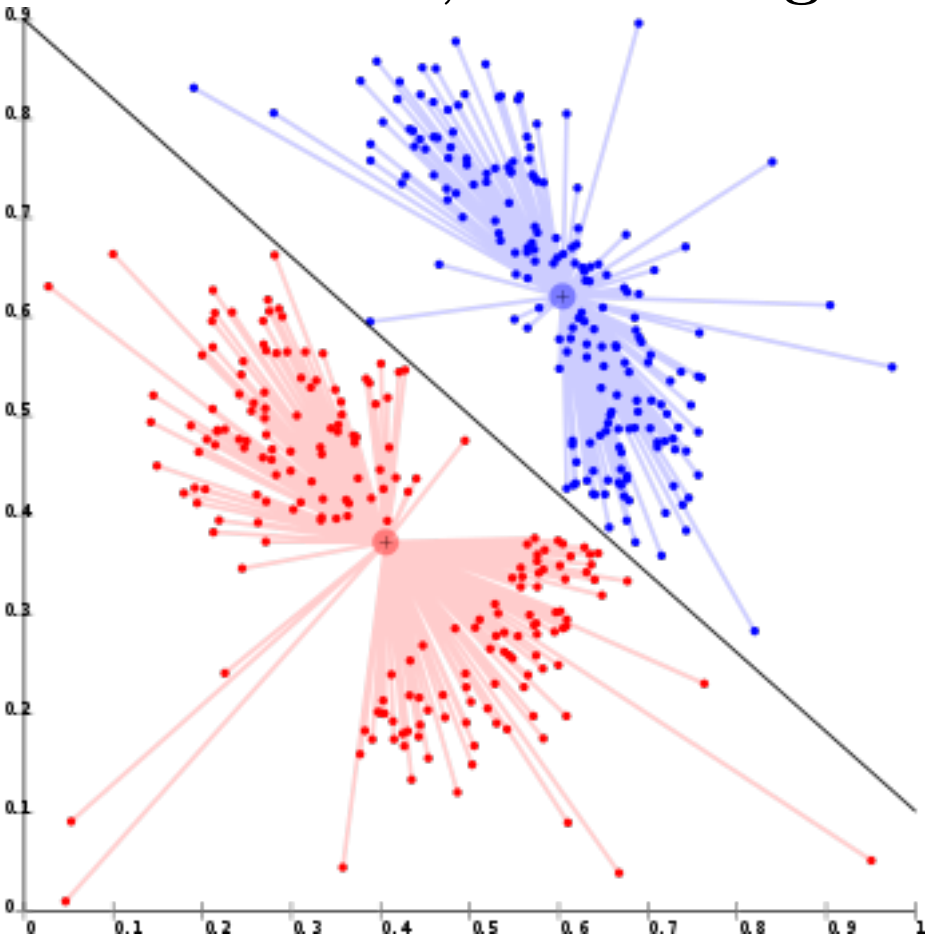
Práctica 6, Clustering



Minería de datos

Andoni Martín Reboredo
David Ramirez Ambrosi

24 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.

