APRENDIZAJE AUTOMÁTICO AVANZADO

INFORME TÉCNICO UNIDAD III – APRENDIZAJE NO SUPERVISADO

PRESENTADO POR:

Edgar Leandro Jiménez Jaimes

Santiago Echeverri Calderón

DOCENTE:

José Lisandro Aguilar Castro

UNIVERSIDAD EAFIT

MEDELLÍN

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

FEBRERO DE 2020

1. Objetivo de la iteración

Analizar y aplicar en un conjunto de datos diferentes técnicas de Aprendizaje No Supervisado con el fin de encontrar agrupaciones que expliquen relaciones entre las observaciones.

2. Contextualización del problema

Las técnicas de Aprendizaje No Supervisado se implementarán en un conjunto de datos proveniente de un experimento de reconocimiento de actividades humanas mediante el uso de los sensores en teléfonos inteligentes.

"El experimento se llevó a cabo con un grupo de 30 voluntarios dentro de un rango de edad de 19-48 años. Cada persona realizó seis actividades (caminar, subir escalas, bajar escalas, estar sentado, estar acostado, estar de pie) usando un teléfono Samsung Galaxy S II en la cintura. Usando el acelerómetro y giroscopio, se capturó la aceleración lineal y la velocidad angular a una velocidad constante de 50Hz. Los experimentos fueron grabados en video y posteriormente se etiquetaron los datos manualmente".[1]

El conjunto de datos se encuentra disponible en:

https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

Aunque los datos están etiquetados, el propósito de la presente iteración es probar diferentes técnicas de Aprendizaje No Supervisado. Para esto se descartarán las etiquetas del conjunto de atributos al momento de usar los algoritmos de clustering, pero serán usadas posteriormente como criterios externos que permitan calcular métricas para evaluar los algoritmos.

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

3. Diseño del modelo

El modelo consta de 3 etapas:

i. Exploración y preprocesamiento de los datos.
ii. Implementación de algoritmos de clustering.
iii. Evaluación del agrupamiento.

3.1. Exploración y preprocesamiento de los datos.

El dataset original fue suministrado en 2 archivos, train y test, pues el propósito original del conjunto es un modelo de clasificación supervisada. Para el presente ejercicio no se requería la partición de los datos así que los 2 archivos se unificaron en un solo dataframe. Se encontraron 561 atributos numéricos, más la etiqueta de la actividad y una identificación del individuo; y 10,229 observaciones.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10299 entries, 0 to 10298
Columns: 563 entries, tBodyAcc-mean()-X to Activity
dtypes: float64(561), int64(1), object(1)
memory usage: 44.2+ MB
```

Al dataframe se le removieron los campos 'Subject' y 'Activity' pues no debían considerarse para la agrupación. Y el campo 'Activity' que contenía las etiquetas de la actividad se guardó en un dataframe separado.

