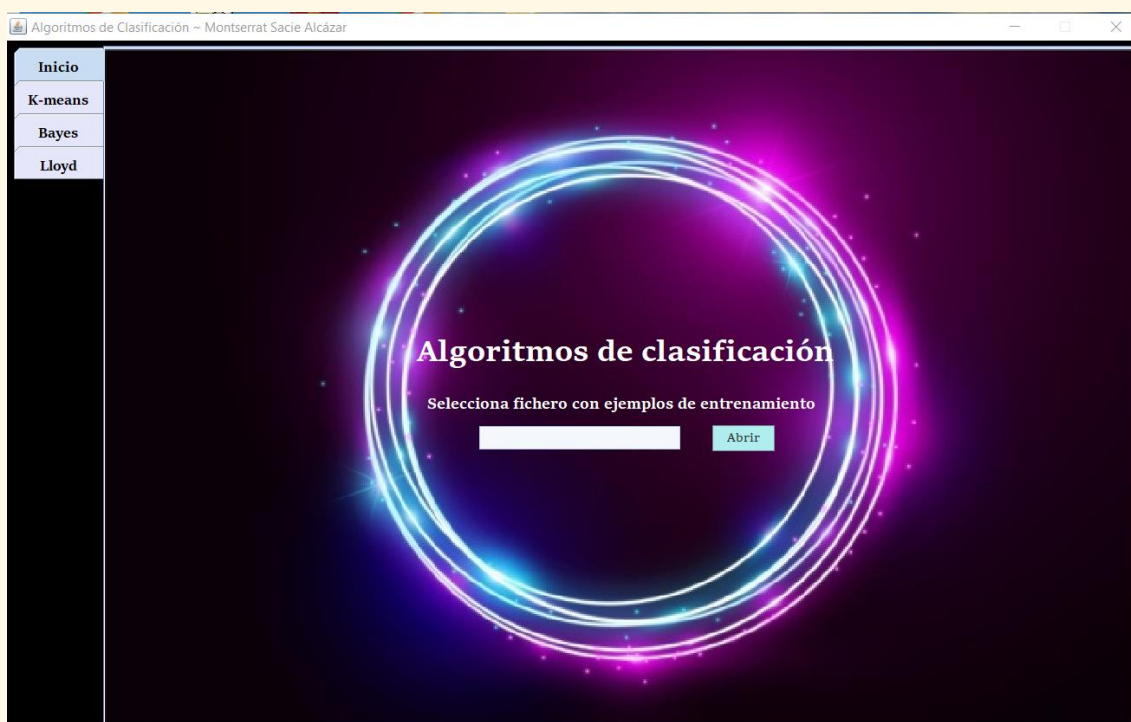




PRÁCTICA 3

IMPLEMENTACIÓN DE ALGORITMOS DE CLASIFICACIÓN



Montserrat Sacie Alcázar

Asignatura: Ingeniería del Conocimiento

Ingeniería del Software

UCM, 2020



Índice

1.Introducción

1.1 Explicación teórica de los algoritmos

1.1.1 K_Medias borroso

1.1.2 Bayes

1.1.3 Lloyd

2.Detalles de la implementación

3.Manual de Usuario

4.Ejecución y resultados



1.Introducción

El objetivo de la práctica 3 ha sido implementar 3 algoritmos de clasificación que nos permiten calcular los parámetros de clasificación dado un fichero con los ejemplos para el aprendizaje y posteriormente clasificar nuevos ejemplos dados.

En la memoria se explican:

- En este **primer apartado**, una introducción teórica de cada uno de los algoritmos de clasificación implementados; para facilitar la comprensión de la implementación explicada en el siguiente apartado.
- En el **apartado 2**, los **detalles de la implementación** y la **distribución de clases de código del proyecto** implementado en Eclipse.
- El **apartado 3** explica los **detalles de la interfaz gráfica para ejecutar la práctica**. En la entrega se añade un video explicativo de ejecución del programa para mayor claridad.
- Por último, en el **apartado 4** se muestra el resultado de la ejecución para el fichero de entrada de este programa, cuya estructura también será expuesta.

1.1 Explicación teórica de los algoritmos

En esta práctica se han implementado tres algoritmos de clasificación:

- Algoritmo "K_means" o K_medias (borroso)
- Algoritmo de Bayes
- Algoritmo de Lloyd

1.1.1 K_Medias borroso

El **algoritmo K_medias** es un método de agrupamiento, cuyo objetivo es particionar un conjunto de n observaciones en k grupos. Así cada observación pertenece al grupo cuyo centro o valor medio es más cercano

Dado que en el fichero de ejemplos se proporcionan muestras clasificadas en dos clases, tenemos que $k = 2$.

Desde un punto de vista teórico estricto, este algoritmo se encuentra dentro del grupo de **Aprendizaje no Supervisado** o *Unsupervised Learning*. Sin embargo, en la práctica nosotros hemos establecido el número de clases k previamente, luego ya lo estamos supervisando de alguna forma.

