

Laboratory 7: BSDS500

Fabian Martínez
Universidad de los Andes
Bogota

fe.martinez10@uniandes.edu.co

Juan Felipe Pérez
Universidad de los Andes
Bogota

jf.perez10@uniandes.edu.co

Abstract

En el siguiente informe se presentan los procesos y resultados obtenidos al realizar la segmentación de imágenes utilizando algoritmos de kmeans y GMM en el espacio de color Lab en el database del BSDS500, y compararlos con los resultados obtenidos por un grupo de investigación de la Universidad de Berkeley en el que participó el profesor Pablo Arbeláez.

1. Introducción

Uno de los grandes problemas en los problemas relacionados a la visión en computador corresponde a la segmentación de una imagen, con el fin de representarla de una manera más significativa y más fácil para realizar un posterior análisis.

Este proceso de segmentación tiene como objetivo el dividir una imagen en varias partes que tengan características comunes, o, en lo posible, que pertenezcan a cierta clase de objetos de interés.

Una de las maneras más comunes de evaluar y de comparar los diferentes métodos de segmentación propuestos por los investigadores consiste en el uso de las bases de datos. Estas bases de datos contienen una gran cantidad de imágenes sobre las cuales se puede realizar tanto el entrenamiento, como la prueba y la validación del método utilizado. Para el presente laboratorio se utiliza una serie de imágenes de la base de datos BSDS500 de la universidad de Berkeley.

1.1. Objetivos

- Mejorar e implementar el código de segmentación realizado en el laboratorio pasado para segmentación por clustering.
- Comparar los resultados obtenidos en los métodos de clusterización optimizados del anterior laboratorio contra los resultados obtenidos por los investigadores de Berkeley.

- Utilizar una parte de la base de datos conocida como BSDS500 de la Universidad de Berkeley, utilizando las anotaciones en cada una de las imágenes para evaluar los resultados obtenidos.

2. Materiales y métodos

Para el siguiente laboratorio se realizan algunas mejoras a las funciones realizadas anteriormente, con el fin de tener unos mejores resultados en el proceso de segmentación de las imágenes.

2.1. Materiales

Los materiales utilizados para el siguiente laboratorio son:

- Matlab para el procesamiento de las imágenes
- Las imágenes de la base de datos BSDS500
- Funciones de Matlab desarrolladas por Pablo Arveláez para la evaluación de la segmentación

2.2. Base de datos

Las imágenes que se utilizaron en el laboratorio hacen parte de la base de datos de la Universidad de Berkeley. Estas imágenes son separadas en tres grupos distintos, uno correspondientes a entrenamiento, otro correspondiente a prueba y otro correspondiente a validación, los cuales tienen 200, 200 y 100 imágenes respectivamente.

Entre las propiedades de las imágenes mas sobresalientes es su tamaño, el cual es de 321×481 píxeles (sin embargo el sentido de las imágenes puede estar invertido, es decir tienen un tamaño de 481×321 píxeles). Por otra parte, esta base de datos tiene las anotaciones correspondientes a cada imagen, realizadas por humanos.

2.3. Funciones utilizadas

Teniendo en cuenta los resultados del laboratorio anterior, se decide utilizar como los métodos de segmentación

preferidos k-means y GMM, debido a la facilidad de implementación y a las segmentaciones apropiadas (las características específicas se detallan a continuación). A pesar de que pueden obtenerse buenos resultados con métodos como la segmentación jerárquica, debido al número y al tamaño de las imágenes este proceso se torna complicado computacionalmente.

2.4. K-Means

Este método es un método de cuantización de vectores, el cual tiene como ventaja su simplicidad. Debido a que este método trabaja con vectores es necesario realizar una transformación de la imagen en vectores. Esta transformación se realiza por medio de una concatenación de la información presentada en los diferentes canales del espacio de color así como de la información espacial (opcional).

El algoritmo inicia con un grupo inicial de centroides al azar, asigna cada píxel (vector con las características) al centroide mas cercano e itera los centroides hasta que no cambie la solución obtenida. Para realizar este procedimiento Matlab cuenta con la función kmeans, la cual recibe una matriz $m \times 1$ y k clusters en los que se quieren agrupar. Una vez obtenida la matriz solución, se reconstruye de nuevo la imagen con las etiquetas correspondientes a cada cluster.

