6. PRÁCTICA: Clustering

Índice

1. Objetivos **1**

2. Material **2**

3. Metodología de trabajo **2**

4. Guión: algoritmo *k-means clustering* **2**

4.1. Especificaciones mínimas 2

4.2. Procedimiento 3

5. Resultados **3**

5.1. Entregables 3

5.2. Aspectos a considerar 5

5.3. Plazos 6

1. Objetivos

Trabajar las siguientes competencias:

- **Competencias transversales:**
  - Trabajo en equipo
  - Habilidad para comunicar y transmitir procedimientos y resultados en base a razonamiento crítico
- **Competencias específicas:**
  - Reconocer las posibilidades que ofrece el uso sistemático de técnicas de extracción de conocimiento
  - Capacidad para sintetizar una técnica de aprendizaje automático no-supervisado, conocer su coste computacional así como sus limitaciones de representación y de inteligibilidad

2. Material

- PC con aplicación Weka
- Recursos accesibles desde Moodle:
  - Manual de la aplicación
  - Bibliografía
  - Fichero de datos para la práctica: se pide trabajar con los ficheros `food.arff` y `colon.arff`. En el directorio de datos asociados a la práctica en Moodle, encontrarás más ficheros de interés para validar el software o para ayudarte a ampliar funcionalidades.
    - `food.arff`: este fichero no tiene la clase asociada, comenzar por aquí para explorar si el software funciona correctamente y ofrece resultados esperados. En este fichero hay un atributo de tipo `string`, explora la posibilidad de no emplear este atributo, si lo consideras oportuno y razonando la respuesta, puedes obviarlo.
    - `colon.arff`: este fichero sí tiene la clase asociada, no se puede emplear el atributo clase para hacer el *clustering*, pero sí para evaluar la calidad del *clustering* obtenido tomando esta etiqueta como referencia.
    - Otros ficheros que no están en formato `.arff`:
      - ◇ En formato `.txt`: `ClusterData.atributos.txt` (este fichero sí tiene la clase asociada para evaluar la calidad del *clustering* en `ClusterData.clase.txt`).
      - ◇ En formato `.csv`: `bank-data.csv`

3. Metodología de trabajo

- Naturaleza: diseño e implementación de software
- Carácter: individual o grupos reducidos (máximo 2 personas)

4. Guión: algoritmo *k-means clustering*

4.1. Especificaciones mínimas

El software implementará el algoritmo *k-means clustering* permitiendo:

1. Modificar inicializaciones:
   a) aleatoria
   b) por división del espacio
2. Seleccionar cualquier distancia de Minkowski ($\forall m \in \mathbb{R}$)
3. Modificar criterios de convergencia:
   a) Un número de iteraciones fijo (especificado como una constante)
   b) Disimilitud entre codebooks sucesivos menor a un umbral prefijado (especificado como una constante)

4.2. Procedimiento

1. Escribir el algoritmo en pseudo-código
2. Planificar el diseño del software dibujando un esquema de dependencias.
3. Implementación de software
4. Validación de software:
   a) Diseñar banco de pruebas para comprobar que la herramienta da los resultados esperados en entornos controlados, tanto en casos generales como en casos extremos
   b) Comprobar resultados con los ofrecidos empleando la herramienta Weka
5. Análisis de resultados:
   a) Modificando inicializaciones: aleatoria, por división del espacio
   b) Modificando distancia Minkowski con valores $m \in 1, 2, 3$
   c) Criterios de convergencia:
      1) Un número de iteraciones fijo (especificado como una constante)
      2) Disimilitud entre codebooks sucesivos menor a un umbral prefijado (especificado como una constante)

5. Resultados

5.1. Entregables

Se pide un paquete con el código fuente diseñado y debidamente documentado (`src.gzip`) así como un informe de prácticas en formato PDF (`InformeP6.pdf`) que incluya las siguientes secciones:

1. **Introducción:**
   a) **Definición:** clasificación no-supervisada
   b) **Objetivo:** definir el problema concreto a abordar en el contexto de la minería de datos. Incluir, además, los siguientes detalles:
      - Instancias: ¿de cuántas instancias se dispone? (N=)
      - Atributos: ¿cuántos atributos se emplean para describir las instancias? (n=) ¿de qué tipo son?
2. **Algoritmo:** *k-means clustering* en pseudo-código
3. **Diseño:** mapa de diseño donde se muestran las dependencias y se documentan las rutinas
4. **Resultados experimentales:**
   a) Describir banco de pruebas diseñado para validar el software y resultados obtenidos
   b) Sobre los datos de aplicación de la tarea se pide obtener los siguientes resultados:

1) Distinto número de clusters a considerar (eg. k: 2, 3, 12)
2) Distintas métricas:
   - Distancia de Manhattan ($m = 1$)
   - Distancia Euclídea ($m = 2$)
   - Distancia de Minkowski con $m = 7'5$
3) Distintas inicializaciones
4) Distintos criterios de convergencia

c) **Clasificación supervisada respecto de otro software de referencia:** utilizar una de las dos alternativas siguientes como software de referencia

- **Clasificación supervisada respecto de Weka** de las variantes implementadas ¿cuáles permite Weka? Para alguna de esas variantes (detallar cuál) supongamos que tomamos los resultados del *clustering* de Weka a modo de clase de referencia. Se desea comparar la clase obtenida con Weka respecto de la etiqueta-cluster obtenida con nuestro software para obtener figuras de mérito como si de un problema de clasificación supervisada se tratara (no es necesario obtener exactamente la etiqueta que Weka ofrece como clase sino obtener la misma clase para las mismas muestras). De este modo, tratamos de validar el software implementado respecto de un software de referencia mediante las figuras de mérito. Es importante detallar el método de evaluación seguido (seleccionar el que se considere oportuno). También se pide describir el procedimiento a seguir para determinar la clase mediante Weka y presentar un análisis cuantitativo.
- **Clasificación supervisada respecto de variantes del software propio:** del mismo modo, se pueden tomar uno de los esquemas de *clustering* como referencia para etiquetar las muestras y obtener los resultados de los demás esquemas de *clustering* explorados. Por ejemplo: tomar como referencia eg. k=2, distancia Euclídea, inicialización aleatoria, convergencia por similitud y evaluar los siguientes:
  - k=2, distancia Manhattan, inicialización aleatoria, convergencia por similitud
  - k=2, distancia Euclídea, inicialización por partición del espacio, convergencia por similitud

d) **Análisis crítico y discusión de resultados**

e) **Rendimiento del software:** discutir el rendimiento del software diseñado en términos de coste temporal real y espacio en memoria requerido. Dar el perfil de tiempo y memoria consumido por cada modelo explorado. Comparar estos resultados con los que se obtienen con Weka.

5. **Conclusiones:** argumentar las siguientes cuestiones.

a) Describir muy brevemente la motivación para llevar a cabo técnicas de *clustering*

b) Describir muy brevemente las conclusiones obtenidas a la vista de los resultados más relevantes

c) Conclusiones generales (análisis de fortalezas del software implementado y reflexiones más importantes sobre la tarea)

d) Propuestas para mejorar o ampliar la funcionalidad del software diseñado en trabajo futuro (análisis de puntos débiles y propuesta para solucionarlos)

6. **Bibliografía:** se cita la fuente empleada en el punto en el que se recurre a ella para fundamentar una argumentación (detallando la página o sección). En esta última sección del informe se detalla la fuente de las fuentes citadas (y sólo esas, es decir, no aparecerán más referencias que las citadas). Deben aparecer tantos detalles sobre la fuente como sean posibles, y al menos: título, autor, editorial, año.
7. **Valoración subjetiva**: (voluntaria) escribe una valoración subjetiva sobre la tarea realizada, los siguientes puntos pueden servir de guía.
   a) ¿Has alcanzado los objetivos que se plantean?
   b) ¿Te ha resultado de utilidad la tarea planteada?
   c) ¿Qué dificultades has encontrado? Valora el grado de dificultad de la tarea
   d) ¿Cuánto tiempo has trabajado en esta tarea? Desglosado:
      1) Tiempo de diseño de software
      2) Tiempo de implementación de software
      3) Tiempo trabajando con Weka
      4) Tiempo dedicado a búsqueda bibliográfica
      5) Tiempo redactando el informe
   e) Sugerencias para mejorar la tarea. Sugerencias para que se consiga despertar mayor interés y motivación en los alumnos.
   f) Críticas (constructivas).

5.2. Aspectos a considerar

Se valorarán los siguientes aspectos:

1. Software:
   a) Algoritmo y Diseño del software
   b) Implementación:
      - Modularidad
      - Portabilidad
      - Eficiencia
      - Documentación de cada rutina (recordar metodología de la programación: pre- y post- condiciones)
   c) Documentación del paquete de software: descripción del modo de uso del software (documentación tipo `leeme.txt`) donde se indica, entre otros, modo de compilación (empleando `Makefile`) y ejemplo de ejecución
   d) Innovación: el proyecto debería ir más allá de los mínimos detallados en este guión, debería aportar un punto original, innovador (e.g. en términos de capacidades, representación, visualización, robustez, eficiencia, etc.).
2. Informe:
   a) Completitud
   b) Formalismo y fundamento teórico, expresión escrita en el ámbito científico-técnico
   c) Razonamiento crítico del análisis de resultados
   d) Presentación

5.3. Plazos

Los plazos de entrega están indicados en Moodle.