```
# Remoción de las columnas Activity y subject
df_etiquetas = df_unido['Activity'].to_frame(name = 'Activity')
df_atributos = df_unido.drop(['subject', 'Activity'], 1)
```

Sobre el campo 'Activity' se usó un label enconder para convertir las etiquetas de texto a un código numérico y se realizó una agrupación y conteo de observaciones por actividad:

	Activity	Activity_code	count
0	LAYING	0	1944
1	SITTING	1	1777
2	STANDING	2	1906
3	WALKING	3	1722
4	WALKING_DOWNSTAIRS	4	1406
5	WALKING_UPSTAIRS	5	1544

Posteriormente se realizó una estadística descriptiva básica de los datos en la cual se encontró que los datos ya habían sido normalizados y que no había valores nulos, por lo tanto, no fue necesario hacer mayor preprocesamiento adicional.

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y
count	10299.000000	10299.000000	10299.000000	10299.000000	10299.000000
mean	0.274347	-0.017743	-0.108925	-0.607784	-0.510191
std	0.067628	0.037128	0.053033	0.438694	0.500240
min	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
25%	0.262625	-0.024902	-0.121019	-0.992360	-0.976990
50%	0.277174	-0.017162	-0.108596	-0.943030	-0.835032
75%	0.288354	-0.010625	-0.097589	-0.250293	-0.057336
max	1.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 561 columns

Antes de hacer la agrupación se realizó una reducción de dimensionalidad mediante PCA. Se escogieron las 68 primeras componentes, las cuales explican el 94.9% de la varianza de los datos. Este número se escogió probando posteriormente el resultado de los algoritmos, procurando que en cada cluster la clase de mayor frecuencia fuera diferente, esto se explicará con más detalle en la siguiente sección.

3.2. Implementación de algoritmos de clustering y pruebas

Con el fin de probar diferentes técnicas de clustering se decidió implementar un modelo de cada tipo: un método de particiones (k-means), uno de densidad (DBSCAN) y uno de jerarquía (aglomerativo).

Clustering mediante k-means

El primer método que se implementó fue k-means y fue el método que se usó como base. Como ya se conocía el número de clases en el dataset (6 actividades), se configuró el algoritmo con k=6. Los resultados se unieron con el dataframe de las etiquetas (actividades) y se realizó una agrupación por cluster asignado y actividad, con un conteo por grupo para observar el desempeño del algoritmo. A continuación, se presentan la tabla con dichas agrupaciones:

Cluster	Activity	count
0	LAYING	12
0	SITTING	3
0	WALKING	700
0	WALKING_DOWNSTAIRS	187
0	WALKING_UPSTAIRS	1,166
1	SITTING	1,257
1	STANDING	1,240
2	WALKING	153
2	WALKING_DOWNSTAIRS	456
2	WALKING_UPSTAIRS	82
3	LAYING	172
3	SITTING	468
3	STANDING	666
4	LAYING	1,760
4	SITTING	49
5	WALKING	869
5	WALKING_DOWNSTAIRS	763
5	WALKING_UPSTAIRS	296

Se puede observar que, aunque los clusters reúnen diferentes actividades, hay una relación entre las actividades agrupadas. Por ejemplo, en el cluster 0, las actividades de mayor frecuencia son caminando (99% dentro del cluster), sin embargo, al algoritmo le cuesta diferenciar si el caminar es en plano o en escalas. Y en todos los clusters se puede observar este mismo comportamiento, agruparon actividades únicamente estáticas o únicamente en movimiento.

Posteriormente, para determinar a qué actividad corresponde cada cluster se calculó la clase con mayor frecuencia dentro de cada agrupación:

cluster	Activity	Activity_code	count
0	WALKING_UPSTAIRS	5	1,166
1	SITTING	1	1,257
2	WALKING_DOWNSTAIRS	4	456
3	STANDING	2	666
4	LAYING	0	1,760
