Trabajo práctico 1: Bayes Ingenuo

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

25 de abril de 2024

Fecha de entrega: Domingo 12 de Mayo
Entrega: Un archivo .zip con el código fuente LaTeX o Lyx, el pdf, y un notebook Jupyter, 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 se introduce la implementación de redes bayesianas. El trabajo practico consta de 120 puntos, donde 20 son extra.

1. (40 puntos) Implementación de la clasificación multiclase de imágenes con Bayes ingenuo usando histogramas

1. Para el presente ejercicio, se implementará la clasificación de imagenes naturales con $K = 10$ clases. La Figura 1 muestra algunas observaciones del conjunto de datos. El código provisto lee las imágenes del conjunto de datos, y los transforma a matrices binarias de $32 \times 32 = 1024$ pixeles. El objetivo de su equipo de desarrollo es utilizar el teorema de Bayes para construir un modelo conocido como Bayes ingenuo, el cual permita estimar la clase a la que pertenece una nueva observación.

2. En el material del curso, se discute el algoritmo de Bayes ingenuo, el cual tiene por objetivo estimar la **probabilidad posterior** de que una observación (imagen en este caso) $\overrightarrow{m} \in \mathbb{N}^D$,donde en este caso $D = 1024$, pertenezca a una clase k como:

   $$p\left(t = k|\overrightarrow{m}\right)$$

   Para aproximarla, se utiliza el teorema de Bayes, el cual luego de desarrollar y simplificar la expresion de tal probabilidad posterior, se concluye

Figura 1: Imagenes de CIFAR-10.

que esta es proporcional a la multiplicación de la probabilidad a priori de $p\left(t=k\right)$ y la verosimilitud de un pixel $p\left(m_i|t=k\right)$:

$$p\left(t=k|\overrightarrow{m}\right) \propto \prod_{i=0}^{D} p\left(m_i|t=k\right) p\left(t=k\right).$$

Por ejemplo, la verosimilitud del pixel i negro (0), $p\left(m_i|t=k\right)$ se implementa como la probablididad de que $p\left(m_i=0|t=k\right)$ en caso de que ese pixel i de la observación a evaluar en el modelo sea negro (0). Es necesario calcular la verosimilitud de cada intensidad de pixel $p\left(m_i=0|t=k\right)$,$p\left(m_i=1|t=k\right)$,$p\left(m_i=2|t=k\right)$ hasta $p\left(m_i=255|t=k\right)$ ($Z=255$).

a) **(10 puntos)** Implemente el cálculo de las probabilidades a priori $p\left(t\right)$ para las $K=10$ clases en el conjunto de datos de entrenamiento en la función *calcular_probabilidad_priori*. Realice tal calculo dentro de la funcion *train_model.*

1) Diseñe y muestre el resultado de una o más pruebas unitarias de tal función objetivo, entradas, salidas esperadas y resultados).

b) Para evaluar la verosimilitud $p\left(m_d|t\right)$, es necesario estimar las densidades $p\left(m_d=0|t\right)$,... $p\left(m_d=255|t\right)$, para todos los pixeles $d=1,\ldots 1024$ pixeles. Para ello, su equipo considerará las siguientes dos variantes:

1) **Enfoque basado en histogramas**: Siguiendo la simplificación sugerida por su colega Josef, usando las imágenes binarizadas, se sugiere lo siguiente. Cree un **tensor de dimensiones *dataset_densities*** $D \times Z \times K$, el cual represente las densidades de

cada pixel (1024 en total) para cada una de las intensidades de pixel posibles ($Z = 255$ maximo) para cada una de las clases (K clases en total), por lo que entonces cada columna corresponde a la densidad de cada pixel. **Para realizar este calculo solo se le permite usar un ciclo *for*, con una iteracion por clase k, como maximo**. Estime los valores de tal matriz usando el conjunto de datos de entrenamiento.

a' **(20 puntos)** Implemente los dos puntos anteriores en la función *train_model_histogram* y retorne *dataset_densities*, junto con el arreglo de probabilidades a priori para todas las clases.

