UNIVERSIDAD DE GRANADA

Clustering con Knime

Tratamiento Inteligente de Datos
Master Universitario en Ingeniería Informática

En esta práctica veremos el uso de algoritmos de agrupamiento o clustering utilizando Knime. Se trabajará con un conjunto de datos reales sobre el que se emplearán diferentes técnicas de clustering y a la luz del conocimiento descubierto se podrán concluir diferentes aspectos sobre los datos. Para ello, se deberán crear informes de resultados y análisis lo suficientemente profundos para resultar de utilidad. Obviamente, se valorará el acierto en los recursos de análisis gráficos empleados, la complejidad de los experimentos realizados, la interpretación de los resultados, la organización y redacción del informe, etc.

1. Clustering en KNIME

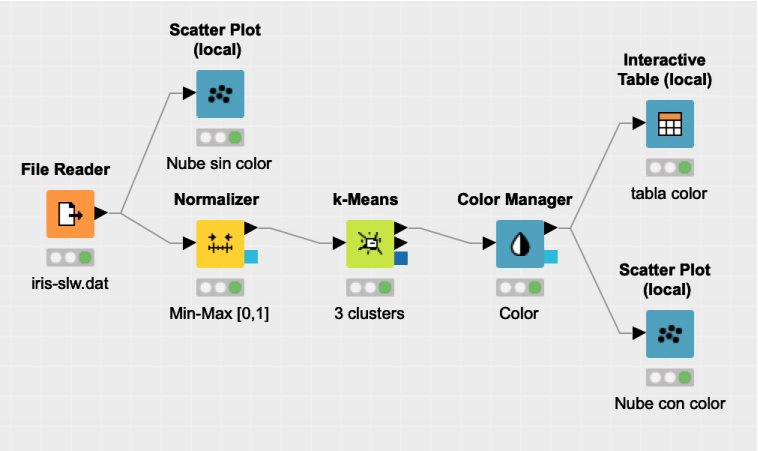
Los algoritmos de clustering en KNIME se encuentran en la carpeta `Analytics > Mining > Clustering` del repositorio de nodos. Sigue los pasos siguientes para ver un pequeño ejemplo de su uso.

Crea un proyecto en KNIME con los siguientes nodos que nos permitan agrupar los datos de `iris-slw.dat`. Este fichero contiene los datos de instancias de plantas de la familia iris (lirios) a las que se les ha calculado longitud y anchura del sépalo. Este fichero es una manipulación de `iris.data` en el que cada instancia contiene cuatro características (longitud y anchura del sépalo, y del pétalo) y también el tipo de planta (la solución al agrupamiento). Por simplicidad, se ha utilizado está versión reducida pero se puede utilizar como ejercicio el archivo original. Basta eliminar de las instancias la última columna para realizar verdaderamente un aprendizaje no supervisado y comparar después con los agrupamientos verdaderos.

Se han utilizado los siguientes nodos.

- Un nodo para leer el fichero.
- Un nodo `Manipulation > Column > Transform > Normalizer` para normalizar los datos, ya que KNIME implementa k-Means sin normalizar previamente las variables.
- Un nodo `Analytics > Mining > Clustering > k-Means` para realizar el clustering. Este nodo añadirá una columna Cluster, indicando el agrupamiento asignada a cada tupla de nuestro conjunto de datos. En su configuración, debemos indicar los atributos que se usarán para establecer los clusters. En nuestro ejemplo, usaremos ambos.
- Un nodo `Views > Property > Color Manager` para colorear los datos correspondientes a la columna Cluster del nodo anterior.
- Un nodo `Views > Local > Interactive Table` para ver los resultados.
- Dos nodos `Views > Local > Scatter Plot` para ver la nube de puntos original y la coloreada con los clusters obtenidos.

El diagrama resultante debe quedar de la siguiente manera:

