Universidad de Huelva

Universidad de Huelva

Grado en Ingeniería Informática

Tema 8. Aprendizaje No Supervisado

Resumen

Autor: Alberto Fernández Merchán
Asignatura: Aprendizaje Automático

1. Introducción

El aprendizaje no supervisado (o clustering) consiste en organizar datos sin etiquetar forman grupos de similitud llamados **clusters**. Un cluster es una colección de elementos de datos que son similares entre sí y diferentes a los elementos de datos de otros grupos.
Para poder hacer **clustering** necesitamos:

- Una **medida** de proximidad o similaridad. Podemos definir la distancia entre objetos y la similitud entre estos de la siguiente forma:
  - medida **distancia**: $d(x_i, x_k)$ será pequeña si x_i y x_k son similares.
  - medida de **similitud**: $s(x_i, x_k)$ será grande si x_i y x_k son similares.
  - Podemos relacionarlas de la siguiente forma: $s(x_i, x_k) = \frac{1}{1+d(x_i,x_k)}$. De esta forma:
    - Si los puntos son iguales, la distancia será 0 y, por tanto, la similitud será de 1.
    - Si los puntos son muy diferentes, la distancia tenderá a ∞ y, por tanto, la similitud será 0.
  - Hay diferentes formas de calcular la distancia:
    - **Distancia Euclídea**: Es invariante a la traslación.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} \left(x_i^{(k)} - x_j^{(k)}\right)^2}$$

    - **Distancia Manhattan**: Es una aproximación a la distancia euclídea, pero más económica de calcular:

$$d(x_i, x_j) = \sum_{k=1}^{n} \left|x_i^{(k)} - x_j^{(k)}\right|$$

    - **Distancia de Minkowsky**:

$$d(x_i, x_j) = \left(\sum_{k=1}^{n} \left|x_i^{(k)} - x_j^{(k)}\right|^p\right)^{1/p}$$

    - **Distancia del coseno**:

$$d(x_i, x_j) = \cos(\theta_{i,j}) = \frac{x_i \cdot x_j}{||x_i|| \cdot ||x_j||} = \frac{\sum_{k=1}^{n} x_i^k * x_j^k}{\sum_{k=1}^{n} (x_i^k)^2 * \sum_{k=1}^{n} (x_j^k)^2}$$

- Una **función de evaluación** de las agrupaciones para evaluar los clústeres a 2 niveles. Es un problema duro computacionalmente. Existen dos tipos de evaluaciones:
  - Cohesión **intra-cluster** (compacidad): Mide qué tan cerca están los puntos de datos en un grupo al centroide del grupo. Comúnmente se utiliza la suma del error cuadrático (SSE).
  - Separación **inter-cluster** (aislamiento): La separación significa que los diferentes clústeres deben estar muy alejados entre sí.

  Para evaluar como de buena es la clasificación podemos utilizar la siguiente función de evaluación:

$$f(x, y) = \alpha \cdot x + \beta \cdot y$$

  donde:

  - α y β son factores de influencia entre 0 y 1.
  - x es la distancia intracluster.
  - y es la distancia intercluster.

- Un **algoritmo** para calcular la agrupación, por ejemplo, optimizando la función de criterio. Existen 3 tipos de algoritmos para realizar clustering:

  1. **Particionales**: Normalmente determinan todos los clústeres a la vez, pero también se pueden usar como algoritmos decisivos en el clúster jerárquico.
  2. **Jerárquicos**: Encuentran agrupaciones sucesivas utilizando agrupaciones previamente establecidas. Pueden ser:
     - **Aglomerativos**: Comienzan con cada ejemplo como un grupo separado y los fusionan en grupos sucesivamente más grandes.
     - **Divisivos**: Comienzan con el conjunto completo y proceden a dividirlo en grupos sucesivamente más pequeños.
  3. **Bayesianos**

2. Ejemplos de algoritmos

2.1. K-Means

K-means (MacQueen, 1967) es un algoritmo de agrupamiento particional. Sea el conjunto de puntos de datos $D\{x^1, x^2, \ldots, x^m\}$, donde $x^i = (x_1^i, x_2^i, \ldots, x_n^i)$ es un vector y **n** es el número de atributos o dimensiones. El algoritmo divide los datos en **k** grupos. Cada grupo tiene un centro de grupo llamado **centroide** y el **k** lo especifica el usuario.

```
Init k
Choose k (random) data points (seeds) to be the initial centroids, cluster centers
Repeat until convergence criteria:
    Assign each data point to the closest centroid
    Re-compute the centroids using the current cluster memberships
```

Figura 1: Pseudocódigo del algoritmo K-Means

2.1.1. Criterio de Convergencia del K-Means

Existen varios criterios que podemos usar para considerar que el algoritmo K-means ha convergido:

- Número de iteraciones fija.
- Que tenga ninguna (o mínima) reasignación de puntos a diferentes grupos.
- Ningún (o mínimo) cambio en los centroides.
- Una disminución mínima en la suma del error cuadrático:

$$SSE = \sum_{j=1}^{k} \sum_{x \in C_j} d(x, m_j)^2$$

donde:

- C_j: es el clúster j.
- m_j: es el centroide del clúster j
- $d(x, m_j)$: es la distancia entre el punto x y el centroide m_j.

