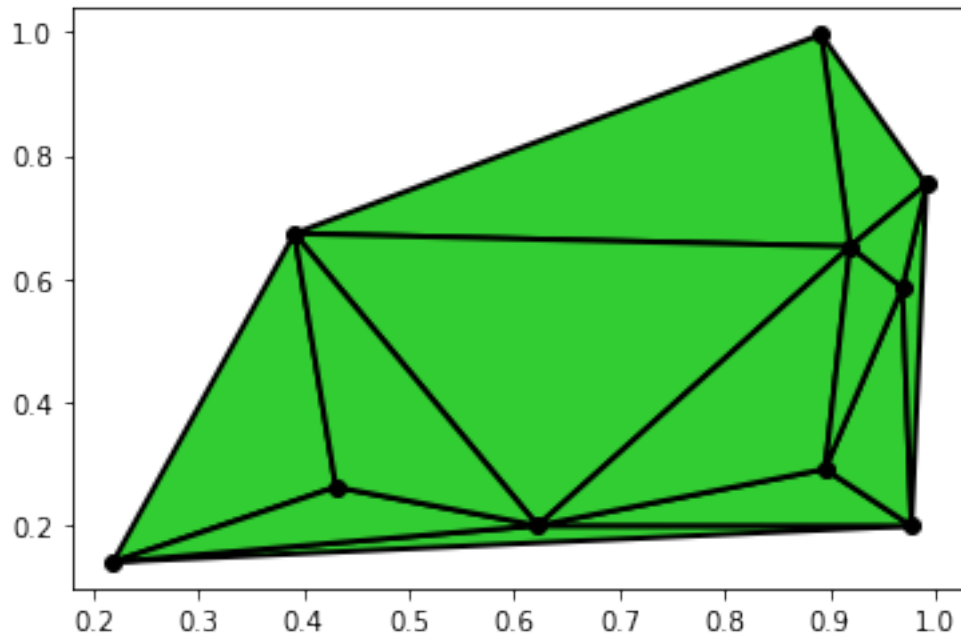
El siguiente comando accede al array de símlices maximales (triángulos) de la triangulación de Delaunay. ¡Ojo! Los vértices de los símlices no salen ordenados en orden lexicográfico

```
[55]: Del.simplices
```

```
[55]: array([[6, 3, 9],
            [1, 3, 6],
            [0, 3, 1],
            [2, 0, 7],
            [8, 0, 2],
            [0, 8, 3],
            [5, 1, 7],
            [0, 5, 7],
            [5, 0, 1],
            [4, 8, 2],
            [8, 4, 3],
            [4, 2, 9],
            [3, 4, 9]], dtype=int32)
```

Para representar gráficamente la triangulación de Delaunay, imprimiendo los triángulos en color verde y las aristas y vértices de color negro utilizamos la siguiente línea de comandos.

```
[56]: c=np.ones(len(points))
      cmap = matplotlib.colors.ListedColormap("limegreen")
      plt.tripcolor(points[:,0],points[:,1],Del.simplices, c, edgecolor="k", lw=2,
      ↪cmap=cmap)
      plt.plot(points[:,0], points[:,1], 'ko')
      plt.show()
```



Si queremos representar la triangulación de Delaunay sobre el diagrama de Voronoi hacemos lo siguiente.

```
[57]: fig = voronoi_plot_2d(vor,show_vertices=False,line_width=2, line_colors='blue' )
      c=np.ones(len(points))
      cmap = matplotlib.colors.ListedColormap("limegreen")
      plt.tripcolor(points[:,0],points[:,1],Del.simplices, c, edgecolor="k", lw=2,
      ↪cmap=cmap)
      plt.plot(points[:,0], points[:,1], 'ko')
      plt.show()
```

