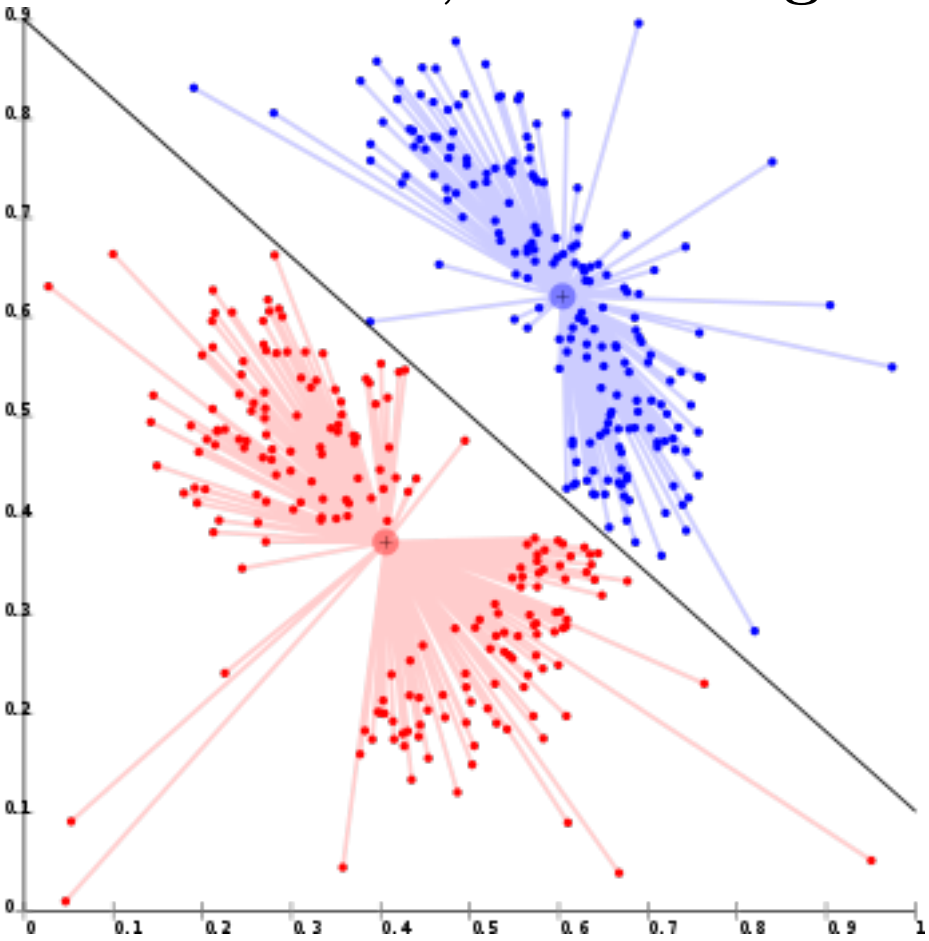
Práctica 6, Clustering



Minería de datos

Andoni Martín Reboredo
David Ramirez Ambrosi

19 de octubre de 2014

Índice general

Índice de figuras

Capítulo 1

Introducción

El presente documento constituye el resultado de la práctica realizada en base a la implementación del algoritmo de clasificación no supervisada **K-Means clustering**. Este algoritmo trata el agrupamiento de un conjunto de instancias en base a su proximidad con las demás instancias contenidas en el espacio de muestra proporcionado al algoritmo para su ejecución.

Dentro del algoritmo cabe el estudio de diferentes variaciones en los distintos parámetros de que dispone. Nosotros hemos considerado variaciones sobre dos parámetros, el método de cálculo de la distancia entre los distintos elementos que posee el cluster y la inicialización de los distintos clusters. Esta inicialización servirá como base de las sucesivas iteraciones que conforman el algoritmo.

1.1. Clasificación no-supervisada

La clasificación no supervisada es aquella que se lleva a cabo mediante el estudio de las diversas instancias que conforman el espacio de aplicación del algoritmo sin que estas instancias tengan que estar previamente clasificadas dentro de una clase [1].

Se trata de una técnica de exploración de los datos en la que se intentan detectar estas clases desconocidas. Dependiendo de el algoritmo de clasificación utilizado, el número de clases debe o no ser especificado. Por ejemplo, en el algoritmo en que se basa este trabajo debe ser especificado, sin embargo en técnicas de **clusterig jerárquico** no.