2.4.1 Mezcla de Gaussianas (GMM)

Para este método se tiene nuevamente la entrada de la matriz, similar a la de k-means. Sin embargo, se utiliza la función fitgmdist de MATLAB, la cual es una función que busca encontrar una mezcla de distribuciones gaussianas con las cuales pueda explicarse la distribución que presentan los datos. Debido a que los resultados son unas distribuciones de probabilidad, es necesario realizar una clusterización fuerte para separar los datos en segmentos.

2.4.2 Mejora

Una vez obtenidos los resultados de la segmentación, se observan algunas zonas inestables en las imágenes, por lo que se realizó una modificación al código usado anteriormente para que el resultado terminara con regiones más uniformes. Al resultado de la clusterización se le aplicó un filtro mediano de tamaño definido por el usuario, de forma que ciertas regiones con muchas etiquetas en ellas resultaran con solamente una etiqueta (la de mayor frecuencia en una ventana).

El único parámetro adicional es entonces el tamaño de la ventana con la que se hará el filtro mediano. Este deberá depender de que tanta variación se tenga en regiones de interés de la imagen. Si el resultado de la clusterización es que en una misma región hay mas de un label, y si se quiere segmentar en dicha región un cuerpo sin muchos detalles, el tamaño de ventana debe ser grande, pues de esta manera

se reduce la variabilidad de los labels en la región. Por el contrario, si lo que desea segmentar tiene varios detalles, lo mejor será usar tamaños de ventana pequeños, pues de lo contrario se perderían dichos detalles.

3. Resultados

Algunos de los resultados obtenidos fueron:

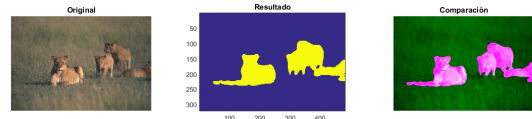


Fig. 1. Resultados en la segmentación por kmeans, con k = 2

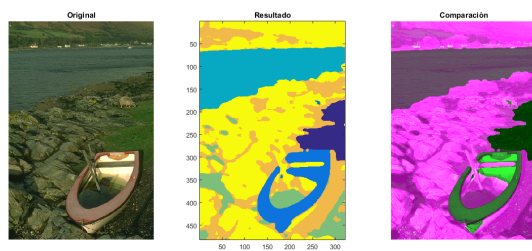


Fig. 2. Resultados en la segmentación por kmeans, con k = 4

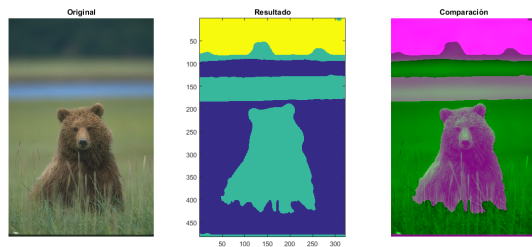


Fig. 3. Resultados en la segmentación por GMM, con k = 3

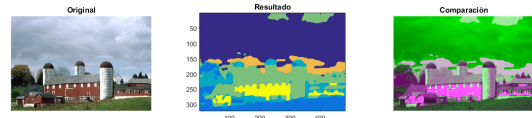


Fig. 4. Resultados en la segmentación por GMM, con k = 6

El método de segmentación que resultó mejor fue el de GMM, pues las regiones segmentadas correspondían bastante mejor a las anotaciones que fueron realizadas por medio de kmeans. En nuestro caso, variamos los k de manera que se pudiera determinar con cuales se obtenía el mejor resultado, de manera que efectivamente del k dependía que tan buena o mala podía ser la segmentación, ya que con un k muy bajo, no realizaba buena segmentación debido a que surgía una muy poca cantidad de bordes, mientras que con

un k demasiado alto, las imágenes se segmentaban en demasiadas regiones, afectando de esta manera la precisión y la cobertura del método.

Lamentablemente no pudimos superar a Pablo, y esto puede ser debido en mayor parte a que el algoritmo de Pablo involucra mucha más información que el nuestro para segmentar la imagen, como por ejemplo información de los bordes. Sin embargo, en comparación con el lab anterior, el método de GMM sigue siendo el que produce mejores resultados frente con los demás métodos de segmentación por clusterización. Esto es debido a que, por lo menos en comparación con kmeans, el método de GMM es mucho más robusto al permitir asignar a cada píxel una probabilidad de pertenecer a un cluster y determinar estos últimos a partir de distribuciones gaussianas (normales) de probabilidad.