Los **parámetros** que deben ser inicializados antes de la ejecución del algoritmo son:

- Centros de las clases iniciales
- Tolerancia (ϵ)
- Peso exponencial (b)



La **entrada del algoritmo** consiste en un fichero con la lista de ejemplos que toman una serie de valores por cada variable de entrada de interés.

Este **algoritmo iterativo** actualiza en cada iteración los centros de las clases y la matriz de grados de pertenencias de los ejemplos o muestras a cada una de las clases (U). Se deja de iterar cuando se cumple el criterio de finalización.

Las **ecuaciones** que permiten estos cálculos y que han sido implementadas de forma precisa son las siguientes:

Ecuaciones:

$$v_i = \frac{\sum_{j=1}^n [P(c_i / \mathbf{x}_j)]^b \mathbf{x}_j}{\sum_{j=1}^n [P(c_i / \mathbf{x}_j)]^b} \quad i = 1, \dots, c \quad (1)$$
$$P(v_i / \mathbf{x}_j) = \frac{1/d_{ij}^{1/(b-1)}}{\sum_{r=1}^c 1/d_{rj}^{1/(b-1)}} \quad \text{y} \quad d_{ij} = \|\mathbf{x}_j - v_i\|^2 \quad i = 1, \dots, c; j = 1, \dots, n \quad (2)$$
$$\Delta = \|\mathbf{v}_i^{(t+1)} - \mathbf{v}_i^{(t)}\| < \varepsilon \quad \text{criterio finalización}$$

Ingeniería del Conocimiento
Gonzalo Pajares 7

Imagen capturada de las transparencias de la Asignatura de IC, creadas por Gonzalo Pajares

1.1.2 Bayes

El **algoritmo de clasificación de Bayes** es otro de los algoritmos de *Machine Learning*, enmarcado dentro de los algoritmos de Aprendizaje Supervisado.

No necesita una inicialización previa de parámetros y la entrada del algoritmo es el fichero con los **ejemplos multivariable y las clases a las que pertenecen** dichos ejemplos (conjunto de x)

Los **parámetros aprendidos** por el algoritmo son:

- Los **centros iniciales** de las clases. Son los puntos medios de entre el conjunto de muestras asociados a cada clase, dados por la siguiente expresión:

$$\hat{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

Imagen capturada de las transparencias de



la Asignatura de IC, creadas por Gonzalo Pajares

- La **matriz de Covarianzas**, **C**, asociada a cada clase, que se calcula de la siguiente forma:

$$C = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mathbf{m}})(\mathbf{x}_i - \hat{\mathbf{m}})^t$$

Para clasificar un nuevo ejemplo se ha implementado la **función de máxima verosimilitud**

$$f(\mathbf{x}_i, \mathbf{w}) = p(\mathbf{x}_i / \mathbf{m}, C) = \frac{1}{(2\pi)^{d/2} |C|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_i - \mathbf{m})^t C^{-1} (\mathbf{x}_i - \mathbf{m}) \right\}$$

Imagen capturada de las transparencias de la Asignatura de IC, creadas por Gonzalo Pajares

1.1.3 Lloyd

El **algoritmo de Lloyd** se encarga de encontrar conjuntos espaciados uniformes de puntos en subconjuntos de espacios euclídeos.

Los **parámetros de entrada** que recibe el algoritmo son:

- Centros de las clases iniciales
- Tolerancia (ϵ)
- Razón de aprendizaje = 0.1

Y en este caso establecemos el número máximo de iteraciones *kmax* o recorrido de todos los ejemplos del fichero de entrada (actualizando los centros cada vez que se lee un nuevo ejemplo).

Se establece un **criterio de finalización** de forma que si se cumple antes de *kmax* el algoritmo no realiza más iteraciones.



Algoritmo de Lloyd (algoritmo competitivo)

- 1) **Inicio:** dados los puntos de datos $\mathbf{x}(k)$, $k = 1, 2, \dots$,
y centros de salida iniciales $\mathbf{c}_j(0)$, $j = 1, \dots, m$.
- 2) Determinar el centro $\mathbf{c}_j(k)$ más próximo al punto $\mathbf{x}(k)$ (competición)
$$j = \arg \min_j \|\mathbf{x}(k) - \mathbf{c}_j(k)\|$$
- 3) Actualizar el centro de salida utilizando las ecuaciones,
$$\mathbf{c}_j(k+1) = \mathbf{c}_j(k) + \gamma(k) [\mathbf{x}(k) - \mathbf{c}_j(k)]$$
$$k = k + 1$$

 $\gamma(k)$ razón de aprendizaje

Ejecución del algoritmo. Imagen capturada de las transparencias de la Asignatura de IC, creadas por Gonzalo Pajares

2. Detalles de la implementación

Los algoritmos se han implementado en el lenguaje de programación **Java** y usando el entorno de desarrollo **Eclipse**.