1.2. Objetivo

Esta práctica tiene como objetivo principal la comprensión de los procesos internos que realiza un algoritmo de agrupamiento cualquiera como puede ser el K-Means clustering. El aprendizaje se realizará de forma práctica a través de la implementación del algoritmo K-Means clustering junto con diversas opciones con las que realizar algunos pasos del mismo, como son el uso de métricas o inicializaciones del algoritmo diferentes. Estas variaciones requieren que el algoritmo sea entendido plenamente para poder hacer contribuciones que tengan utilidad para la realización del proceso.

Capítulo 2

Algoritmo

2.1. K-means, algoritmo principal

```
inicializar
divergencia = infinito

Mientras(numiteraciones <= iteracionesIndicadas AND divergencia < delta)
{
    centroides = centroidesNuevos

    calcularPertenencias
    centroidesNuevos = calcularNuevosCentroides

    calcularDivergencia(centroidesNuevos)
}
```

2.2. Subrutina inicialización

2.2.1. Inicialización aleatoria

```
Para cada dimensión
{
   mientras extraiga una instancia ya evaluada
   {
      extraigo una instancia nueva
   }
   añado la instancia extraida a las evaluadas
   establezco la instancia como centroide de un cluster
}
```

2.2.2. Pertenencia aleatoria

```
Mientras haya instancias que asignar
```

```
{
   Calculo un número de cluster aleatorio
   Extraigo la siguiente instancia
   Añado la instancia al cluster aleatorio
}
Calculo los centroides del cluster
```

2.2.3. División de espacio

```
Obtengo los rangos máximos y mínimos de cada subespacio
Mientras no haya creado k divisiones
{
   Mientras no haya establecido todos los atributos(recorrido los subespacios)
   {
      Divido el subespacio en K
      Asigno el centro del subespacio dividido correspondiente al índice del bucle
   }
   Añado el centroide resultante de dividir el espacio
}
```

2.2.4. Generación aleatoria de codewords

```
Obtengo los máximos y mínimos de cada dimensión
Para cada cluster
{
   Evaluo cada dimensión
   {
      Calculo un valor aleatorio para ese centroide en esa dimensión
   }
   Añado el centroide generado al cluster
}
```

2.3. Subrutina calcularPertenencias

```
Crear nuevos clusters

Para cada instancia
{
    Para cada centroide
    {
        distancia entre la instancia y cada cluster
        Guardamos los mínimos
    }
    Para cada centroide obtenido
        Guardamos la instancia en el cluster correspondiente al centroide
}
```

2.4. Subrutina calcularCentroides

```
Para cada cluster

   Calcular la instancia media

return nuevosCentroides
```

2.5. Subrutina calcularDivergencia

```
Para cada cluster

   Calcular la distancia entre el centroide antiguo y el nuevo

return divergenciaAcumulada
```

Capítulo 3

Diseño

En la figura 3.1 se muestra el diagrama de clases final de la aplicación, extraido a partir del código mediante el software *Visual Paradigm*. La especificación de cada uno de los métodos se puede encontrar en la documentación adjunta generada con *Eclipse*, la carpeta doc contiene la *javadoc* generada en forma de *HTMLs* navegables.

