Minería de datos Práctica 1:Clustering knn-means

Jose Ignacio Sánchez
Josu Rodríguez

4 de octubre de 2014

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. Introducción

El objetivo principal de esta práctica es obtener la capacidad de formular un algoritmo de aprendizaje automático de clasificación ***No-Supervisada***. Por otra parte, se trabajarán la capacidad de sintetizar uns técnica de aprendizaje automático no-supervisado, conocer su coste computacional así como sus limitaciones de representación y de inteligibilidad

2. Recursos

- PC con aplicación Weka.
- Bibliografía.
- Librerías de Weka.
- Manual de Weka.
- Guía de la práctica.
- Ficheros para los datos de la práctica: food.arff, colon.arff.
- Otros ficheros que no están en formato *.arff*:
  - En formato *.txt*: ClusterData.atributos.txt (este fichero si tiene la clase asociada para evaluar la calidad del *clustering* en ClusterData.clase.txt).
  - E formato *.csv* bank-data.csvclustering

3. Clasificación NO-supervisada o *Clustering*

(*Definición) 3.1 Se considera* ***clasificación no-supervisada*** *cuando el conjunto de entrenamiento no están las instancias etiquetadas con el valor de la clase. Es un experimento exploratorio, que trata de agrupar las instancias en grupos definidos por similitud entre las características de las instancias que pertenecen al mismo grupo y disimilitud entre las que pertenecen a grupos distintos. Tecnicamente estos grupos son llamados Clusters.*

3.1. Clustering *k-means*

4. Diseño

Estructuramos la ejecución del algoritmo en fases como se puede ver en la figura 1 , las cuales se detallan a continuación.

Primera fase: carga de datos y configuración

Inicialmente se encarga de cargar el fichero en una estructura de datos adecuada para el cálculo del algoritmo.Además se carga la configuración establecida por el usuario, es decir que tipo de inicialización para el *codebook*, número de clusters, distancia a utilizar...

Segunda fase: Preproceso de datos

En el preproceso se normaliza el espacio, haciendo uso de la función estadística zscore, es decir a cada *Feature* de la instancia se le resta su media y dividiendo el resultado por la varianza de dicha *Feature* se consigue que todos los valores de los atributos se encuentren dentro del intervalo [-1,1]. Con ésto conseguimos evitar la mayor influencia de algunos atributos debido a que se encuentren dentro de un dominio de valores mayor. Por ejemplo si vamos a agrupar viviendas y utilizamos como características el número de habitaciones y el precio, seguramente el número de habitaciones no será mucho mayor de 3 y en cambio el precio puede ser mayor incluso de 300000 €.[1]

Tercera Fase: Algoritmo K-means

En esta fase se implementa el algoritmo **K-means**.

1. En primer lugar inicia los *centroides* con el criterio establecido por el usuario, o la matriz de bits de pertenencias.
2. Recorre las instancias del conjunto y calcula la distancia a cada uno de los *codeword* actualizando la matriz de bits de pertenencia,el valor del bit es uno si es el centroide más cercano a la instancia.
3. Se calcula de nuevo el vector promedio para cada cluster.
4. Iterar los pasos dos y tres hasta converger.

Cuarta Fase: Evaluación

En esta fase se tratará de automatizar la evaluación del algoritmo frente a los datos obtenidos con distintas ejecuciones, variando los parámetros o incluso con los resultados obtenidos con el modelo de **K-means** que ya implementa el API de **weka**.

Dependencias

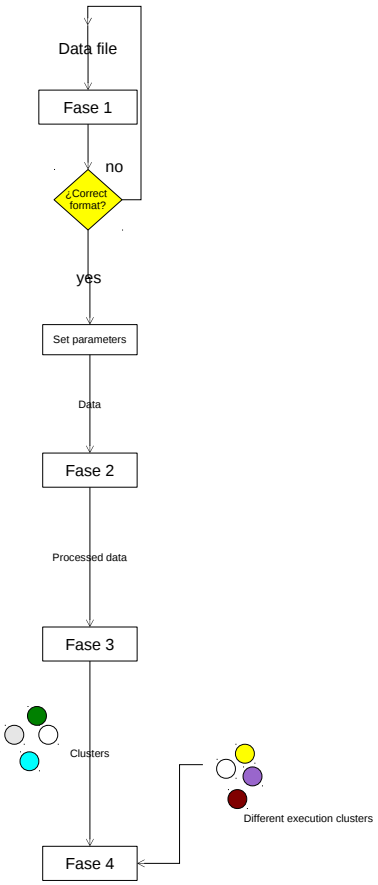

Figura 1: Dependencias del sistema

4.1. Algoritmo en pseudocódigo

```
Let k be the number of clusters to partition the data set
Let X = x_1, x_2, ..., x_n be the data set to be analyzed
Let M = m_1, m_2, ..., m_k be the code-book associated to the clusters
Let dist(a, b) be the desired distance metric
Let B = B_11, B_12, ..., B_nk be the temporary pertenece bit matrix

Ensure: C = C_1, C_2, ..., C_k set of clusterized instances

Begin:

  //randomly initialize the first centroids
  for each m_j
    m_j = randomsample(X)
  end

  //assign dataset instances to each cluster generated by the centroids
  for each x_n
    B_nj = 1 si argmin dist(x_n, m_j) = m_j \foreach m_j si no B_nj = 0
  end

  for each B_nj
    if B_nj == 1
      C_j.add(x_i)
    end
  end

  //iterate the algorithm generatin new centroids based on previously clusterized instances until
      there are no changes between iterations
  while changes in M do
    for each m_j
      m_jnew = calculatecentroid(C_j)
      if m_jnew == m_j
        changes = false
      else
        changes = true
      end
      m_j = m_jnew
    end

    for each x_n
      B_nj = 1 si argmindist(x_n, m_j) = m_j \foreach m_j si no B_nj = 0
    end

    for each B_nj
      if B_nj == 1
        C_j.add(x_i)
      end
    end
  end

  return C = C_1, C_2, ..., C_k
end
```

5. Implementación

5.0.1. Problemas encontrados

5.0.2. Soluciones adoptadas

6. Validación del *software*

6.1. Diseño del banco de pruebas

7. Análisis de resultados

7.1. Modificando inicializaciones

7.2. Modificando distancia Minkowski

7.3. Criterios de convergencia

7.3.1. Número fijo de iteraciones

7.3.2. Disimilitud entre *codebooks*

7.4. Distintas métricas

7.4.1. Manhattan

7.4.2. Euclídea

7.4.3. Minkowski

8. Clasificación supervisada respecto de

9. Conclusiones

- Breve descripción de las motivaciones para llevar a cabo técnicas de clustering.
- Conclusiones a la vista de los resultados más relevantes.
- Conclusiones generales.(Análisis de fortalezas del sw y reflexiones sobre la tarea.
- Análisis de puntos débiles y propuestas de mejoras.

10. Valoración subjetiva

1. ¿Has alcanzado los objetivos que se plantean?
2. ¿Te ha resultado de utilidad la tarea planteada?
3. ¿Qué dificultades has encontrado?Valora el grado de dificultad de la tarea.
4. ¿Cuánto tiempo has trabajado en esta tarea? Desglosado:

Coste temporal	
Diseño de software	
Implementación de software	
Tiempo trabajando con Weka	
Búsqueda bibliográfica	
Informe	1

5. Sugerencias para mejorar la tarea. Sugerencias para que se consiga despertar mayor interés y motivación en los alumnos.

6. Críticas(constructivas).

Referencias

[1] Andrew Ng. Machine learning, 2014.