

Evaluación de técnicas de agrupamiento para segmentación de nubes de puntos LIDAR

Nicolás Urbano Pintos^{1,2}, Héctor Lacomí^{2,3} y Mario Lavorato¹

¹ Grupo TAMA, UTN FR Haedo, Haedo, Buenos Aires, Argentina
nurbano@frh.utn.edu.ar

² DRL- CITEDEF, Villa Martelli, Buenos Aires, Argentina

³ Grupo ASE, UTN FR Haedo, Haedo, Buenos Aires, Argentina

Resumen Las nubes de puntos LIDAR suministran información precisa sobre distancias, y son de gran utilidad para robots de búsqueda y rescate urbano, vehículos autónomos y robots de uso industrial en general. Estos conjuntos de datos representan escenarios, ya sean interiores o exteriores, formados por edificios, personas, vehículos y objetos en general. La información se presenta de forma densa y no estructurada en coordenadas cartesianas de 3 dimensiones. Para analizar el entorno representado, una de las primeras tareas es la segmentación de los escenarios. En este trabajo se evalúan técnicas de agrupamiento no supervisado para segmentar conjuntos de datos de nubes de puntos LIDAR. A partir de una observación visual de las clases y el cálculo de métricas de agrupamiento, se compara el comportamiento de los algoritmos K-means, DBSCAN, BIRCH y Mean Shift en diferentes escenarios interiores y exteriores.

Keywords: segmentación · nubes de puntos LIDAR · agrupamiento

1. Introducción

En aplicaciones como la navegación autónoma [5], la robótica industrial [9], o los robots de rescate y búsqueda urbana [1], es necesario conocer el entorno circundante. Es por ello que estas tecnologías incluyen sensores, como son cámaras digitales, sensores radar, sensores de ultrasonido y sistemas LIDAR (Light Detection and Range). Los sensores LIDAR son de gran utilidad, ya que proveen información sobre distancias e intensidad. Los mismos funcionan gracias al concepto de tiempo de vuelo (ToF- Time of Flight) [4], el cual se basa en la emisión de un pulso láser, que refleja en un objetivo, y luego el pulso reflejado es captado por un sensor, generalmente un fotodiodo. El pulso láser viaja a la velocidad de la luz, y recorre una distancia que representa el doble de la distancia entre el dispositivo LIDAR y el objetivo. Por lo tanto, midiendo el tiempo transcurrido entre cada pulso se puede calcular la distancia al objetivo.

Las nubes de puntos LIDAR son obtenidas a partir de dispositivos de mapeo en 3 dimensiones, que realizan generalmente un barrido mecánico del haz de luz. Los sistemas LIDAR basados en ToF se utilizan para escanear objetos

físicos exteriores e interiores de edificios, plantas de procesamiento, topografías, escenarios de navegación autónoma, entre otros.

Una de las principales tareas para analizar estos datos es la segmentación, para ello es necesario agrupar puntos con características similares en regiones homogéneas. Este proceso es vital a la hora de realizar un análisis del entorno descrito por las nubes de puntos, y puede ser el puntapié de aplicaciones más complejas, como pueden ser detección y/o clasificación de objetos y tareas de extracción de características.

Si bien las nubes de puntos lidar nos dan información precisa acerca de las formas y el tamaño de los objetos de la escena, segmentar dicha información presenta dificultades. Las nubes de puntos son generalmente ruidosas, dispersas y no organizadas. Asimismo, la densidad de muestra de los puntos tiene variaciones debido a las velocidades lineales y angulares del dispositivo LIDAR. Además, debido a las limitaciones de los escáneres 3D, el primer plano suele estar contaminado por el fondo, lo cual trae una dificultad adicional a la hora de segmentar.

Entre los métodos más comunes para la segmentación de nubes de puntos LIDAR se encuentran los basados en bordes, en regiones, en atributos y en modelos. Los métodos basados en regiones utilizan la información de las vecindades para combinar puntos cercanos que tengan propiedades similares para obtener regiones aisladas. Estos métodos son menos afectados por el ruido que los métodos basados en borde, aunque tienen el problema de la segmentación excesiva o insuficiente y la determinación imprecisa de los límites de cada región. Si bien los métodos que detectan atributos, obtienen mejores resultados en escenarios complejos, los métodos basados en región tienen buen comportamiento en escenarios simples y permiten ejecutarlos a mayor velocidad [6].