5	WALKING	3	869

Un aspecto positivo que se observa en la clasificación es que sí hay una correspondencia entre los clusters y las clases, ya que no hay múltiples clusters en los que la misma clase sea la de mayor frecuencia. Cuando se definió el valor de 94.9% de varianza explicada para la selección de variables mediante PCA, se hizo validando este resultado de agrupación, procurando que cada cluster correspondiera a una clase.

Para evaluar el desempeño del algoritmo se usaron 2 métricas, una1 interna (coeficiente de silueta) y otra externa (coeficiente de pureza total). Los resultados fueron los siguientes:

Coeficiente de pureza total: 59.9%

Coeficiente de silueta: 0.15

Cluster	Actividad	Coeficiente de pureza
4	WALKING_UPSTAIRS	90.5%
0	SITTING	75.5%
1	WALKING_DOWNSTAIRS	70.7%
5	STANDING	50.5%
3	LAYING	34.9%
2	WALKING	32.4%

El coeficiente de pureza total indica el porcentaje de observaciones que fueron clasificadas en el cluster donde predominaban las observaciones de la misma clase. Con un 59.9% de pureza total el desempeño de la agrupación es bajo. Uno de los motivos que puede explicar el bajo desempeño se evidencia con el coeficiente de silueta, el cual indica la relación de la distancia media entre elementos de un cluster con la distancia media a los elementos que nos están en ese cluster. Un valor 0.15 indica que hay superposición entre los clusters, y esta superposición impacta en la pureza de las agrupaciones.

A nivel de pureza por clusters se observa que las actividades de subir escalas, estar sentado y bajar escalas son más diferenciables de las demás y tuvieron un mejor desempeño.

Clustering mediante DBSCAN

Posteriormente se implementó una agrupación basada en densidades con el algoritmo DBSCAN. Sin embargo, y a pesar de que se realizaron múltiples iteraciones modificando los hiperparámetros *eps* y *min_samples*, el resultado siempre fue un cluster único y el cluster -1 con unas cuantas observaciones de ruido.

Este mal desempeño se debe a que los dominios de alta dimensión son muy exigentes para este tipo de algoritmo. Adicionalmente DBSCAN produce cluster mutuamente excluyentes, y como se observó en la agrupación con k-means el conjunto de datos es muy denso y tiene clusters superpuestos, por esto el algoritmo DBSCAN agrupa todos los puntos en un único cluster.

Clustering mediante Jerarquía Aglomerativa

Para finalizar se usó un método basado en jerarquía. Así como en k-means se usó k=6, por el número de clases ya conocido. Este algoritmo ofrece un hiperparámetro llamado criterio de vinculación o enlace, el cual determina la distancia entre conjuntos de observaciones en función de las distancias por pares entre observaciones, entonces se evaluó el coeficiente de pureza total para cada criterio de enlace y los resultados fueron los siguientes:

Criterio de enlace	Pureza total
Ward (minimiza la varianza entre los clusters)	64.53%
Completo	37.70%
Promedio	40.02%
Único	18.90%

Se eligió el tipo de enlace *Ward* por ser el de mejor desempeño.
Los siguientes son los resultados agrupados por cluster y actividad con criterio de enlace *Ward*:

Cluster	Activity	Count
0	LAYING	1,944
0	SITTING	65
0	STANDING	2
1	WALKING	1,421
1	WALKING_DOWNSTAIRS	1,021
1	WALKING_UPSTAIRS	497
2	SITTING	1,021
2	STANDING	923
3	WALKING	250
3	WALKING_DOWNSTAIRS	138
3	WALKING_UPSTAIRS	1,032
4	WALKING	51
4	WALKING_DOWNSTAIRS	247
4	WALKING_UPSTAIRS	15
5	SITTING	691
5	STANDING	981

