Trabajo práctico 0: Algoritmo de Maximización de la Esperanza

Ph. D. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Computación
PAttern Recongition and MAchine Learning Group (PARMA-Group)

4 de septiembre de 2023

Fecha de entrega: Lunes 25 de Setiembre.
Entrega: Un archivo .zip con el código fuente LaTeX o Lyx, el pdf, y un jupyter en Pytorch, debidamente documentado, con una función definida por ejercicio. A través del TEC-digital.
Modo de trabajo: Grupos de 3 personas.

Resumen

En el presente trabajo práctico introduce al algoritmo de maximización de esperanza, y su aplicación para la segmentacion de imagenes.

1. (100 puntos) Algoritmo de Maximización de la Esperanza con datos artificiales

A continuación, implemente el algoritmo de maximización de la esperanza (descrito en el material del curso), usando la definición y descripción de las siguientes funciones como base:

1. **(15 puntos)** Implemente la función *generate_data* la cual reciba la cantidad de observaciones unidimensionales total a generar N, y los parámetros correspondientes a $K = 2$ funciones de densidad Gaussianas. Genere los datos siguiendo tales distribuciones, y retorne tal matriz de datos $X \in \mathbb{R}^{N \times 1}$.

   a) Grafique los datos usando un *scatter plot* junto con las gráficas de los histogramas de los datos y las funciones de densidad de probabilidad Gaussianas usando los parámetros para inicializar los datos, en la misma figura (gráfico).

2. **(5 puntos)** Implemente la función *init_random_parameters* la cual genere una matriz de $P \in \mathbb{R}^{K\times 2}$ dimensiones, con los parámetros de las funciones de densidad Gaussiana generados completamente al azar.

    a) Muestre un pantallazo donde verifique su funcionamiento correcto con los comentarios asociados.

3. **(20 puntos)** Implemente la función *calculate_likelihood_gaussian_observation(x_n, mu_k, sigma_k)* la cual calcule la verosimilitud de una observación específica x_n, para una función de densidad Gaussiana con parámetros μ_k y σ_k. Realice la correción pertinente al calculo de la función de verosimilitud para evitar el problema de *under flow.*

    a) Diseñe y ejecute una prueba unitaria donde verifique su funcionamiento correcto con los comentarios asociados.

4. **(10 puntos)** Implemente la función *calculate_membership_dataset(X_dataset, Parameters_matrix)*, la cual, usando la matriz de parámetros P y la función anteriormente implementada *calculate_likelihood_gaussian_observation*, defina por cada observación $x_n \in X$ la pertenencia o membresía a cada cluster $k = 1, \dots K$, en una matriz binaria $M \in \mathbb{R}^{N\times K}$. Retorne tal matriz de membresía M.

    a) Diseñe y ejecute una prueba unitaria donde verifique su funcionamiento correcto con los comentarios asociados.

5. **(30 puntos)** Implemente la función *recalculate_parameters(X_dataset, Membership_data)*, la cual recalcule los parámetros de las funciones de densidad Gaussianas representandas en la matriz P, de acuerdo a lo representado en la matriz de membresía M.

    a) Use las funciones *mean* y *std* de pytorch para ello. Intente prescindir al máximo de estructuras de repetición tipo *for*.

6. **(10 puntos)** Ejecute 5 corridas diferentes del algoritmo, donde por cada una documente los parámetros a los que se arribó.

    a) Grafique las funciones de densidad de probabilidad a las que convergió el algoritmo. Puede graficar también las funciones de densidad obtenidas en 2 o 3 pasos intermedios. Presente una tabla de gráficas donde en cada entrada se identifique el número de iteración y los parámetros iniciales.

    b) Comente los resultados.

7. **(10 puntos)** Proponga una mejor heurística para inicializar los parámetros del modelo aleatoriamente.

    a) Compruebe la mejora obtenida con el método propuesto, corriendo las pruebas del punto anterior.