Las **funcionalidades básicas** implementadas son:

- a) Lectura del fichero de entrada (se selecciona del directorio en el que lo tengamos desde la interfaz gráfica) y agrupación de los datos como pertenecientes a cada una de las clases correspondientes.
- b) Implementación de un método de clasificación.
- c) Proporcionar al programa nuevos ejemplos para su clasificación, almacenados en los ficheros *TestIris01.txt*, *TestIris02.txt* y *TestIris03.txt*

Por otro lado, las ampliaciones **implementadas** son:

- d) Implementación de los 3 algoritmos propuestos.
- e) Comprobar el correcto funcionamiento de cada algoritmo para los nuevos ejemplos almacenados en los tres ficheros mencionados anteriormente.

Extra:

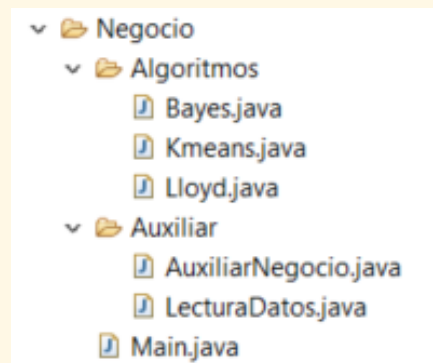
- f) Comprobación del funcionamiento de los algoritmos desde el *Main* de la aplicación, cogiendo como entrada los datos de la Hoja 2 de ejercicios.

Para la programación de los algoritmos se han implementado las fórmulas completas expuestas en el apartado 1 de la memoria



Estructura del proyecto

Negocio: clases con la implementación de la lógica del programa

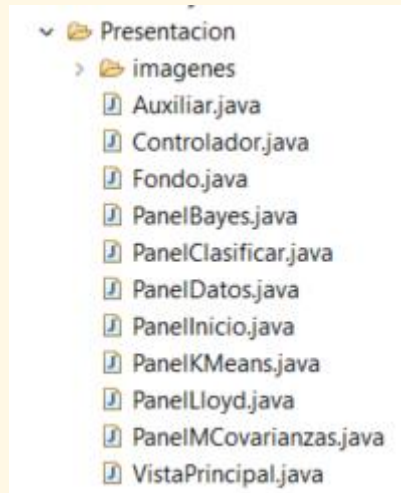


- **Main.java:** clase principal a ejecutar por el usuario en Eclipse para lanzar el programa. Esta invoca a la ventana principal de la interfaz gráfica. En el main también está incluida la ejecución de los ejercicios de la Hoja 2 de clase, cuyos resultados se muestran por consola.
- **Bayes.java:** reúne todos los métodos para la implementación del algoritmo de Bayes. Los datos se guardan en un `HashMap<String, ArrayList<double[]>>`, donde se guarda para cada clase (String) el conjunto de muestras o ejemplos. También contiene los centros (`double[]`) y las matrices de Covarianza (de tipo Matrix) almacenadas en dos `HashMap` según la clase a la que están asociados.
- **Kmeans.java:** Tiene atributos para cada uno de los parámetros y datos de entrada y métodos para los cálculos implicados en el algoritmo de K_medias borroso. Por defecto, la tolerancia toma el valor 0.01 y la b (peso exponencial) vale 2.
- **Lloyd.java:** implementación del algoritmo de Lloyd. Por defecto la tolerancia vale 10^{-10} , el número máximo de iteraciones es 10 y la razón de aprendizaje es 0.1.
- **AuxiliarNegocio.java:** Dado que se ha trabajado con objetos de tipo Matrix en la implementación de Bayes y K_means, se han implementado en esta clase métodos auxiliares para obtener por ejemplo una matriz columna a partir de una lista de double (`double[]`) o inicializar una matriz cuadrada dado la dimensión, con ceros. Estos métodos son complementarios a los ya implementados en la librería Jama instalada para el uso de Matrix. También se incluye un método para el redondeo de decimales.



- **LecturaDatos.java:** Contiene un método que reciben un String con la ruta del fichero.txt con los ejemplos de entrenamiento, lo abre, lo lee y devuelve un ArrayList de double[] con los ejemplos.

Presentación: para la implementación de la interfaz gráfica se ha utilizado la herramienta *WindowBuilder* en Eclipse.