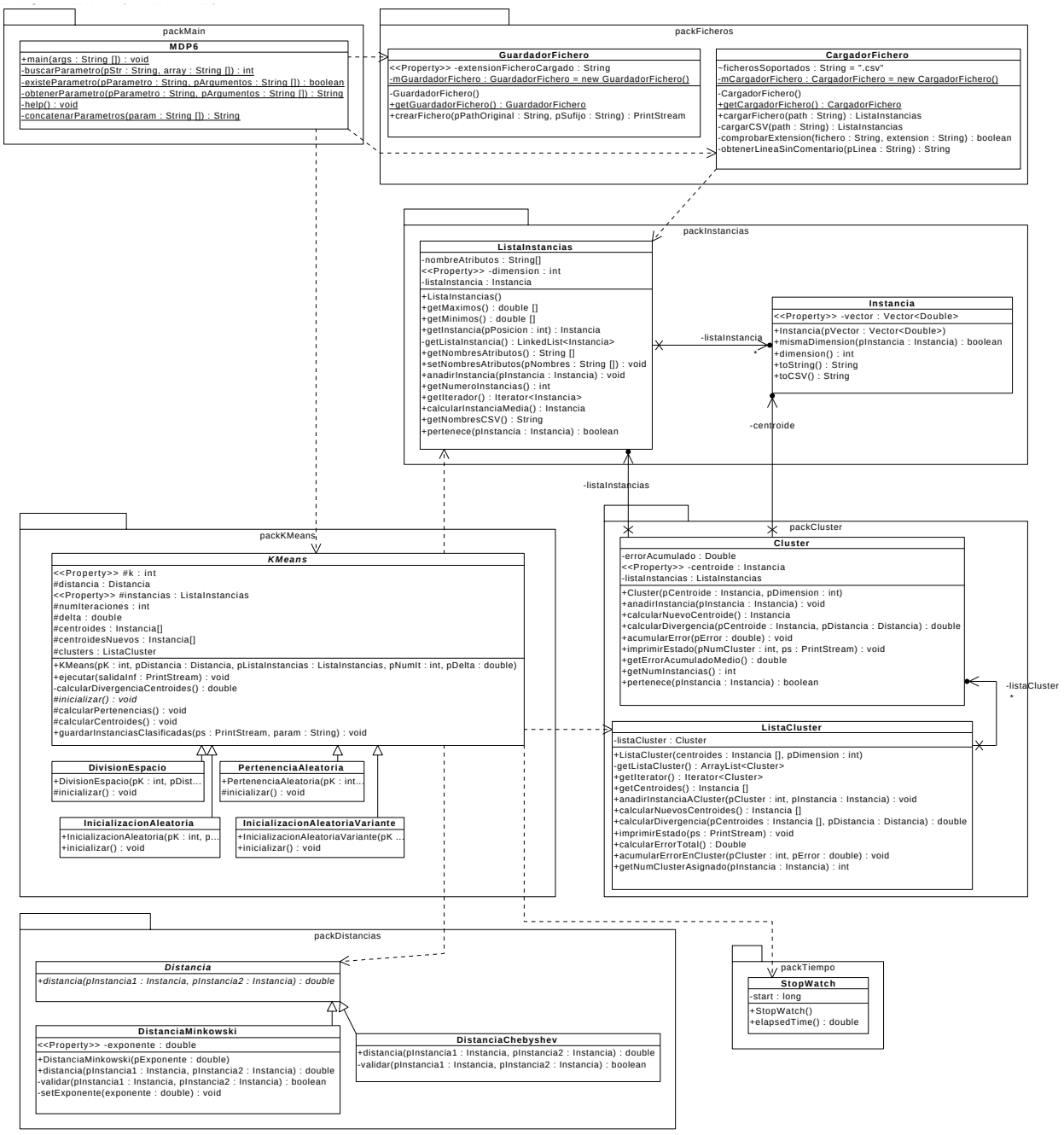


Figura 3.1: Diagrama de clases y paquetes

Capítulo 4

Resultados experimentales

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

Para la validación de nuestro software, MDP6, utilizaremos el software Weka usado en la asignatura. Ambos softwares, el propio y Weka, difieren en capacidad y soporte de funciones. Para la realización de pruebas, describimos las capacidades de cada uno de ellos:

MDP6

- **Inicializaciones**. Disponemos de 4 tipos de inicializaciones diferentes:
  - **Inicialización aleatoria**. Elige al azar k instancias del conjunto de datos para actuar de centroides
  - **Inicialización por petenencia aleatoria**. Por cada instancia, la incluye al azar en uno de los k clusters. Posteriormente se calculan los centroides.
  - **Inicialización por generación aleatoria de centroides**. Se recogen los rangos en que varían los atributos de todas las instancias. Posteriromente se crean k centroides con valores al azar incluidos en los diferentes intervalos identificados.
  - **Inicialización por división de espacio**. Se divide el espacio muestral de acuerdo al número de clusters a crear y se identifican los centroides.
- **Distancias**. Se han implementado dos tipos de métricas capaces de trabajar con el algoritmo
  - **Distancia Minkowski**. Implementada para cualquier valor del parámetro m (siempre que sea un real positivo mayor o igual de 1).
  - **Distancia Chebyshev**.
- **Criterio de parada**. Se ofrece la posibilidad de establecer un límite de iteraciones para la ejecución del algoritmo o bien el uso de un valor a satisfacer para la divergencia entre conjuntos de centroides de dos iteraciones diferentes. También se pueden combinar ambos valores de forma que el primero en satisfacerse determine el fin del algoritmo.
- **Tipos de atributos**. Únicamente se permiten valores numéricos y conocidos para los atributos de las instancias. Todo fichero que vaya a ser cargado debe satisfacer esta condición. No se ofrece funcionalidad como la normalización de atributos.

