APRENDIZAJE AUTOMÁTICO

Trabajo.2: Programación

Fecha límite de entrega:3 de Mayo 2016

Valoración: 10 puntos + 2 puntos de bonus para los apartados opcionales

NORMAS DE DESARROLLO Y ENTREGA DE TRABAJOS

Para este trabajo como para los demás es obligatorio presentar un informe escrito con sus valoraciones y decisiones adoptadas en el desarrollo de cada uno de los apartados. Incluir en el informe los gráficos generados. También deberá incluirse una valoración sobre la calidad de los resultados encontrados. (obligatorio en pdf). **Sin este informe se considera que el trabajo NO ha sido presentado.**

Normas para el desarrollo de los Trabajos: EL INCUMPLIMIENTO DE ESTAS NORMAS SIGNIFICA PERDIDA DE 2 PUNTOS POR CADA INCUMPLIMIENTO.

- El código se debe estructurar en un único script R con distintas funciones o apartados, uno por cada ejercicio/apartado de la práctica.
- Todos los resultados numéricos o gráficas serán mostrados por pantalla, parando la ejecución después de cada apartado. No escribir nada en el disco.
- El path que se use en la lectura de cualquier fichero auxiliar de datos debe ser siempre "datos/nombre_fichero". Es decir, crear un directorio llamado "datos"dentro del directorio donde se desarrolla y se 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 fijar al comienzo los parámetros por defecto que considere que son los óptimos.
- El código debe estar obligatoriamente comentado explicando lo que realizan los distintos apartados y/o bloques.
- Poner puntos de parada para mostrar imágenes o datos por consola.
- Todos los ficheros (*.R *.pdf) se entregan juntos dentro de un único fichero zip, sin ningún directorio que los contenga.
- ENTREGAR SOLO EL CODIGO FUENTE, NUNCA LOS DATOS.
- **Forma de entrega**: Subir el zip al Tablón docente de CCIA.

1. Modelos Lineales

1. (1 punto) **Gradiente Descendente**. Implementar el algoritmo de gradiente descendente.
   a) Considerar la función no lineal de error $E(u,v) = (ue^v - 2ve^{-u})^2$. Usar gradiente descendente y minimizar esta función de error, comenzando desde el punto $(u,v) = (1,1)$ y usando una tasa de aprendizaje $\eta = 0{,}1$.
      1) Calcular analíticamente y mostrar la expresión del gradiente de la función $E(u,v)$
      2) ¿Cuántas iteraciones tarda el algoritmo en obtener por primera vez un valor de $E(u,v)$ inferior a 10^{-14} . (Usar flotantes de 64 bits)
      3) ¿Qué valores de (u,v) obtuvo en el apartado anterior cuando alcanzó el error de 10^{-14}.

   b) Considerar ahora la función $f(x,y) = x^2 + 2y^2 + 2\sin(2\pi x)\sin(2\pi y)$
      1) Usar gradiente descendente para minimizar esta función. Usar como valores iniciales $x_0 = 1, y_0 = 1$, la tasa de aprendizaje $\eta = 0{,}01$ y un máximo de 50 iteraciones. Generar un gráfico de cómo desciende el valor de la función con las iteraciones. Repetir el experimento pero usando $\eta = 0{,}1$, comentar las diferencias.
      2) Obtener el valor mínimo y los valores de las variables que lo alcanzan cuando el punto de inicio se fija: (0,1, 0,1), $(1,1)$,$(-0{,}5, -0{,}5)$,$(-1,-1)$. Generar una tabla con los valores obtenidos ¿Cuál sería su conclusión sobre la verdadera dificultad de encontrar el mínimo global de una función arbitraria?