En este trabajo se estudia el enfoque de la segmentación en regiones a partir de técnicas de agrupamiento no supervisado. Se evalúan los algoritmos DBSCAN [10], K-means [11], BIRCH [16] y Mean Shift [15] en cuatro conjuntos de datos diferentes, a partir de una representación visual de las clases predichas y del cálculo de métricas de agrupamiento. Las contribuciones de este trabajo son:

- Evaluación del comportamiento de algoritmos de agrupamiento no supervisado en diferentes conjuntos de datos de nubes de puntos LIDAR de escenarios interiores y exteriores, implementado una búsqueda iterativa de hiperparámetros.
- Desarrollo de una aplicación web [14] para la evaluación de los diferentes algoritmos de agrupamiento en los conjuntos de datos analizados en el trabajo. A su vez, se dispuso en línea el conjunto de datos modificado [13] para este trabajo, para futuras evaluaciones.

2. Experimentación

2.1. Conjunto de datos

En este trabajo se utilizan 4 conjuntos de datos de nubes de puntos que representan escenarios interiores y exteriores. En el caso de los interiores, los es-

cenarios representan a una mesa con cuatro sillas, una escalera con la información del suelo, y un piano con un banco y un florero. En el caso del escenario exterior, se presentan a dos aviones detenidos en un aeropuerto con la información del suelo. Estos conjuntos de datos fueron obtenidos delimitando espacialmente escenarios del conjunto de datos Point Cloud Sample [8].

En primer lugar, se obtuvo la información de las nubes de puntos de los conjuntos de datos, que se encuentra en formato .xyz. Para este trabajo se utilizaron solo las coordenadas cartesianas x , y , z . Luego, se graficó cada nube de puntos en 3 dimensiones, y se realizó una segmentación manual, a partir de restricciones espaciales. Cada segmento fue asignado a clases numéricas, como por ejemplo 0, 1, 2. En la Fig. 1a se observa el conjunto de datos mesa segmentada a partir de las clases 0, 1, 2, 3 y 4. Se representó cada clase con un color, y se mantuvo la paleta en todos los gráficos.

2.2. Agrupamiento

El agrupamiento se realizó a partir de la biblioteca sklearn [7] codificada en Python. Del abanico de algoritmos que ofrece la biblioteca, se evaluaron cuáles eran los más aptos, de acuerdo a las características del conjunto de datos y se seleccionaron los siguientes:

- **K-means** [11]: Agrupa puntos minimizando la suma de distancias entre cada objeto y el centroide del clúster. El parámetro principal es la cantidad de clústeres. Se utiliza para conjuntos de datos con gran cantidad de muestras, y una cantidad mediana de clústeres.
- **DBSCAN** [10]: Agrupa áreas de alta densidad separadas por áreas de baja densidad. Los parámetros principales son la cantidad mínima de puntos para formar un clúster (`min_samples`), y la distancia máxima para considerar un punto en una vecindad (`eps`). Se utiliza para conjunto de datos con gran cantidad de muestras, y una cantidad mediana de clústeres. Evalúa la distancia entre puntos más cercanos.
- **Mean Shift** [15]: Busca manchas en una densidad uniforme de muestras. Se basa en el centroide que funciona mediante la actualización de candidatos para que dichos centroides sean la media de los puntos dentro de una región determinada, llamada ancho de banda. Dichos candidatos luego son filtrados para eliminar los duplicados y formar el conjunto final de centroides.
- **BIRCH** [16]: Construye gradualmente un árbol de características de agrupamiento a través de un proceso de 2 etapas. En la primera etapa, analiza secuencialmente las muestras y construye un árbol de características de agrupamiento inicial. En la segunda etapa utiliza un algoritmo de agrupamiento arbitrario para agrupar a las hojas del árbol.

2.3. Selección de Hiperparámetros

El parámetro k de K-means se definió en cada escenario por la cantidad de clases determinadas en la separación manual.

En el caso de DBSCAN, se realizó un barrido de `min_samples` entre 2 y 20 con pasos unitarios, y de `eps` entre 0 y 2 con pasos de 0.01. En la totalidad de las combinaciones de parámetros, se calcularon la cantidad de clústeres generados, y se eligieron como candidatos a las combinaciones en donde los clústeres eran iguales a las clases reales de cada escenario. Luego, dentro de los candidatos se evaluó con el índice Rand [12], y se seleccionó la combinación de `min_samples` y `eps` que mayor índice obtuvo.

Para Mean Shift, se realizó un barrido del parámetro cuantil en la estimación del ancho de banda, de 0.1 a 1 con saltos de 0.01. Se seleccionaron aquellos valores de los cuantiles que generaron una cantidad de grupos igual al de las clases de cada nube de puntos. Luego se evaluaron cada candidato con el índice Rand y se determinó el cuantil que mayor índice obtuvo.