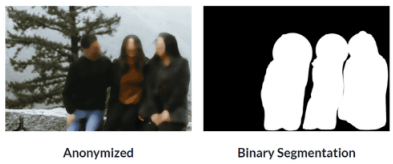

Figura 1: Segmentación binaria.

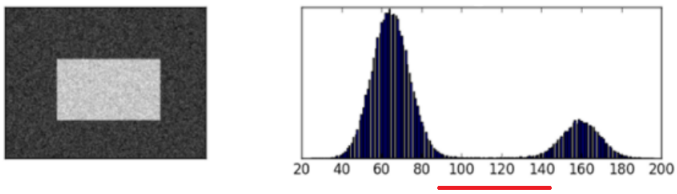

Figura 2: Histograma de una imagen con un fondo y un objeto de interés.

2. (30 puntos extra) Umbralización con el algoritmo de Maximización de la Esperenza

En procesamiento de imágenes, un problema muy frecuente es la segmentación binaria de una imagen, para detectar objetos de interés respecto a un fondo. Este problema se pueden enfocar como el de clasificar cada valor de pixel z_i de una imagen $U \in \mathbb{R}^{a \times b}$ de a filas y b en dos categorias: fondo ($t_i = 0$) y objeto de interés ($t_i = 1$). La Figura 1 muestra un ejemplo de una segmentación binaria.

Un enfoque para clasificar los pixeles en dos clases, es crear una función $\tilde{t}_i = f(z_i)$, la cual tome la decisión de si $\tilde{t}_i = 0$ o si $\tilde{t}_i = 1$, a partir de un umbral óptimo τ:

$$f(p_i) = \begin{cases} 1 & \text{si } z_i > \tau \\ 0 & si z_i \leq \tau \end{cases}$$

Para calcular ese umbral óptimo, se puede usar el histograma del conjunto de pixeles en la imagen U. Observe por ejemplo la imagen de la Figura 2. El histograma h_U de tal imagen U es una aproximación a su función de densidad $\tilde{p}_U$. A partir de ese histograma, se puede observar que una función mixta Gaussiana, definida como la combinación lineal de dos funciones Gaussianas, puede ser un modelo con un buen ajuste a tal histograma h_U:

$$p(z \mid \mu_1, \mu_2, \sigma_1, \sigma_2) = P_1 \mathcal{N}(z \mid \mu_1, \sigma_1) + P_2 \mathcal{N}(z \mid \mu_2, \sigma_2)$$

Donde $P_1 + P_2 = 1$ son los pesos relativos de cada «campana Gaussiana».

Con un algoritmo como por ejemplo la Maximización de la Esperanza, es posible estimar los parámetros $\mu_1, \mu_2, \sigma_1, \sigma_2$. Por ello:

- **(10 puntos)** Para la imagen provista *cuadro1_005.bmp*, calcule su histograma y usando el algoritmo de Maximización de la Esperanza, estime los parámetros $\mu_1, \mu_2, \sigma_1, \sigma_2$. Hágalo para 3 corridas, y para cada una de ellas muestre el histograma de la imagen U con la función de densidad estimada $p(z \mid \mu_1, \mu_2, \sigma_1, \sigma_2)$ superpuesta (en la misma gráfica).

Note ahora que una vez estimado el modelo con mejor ajuste $p(z \mid \mu_1, \mu_2, \sigma_1, \sigma_2)$, es posible estimar el umbral óptimo τ, tomando en cuenta el «valle» entre ambas Gaussianas.

- **(20 puntos)** Diseñe e implemente un algoritmo sencillo, el cual, tomando en cuenta al modelo $p(z \mid \mu_1, \mu_2, \sigma_1, \sigma_2)$, estime el o los umbrales óptimos τ. Realice y documente las pruebas asociadas y aplique el umbral o los umbrales óptimos obtenidos en la imagen de prueba. Muestre y comente los resultados.