

Plataforma web para medir la efectividad del RTree de libspatialindex de Python en reconocimiento facial

Por José Ignacio Huby, Renato Bacigalupo y Juan Gálvez

Introducción

Objetivo: implementar una página web que permita visualizar la rapidez del RTree en un escenario de comparación de vectores de características de rostros humanos

No es un objetivo: implementar una aplicación de reconocimiento facial para uso comercial

Diseño

Tipo de búsqueda: KNN eucl
(los k vecinos más cercanos)

Comparador: KNN secuencial
(uno contra todos)

Cantidades: 100, 200, 400, 800,
1600, 3200, 6400, 13171

(~ el doble del anterior)

(permite evaluar evolución exponencial)

Metodología

Paradigma: Orientado a objetos

Frontend: Vue

Backend: Flask

Implementación

class KNNClassifierRTree

Los k más
ceranos

El que más
se repite

```
def classify(self, vec, k):
    nearest = list(self.idx.nearest(vec.tolist()*2, k))
    if len(nearest) > 0:
        fr_dict = {}
        for n in nearest:
            label = self.train_set[n][0]
            if label in fr_dict:
                fr_dict[label] += 1
            else:
                fr_dict[label] = 1
        return sorted(fr_dict.items(), key = lambda x : x[1])[-1]
    else:
        return "No se parece a nadie", 0
```

class KNNClassifierSeq

Los k más
ceranos

El que más
se repite y
tiene menor
distancia

```
def classify(self, vec, k, max_dist):
    max_heap = []
    for i in range(k):
        max_heap.append((-max_dist, "none"))
    for train_label, train_vec in self.train_set:
        distance = -((sum([(x1 - x2)**2 for x1, x2 in zip(vec, train_vec)]))**(1/2))
        if max_heap[0][0] < distance:
            heappop(max_heap)
            heappush(max_heap, (distance, train_label))
    nearest = [(distance, label) for distance, label in max_heap if label != "none"]
    if len(nearest) > 0:
        fr_dict = {}
        for distance, label in nearest:
            if label in fr_dict:
                fr_dict[label][0] += 1
                if fr_dict[label][1] < distance:
                    fr_dict[label][1] = distance
            else:
                fr_dict[label] = [1, distance]
        fr_list = sorted(fr_dict.items(), key = lambda x : x[1])
        return fr_list[-1][0], fr_list[-1][1][0]
    else:
        return "No se parece a nadie", 0
```

Inicialización

```
train_set = read_train_set()

knn_s = RangeContainers(KNNClassifierSeq, train_set, [(0, 100), (0, 200), (0, 400), (0, 800), (0, 1600),
(0, 3200), (0, 6400), (0, 13171)])

knn_r = RangeContainers(KNNClassifierRTree, train_set, [(0, 100), (0, 200), (0, 400), (0, 800),
(0, 1600), (0, 3200), (0, 6400), (0, 13171)])
```


Uso

```
obj.call_container(rango, lambda x : x.classify(args))
```



knn_r



(0, 100)



vec, k

Extra

class HyperCubeClassifierRTree

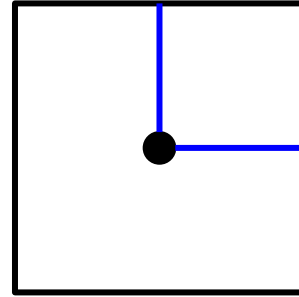
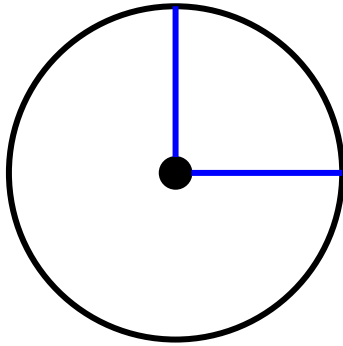
Los k más
ceranos

El que más
se repite

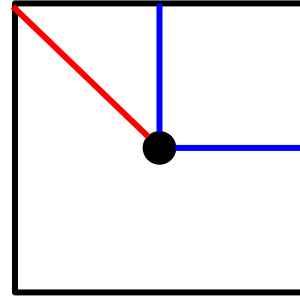
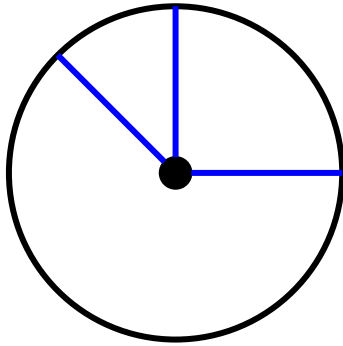
```
def classify(self, vec, r):
    lis = vec.tolist()
    nearest = list(self.idx.intersection([x-r for x in lis] + [x+r for x in lis]))
    if len(nearest) > 0:
        fr_dict = {}
        for n in nearest:
            label = self.train_set[n][0]
            if label in fr_dict:
                fr_dict[label] += 1
            else:
                fr_dict[label] = 1
        return sorted(fr_dict.items(), key = lambda x : x[1])[-1]
    else:
        return "No se parece a nadie", 0
```

Es una búsqueda por rango...