Aún así, una limitación del algoritmo que desarrollamos es precisamente que, a diferencia del método de Pablo, nosotros no usamos información de bordes u orientaciones en la imagen, por lo que una posible manera de mejorar el método sería teniendo en cuenta mucha más información que ayude a separar regiones de una imagen, y no solamente por el color.

3.1. Curvas PR

A continuación se presentan las curvas de PR obtenidas para cada uno de los métodos, las cuales se obtuvieron utilizando las funciones de evaluación.

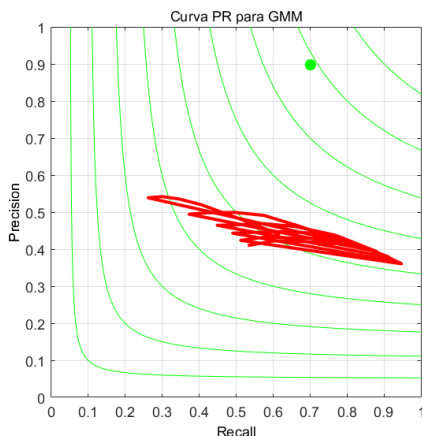


Fig. 5. Curva de PR con el método de K-means

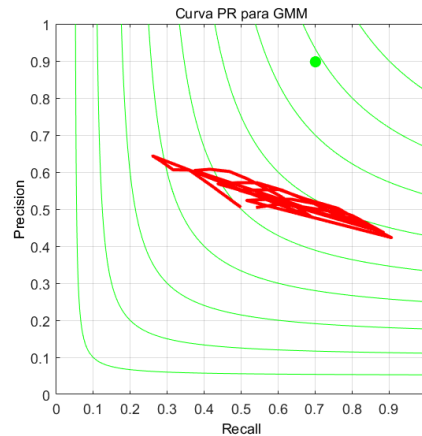


Fig. 6. Curva de PR con el método de GMM

Una curva de Precision-Recall, es una medida que tan bueno fue el desempeño de una clasificación. En general, en una curva de PR, tiene dos ejes, uno en el cual se evalúa la precisión, que es el número de verdaderos positivos sobre el número de verdaderos positivos más el número de falsos positivos, y otro en el que se evalúa la cobertura, que es el número de verdaderos positivos más el número de falsos negativos. El objetivo de una curva es tener una mayor área bajo la curva, ya que esto indica que se alcanza una mayor precisión a medida que se tiene una mayor cobertura, siendo el resultado ideal aquel que tiene un valor de 1 tanto en el eje P como en el eje R.

Como se puede observar en las gráficas anteriores, se tiene en general un mayor precisión y cobertura cuando la segmentación se realiza por el método de GMM que con el método de k-means.

3.2. Comentarios adicionales

Finalmente, en la siguiente imagen se presentan los tiempos que se obtuvieron al realizar la evaluación de las imágenes para uno de los métodos.

```

New to MATLAB? See resources for Getting Started.
Processing image 80085 (191/200)...
Processing image 80090 (192/200)...
Processing image 8068 (193/200)...
Processing image 81066 (194/200)...
Processing image 81090 (195/200)...
Processing image 81095 (196/200)...
Processing image 87015 (197/200)...
Processing image 92014 (198/200)...
Processing image 94095 (199/200)...
Processing image 97010 (200/200)...
Warning: File not found or permission denied
> In allBench_fast (line 50)
    In test_benchs_fast (line 12)
Warning: File not found or permission denied
> In allBench_fast (line 53)
    In test_benchs_fast (line 12)
Elapsed time is 274.381898 seconds.
fx >>

```

Fig. 7. Tiempo utilizado para la evaluación

Como se puede apreciar el tiempo de ejecución de los métodos es bastante alto, incluso requiere de una máquina con buenas características de Ram y procesador. Por otro lado, la segmentación de imágenes es un proceso mucho mas complejo que requiere aun mejores especificaciones de memoria para poderlo realizar.

4. Conclusiones

1. Variar el número de clusters en los métodos de segmentación utilizados permiten controlar el número de bordes que se obtienen a la hora de realizar la segmentación, en los que a mayor número de clusters la cantidad de bordes y clases de objetos detectados es mayor.
2. Dependiendo del tipo de método utilizado para la segmentación se puede obtener resultados diferentes, lo que afecta la precisión y la cobertura a la hora de segmentar las imágenes.
3. Utilizar un procesamiento de las imágenes antes y despues de realizar el proceso de segmentación permite tener mejores resultados, como lo es el caso del uso de filtros como el utilizado en el presente laboratorio.