Práctica 2 - Aprendizaje Automático
Complejidad de H y Modelos Lineales

Límite de entrega: **1 de Mayo de 2022 a las 23:59 (PRADO)**

Valoración máxima: 12 puntos (+1.5 puntos de Bonus)

Materiales a entregar: un fichero Python (.py o .ipynb) y un informe describiendo y analizando el trabajo desarrollado, los resultados obtenidos y las conclusiones extraídas. En caso de que se opte por entregar un Colab Notebook, el informe debe estar integrado en el mismo cuaderno (intercalando texto, código y resultados).

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Es obligatorio presentar un informe con las valoraciones y decisiones adoptadas en el desarrollo de cada uno de los apartados. En dicho informe se incluirán los gráficos generados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. Sin este informe se considera que el trabajo NO ha sido presentado.

El incumplimiento de las normas que se listan a continuación puede implicar la pérdida de 2 puntos por cada norma incumplida:

- Cada ejercicio/apartado de la práctica debe quedar perfectamente identificado en el material entregado (código y memoria).
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado.
- El codigo NO puede escribir nada a disco.
- El path que se use en la lectura de cualquier fichero auxiliar de datos debe ser siempre "datos/nombre_fichero". Es decir, se espera que el código lea de un directorio llamado "datos", situado dentro del directorio donde se desarrolla y ejecuta la práctica.
- Un código es apto para ser corregido si se puede ejecutar de principio a fin sin errores.
- No es válido usar opciones en las entradas. Para ello, se deben fijar al comienzo los valores por defecto que se consideren óptimos.
- El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
- En caso de que haya más de un fichero (por ejemplo, *.py y *.pdf), estos se entregarán juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- Se entrega solamente el código fuente, y no los datos empleados.

1. EJERCICIO SOBRE LA COMPLEJIDAD DE H Y EL RUIDO (5 puntos)

En este ejercicio debemos aprender la dificultad que introduce la aparición de ruido en las etiquetas a la hora de elegir la clase de funciones más adecuada. Haremos uso de tres funciones incluidas en el fichero *template_trabajo2.py*:

- $simula\_unif(N, dim, rango)$, que calcula una lista de N vectores de dimensión dim. Cada vector contiene dim números aleatorios uniformes en el intervalo $rango$.
- $simula\_gauss(N, dim, sigma)$, que calcula una lista de longitud N de vectores de dimensión dim, donde cada posición del vector contiene un número aleatorio extraído de una distribucción Gaussiana de media 0 y varianza dada, para cada dimension, por la posición del vector $sigma$.
- $simula\_recta(intervalo)$, que simula de forma aleatoria los parámetros, $v = (a, b)$ de una recta, $y = ax + b$, que corta al cuadrado $[-50, 50] \times [-50, 50]$.

1. (1 punto) Dibujar gráficas con las nubes de puntos simuladas con las siguientes condiciones:
   a) Considere $N = 50$, $dim = 2$, $rango = [-50, 50]$ con $simula\_unif(N, dim, rango)$.
   b) Considere $N = 50$, $dim = 2$ y $sigma = [5, 7]$ con $simula\_gauss(N, dim, sigma)$.
2. Vamos a valorar la influencia del ruido en la selección de la complejidad de la clase de funciones. Con ayuda de la función $simula\_unif(100, 2, [-50, 50])$ generamos una muestra de puntos 2D a los que vamos añadir una etiqueta usando el signo de la función $f(x, y) = y - ax - b$, es decir el signo de la distancia de cada punto a la recta simulada con $simula\_recta()$.
   a) (1 punto) Dibujar un gráfico 2D donde los puntos muestren (use colores) el resultado de su etiqueta. Dibuje también la recta usada para etiquetar. Observe que todos los puntos están bien clasificados respecto de la recta.
   b) (0.5 puntos) Modifique de forma aleatoria un 10 % de las etiquetas positivas y otro 10 % de las negativas y guarde los puntos con sus nuevas etiquetas. Dibuje de nuevo la gráfica anterior. Ahora habrá puntos mal clasificados respecto de la recta.
   c) (2.5 puntos) Supongamos ahora que las siguientes funciones definen la frontera de clasificación de los puntos de la muestra en lugar de una recta
      - $f(x, y) = (x - 10)^2 + (y - 20)^2 - 400$
      - $f(x, y) = 0{,}5(x + 10)^2 + (y - 20)^2 - 400$
      - $f(x, y) = 0{,}5(x - 10)^2 - (y + 20)^2 - 400$
      - $f(x, y) = y - 20x^2 - 5x + 3$

      Visualizar el etiquetado generado en el apartado 2b junto con la gráfica de cada una de las funciones. Comparar las regiones positivas y negativas de estas nuevas funciones con las obtenidas en el caso de la recta. Argumente si estas funciones más complejas son mejores clasificadores que la función lineal. Observe las gráficas y diga qué consecuencias extrae sobre la influencia de la modificación de etiquetas en el proceso de aprendizaje. Explique el razonamiento.