Para el algoritmo BIRCH, se determinó la cantidad de sub clústeres de acuerdo a la cantidad de clases, y se realizó un barrido de los umbrales que permiten detectar la cantidad de grupos coincidente a la cantidad de clases, y aquellos umbrales candidatos fueron evaluados con el índice Rand, para determinar el mejor umbral.

3. Evaluación

Para evaluar los resultados de las diferentes técnicas de agrupamiento estudiadas en el apartado anterior se utiliza el índice Jaccard [3] y el índice Rand [12]. El índice Rand (RI- Rand Index) es una medida de la similitud entre dos agrupaciones de datos y se define según la Ec. 1.

$$RI = \frac{a + b}{C_2^{n_{muestras}}} \quad (1)$$

Siendo C las clases reales y K el agrupamiento, se define a y b como:

- a , el número de pares de elementos que están en el mismo conjunto que C y en el mismo conjunto que K .
- b , el número de pares de elementos que están en diferente conjunto que C y en diferente conjunto que K

Donde $C_2^{n_{muestras}}$ es el número total de pares en el conjunto de datos.

El índice de Jaccard (JI- Jaccard Index) mide la similitud entre un conjunto finito de muestras, y se define como el tamaño de la intersección dividido el tamaño de la unión del conjunto de muestras. El mismo se describe en la Ec. 2

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Siendo A y B dos conjuntos de datos, $A \cap B$ la intersección de ambos conjuntos y $A \cup B$ la unión.

En la Tabla 1 se observan los parámetros seleccionados, y las métricas Rand y Jaccard obtenidas con K-Means, DBSCAN, Mean Shift y BIRCH para cada escenario, en negrita se aprecia el mejor método para cada escenario.

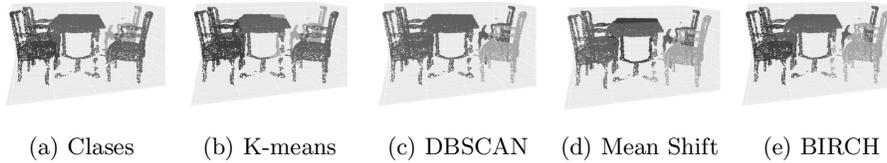


Figura 1: Conjunto de datos de mesa segmentado con 4 algoritmos

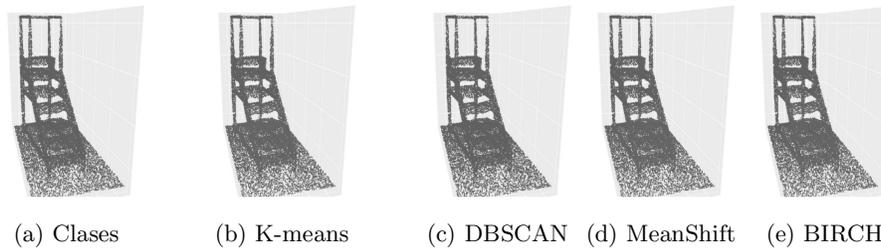


Figura 2: Conjunto de datos de escalera segmentado con 4 algoritmos

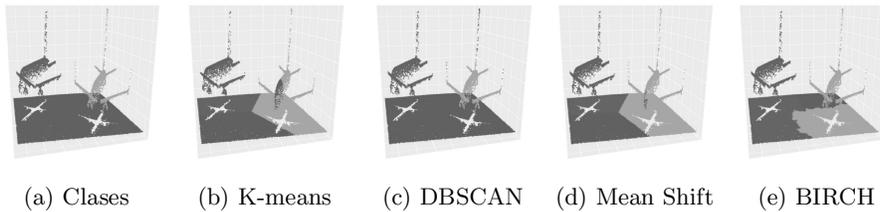


Figura 3: Conjunto de datos de aviones segmentado con 4 algoritmos

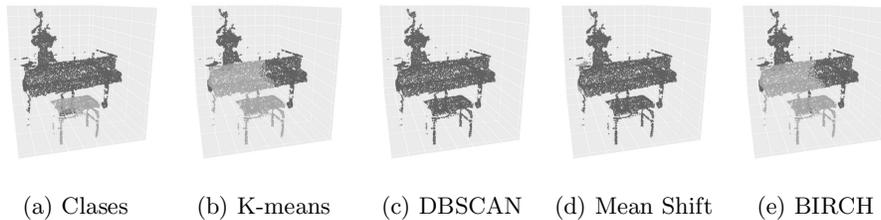


Figura 4: Conjunto de datos de piano segmentado con 4 algoritmos

4. Análisis de los resultados

4.1. Análisis visual

En el conjunto de datos denominado mesa, el agrupamiento con K-means, que se observa en la Fig. 1b, separa correctamente cada una de las sillas, pero