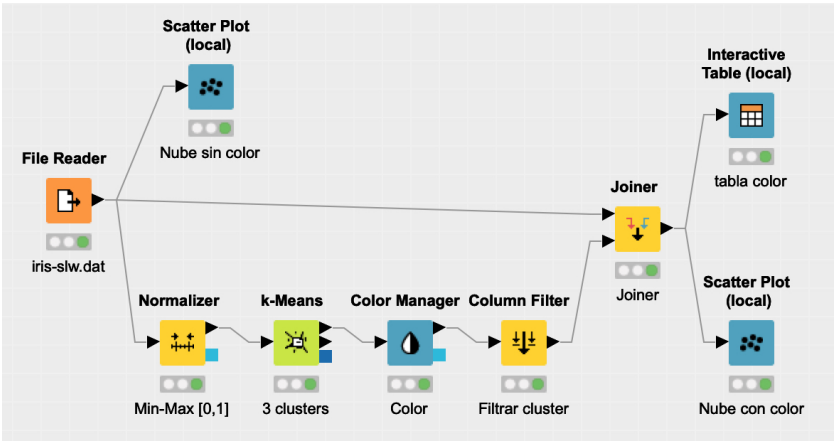


Para ver los centroides obtenidos, seleccione con la derecha el nodo `k-Means` y muestre los resultados (`View: Cluster View`). Por otro lado, si abrimos ahora los nodos `Interactive Table` y `Scatter Plot` y pinchamos sobre un punto en la nube de puntos y con el ratón derecha seleccionamos `Hilite`, dicho punto se marca como naranja y, además, también se marca el dato correspondiente en la tabla interactiva.

Observa que, en nuestro proyecto actual, tanto la tabla como la nube de puntos nos muestran los datos normalizados. Si queremos ver los datos originales, basta hacer lo siguiente:

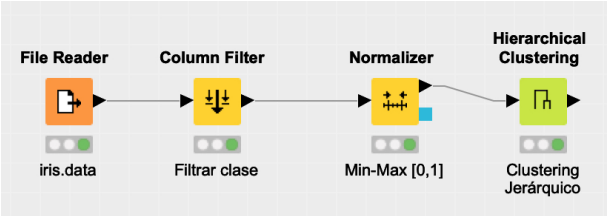
- De la salida de `k-Means`, mantenga únicamente el atributo Cluster y el identificador de fila (RowID). Esto lo podemos hacer con un nodo de tipo `Column Filter` aplicado sobre la salida de `Color Manager`.
- A continuación, combina el resultado del `Column Filter` con los datos originales mediante un nodo `Joiner` (eligiendo el RowID como Join Column e Inner Join como método de reunión).

Así quedaría nuestro proyecto KNIME tras realizar estas modificaciones:

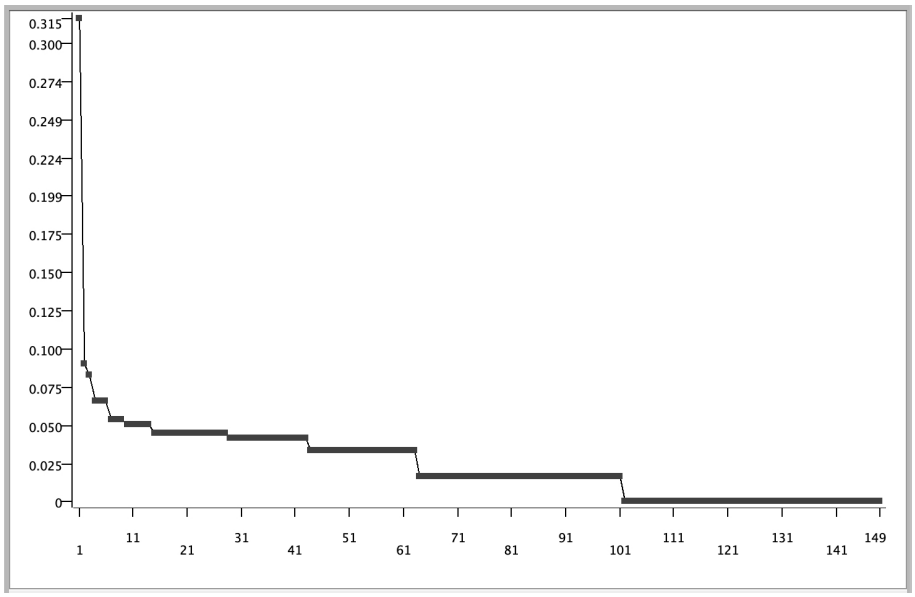


Este fichero de datos es parte de un conjunto de datos muy conocido que se utiliza normalmente como ejemplo de clasificación. Las instancias corresponden a tres categorías diferentes: Iris Setosa, Iris Versicolour e Iris Virginica. Por lo tanto, es lógico que hayamos utilizado tres cluster en el algoritmo k-medias. Sin embargo, a priori, sin conocer esta información, no está claro el número de cluster a utilizar. Para intentar determinarlo, podemos realizar un clustering jerárquico usando el nodo `Hierarchical Clustering` sobre el archivo `iris.data` que

contiene el conjunto de datos completo de atributos e instancias.