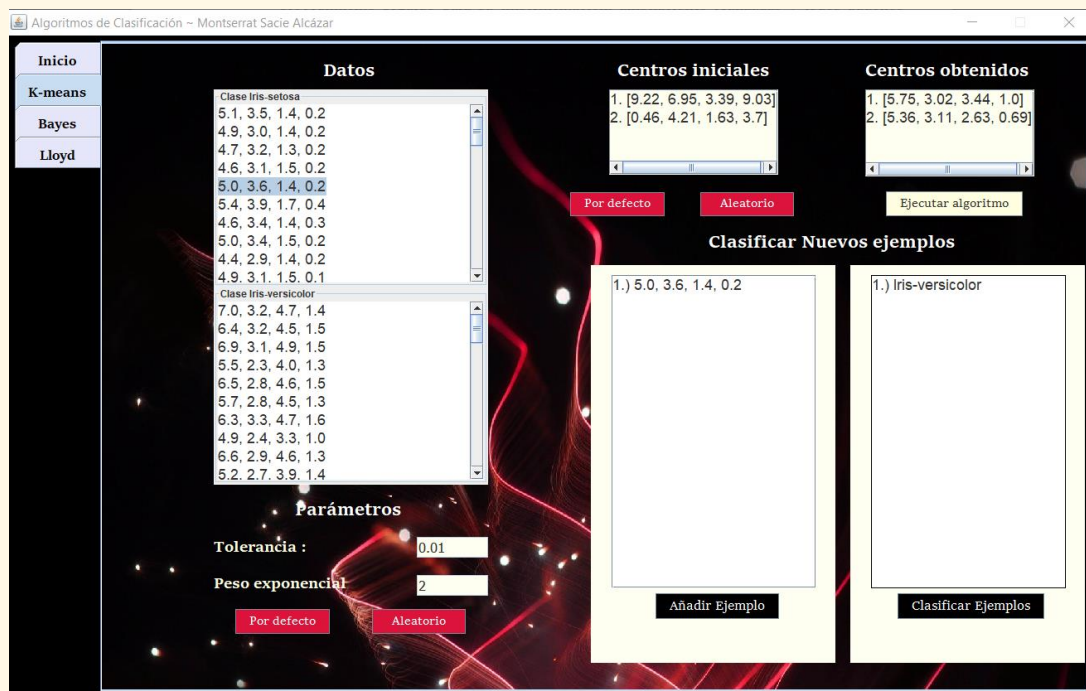
En la carpeta de imágenes se encuentran las imágenes .png utilizadas como fondo de los paneles principales incluidos en la vista Principal.

- **Auxiliar:** Incluye métodos para obtener una String a partir de una matriz, un ArrayList con los centros o el ArrayList de datos.
- **Controlador.java:** Clase que comunica Negocio con Presentación. Contiene una instancia actualizada de los datos cargados del fichero Iris2Clases.txt así como almacena los nuevos ejemplos añadidos por el usuario. Se encarga de inicializar y actualizar los atributos de cada uno de los algoritmos (también instanciados como atributos en esta clase) ante cualquier cambio realizado en la interfaz por el usuario.
- **Fondo.java:** JPanel cuya única utilidad es colocar una imagen de fondo.
- **PanelBayes.java:** Clase que extiende a Fondo (al igual que los paneles de k-Means y Lloyd). Este panel incluye un botón para ejecutar el algoritmo de Bayes, así como JTextArea para mostrar los centros y matrices de Covarianza obtenidos tras la ejecución. Incluye un PanelDatos así como un PanelClasificar explicados a continuación.



- **PanelClasificar.java:** Implementación de un panel en el que si pulsamos el botón “Añadir Ejemplo” aparece una ventana pop-up donde poder añadirlo. Los ejemplos añadidos se muestran uno tras otro en un JTextArea y no se eliminan (tanto visualmente como en memoria) hasta que el usuario cambie a un panel de otro algoritmo o el panel de Inicio.
- **PanelDatos.java:** Incluye de forma dinámica verticalmente un JTextArea por cada clase con datos cargada. Si no hemos cargado ningún fichero con ejemplos aparecerá vacío.
- **PanelInicio.java:** panel donde se permite abrir un fichero con los ejemplos de entrenamiento. Deben tener el formato especificado en esta memoria.
- **PanelKMeans.java:** es uno de los paneles principales. Además de ejecutar el algoritmo o clasificar nuevos ejemplos, permite poder generar aleatoriamente o recuperar los valores por defecto de los parámetros (tolerancia y peso exponencial) así como generar aleatoriamente o por defecto los centros iniciales*.
- **PanelLloyd.java:** panel prácticamente igual que el PanelKMeans.java. El único cambio es el nombre de uno de los parámetros que recibe (razón de aprendizaje)
- **PanelMCovarianzas.java:** panel que añade dinámicamente en horizontal un JTextArea por cada matriz de Covarianzas calculada. Este panel se incluye únicamente en PanelBayes.java
- **VistaPrincipal.java:** JFrame que contiene un JTabbedPane donde se añaden los paneles principales:
 - PanelInicio
 - Panel K-means
 - Panel Bayes
 - Panel Lloyd

***Nota:** Aunque la implementación del programa permite generar los centros aleatoriamente esto puede llevarnos a confusión. Al modificar los centros c1 y c2, puede ocurrir que un punto clasificado como “Iris-setosa” por estar más próximo al centro por defecto c1, pase a estar más cerca del nuevo centro c2 que del nuevo c1. Por lo tanto, el programa lo clasifica como “iris-versicolor”. Este funcionamiento es el esperado como correcto, pero los ejemplos ya no se encuentran clasificados tal y como aparece el fichero *Iris2Clases.txt*.