2. (0.5 puntos) **Coordenada descendente**. En este ejercicio comparamos la eficiencia de la técnica de optimización de "coordenada descendente" usando la misma función del ejercicio 1.1*a*. En cada iteración, tenemos dos pasos a lo largo de dos coordenadas. En el Paso-1 nos movemos a lo largo de la coordenada u para reducir el error (suponer que se verifica una aproximación de primer orden como en gradiente descendente), y el Paso-2 es para reevaluar y movernos a lo largo de la coordenada v para reducir el error (hacer la misma hipótesis que en el paso-1). Usar una tasa de aprendizaje $\eta = 0{,}1$.
   a) ¿Qué valor de la función $E(u,v)$ se obtiene después de 15 iteraciones completas (i.e. 30 pasos) ?
   b) Establezca una comparación entre esta técnica y la técnica de gradiente descendente.

3. (1 punto) **Método de Newton** Implementar el algoritmo de minimización de Newton y aplicarlo a la función $f(x,y)$ dada en el ejercicio.1*b*. Desarrolle los mismos experimentos usando los mismos puntos de inicio.
   - Generar un gráfico de como desciende el valor de la función con las iteraciones.
   - Extraer conclusiones sobre las conductas de los algoritmos comparando la curva de decrecimiento de la función calculada en el apartado anterior y la correspondiente obtenida con gradiente descendente.

4. (1.5 puntos) **Regresión Logística**: En este ejercicio crearemos nuestra propia función objetivo f (probabilidad en este caso) y nuestro conjunto de datos $\mathcal{D}$ para ver cómo funciona regresión logística. Supondremos por simplicidad que f es una probabilidad con valores 0/1 y por tanto que y es una función determinista de $\mathbf{x}$.

   Consideremos $d = 2$ para que los datos sean visualizables, y sea $\mathcal{X} = [-1,1] \times [-1,1]$ con probabilidad uniforme de elegir cada $\mathbf{x} \in \mathcal{X}$. Elegir una línea en el plano como la frontera entre $f(\mathbf{x}) = 1$ (donde y toma valores $+1$) y $f(\mathbf{x}) = 0$ (donde y toma valores -1), para ello seleccionar dos puntos aleatorios del plano y calcular la línea que pasa por ambos.

Seleccionar $N = 100$ puntos aleatorios $\{\mathbf{x}_n\}$ de $\mathcal{X}$ y evaluar las respuestas de todos ellos $\{y_n\}$ respecto de la frontera elegida.

a) Implementar Regresión Logística(RL) con Gradiente Descendente Estocástico (SGD) bajo las siguientes condiciones:

- Inicializar el vector de pesos con valores 0.
- Parar el algoritmo cuando $||\mathbf{w}^{(t-1)} - \mathbf{w}^{(t)}|| < 0{,}01$, donde $\mathbf{w}^{(t)}$ denota el vector de pesos al final de la época t. Una época es un pase completo a través de los N datos.
- Aplicar una permutación aleatoria de $1, 2, \ldots, N$ a los datos antes de usarlos en cada época del algoritmo.
- Usar una tasa de aprendizaje de $\eta = 0{,}01$

b) Usar la muestra de datos etiquetada para encontrar g y estimar E_{out} usando para ello un número suficientemente grande de nuevas muestras.

c) (*Opcional 0.2*) Repetir el experimento 100 veces con diferentes funciones frontera y calcule el promedio.

1) ¿Cuál es el valor de E_{out} para $N = 100$?
2) ¿Cuántas épocas tarda en promedio RL en converger para $N = 100$, usando todas las condiciones anteriormente especificadas?

5. (1 punto) **Clasificación de Dígitos**. Considerar el conjunto de datos de los dígitos manuscritos y seleccionar las muestras de los dígitos 1 y 5. Usar los ficheros de entrenamiento (training) y test que se proporcionan. Extraer las características de **intensidad promedio** y **simetría** en la manera que se indicó en el ejercicio 3 del trabajo 1.

   Plantear un problema de clasificación binaria que considere el conjunto de entrenamiento como datos de entrada para aprender la función g. Usando el modelo de Regresión Lineal para clasificación seguido por PLA-Pocket como mejora. Responder a las siguientes cuestiones.

   a) Generar gráficos separados (en color) 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) Obtener cotas sobre el verdadero valor de E_{out}. Pueden calcularse dos cotas una basada en E_{in} y otra basada en E_{test}. Usar una tolerancia $\delta = 0{,}05$. ¿Que cota es mejor?

   d) (*Opcional 0.1*) Repetir los puntos anteriores pero usando una transformación polinómica de tercer orden($\Phi_3(\mathbf{x})$ en las transparencias de teoría).

   e) (*Opcional 0.1*) Si tuviera que usar los resultados para dárselos a un potencial cliente ¿usaría la transformación polinómica? Explicar la decisión.