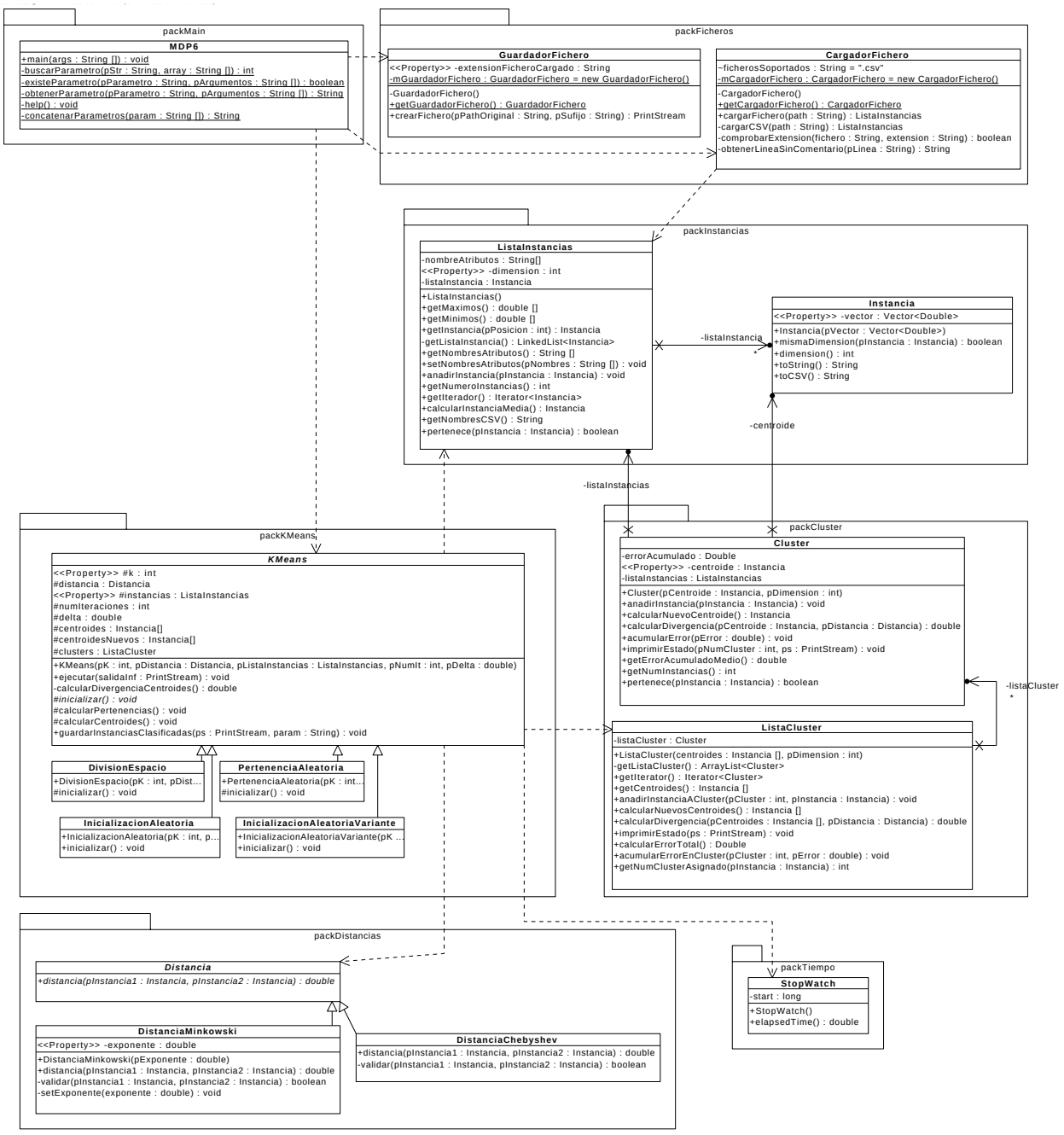


Figura 3.1: Diagrama de clases y paquetes

Capítulo 4

Resultados experimentales

4.1. Banco de pruebas para la validación de software y resultados

4.2. Resultados

4.3. Clasificación supervisada respecto de otro software de referencia

4.4. Variabilidad de los resultados

El uso de generadores de valor aleatorios sin utilizar ningún tipo de mapeo, a diferencia de Weka, hace que la calidad de **la solución varíe** en exceso de una ejecución a otra. El uso de aleatorios se da en los casos de inicialización aleatoria, pertenencia aleatoria y generación aleatoria de codewords. Este problema no se da en el último tipo de inicialización, división de espacio, ya que ésta generará los mismos k centroides para un mismo conjunto de datos.

Para probar la variabilidad, se ha utilizado el fichero de pruebas food.csv con la inicialización aleatoria y la distancia euclídea generando 4 clusters. Se ha repetido esta misma operación 10000 veces y se han obtenido los resultados siguientes:

```
Min.:      4.148
1st Qu.:   6.014
Mediana:   6.335
Media:     6.181
3rd Qu.:   7.145
Max.:      7.843
Varianza:
```

Los valores del error en las 10000 repeticiones varían en el intervalo [4.148, 7.843], siendo la media 6.181. En la figura 4.1 se muestran de forma gráfica los resultados mediante un histograma con la densidad de los diferentes valores

