FLOP A Free Laboratory Of Programming

Skip Menu

- Home
- Practice Area
- Manage Collections
- Newbies Area
- Suported programming languages
- Join & us
- Resources

Geometría Computacional 2016
Práctica 1

- Fecha límite de entrega: **viernes 11 de marzo de 2016 a las 23:55**
- Test de conocimientos sobre la materia vista hasta ahora: **miércoles 16 de marzo** en horario de clase.
- Forma de entrega: se informará oportunamente en el Campus Virtual
- La práctica se realizará en un módulo python cuyo nombre será `fisher.py`
- El módulo comenzará importando la división y numpy (y ningún otro módulo de python) y contendrá la siguiente clase con los siguientes dos métodos (aparte de otros atributos y métodos que se quieran incluir en ella) como se indica a continuación. No debe haber nada en el módulo que permita la identificación de sus autores dado que, posteriormente, se llevará a cabo un sistema de evaluación

por pares.

```
# -*- coding: utf-8 -*-

from __future__ import division
import numpy as np

class Fisher:

    def train_fisher(self, X0, X1):
        pass #incluye aquí tu código

    def classify_fisher(self, X):
        pass #incluye aquí tu código
```

La clase `Fisher` implementará el algoritmo de Fisher para clasificación de dos clases de puntos con el umbral que minimiza la probabilidad de error, tal y como se vio en clase. El método train_fisher no devolverá nada, se limitará a hacer los cálculos que sean necesarios y guardarlos en atributos del objeto que se cree.

- Los datos de entrenamiento $X0$, $X1$ y X serán dos listas de puntos de dimensión y longitudes arbitrarias, es decir, de la forma `[[x0,...,xD],...,[z0,...,zD]]`
- El resultado que devolverá el método `classify_fisher` será una lista de clases a las que pertenece cada punto de X, es decir, una lista de la forma `[c0,c1,...,cM]` en donde cada `ci` es 0 o 1.
- Se valorará la corrección de los resultados obtenidos y la velocidad de ejecución. Se valorará también la claridad del código: si los nombres de variables, métodos, etc. son adecuados, si hay funciones demasiado complejas que deberían haberse escrito en partes conceptualmente más simples, si se han respetado las directrices de PEP8, etc.

File Examinar… No se ha seleccionado ningún archivo. Submit