Es una búsqueda por rango...
solo que no es circular :)



Es una búsqueda por rango...
solo que no es circular :))



Es más rápido, pero reduce un poco la efectividad

Demo

Resultados

PROYECTO 3 - FACE RECOGNITION

prueba.jpg



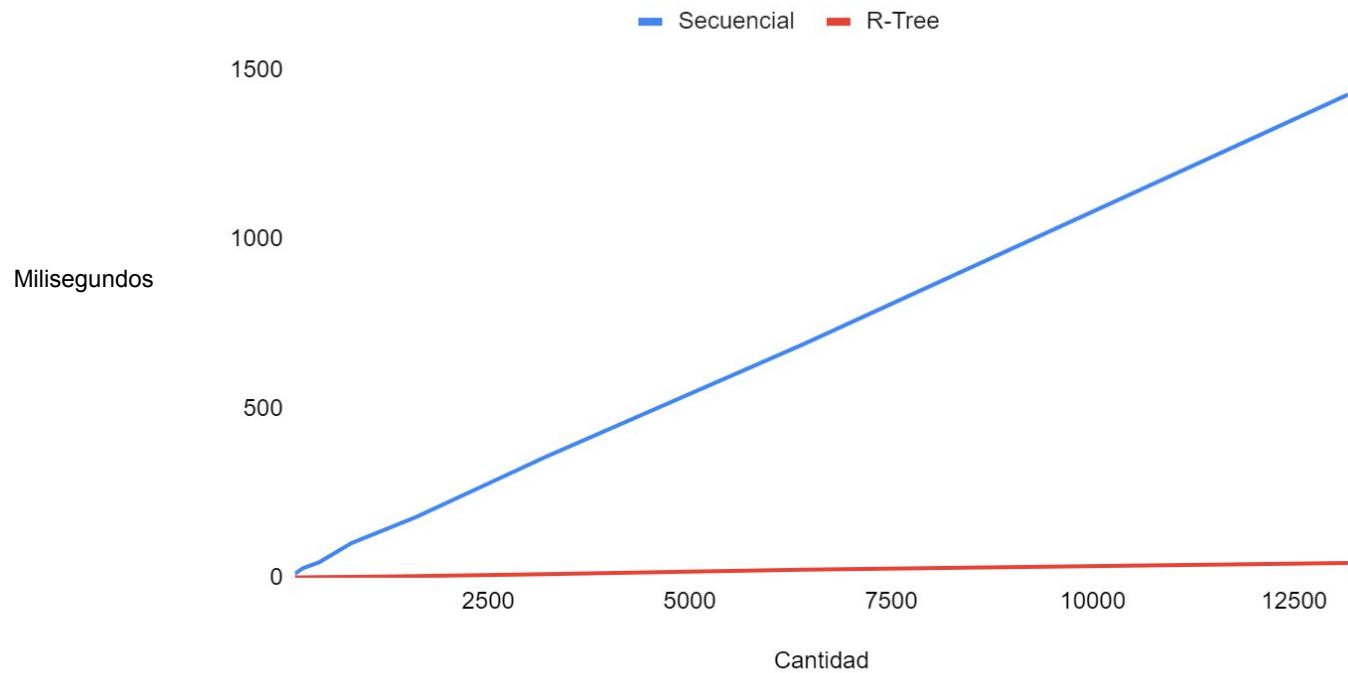
Renato Bacigalupo

Juan Garzón

Jose Huby

Tiempos en ms para $K = 8$

Tam de la BD	Secuencial	R-Tree
100	12	1
200	28	1
400	45	2
800	102	3
1600	179	5
3200	356	11
6400	689	24
13171	1428	44



Medición extra en ms...

Tam de la BD	HyperCube
100	< 1
200	< 1
400	< 1
800	< 1
1600	1
3200	3
6400	6
13171	10

Cuándo usar qué

¿Tu base de datos está dividida en pequeños grupos?

Secuencial

Asistencia a clases con reconocimiento facial

¿Tu base de datos es inmensa?

RTree

Sistema de hipervigilancia a nivel de ciudad o país

RTree sobre
secuencial, pero
considerar otras
alternativas (ej: **KD**)

Seguimiento a empleados en una gran fábrica