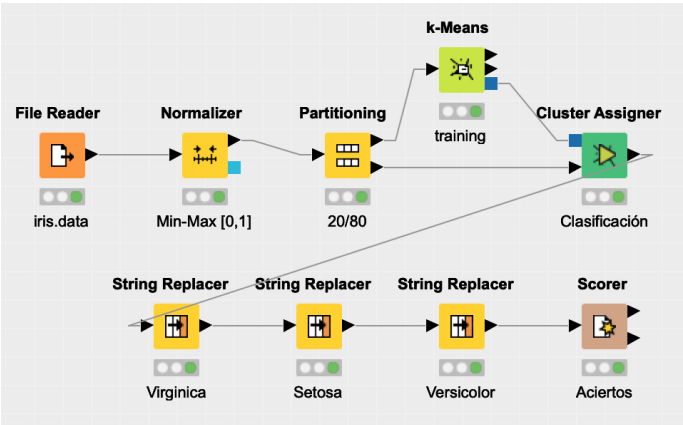


Hemos filtrado el atributo clase puesto que es la solución al problema. En la configuración del nodo `Hierarchical Clustering` seleccionamos, por ejemplo, la distancia euclídea y el tipo de enlace `SINGLE`. Entonces obtenemos una distancia según clusters como especifica el siguiente gráfico (se puede obtener pinchar con el botón derecho en `Hierarchical Clustering` y seleccionar `View: Dendogram /Distance View`)



El gráfico parece sugerir que el número de cluster debería ser tres, cuatro o cinco.

Por otro lado, los algoritmos de clustering pueden utilizarse para clasificación. En KNIME esta tarea se puede realizar con el nodo `Cluster Assigner`. Por ejemplo, vamos a considerar de nuevo el archivo `iris.data` y vamos a dividir los datos en dos conjuntos: uno de entrenamiento y otro de validación.



Con el conjunto de entrenamiento, utilizamos el algoritmo k-medias (para tres cluster con la distancia Euclídea) para crear el modelo de prototipos. Comparando con los datos reales, vemos que el modelo nombra como `Cluster_0` al cluster formado por `iris-virginica`, `Cluster_1` al cluster formado por `iris-setosa` y

`Cluster_2` al cluster formado por `iris-versicolor`. Una vez clasificado el conjunto de validación, cambiamos los nombres con el nodo `String Replacer` para poder comparar con los datos reales. Este proceso nos da un acierto del 85.8 %.

Por último, los algoritmos de clustering sirven para detectar outliers. Si usamos un algoritmo de clustering jerárquico, un caso extremo o outlier se unirá en algún momento a aquel cluster que esté más cercano a él. Si bien que un cluster contenga muchos ejemplos no quiere decir que necesariamente contenga outliers, lo que sí es cierto es que un outlier se unirá siempre a un cluster en las fases finales del agrupamiento jerárquico: los outliers serán aquellos valores que, de forma aislada, se unen a algún cluster en las últimas iteraciones del algoritmo jerárquico. Considera el archivo `iris-slw_outliers.dat`.

- Realiza un clustering jerárquico similar a lo anteriormente explicado.
- Analiza la existencia de outliers.
- Dibuja los puntos en el plano y compara el análisis del punto anterior con lo visto en el gráfico
- En el caso de existir outliers, elimínalos y repite el estudio. Compara ambos experimentos.
- Del archivo con los outliers eliminados, estudia como se realizaría (qué nodos utilizar) en KNIME un agrupamiento utilizando el algoritmo k-medoides (nodo `k-Medoids`) para tres cluster y compara con el punto anterior.

2. Vino

El archivo `wine.data` contiene los datos reales de 178 vinos de una misma región de Italia. Cada instancia está compuesta por trece atributos numéricos más una clase (la primera columna) que determina el nivel de alcohol del vino (tres tipos; 1, 2 y 3), y es la solución al proceso de clustering (por lo que se debe eliminar para realizar la tarea de minería de datos). La descripción de los atributos se encuentra en el fichero `wine_names.txt` (por algún motivo, falta la descripción de una de las columnas). Se deben realizar las siguientes actividades:

- Realizar un algoritmo de clustering jerárquico para analizar en cuántos clusters diferentes podríamos agrupar los datos.
- Analiza la existencia de outliers y elimínalos si consideras que existe alguno. Repite el clustering jerárquico y vuelve a analizar el número de cluster a considerar.
- Aplica el algoritmo k-medias al archivo del punto anterior con el número de clusters elegido. Compara cómo se distribuyen los clusters respecto a las clases de la primera columna.
- Aplica algún tipo de reducción de dimensionalidad que consideres oportuno (filtrando columnas, correlacción, análisis de componentes principales, etc...) y aplica el algoritmo k-medias para tres cluster. De nuevo, compara cómo se distribuyen los clusters respecto a las clases de la primera columna.
- Aplica el algoritmo DBSCAN utilizando el nodo contenido en la carpeta de weka al archivo original para varios parámetros de radio y puntos mínimos. Compara cómo se distribuyen los clusters respecto a la primera columna ¿Se pueden identificar los outliers?