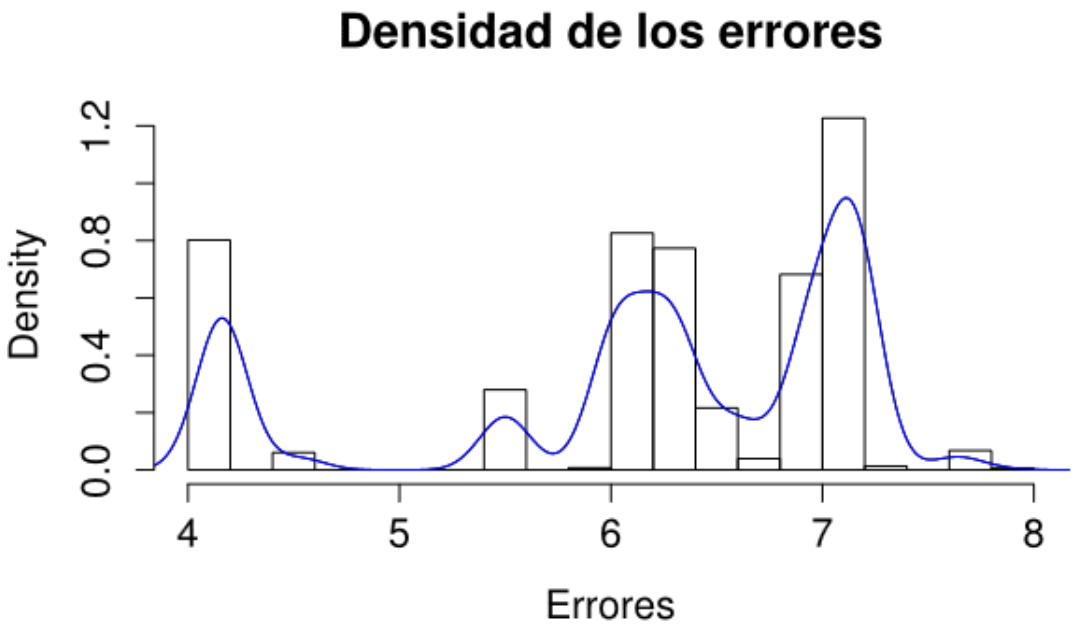


Figura 4.1: Histograma y frecuencia de los errores obtenidos

4.5. Análisis crítico y discusión de resultados

4.6. Rendimiento del software

Capítulo 5

Conclusiones

5.1. Motivación para la realización de *Clustering*

Explorar un conjunto de instancias con el objetivo de

5.2. Conclusiones de los resultados

5.3. Conclusiones generales

El tiempo invertido en el proyecto ha sido mucho mayor conforme se acercaba la fecha de entrega. Debemos mejorar en organización, especialmente al principio de la práctica, donde un diseño poco especificado con ciertas cosas *."en el aire"* nos llevaron a diseñar al implementar.

También el hecho de probar y perfeccionar el funcionamiento de **todo** el código antes de comenzar con la extracción de resultados experimentales nos limitó en el análisis de los mismos y en la retroalimentación del algoritmo implementado.

Resumiendo, hay que mejorar la planificación, dividir el proyecto en partes para poder mejorarlo incrementalmente.

Pese a los problemas, la solución implementada ofrece una variedad bastante amplia de parámetros posibles a utilizar, teniendo 4 tipos de inicialización, dos métricas diferentes - en el caso de la métrica Minkowski ofrece otro parámetro más que variar- y la posibilidad de controlar el número de iteraciones tanto por la especificación de un maximo de vueltas, como por la especificación del margen de variación que se debe alcanzar hasta dar por válidos los centroides identificados.

5.4. Propuestas de mejora

La entrada a nuestro programa es un factor un tanto restrictivo en el sentido de que el único tipo de fichero soportado es el **CSV**. Inicialmente, la intención era implementar también la carga de ficheros **ARFF**, que no ha sido posible hacerla por falta de planificación principalmente.

En cuanto a inicializaciones, hemos implementado 4 tipos diferentes. De los tres, el correspondiente a la división de espacio es el que más puede mejorar, ya que para su implementación no hemos llegado a consultar bibliografía extra.

La **estimación del error** es otro apartado en el que nos ha quedado duda, ya que el proceso de cálculo utilizado no ha sido el mismo que el usado por Weka y nos dificultaba la comparación de ambos algoritmos utilizando esta medida.

Otra restricción se da por el tipo de datos tratados, el algoritmo solo es capaz de lidiar con atributos de tipo numérico, siendo imposible realizar la carga de instancias que contengan valores no numéricos. De hecho, algunos de los ficheros han tenido que ser preprocesados para poder trabajar con ellos.

Otro problema ya arrastrado de trabajos anteriores, es la **presentación de resultados**. No en cuanto a la información presentada, sino a como se presentan concretamente los **valores reales**, que deberían ser redondeados a, por ejemplo, 3 decimales. Esto permitiría que los informes generados sean más fáciles de leer e interpretar.

Bibliografía

[1] Alicia Pérez. Minería de datos, tema 9: Clasificación no-supervisada (clustering).

Capítulo 6

Valoración subjetiva

Alcance de objetivos

Utilidad de la tarea

Dificultad

Tiempo de trabajo

Sugerencias de mejora

Críticas