Datasets para experimentación en Big Data

Todos los datos se estandarizan con media 0 y varianza unidad. Para las ejecuciones, los parámetros utilizados son:

- `numPartitions`: $20 \cdot 19 = 380$
- `numPartitionsPerGroup`: 19
- `num-executors`: 20
- `executor-cores`: 19
- `executor-memory`: 46g

Clasificación

Disponemos de los siguientes algoritmos para resolver un problema de clasificación multiclase:

- **Algoritmo de Chiu**. Versión adaptada a clasificación del algoritmo *subtractive clustering* de Chiu et al., donde se asigna a cada punto la clase del centroide al que tenga mayor pertenencia. Viene determinado por un parámetro r_a que define el tamaño efectivo del vecindario de cada punto. A su vez se consideran tres variantes:
  - La versión global, que utiliza todos los datos. Nos referiremos a ella como `ChiuG`.
  - La versión local, que trabaja por particiones y luego concatena los resultados, llamada `ChiuL`.
  - Una versión intermedia, que trabaja por particiones y luego integra los resultados de varias particiones aplicando sobre ellos una variante del algoritmo global. La llamamos `ChiuI`.
- **Fuzzy C Means**. Versión adaptada a clasificación del conocido algoritmo de *clustering*, donde se asigna a cada punto la clase mayoritaria del centroide al que tenga mayor pertenencia, tras un α-corte de 0.6. Existen cuatro versiones:
  - Una primera versión en la que la inicialización de los centroides es aleatoria, y se eligen tantos como el número de clases. Lo llamamos simplemente `FCM`.
  - Tres versiones en las que la inicialización se realiza mediante una de las tres versiones del algoritmo de Chiu: `FCM + ChiuG`, `FCM + ChiuL` y `FCM + ChiuI`.
- **Random Forest**. Primer algoritmo de comparación, con 200 árboles, referido como `RF`.

- **Regresión Logística con SGD.** Segundo algoritmo de comparación, referido como `RLog`.
- **Perceptrón multicapa.** Tercer algoritmo de comparación, abreviado como `MLP`.

La métrica utilizada para evaluar la bondad de los modelos será el porcentaje de instancias mal clasificadas o *error de clasificación* en el conjunto de test, abreviado como `eclass`. Mostraremos una tabla como la siguiente para cada conjunto de datos:

Algoritmo		**Tiempo (m)**	**eclass (%)**	**Nº Centroides**
`ChiuG`	$r_a = 0.3$			
	$r_a = 1.0$			
	$r_a = ...$			
...		...	...	...

Tabla 1: Ejemplo de resultados para clasificación en un conjunto de datos.

Kitsune

Disponible en el repositorio UCI. Se trata de un conjunto de ciberseguridad que contiene información sobre el tráfico de paquetes relacionado con 9 ataques distintos sobre un sistema de IoT.

- Hay un total de **23.788.873** instancias.
- Hay *dos etiquetas* de clase: paquete malicioso (1) o benigno (0).
- Cada instancia tiene **115** características numéricas que representan información estadística de los ataques. En concreto, se realizan 23 medidas diferentes en 5 ventanas de tiempo.

HEPMASS

Disponible en el repositorio UCI. Se trata de una serie de observaciones sobre colisiones de partículas usadas para detectar nuevas partículas. El objetivo es distinguir qué colisiones producen partículas y cuáles no. Las características de este conjunto son:

- Hay un total de **10.500.000** instancias.
- Hay **dos etiquetas** de clase: 1 para colisión exitosa, 0 para no exitosa.
- Cada instancia tiene **27 características** pertinentes al experimento (22 de bajo nivel y 5 de alto nivel).
- Se divide en **7.000.000** de ejemplos de entrenamiento y **3.500.000** ejemplos de test.

Los resultados obtenidos son los siguientes:

Algoritmo		Tiempo (m)	eclass (%)	Nº Centroides
RF		1.17	9.317	-
RLog		0.55	9.351	-
FCM		1.91	50.022	2
ChiuI	$r_a = 2.0$			
FCM + ChiuI	$r_a = 2.0$			

Tabla 2: Resultados para clasificación en HEPMASS.

COMET_MC

Disponible en openml. Se trata como en el caso anterior de estudiar un proceso físico entre partículas que puede resultar en una señal de activación o no.

- Hay un total de **7.619.400** instancias.
- Hay **dos etiquetas** de clase: 1 para activación, 0 para *background*.
- Cada instancia tiene **2 características** relacionadas con el experimento: la energía y el tiempo relativo.
- Lo dividimos en **5.333.580** ejemplos de entrenamiento y **2.285.822** ejemplos de test.

Regresión

Consideramos los siguientes algoritmos de regresión:

- **Algoritmo de Chiu**. Adaptación a un sistema de regresión e identificación de modelos a partir del algoritmo de *subtractive clustering*. Se trata del modelo de "orden 0" mencionado en el paper original. Nos referiremos a cada una de las tres versiones disponibles como `MI + ChiuG`, `MI + ChiuL` y `MI + ChiuI`.
- **Algoritmo de Wang-Mendel**. Construye de forma simple un sistema basado en reglas difusas para predecir una salida a partir de unos datos de entrada. Lo abreviamos como `WM`.
- **Regresión lineal**. Primer algoritmo de comparación, `RL`.
- **Random Forest**. Segundo algoritmo de comparación, `RF`.

La métrica para evaluar la precisión del modelo será el *error cuadrático medio* entre las predicciones y los valores reales en el conjunto de test, denotado como `mse`. Mostraremos una tabla como la siguiente para cada conjunto de datos:

Algoritmo		Tiempo (m)	mse (%)	Nº Centroides
MI + ChiuG	$r_a = 0.3$			
	$r_a = 1.0$			
	$r_a = ...$			
...		...	...	...

Tabla 3: Ejemplo de resultados para regresión en un conjunto de datos.

Gas mixtures

Disponible en el repositorio UCI. Contiene datos de 16 sensores químicos expuestos a dos mezclas de gases en varias concentraciones: una basada en etileno + CO y otra en etileno + metano.

- Hay un total de **4.178.504** instancias.
- Podemos buscar hacer regresión sobre **una o dos variables**: la concentración de etileno y/o la concentración de CO/metano.
- Cada instancia consta de **17 atributos**: 16 medidas de de los sensores junto con el tiempo de medición.