b' Diseñe y muestre el resultado de dos o más pruebas unitarias de tal función (objetivo, entradas, salidas esperadas y resultados).

c' Grafique los histogramas de los primeros 5 pixeles y las primeras 5 intensidades de pixeles para la clase 1 y 2. Siguen alguna distribucion conocida?

d' **(5 puntos)** Implemente la función *test_model_histogram(input_torch, dataset_densities, num_classes = 10)* la cual realice la estimación de a cual clase pertenece una observación contenida en el vector *input_torch*, para un modelo representado en *dataset_densities (*obtenido en el paso anterior*).* Para ello, el enfoque de Bayes ingenuo estima la función de densidad posterior como sigue:

$$p\left(t = k|\overrightarrow{m}\right) \propto \prod_{d=0}^{D} p\left(m_d|t = k\right) p\left(t = k\right).$$

La clase estimada a la que pertenece la observación $\overrightarrow{m}$ corresponde entonces a la clase k con mayor probabilidad posterior $p\left(t = k|\overrightarrow{m}\right)$.

e' Diseñe y muestre el resultado de una o más pruebas unitarias de tal función, siguiendo las pautas del diseño anteriores. Explique el diseño de la misma.

f' **(5 puntos)** Implemente la función *test_model_batch_histogram(test_set, labels, dataset_densities, p_t_tensor)* la cual calcule y retorne la tasa de aciertos para un conjunto de observaciones, basado en la función anteriormente implementada *test_model.*

g' Diseñe y muestre los resultados de al menos 2 pruebas unitarias para validar su correcto funcionamiento. Detalle el diseño y documente los resultados obtenidos de las dos pruebas unitarias.

1.1. (30 puntos) Prueba del modelo

1. **(10 puntos)** Entrene el modelo propuesto, con el conjunto de observaciones contenido en la carpeta *train*, y reporte la tasa de aciertos al utilizar la función anteriormente implementada*test_model_batch_histogram*. Verifique y comente los resultados. Es posible que observe valores nulos en el resultado de evaluar la función posterior a través de la función *test_model* la cual implementa la ecuación:

$$p\left(t=k|\overrightarrow{m}\right) \propto \prod_{d=0}^{D} p\left(m_d|t=k\right) p\left(t=k\right).$$

   Si observa valores de 0 o nulos en la evaluación de la función, argumente el porqué puede deberse este comportamiento. ¿Cómo se puede corregir el problema detectado, según las herramientas matemáticas estudiadas en clase? Implemente tal enfoque y compruebe los resultados.

2. **(5 puntos)** Entrene el modelo usando todos los datos de train, pero ahora pruebelo con los datos en la carpeta de test, reporte los resultados y comentelos.

3. **(15 puntos)** Particione los datos de forma aleatoria con 70 % de las observaciones para entrenamiento y 30 % para prueba (a partir de la carpeta *train*). Calcule la tasa de aciertos para 10 corridas (**idealmente 30**), cada una con una partición de entrenamiento y otra de prueba distintas. Reporte los resultados de las corridas en una tabla, además de la media y desviación estándar de la tasa de aciertos para las 10 corridas. Para realizar las particiones puede usar la libreria *sklearn.*

2. (30 puntos) Implementación de la clasificación multiclase de imágenes con Bayes ingenuo usando un modelo Gaussiano

1. **Enfoque basado en un modelo Gaussiano**: Su colega Samir cuestiona el usar todos los valores de intensidad y calcular los histogramas como aproximacion de las densidades, puesto que según su punto de vista, puede sobre-ajusterse a los datos. Como alternativa, Samir propone ajustar un modelo Gaussiano para cada pixel de la imagen d, dada cada categoría k posible. Es por ello que entonces cada función de densidad condicional:

$$p\left(m_d|t=k\right) = \frac{1}{\sqrt{2\pi\sigma_{d,k}^2}} e^{\frac{-1}{2}\left(\frac{m_d-\mu_{d,k}}{\sigma_{d,k}}\right)^2}$$

   consistirá en un un modelo Gaussiano con parámetros $\mu_{d,k}$ y $\sigma_{d,k}$ para un pixel y clase especificos. Ello hará posible, con solamente conocer esos

parámetros, estimar la verosimilitud de una intensidad de pixel $m_d \in [0 - 255]$, usando el modelo Gaussiano.

a) **(20 puntos)** Implemente función *train_model_gaussian* la cual tome las entradas necesarias para retornar las matrices *Mu_d_k* y *Sigma_d_k,* las cuales corresponden a $\mathcal{M}^{D\times K}$ y $\Sigma^{D\times K}$. Al ajustar estos conjuntos de parámetros, ya es posible entonces estimar la verosimilitud definida anteriormente.