Cuadro 1: Resultados de métricas

Conjunto de datos	Método	Parámetros		RI	JI
aviones	K-means	K:3		0.421	0.213
	DBSCAN	eps:1.25	n_samples: 24	0.999	0.742
	Mean Shift	quantile: 0.18	n_samples: 500	0.426	0.202
	BIRCH	n_cluster:3	th:0.5	0.423	0.138
mesa	K-means	K:5		0.833	0.270
	DBSCAN	eps: 0.08	n_samples: 4	0.682	0.044
	Mean Shift	quantile: 0.06	n_samples: 500	0.834	0.135
	BIRCH	n_cluster:3	th:0.2	0.826	0.317
escalera	K-means	K:2		0.601	0.137
	DBSCAN	eps: 0.038	n_samples: 18	0.546	0.174
	Mean Shift	quantile: 0.35	n_samples: 500	0.598	0.150
	BIRCH	n_cluster:3	th:0.5	0.514	0.259
piano	K-means	K:3		0.666	0.399
	DBSCAN	eps: 0.24	n_samples: 26	0.766	0.408
	Mean Shift	quantile: 0.21	n_samples: 500	0.755	0.414
	Birch	n_cluster:3	threshold:0.5	0.514	0.260

no identifica a la tabla de la mesa como un grupo, sino como parte de los grupos que contiene a las sillas. Respecto a DB-SCAN, que se observa en la Fig. 1c, el algoritmo es capaz de determinar 3 clústeres bien definidos, 2 de las sillas y un tercer conjunto formado por la mesa y 2 sillas, esto se debe principalmente a que DB-SCAN utiliza la densidad de los datos para determinar los puntos, y las sillas encimadas entre sí, y cercanas a la mesa, no presentan un cambio de densidad detectable. En el caso de Mean Shift, que se observa en 1d y de BIRCH, que se observa en la Fig. 1e, se comportaron de forma similar a K-means.

El conjunto de datos escalera tiene la característica de estar integrado por un solo objeto, la escalera y el suelo. En la Fig. 2b se observa el agrupamiento con K-means, en la Fig. 2c DBSCAN. En la Fig. 2d se visualiza el resultado de Mean Shift y en la Fig. 2e el agrupamiento del algoritmo BIRCH. En ninguno caso se observaron resultados satisfactorios. Por lo que se recomienda extraer el suelo, antes de hacer el agrupamiento.

Respecto al conjunto de datos denominado aviones, tanto K-means (Fig. 3b), Mean Shift (Fig. 3d) y BIRCH (Fig. 3e) no pudieron diferenciar correctamente los aviones del suelo. En el caso de DBSCAN (Fig. 3c) se pudo diferenciar correctamente a los 2 aviones, como así también a los puntos que representa el suelo. Aunque con la particularidad de no poder agrupar a los puntos pertenecientes a la cola de los aviones, que se visualiza en color verde.

Respecto al conjunto de datos piano, que tiene la particularidad de presentarclusiones o datos incompletos, ya que no se tiene información sobre la tapa del piano. En la Fig. 4b se observa el agrupamiento K-means, donde el algoritmo separó de forma correcta a la maceta, pero identificó al piano como dos clases. El algoritmo BIRCH (Fig. 4e) se comportó de manera similar a K-means. El caso de DBSCAN (Fig. 4c), el algoritmo no pudo diferenciar al banco del piano. De

manera similar se comportó Mean Shift (Fig. 4d). En todos los casos evaluados se observó que es posible diferenciar a la planta del piano, pero no así al banco.

4.2. Análisis cuantitativo

En el caso del conjunto de datos mesa, el algoritmo Mean Shift y BIRCH son los que mayor índice Rand obtienen. Si bien el índice Rand de BIRCH está unas décimas por debajo de Mean Shift, este obtiene mayor puntuación en el índice de Jaccard, debido a que distinguió a las 4 sillas. Si bien el índice RAND con un valor cercano a 0,83 parece indicar que el agrupamiento fue correcto, al analizar el índice de Jaccard se saca la misma conclusión que en la inspección visual, ningún algoritmo fue capaz de distinguir a los puntos correspondientes con la mesa como un único grupo o clase.

En el conjunto de datos escalera, ninguno de los métodos pudo diferenciar correctamente a los puntos pertenecientes a la clase escalera, con los datos pertenecientes al suelo. Las pobres puntuaciones en las métricas reflejan lo mismo que se observó visualmente.