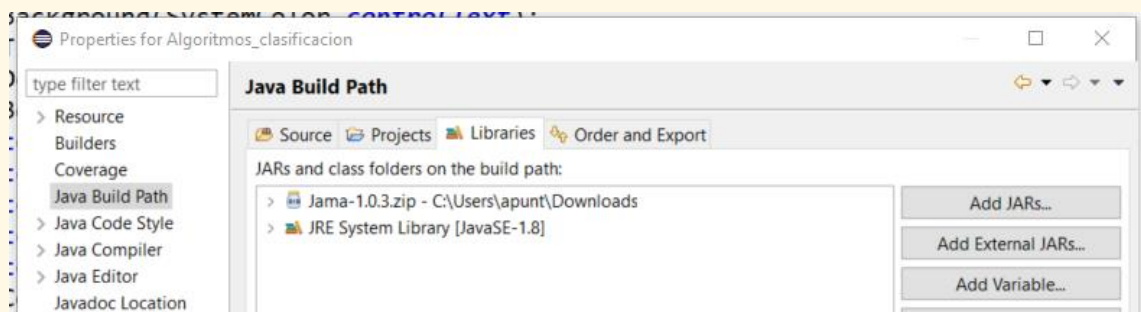


caso que puede llevar a confusión


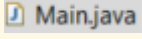
3. Manual de Usuario

Importar el proyecto Algoritmos_clasificacion en Eclipse

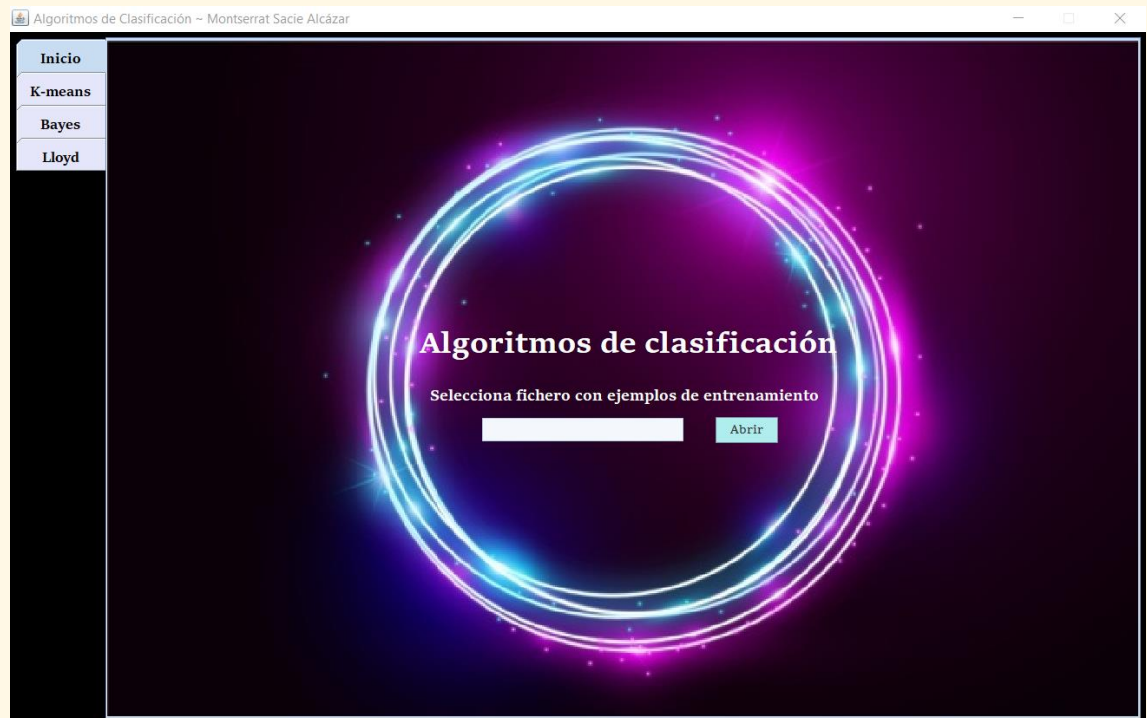
Incluir la librería Jama.jar que nos permite trabajar con matrices. Esta se incluye en la entrega.



En la página <https://math.nist.gov/javanumerics/jama/doc/Jama/Matrix.html> se puede consultar la documentación y detalles de los métodos de la librería

Arrancar la aplicación pulsando Run  colocados sobre 

Nos aparece:



Primero cargamos el fichero *Iris2Clases.txt* que contiene la lista de ejemplos junto a su clase. Si metemos otro fichero que tenga un formato incorrecto nos avisará y no podremos cargarlo.