2. Sobreajuste

1. (2 puntos) **Sobreajuste.** Vamos a construir un entorno que nos permita experimentar con los problemas de sobreajuste. Consideremos el espacio de entrada $\mathcal{X} = [-1, 1]$ con una densidad de probabilidad uniforme, $\mathbb{P}(x) = \frac{1}{2}$. Consideramos dos modelos $\mathcal{H}_2$ y $\mathcal{H}_{10}$ representando el conjunto de todos los polinomios de grado 2 y grado 10 respectivamente. La función objetivo es un polinomio de grado Q_f que escribimos como $f(x) = \sum_{q=0}^{Q_f} a_q L_q(x)$, donde $L_q(x)$ son los polinomios de Legendre (ver la relación de ejercicios.2). El conjunto

de datos es $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ donde $y_n = f(x_n) + \sigma\epsilon_n$ y las $\{\epsilon_n\}$ son variables aleatorias i.i.d. $\mathcal{N}(0,1)$ y σ^2 la varianza del ruido.

Comenzamos realizando un experimento donde suponemos que los valores de Q_f, N, σ, están especificados, para ello:

- Generamos los coeficientes a_q a partir de muestras de una distribución $\mathcal{N}(0,1)$ y escalamos dichos coeficientes de manera que $\mathbb{E}_{\mathbf{a},x}[f^2] = 1$ (Ayuda: Dividir los coeficientes por $\sqrt{\sum_{q=0}^{Q_f} \frac{1}{2q+1}}$)
- Generamos un conjunto de datos, $x_1, \ldots, x_N$ muestreando de forma independiente $\mathbb{P}(x)$ y los valores $y_n = f(x_n) + \sigma\epsilon_n$.

Sean g_2 y g_{10} los mejores ajustes a los datos usando $\mathcal{H}_2$ y $\mathcal{H}_{10}$ respectivamente, y sean $E_{\text{out}}(g_2)$ y $E_{\text{out}}(g_{10})$ sus respectivos errores fuera de la muestra.

a) Calcular g_2 y g_{10}

b) ¿Por qué normalizamos f? (Ayuda: interpretar el significado de σ)

c) (*Opcional 0.1*) ¿Cómo podemos obtener E_{out} analíticamente para una g_{10} dada ?

2. (0.5 puntos) Siguiendo con el punto anterior, usando la combinación de parámetros $Q_f = 20, N = 50, \sigma = 1$ ejecutar un número de experimentos (>100) calculando en cada caso $E_{\text{out}}(g_2)$ y $E_{\text{out}}(g_{10})$. Promediar todos los valores de error obtenidos para cada conjunto de hipótesis, es decir

$$E_{\text{out}}(\mathcal{H}_2) = \text{promedio sobre experimentos}(E_{\text{out}}(g_2))$$
$$E_{\text{out}}(\mathcal{H}_{10}) = \text{promedio sobre experimentos}(E_{\text{out}}(g_{10}))$$

Definimos una medida de sobreajuste como $E_{\text{out}}(\mathcal{H}_{10}) - E_{\text{out}}(\mathcal{H}_2)$.

a) Argumentar por qué la medida dada puede medir el sobreajuste.

b) (*Opcional 0.1*) Usando la combinación de valores de los valores $Q_f \in \{1, 2, \ldots, 100\}$, $N \in \{20, 25, \ldots, 100\}$, $\sigma \in \{0; 0{,}05; 0{,}1; \ldots; 2\}$, se obtiene una gráfica como la que aparece en la figura 4.3 del libro "Learning from data", capítulo 4. Interpreta la gráfica respecto a las condiciones en las que se da el sobreajuste. (Nota: No es necesario la implementación).