2.1.2. Ventajas

- Es fácil de implementar y comprender.
- La complejidad temporal es de $O(tkn)$. Siendo **n** el número de puntos, **k** el número de clusteres y **t** el número de iteraciones.
- Como k y t suelen ser pequeños, se considera un algoritmo lineal.
- Es el algoritmo de agrupación más popular.

El algoritmo puede terminar en un óptimo local si se utiliza la suma del error cuadrático (SSE). El óptimo local es difícil de encontrar. Lo que se suele hacer es repetir la ejecución varias veces y elegir la que ofrezca el error más bajo.

2.1.3. Desventajas

- El algoritmo es aplicable, solamente, si se define la media. Para datos categóricos el centroide estará representado por los valores más frecuentes.
- El usuario debe especificar k.
- El algoritmo es sensible a **outliers**. Los outliers son puntos de datos que están muy lejos de otros puntos. Pueden ser errores en el registro de datos o algunos datos especiales con valores muy diferentes. Debemos tratarlos para no obtener clústeres indeseables:
  - Eliminar algunos puntos de datos que estén mucho más lejos de los centroides que otros puntos de datos. Es posible que deseemos monitorear estos posibles valores atípicos en algunas iteraciones y luego decidir si eliminarlos.
  - Realizar un muestreo aleatorio. Al elegir un subconjunto de los puntos de datos, la posibilidad de seleccionar un valor atípico es mucho menor. Asignar el resto de los puntos de datos a los conglomerados por distancia o comparación de similitud o clasificación.
- Es sensible a las semillas iniciales. Dependiendo de la semilla inicial puede generar un clúster u otro.
- No es adecuado para descubrir clústeres que no son hiper-elipsoides.

2.2. Clustering Jerárquico

Para algunos datos, el agrupamiento jerárquico es más apropiado que el agrupamiento plano que hacen los algoritmos particionales.
La forma preferida de representar estas jerarquías es utilizando un **dendrograma**:

- Es un árbol binario
- El nivel **k** corresponde a la partición con $n - k + 1$ clusters.
- Si necesitamos k clsters, tomaremos el nivel $k - n + 1$.
- Si los ejemplos están en el mismo grupo en el nivel k, permanecerán en el mismo grupo para niveles más altos.
- Un dendrograma muestra típicamente la similitud de grupos creados.

Existen dos tipos de agrupaciones jerárquicas:

- **Divisiva** (*top-down*): Comienza con todos los puntos de datos en un clúster, la raíz, luego divide la raíz en un conjunto de clusteres secundarios. Cada grupo secundario se divide más recursivamente. Se detiene cuando solo quedan grupos con un solo punto individual.

  Este algoritmo es menos «*ciego*» que su contraparte a la estructura global de los datos. Al dar el primer paso, tiene acceso a todos los datos y puede encontrar la mejor división posible en dos partes.

- **Aglomerativa** (*bottom-up*): El dendrograma se construye desde el nivel inferior fusionando el par de clústeres más similares o cercanos. Se detiene cuando todos los puntos se fusionan en un solo grupo (la raíz).

  Es más rápido de calcular que el divisivo, en general, al dar el primer paso para fusionar no considera la estructura global de los datos. Solo observa la estructura por pares.

```
Initialize with each example in singleton cluster
while there is more than 1 cluster:
    find the nearest clusters
    merge them
```

Figura 2: Pseudocódigo del algoritmo aglomerativo

Existen cuatro formas para medir la distancia del grupo:

1. Distancia mínima(*single-linkage*): $d_{min}(C_i, C_j) = min_{x \in C_i, y \in C_j} ||x - y||$.
   - Genera un árbol de expansión mínimo.
   - Estimula el crecimiento de grupos alargados.
   - Es muy sensible al ruido.
2. Distancia máxima (*complete-linkage*): $d_{max}(C_i, C_j) = max_{x \in C_i, y \in C_j} ||x - y||$.
   - Fomenta los clusteres compactos.
   - No funciona bien con clusteres alargados.
3. Distancia media: $d_{avg}(C_i, C_j) = \frac{1}{n_i \cdot n_j} \sum_{x \in C_i} \sum_{y \in C_j} ||x - y||$.
4. Distancia centroide (*centroid-linkage*): $d_{cent}(C_i, C_j) = ||\mu_i - \mu_j||$.
   - Favorece distribuciones hiper-esféricas.

Cada distancia genera una distribución diferente de agrupaciones. No existe un método concreto que de buenos resultados siempre.