1) Grafique los histogramas y los modelos Gaussianos de los primeros 5 pixeles y las primeras 5 intensidades de pixeles para la clase 1 y 2. Comente los resultados.

2) Diseñe y muestre los resultados de al menos 2 pruebas unitarias para validar su correcto funcionamiento. Detalle el diseño y documente los resultados obtenidos de las dos pruebas unitarias.

b) **(5 puntos)** Implemente la función test *test_model_gaussian(input_torch, mu_d_k, sigma_d_k)* la cual realice la estimación de a cual clase pertenece una observación contenida en el vector *input_torch,* para un modelo representado por los arreglos recibidos $\mathcal{M}^{D\times K}$ y $\Sigma^{D\times K}$. Recuerde que al momento de evaluación, se implementa el teorema de Bayes para estimar:

$$p\left(t=k|\overrightarrow{m}\right) \propto \prod_{i=0}^{D} p\left(m_d|t=k\right) p\left(t=k\right).$$

Ello similar al enfoque anterior, donde lo que varía es la estimación de $p\left(m_d|t=k\right)$.

1) Diseñe y muestre los resultados de al menos 2 pruebas unitarias para validar su correcto funcionamiento. Detalle el diseño y documente los resultados obtenidos de las dos pruebas unitarias.

c) **(5 puntos)** Implemente la función *test_model_batch_gaussian(test_set,mu_k,sigma_k)* la cual calcule y retorne la tasa de aciertos para un conjunto de observaciones, basado en la función anteriormente implementada *test_model_gaussian.*

1) Diseñe y muestre los resultados de al menos 2 pruebas unitarias para validar su correcto funcionamiento. Detalle el diseño y documente los resultados obtenidos de las dos pruebas unitarias.

2.1. (20 puntos) Prueba del modelo

1. **(5 puntos)** Entrene el modelo propuesto, con el conjunto de observaciones contenido en la carpeta *train*, y reporte la tasa de aciertos al utilizar la función anteriormente implementada *test_model_batch_gaussian*. Verifique y comente los resultados. Es posible que observe valores nulos en el resultado de evaluar la función posterior a través de la función *test_model_gaussian* la cual implementa la ecuación:

$$p\left(t=k|\overrightarrow{m}\right) \propto \prod_{d=0}^{D} p\left(m_d|t=k\right) p\left(t=k\right).$$

   Si observa valores de 0 o nulos en la evaluación de la función, argumente el porqué puede deberse este comportamiento. ¿Cómo se puede corregir el problema detectado, según las herramientas matemáticas estudiadas en clase? Implemente tal enfoque y compruebe los resultados.

2. **(5 puntos)** Entrene el modelo usando todos los datos de train, pero ahora pruebelo con los datos en la carpeta de test, reporte los resultados y comentelos.

3. **(5 puntos)** Particione los datos de forma aleatoria con 70 % de las observaciones para entrenamiento y 30 % para prueba (a partir de la carpeta *train*). Calcule la tasa de aciertos para 10 corridas (**idealmente 30**), cada una con una partición de entrenamiento y otra de prueba distintas. Reporte los resultados de las corridas en una tabla, además de la media y desviación estándar de la tasa de aciertos para las 10 corridas. Para realizar las particiones puede usar la libreria *sklearn.*

3. (20 puntos extra) Implementación de la clasificación multi-clase de imágenes con Bayes ingenuo usando un modelo KDE

Realice los puntos de la seccion anterior usando un modelo KDE. Use un kernel Gaussiano y pruebe al menos 2 valores diferentes de ancho.