Entramos en cualquiera de los paneles para probar el algoritmo deseado
Nos aparece:



Algoritmos de Clasificación - Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Datos

Clase Iris-setosa
5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor
7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Parámetros

Tolerancia : 0.01
Peso exponencial 2

Por defecto Aleatorio

Centros iniciales

1. [4.6, 3.0, 4.0, 0.0]
2. [6.8, 3.4, 4.6, 0.7]

Por defecto Aleatorio Ejecutar algoritmo

Centros obtenidos

Clasificar Nuevos ejemplos

Añadir Ejemplo Clasificar Ejemplos

Si queremos calcular parámetros aleatoriamente pulsamos **Aleatorio**. Si no, dejar como está. Lo mismo con los centros.

Pulsar **Ejecutar algoritmo** para ejecutar K-means.

Si no hemos abierto el fichero; nos pedirá que nos dirijamos a la pestaña Inicio para hacerlo. En caso contrario nos aparece

1. Una ventana gigantesca con la matriz U, que en algún caso podría sernos útil consultar. Si queremos quitarla del medio pulsamos Aceptar

Matriz de grados de pertenencia, U

Matriz

0.647 0.674 0.678 0.701 0.652 0.608 0.684 0.664 0.706 0.682 0.615 0.688 0.684 0.69 0.56 0.554 0.597 0.644 0.578 0.639 0.634 0.639 0.658 0.658 0.712 0.678 0.663 0.643 0.64 0.7 0.695 0.617 0.624 0.586 0.682 0.651 0.604 0.682 0.698 0.655 0.647 0.69 0.695 0.652 0.656 0.678 0.647 0.692 0.626 0.66 0.665 0.12
0.352 0.325 0.321 0.298 0.347 0.391 0.315 0.335 0.293 0.317 0.383 0.311 0.315 0.309 0.439 0.445 0.402 0.355 0.421 0.36 0.365 0.36 0.341 0.287 0.32 0.336 0.356 0.359 0.299 0.304 0.382 0.375 0.413 0.317 0.348 0.395 0.317 0.301 0.344 0.352 0.309 0.304 0.347 0.343 0.321 0.352 0.307 0.373 0.339 0.934

Aceptar

2. En la ventana principal tenemos



Algoritmos de Clasificación - Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Datos

Clase Iris-setosa

5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor

7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Parámetros

Tolerancia : 0.01

Peso exponencial 2

Por defecto Aleatorio

Centros iniciales

1. [4.6, 3.0, 4.0, 0.0]
2. [6.8, 3.4, 4.6, 0.7]

Por defecto Aleatorio Ejecutar algoritmo

Centros obtenidos

1. [5.11, 3.16, 2.12, 0.5]
2. [5.96, 3.0, 3.83, 1.15]

Clasificar Nuevos ejemplos

Añadir Ejemplo Clasificar Ejemplos

Los centros han sido calculados (Centros obtenidos), el resto del panel no ha cambiado.

En la pestaña de Bayes tenemos:

Algoritmos de Clasificación - Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Centros obtenidos

Matrices de Covarianza

Ejecutar algoritmo

Datos

Clase Iris-setosa

5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor

7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Clasificar Nuevos ejemplos

Añadir Ejemplo Clasificar Ejemplos

Podemos ejecutar como en el caso anterior y tras ejecutar, clasificar un nuevo ejemplo. Si no se ha ejecutado el algoritmo (en cualquiera de los paneles dedicados a los algoritmos) no nos permite añadir ejemplos nuevos ni clasificarlos.

Por último en la pestaña de Lloyd tenemos:



Algoritmos de Clasificación ~ Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Datos

Clase Iris-setosa
5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor
7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Centros iniciales
1. [4.6, 3.0, 4.0, 0.0]
2. [6.8, 3.4, 4.6, 0.7]

Centros obtenidos

Por defecto Aleatorio Ejecutar algoritmo

Parámetros
Tolerancia : 1.0E-10
Razón de aprendizaje: 0.1
Por defecto Aleatorio

Clasificar Nuevos ejemplos

Añadir Ejemplo Clasificar Ejemplos

Cada vez que cambiamos de pestaña el algoritmo en el que estábamos se reinicia, por lo que habría que ejecutarlo de nuevo.

Panel para comprobar el resultado para un nuevo ejemplo o para los ejemplos de la lista, por ejemplo para el Algoritmo de Bayes:

Clasificar Nuevos ejemplos

Añadir Ejemplo Clasificar Ejemplos

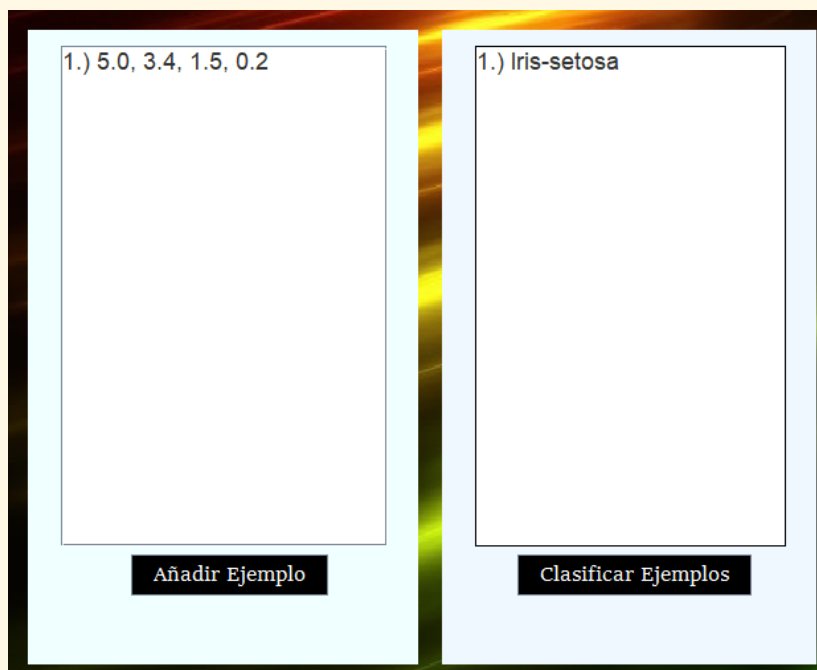
Añadimos un ejemplo pulsando

Añadir Ejemplo



Pulsamos  y 

En caso de no haber escrito nada; nos saldrá un panel emergente informándonos de que no podemos hacer eso. Como no es el caso, nos aparece:





4. Ejecución y resultados

Se ha ejecutado el algoritmo dos veces

1. Desde el Main.java, con los datos proporcionados en los enunciados de la Hoja 2 de ejercicios de clase

```
ArrayList<double []> puntos = new ArrayList<double []>();
puntos.add(new double[] {3,1});
puntos.add(new double[] {2,2});
puntos.add(new double[] {1,0});
puntos.add(new double[] {6,7});
puntos.add(new double[] {7,5});
puntos.add(new double[] {8,6});

ArrayList<String> clases = new ArrayList<String>();//clase de cada punto
clases.add("Clase 1");
clases.add("Clase 1");
clases.add("Clase 1");
clases.add("Clase 2");
clases.add("Clase 2");
clases.add("Clase 2");
```

Resultado

```
Algoritmo de Bayes - Ejercicio 1, Hoja 2
.....
Centros obtenidos:
1. [2.0, 1.0]
2. [7.0, 6.0]

Matrices de Covarianza:
Clase: Clase 1

---Matriz-----
0.666  0.333
0.333  0.666
-----

Clase: Clase 2

---Matriz-----
0.666  -0.334
-0.334  0.666
-----
```




Algoritmo k_means borroso - Ejercicio 2, Hoja 2

.....

Centros obtenidos:

1. [2.0, 1.0]
2. [7.01, 5.98]

Matriz de grados de pertenencia U:

---Matriz-----

0.976	0.976	0.972	0.037	0.023	0.016
0.023	0.023	0.027	0.962	0.976	0.983

La matriz U no la muestra completa, dado el número desmesurado de columnas
(una por cada muestra)

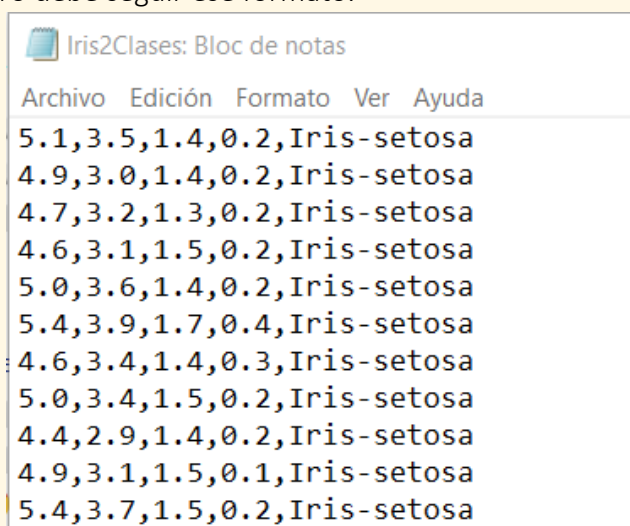
Algoritmo de Lloyd - Ejercicio 3, Hoja 2

.....

Centros obtenidos:

1. [1.98, 0.99]
2. [7.01, 5.99]

2. Desde la interfaz cargamos el fichero *Iris2Clases.txt*. Cada ejemplo toma valores en 4 variables y están clasificados en la clase "Iris-setosa" o "Iris-versicolor". Cualquier fichero debe seguir ese formato.