Se puede observar que el algoritmo realizó agrupaciones de actividades únicamente estáticas o actividades únicamente de movimiento. Y así como en k-means le cuesta diferenciar las sub-actividades. Por ejemplo, los clusters 2 y 5 son muy parecidos, sólo contienen las actividades estar sentado y estar acostado.

Al evaluar la pureza a nivel de cada cluster se observa que el algoritmo fue capaz de agrupar con muy buena precisión las actividades estar acostado (100% de pureza) y caminar (82.5% de pureza), sin embargo, su desempeño es bajo para las demás actividades:

Cluster	Activity	Coeficiente de pureza
0	LAYING	100.0%
1	WALKING	82.5%
3	WALKING_UPSTAIRS	66.8%
2	SITTING	57.5%
5	STANDING	51.5%
4	WALKING_DOWNSTAIRS	17.6%

También se evaluó el coeficiente de silueta de la agrupación, el cual arrojó un valor 0.09, indicando un nivel de superposición muy alto. Esto se puede observar en los clusters 2 y 5 los cuales contienen cantidades de puntos en proporciones muy similares de las actividades estar sentado y estar acostado.

4. Análisis de Resultados

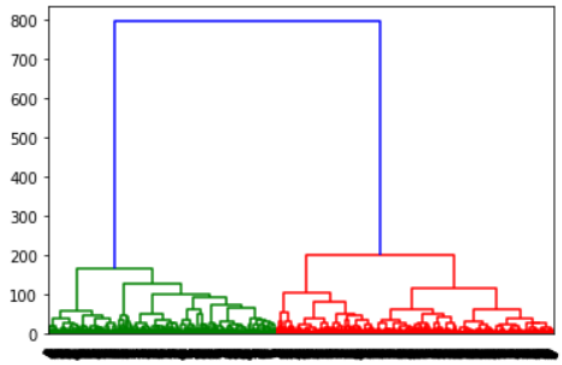
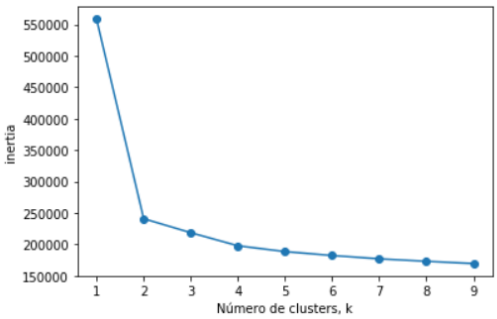
El conjunto de datos por ser de alta dimensionalidad y densidad dificulta la agrupación de las observaciones. Aunque a nivel de pureza por cluster los 2 algoritmos fueron capaces de separar con un buen desempeño algunas de las clases, tuvieron errores en otras y la pureza total fue baja en ambos casos:

Activity	Coeficiente de pureza	
	Jerárquico	k-means
LAYING	100.0%	34.9%
SITTING	57.5%	75.5%
STANDING	51.5%	50.5%
WALKING	82.5%	32.4%
WALKING_DOWNSTAIRS	17.6%	70.7%
WALKING_UPSTAIRS	66.8%	90.5%

Coeficiente de pureza total	
Jerárquico	k-means
64.53%	59.90%

La agrupación jerárquica sin embargo tuvo un mejor desempeño que k-means.

Para entender mejor los resultados de la agrupación se usaron técnicas para hallar el k optimo de cada uno de los algoritmos:

Método con dendograma para clustering jerárquico	Método elbow para k-means
	

Como se puede observar en las gráficas, ambos métodos indican que los algoritmos tendrían un mejor desempeño separando las observaciones en 2 clusters, aunque por el tipo de experimento se sabe que los datos corresponden a 6 clases. Este fenómeno puede ocurrir debido a que en las actividades se pueden encontrar 2 grandes grupos:

Actividades estáticas, comprendidas por las clases:

- LAYING
- SITTING
- STANDING

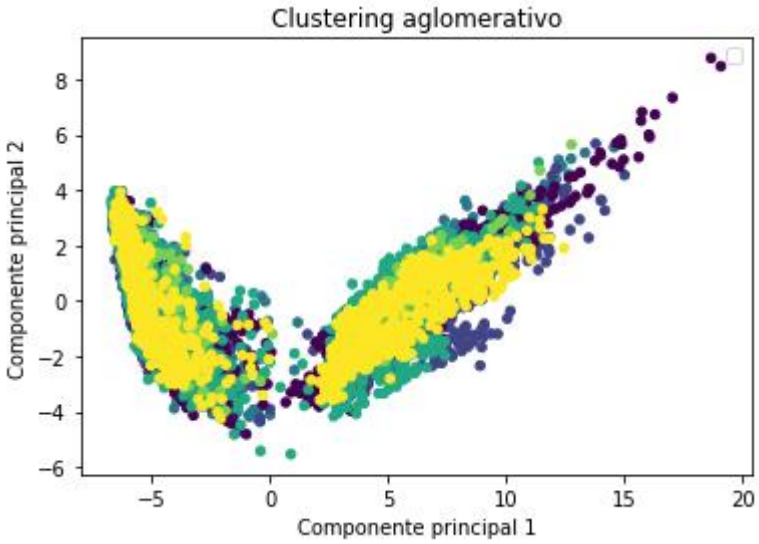
Y actividades en movimiento, comprendidas por las clases:

- WALKING
- WALKING_DOWNSTAIRS
- WALKING_UPSTAIRS

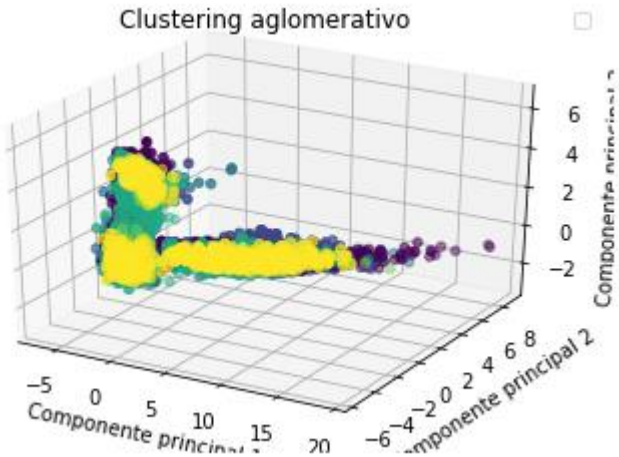
En cada grupo el comportamiento del acelerómetro y el giroscopio es muy diferente, por esto es natural que ambos métodos sugieran que el óptimo es de 2 agrupaciones.

Finalmente, para tratar de comprender mejor los resultados se realizaron visualizaciones de los clusters. Como el conjunto tiene 68 variables se usó el método PCA para graficar las 2 y las 3 componentes más importantes de los clusters obtenidos con el método jerárquico aglomerativo que fue con el que se obtuvo un mayor coeficiente de pureza total.

Visualización con las 2 componentes principales



Visualización con las 3 componentes principales



Con 2 y 3 componentes se puede observar que los clusters no son diferenciables a la vista y que la superposición es muy alta como lo indicaba el coeficiente de silueta.

5. Cuaderno Jupyter

El Notebook con la implementación puede encontrarse en:

https://github.com/santiagoec/CM0891-Aprendizaje-Automatico/blob/master/03_Aprendizaje_No_Supervisado.ipynb