2. MODELOS LINEALES (7 puntos)

1. (3 puntos) **Algoritmo Perceptrón (PLA)**.
   Implementar la función $ajusta\_PLA(datos, label, max\_iter, vini)$ que calcula el hiperplano solución a un problema de clasificación binaria usando el algoritmo PLA. La entrada *datos* es una matriz donde cada item con su etiqueta está representado por una fila de la matriz, *label* el vector de etiquetas (cada etiqueta es un valor +1 o -1), *max_iter* es el número máximo de iteraciones permitidas y *vini* el valor inicial del vector. La función devuelve los coeficientes del hiperplano.

   a) Ejecutar el algoritmo PLA con los datos empleados en el apartado 2a del ejercicio 1. Inicializar el algoritmo con: i) el vector cero y, ii) con vectores de números aleatorios en [0, 1] (10 veces). Anotar el número medio de iteraciones necesarias en ambos para converger. Se deben mostrar en una tabla cada uno de los pesos iniciales empleados, los finales (obtenidos tras el proceso de entrenamiento), y el porcentaje de error de clasificación. Valorar el resultado relacionando el punto de inicio con el número de iteraciones.

   b) Hacer lo mismo usando los datos del apartado 2b del ejercicio 1. ¿Observa algún comportamiento diferente? En caso afirmativo diga cuál y las razones para que ello ocurra.

2. (4 puntos) **Regresión Logística (RL)**.
   En este ejercicio emplearemos nuestra propia función objetivo f y un conjunto de datos $\mathcal{D}$ para ver cómo funciona regresión logística. Consideraremos $d = 2$ para que los datos sean fácilmente visualizables, y emplearemos $\mathcal{X} = [0, 2] \times [0, 2]$ con probabilidad uniforme de elegir cada $x \in \mathcal{X}$. Elegir una línea en el plano que pase por $\mathcal{X}$ como la frontera que separa la región en donde y toma valores +1 y -1. Para ello, seleccionar dos puntos aleatorios de $\mathcal{X}$ y calcular la línea que pasa por ambos.

   Impleméntese RL con Gradiente Descendente Estocástico (SGD) del siguiente modo:

   - Inicializar el vector de pesos con valores 0.
   - Parar el algoritmo cuando $\|w^{(t+1)} - w^{(t)}\| < 0{,}01$, donde $w(t)$ denota el vector de pesos al final de la época t. Recuérdese que una época es un pase completo a través de los N ejemplos de nuestro conjunto de datos.
   - Aplicar una permutación aleatoria de {1, 2,..., N} a los índices de los datos, antes de usarlos en cada época del algoritmo.

   A continuación, empleando la implementación anterior, realícese el siguiente experimento:

   - Seleccione $N = 100$ puntos aleatorios $\{\mathbf{x}_n\}$ de $\mathcal{X}$ y evalúe las respuestas $\{y_n\}$ de todos ellos respecto de la frontera elegida.
   - Ejecute RL para encontrar la función solución g, y evalúe el error E_{out} usando para ello una nueva muestra de datos (> 999). Se debe escoger experimentalmente tanto el learning rate (tasa de aprendizaje η) como el tamaño de batch.
   - Repita el experimento 100 veces, y calcule los valores promedio de E_{out}, de porcentaje de error de clasificación, y de épocas necesarias para converger.

3. BONUS (1.5 puntos)

El BONUS solo se tendrá en cuenta si se ha obtenido al menos el 75 % de los puntos de la parte obligatoria.

Clasificación de Dígitos. Considerar el conjunto de datos de dígitos manuscritos, y seleccionar las muestras de los dígitos 4 y 8. Extraer las características de intensidad promedio y simetría en la manera que se indicó en el ejercicio 3 de la práctica anterior.

1. Plantear un problema de clasificación binaria que considere el conjunto de entrenamiento como datos de entrada para aprender la función g.

2. Compárense los modelos de regresión lineal, PLA, RL y PLA-Pocket.

   a) Generar gráficos separados de los datos de entrenamiento y test junto con la función estimada.

   b) Calcular E_{in} y E_{test} (error sobre los datos de test).

   c) Si se emplean los pesos obtenidos con regresión lineal para inicializar los otros tres métodos (RL, PLA, PLA-pocket), ¿se observa alguna mejora en los resultados a algún nivel? Justifique su respuesta.

   d) Obtener cotas sobre el verdadero valor de E_{out} para los cuatro métodos empleados. Calcúlense dos cotas: una basada en E_{in} y otra basada en E_{test}. Usar una tolerancia $\delta = 0{,}05$. ¿Que cota es mejor? Justifique la respuesta.