Y los ejemplos a clasificar son



TestIris01: Bloc de notas

Archivo Edición Formato Ver Ayuda

5.1,3.5,1.4,0.2,Iris-setosa

TestIris02: Bloc de notas

Archivo Edición Formato Ver Ayuda

6.9,3.1,4.9,1.5,Iris-versicolor

TestIris03: Bloc de notas

Archivo Edición Formato Ver Ayuda

5.0,3.4,1.5,0.2,Iris-setosa

Resultado:

Algoritmo k-Means

Algoritmos de Clasificación ~ Montserrat Sacie Alcázar

Inicio

K-means

Bayes

Lloyd

Datos

Clase Iris-setosa

5.1, 3.5, 1.4, 0.2

4.9, 3.0, 1.4, 0.2

4.7, 3.2, 1.3, 0.2

4.6, 3.1, 1.5, 0.2

5.0, 3.6, 1.4, 0.2

5.4, 3.9, 1.7, 0.4

4.6, 3.4, 1.4, 0.3

5.0, 3.4, 1.5, 0.2

4.4, 2.9, 1.4, 0.2

4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor

7.0, 3.2, 4.7, 1.4

6.4, 3.2, 4.5, 1.5

6.9, 3.1, 4.9, 1.5

5.5, 2.3, 4.0, 1.3

6.5, 2.8, 4.6, 1.5

5.7, 2.8, 4.5, 1.3

6.3, 3.3, 4.7, 1.6

4.9, 2.4, 3.3, 1.0

6.6, 2.9, 4.6, 1.3

5.2, 2.7, 3.9, 1.4

Centros iniciales

1. [4.6, 3.0, 4.0, 0.0]

2. [6.8, 3.4, 4.6, 0.7]

Centros obtenidos

1. [5.11, 3.16, 2.12, 0.5]

2. [5.96, 3.0, 3.83, 1.15]

Por defecto

Aleatorio

Ejecutar algoritmo

Clasificar Nuevos ejemplos

1.) 5.1, 3.5, 1.4, 0.2

2.) 6.9, 3.1, 4.9, 1.5

3.) 5.0, 3.4, 1.5, 0.2

Parámetros

Tolerancia : 0.01

Peso exponencial 2

Por defecto

Aleatorio

Clasificar Ejemplos

1.) Iris-setosa

2.) Iris-versicolor

3.) Iris-setosa



Algoritmo de Bayes

Algoritmos de Clasificación ~ Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Centros obtenidos

1. [5.93, 2.77, 4.26, 1.32]
2. [5.0, 3.41, 1.46, 0.24]

Matrices de Covarianza

Clase Iris-setosa
---Matriz---
0.121 0.098 0.015
0.098 0.142 0.011
0.015 0.011 0.029
0.01 0.011 0.005

Clase Iris-versicolor
---Matriz---
0.261 0.083 0.179
0.083 0.096 0.081
0.179 0.081 0.216
0.054 0.04 0.071

Ejecutar algoritmo

Datos

Clase Iris-setosa
5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor
7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Clasificar Nuevos ejemplos

1.) 5.1, 3.5, 1.4, 0.2
2.) 6.9, 3.1, 4.9, 1.5
3.) 5.0, 3.4, 1.5, 0.2

Añadir Ejemplo

1.) Iris-setosa
2.) Iris-versicolor
3.) Iris-setosa

Clasificar Ejemplos

Algoritmo de Lloyd

Algoritmos de Clasificación ~ Montserrat Sacie Alcázar

Inicio
K-means
Bayes
Lloyd

Datos

Clase Iris-setosa
5.1, 3.5, 1.4, 0.2
4.9, 3.0, 1.4, 0.2
4.7, 3.2, 1.3, 0.2
4.6, 3.1, 1.5, 0.2
5.0, 3.6, 1.4, 0.2
5.4, 3.9, 1.7, 0.4
4.6, 3.4, 1.4, 0.3
5.0, 3.4, 1.5, 0.2
4.4, 2.9, 1.4, 0.2
4.9, 3.1, 1.5, 0.1

Clase Iris-versicolor
7.0, 3.2, 4.7, 1.4
6.4, 3.2, 4.5, 1.5
6.9, 3.1, 4.9, 1.5
5.5, 2.3, 4.0, 1.3
6.5, 2.8, 4.6, 1.5
5.7, 2.8, 4.5, 1.3
6.3, 3.3, 4.7, 1.6
4.9, 2.4, 3.3, 1.0
6.6, 2.9, 4.6, 1.3
5.2, 2.7, 3.9, 1.4

Centros iniciales

1. [4.6, 3.0, 4.0, 0.0]
2. [6.8, 3.4, 4.6, 0.7]

Por defecto **Aleatorio**

Centros obtenidos

1. [4.95, 3.27, 1.66, 0.32]
2. [5.75, 2.75, 4.1, 1.27]

Ejecutar algoritmo

Clasificar Nuevos ejemplos

1.) 5.1, 3.5, 1.4, 0.2
2.) 6.9, 3.1, 4.9, 1.5
3.) 5.0, 3.4, 1.5, 0.2

Añadir Ejemplo

1.) Iris-setosa
2.) Iris-versicolor
3.) Iris-setosa

Clasificar Ejemplos

Parámetros

Tolerancia : 1.0E-10

Razón de aprendizaje: 0.1

Por defecto **Aleatorio**

Se incluye en la entrega un video que muestra la ejecución de estos ejemplos.