- **Medidas de calidad**. Como medida del error tras el proceso de clustering se ofrece tanto el **Error Cuadrático** como el **Error Cuadrático Medio**.
- **Clasificación de instancias**. Se puede generar un archivo conteniendo las instancias cargadas (en el mismo orden al de entrada) en formato CSV. En este fichero de salida, a cada instancia se le ha añadido un atributo que indica el custer al que finalmente fue asociada tras la ejecución del algoritmo.
- **Repetición de pruebas**. Como se explicará posteriormente, las pruebas realizadas con tipos de inicializaciones que utilicen generación de valores aleatorios no pueden ser repetidas ya que los valores variarán en cada ejecución.

Weka

- **Inicializaciones**. No ofrece más que un tipo de inicialización.
- **Distancias**. Las distancias disponibles para el agoritmo son Minkowski 1 y Minkowski 2 (Euclídea y Manhattan respectivamente). Además, cada una de estas dos distancias permite normalizarlas.
- **Criterio de parada**. Permite establecer el número máximo de iteraciones que realizará el algoritmo.
- **Tipos de atributos**. Soporta nominales, numéricos, binarios, unarios, con valores desconocidos o nominales vacíos.
- **Medidas de calidad**. En el caso de la distancia **Euclídea**, se ofrece el **Error Cuadrático** como medida de error. En el caso de la distancia **Manhathan**, la medida ofrecida es la **suma de las distancias dentro del cluster**.
- **Clasificación de instancias**. Tras ejecutar el algoritmo, se puede visualizar el resultado de la agrupación de las instancias y guardarlas en formato **ARFF** con un atributo más correspondiente al cluster con que la instancia ha sido asociada.
- **Repetición de pruebas**. Weka permite que para una misma semilla (parámetro configurable) los valores generados carezcan de variación, de forma que se puede repetir el mismo experimento una y otra vez para una misma semilla.

Los ficheros utilizados como fuente de instancias han sido **previamente procesados** con *Weka* para adaptarlos a nuestro software. Se ha procedido a **eliminar el atributo clase** en aquellos ficheros que la tuvieran y se han guardao las instancias en **formato CVS**.

Así mismo, en algunos casos ha sido necesario utilizar además un **procesador de texto** para poder obtener los resultados deseados. También se ha usado el software **R** para obtener ciertos resultados estadísticos y realizar comparaciones de valores. Para automatizar el proceso de obtención de datos, se han desarrollado programas **JAVA** extra y se ha utilizado un script del interprete **BASH**.

4.1.1. Pruebas realizadas

Se han realizado tres tipos diferentes de pruebas para comprobar la validez y rendimiento del software desarrollado. En cada una de estas pruebas los indicadores de referencia utilizados han sido:

- **Medidas de error**. Comparación directa del error generado en el proceso de clustering. Prueba desarrollada en el apartado 4.2, se utiliza el **Error Cuadrático** para realizar comparar directamente el software con Weka.
- **Medidas de precisión**. Comparación de la **precisión** capaz de alcanzar nuestro software con la que Weka es capaz de alcanzar. El proceso se desarrolla y explica en el apartado 4.3.
- **Medidas estadísticas**. Uso de medidas como máximo, mínimo, media y varianza de un conjunto de datos para determinar la variabilidad de las repeticiones sucesivas de un determinado experimento con nuestro software. Proceso desarrollado en el apartado 4.4.

4.2. Resultados comparativos

4.3. Clasificación supervisada respecto de otro software de referencia

4.4. Variabilidad de los resultados

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