En el caso del conjunto de datos aviones, la evaluación de ambas métricas coinciden con la observación visual, DBSCAN es el método que mejor agrupamiento realizó, ya que pudo diferenciar perfectamente a los aviones del suelo y agrupó a cada avión en un único clúster. Si analizamos el resultado del índice Jaccard para este conjunto de datos, observamos que el valor 0,7419 dista de 1, esto es debido a los puntos correspondientes a la cola del avión que el algoritmo no pudo agrupar, tal como se comentó en el análisis visual.

Para el caso del conjunto piano, DBSCAN y Mean Shift obtuvieron la mejor puntuación en el índice Rand y Jaccard respectivamente. Aunque en ninguno de los casos, se pudo diferenciar al piano del banco.

5. Conclusiones

En escenarios donde existe información del suelo y los objetos están directamente sobre el suelo, como el caso del conjunto de datos escalera, no fue posible diferenciar correctamente los puntos correspondientes al suelo y al objeto. Para solucionar este inconveniente se puede realizar una extracción del suelo, utilizando por ejemplo el algoritmo RANSAC [2], generando un plano que represente al suelo y separando a los valores atípicos.

En cambio, en escenarios donde existe información del suelo, pero los objetos están principalmente por encima del mismo, como el caso del conjunto de datos aviones, el algoritmo DBSCAN pudo agrupar correctamente a los puntos del suelo. En este caso, los puntos correspondientes a los dos aviones, se encuentran separados por zonas de densidad muy diferentes, es por ello que el algoritmo pudo realizar una separación casi perfecta. Aunque, se observó que el algoritmo DBSCAN presenta dificultades a la hora de agrupar puntos correspondientes a objetos muy próximos, como el caso del conjunto de datos mesa. En estos casos, K-means presentó mejores rendimientos. Además, DBSCAN no agrupa

zonas de baja densidad, como el caso de las colas de los aviones o los puntos correspondientes a la zona inferior de los vehículos, si bien esto se nota en la observación visual, afecta de manera significativa a la métrica Jaccard.

Respecto a la selección de parámetros, en todos los casos, fue necesario contar con la información de la cantidad de grupos o clústeres presentes en cada escenario. Por lo tanto, el agrupamiento realizado en todos los casos se puede describir como semi supervisado. En el caso de DBSCAN, Mean Shift y BIRCH se logró obtener los parámetros a partir del número de clústeres e iteraciones, evaluando en cada caso el mejor conjunto de parámetros con el índice Rand.

Referencias

1. Chen, X., Zhang, H., Lu, H., Xiao, J., Qiu, Q., Li, Y.: Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue. SSRN 2017 - 15th IEEE International Symposium on Safety, Security and Rescue Robotics, Conference pp. 41–47 (2017)
2. Derpanis, K.G.: Overview of the RANSAC algorithm. *Image Rochester NY* **4**(1), 2–3 (2010)
3. Dodge, Y.: *The concise encyclopedia of statistics*. Springer Science & Business Media (2008)
4. Li, L., et al.: Time-of-flight camera—an introduction. Technical white paper (SLOA190B) (2014)
5. Milanés, V., Llorca, D.F., Vinagre, B.M., González, C., Sotelo, M.A.: Clavileño: Evolution of an autonomous car. In: 13th International IEEE Conference on Intelligent Transportation Systems. pp. 1129–1134 (2010)
6. Nguyen, A., Le, B.: 3d point cloud segmentation: A survey. In: 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM). pp. 225–230 (2013)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
8. Poux, F.: Point cloud sample dataset, https://drive.google.com/drive/folders/1Ih_Zz9a6UcbU1aA-puEB_is7DYvXrb4w
9. Saudabayev, A., Khassanov, Y., Shintemirov, A., Varol, H.A.: An intelligent object manipulation framework for industrial tasks. 2013 IEEE International Conference on Mechatronics and Automation, IEEE ICMA 2013 pp. 1708–1713 (2013)
10. Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X.: DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* **42**(3), 1–21 (2017)
11. Sculley, D.: Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web. pp. 1177–1178 (2010)
12. Steinley, D.: Properties of the hubert-arable adjusted rand index. *Psychological methods* **9**(3), 386 (2004)
13. Urbano Pintos, N.: Lpc clustering, https://github.com/nurbano/lpc_clustering
14. Urbano Pintos, N.: Lpc-clustering app, <https://lpc-clustering.streamlit.app>
15. Von Luxburg, U.: A tutorial on spectral clustering. *Statistics and computing* **17**(4), 395–416 (2007)
16. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. *ACM sigmod record* **25**(2), 103–114 (1996)