Matemática para ciencias de los datos: Trabajo práctico 4

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Escuela de Ingeniería en Computación,
PAttern Recognition and MAchine Learning Group (PARMA-Group)

5 de junio de 2019

El presente proyecto introduce conceptos de optimización y aprendizaje automático.

- **Fecha de entrega: 12 de Junio**
- **Modo de trabajo**: Grupo de tres personas.
- **Tipo de entrega:** digital, por medio de la plataforma TEC-digital (Jupyter y pdf).

1. (100 puntos) Optimización de funciones

Para las siguientes funciones:

1. $$f_1\left(x_1, x_2\right) = \left(x_1 - 0{,}7\right)^2 + \left(x_2 - 0{,}5\right)^2$$

   con $x_1, x_2 \in [-4, 4]$.

2. $$f_2\left(x_1, x_2\right) = xe^{\left(-x^2 - y^2\right)}$$

   con $x_1, x_2 \in [-2, 2]$.

Realice lo siguiente:

1. **(10 puntos)** Según tales gráficas, grafique las funciones usando la función *meshgrid, y* distinga si las funciones son convexas o no, y los puntos mínimos y regiones o puntos silla.

2. **(50 puntos)** Implemente el algoritmo del **descenso del gradiente**, para cada función:

   a) Escoja un coeficiente de aprendizaje α que permita la convergencia y reporte los resultados para 10 corridas:
      1) la cantidad de iteraciones necesarias para converger,
      2) el punto de convergencia.
      3) Reporte si convergió al punto correcto.
      4) Escoga una de las corridas y en una gráfica muestre los puntos probados por el algoritmo.

   b) Escoja un α relativamente grande respecto al valor seleccionado ¿Qué sucede? ¿Permite un α muy grande la convergencia?
      1) ¿Qué sucede si escoge un α muy pequeño?

   c) Muestre los resultados para 10 corridas:
      1) la cantidad de iteraciones necesarias para converger,
         a′ el punto de convergencia y reporte si fue el correcto.
         b′ Escoga una de las corridas y en una gráfica muestre los puntos probados por el algoritmo, usando una gráfica de las curvas de nivel para la función optimizada.

3. **(40 puntos)** Implemente el algoritmo de **Newton-Raphson**, para cada función:

   a) Calcule la matriz Hessiana demostrando cada paso intermedio.

   b) Reporte los resultados para 10 corridas:
      1) La cantidad de iteraciones necesarias para converger.
      2) El punto de convergencia y reporte si convergió al punto correcto.
      3) Escoga una de las corridas y en una gráfica muestre los puntos probados por el algoritmo, usando una gráfica de las curvas de nivel para la función optimizada.

2. (30 puntos) El algoritmo del Perceptrón con descenso del gradiente

1. Basado en el último inciso del trabajo práctico 0, cree el archivo *twoClassesClassificationSkeleton.*py, el cual implementa la generación de datos aleatorio. Tales datos serán utilizados como datos de prueba. Parametrice la cantidad de muestras, matriz de covarianza y medias.

2. **(15 puntos) Algoritmo del Perceptrón:** Implemente el algoritmo del perceptrón rescindiendo al máximo de estructuras de tipo *for,* usando entonces operaciones matriciales.

3. **(15 puntos)** Para el clasificador:

   a) Realice **2 pruebas con distintas distancias** de separación entre las muestras de las clases, con una **prueba linealmente separable, y otra no**, y documente el número (en una tabla) de muestras mal clasificadas y la cantidad de iteraciones para converger.

      1) Defina el conjunto de muestras de entrenamiento como el 70 % de las muestras aleatoriamente seleccionadas, y el resto utilícelas como muestras de prueba.

   b) Reporte los resultados promedio y varianza para 10 corridas, con datos generados con tres distintas circunstancias de separabilidad de los datos: **linealmente separables, no linealmente separables con mezcla leve de los datos, y mezcla mayor de los dato**s.

   c) Grafique el error o pérdida de entrenamiento de al menos dos corridas, con todas las iteraciones de esas corridas.