3. (*Opcional 0.4*) Repetir el experimento descrito en los puntos anteriores pero para el caso de clasificación donde la función objetivo es un perceptron ruidoso $f(x) = \text{sign}(\sum_{q=1}^{Q_f} a_q L_q(x) + \epsilon)$. Notemos que $a_0 = 0$ y que las a_q deben ser normalizadas para que $\mathbb{E}_{\mathbf{a},x}\left[(\sum_{q=1}^{Q_f} a_q L_q(x))^2\right] = 1$. En este caso los modelos $\mathcal{H}_2$ y $\mathcal{H}_{10}$ contienen el signo de los polinomios de segundo y décimo orden respectivamente. (Atención: los datos no serán separables) (Ayuda: para la normalización adaptar la regla dada en el punto anterior)

3. Regularización y Selección de Modelos

1. (2.5 puntos) Para $d = 3$ (dimensión) generar un conjunto de N datos aleatorios $\{\mathbf{x}_n, y_n\}$ de la siguiente forma. Para cada punto $\mathbf{x}_n$ generamos sus coordenadas muestreando de forma independiente una $\mathcal{N}(0,1)$. De forma similar generamos un vector de pesos de (d+1) dimensiones $\mathbf{w}_f$, y el conjunto de valores $y_n = \mathbf{w}_f^T \mathbf{x}_n + \sigma\epsilon_n$, donde ϵ_n es un ruido que sigue también una $\mathcal{N}(0,1)$ y σ^2 es la varianza del ruido; fijar $\sigma = 0{,}5$.

Usar regresión lineal con regularización "weight decay" para estimar $\mathbf{w}_f$ con $\mathbf{w}_{reg}$. Fijar el parámetro de regularización a $0{,}05/N$.

a) Para $N \in \{d+15, d+25, \ldots, d+115\}$ calcular los errores $\mathrm{e}_1, \ldots, \mathrm{e}_\mathrm{N}$ de validación cruzada y E_{cv}.

b) Repetir el experimento 10^3 veces, anotando el promedio y la varianza de e_1, e_2 y E_{cv} en todos los experimentos.

c) ¿Cuál debería de ser la relación entre el promedio de los valores de e_1 y el de los valores de E_{cv} ? ¿y el de los valores de e_2? Argumentar la respuesta en base a los resultados de los experimentos.

d) ¿Qué es lo que contribuye a la varianza de los valores de e_1 ?

e) Si los errores de validación-cruzada fueran verdaderamente independientes, ¿cual sería la relación entre la varianza de los valores de e_1 y la varianza de los de E_{cv}?

f) Una medida del número efectivo de muestras nuevas usadas en el cálculo de E_{cv} es el cociente entre la varianza de e_1 y la varianza de E_{cv}. Explicar por qué, y dibujar, respecto de N, el número efectivo de nuevos ejemplos(N_{eff}) como un porcentaje de N. NOTA: Debería de encontrarse que N_{eff} está cercano a N.

g) Si se incrementa la cantidad de regularización, ¿debería N_{eff} subir o bajar?. Argumentar la respuesta. Ejecutar el mismo experimento con $\lambda = 2{,}5/N$ y comparar los resultados del punto anterior para verificar la conjetura.

2. (*Opcional* 1.0 punto) La técnica de validación cruzada da una estimación precisa de $\hat{E}_{\mathrm{out}}(N-1)$, pero puede ser demasiado inestable en problemas de selección de modelos. Una heurística común para regularizar validación cruzada en selección de modelos es usar una medida de su error $\sigma_{\mathrm{cv}}(\mathcal{H})$

   a) Una elección para $\sigma_{\mathrm{cv}}(\mathcal{H})$ es el uso de la desviación estándar de los errores por LOO ("leave-one-out") dividida por $\sqrt{N}$, $\sigma_{\mathrm{cv}}(\mathcal{H}) \approx \frac{1}{\sqrt{N}}\sqrt{\mathrm{var}(e_1, e_2, \ldots, e_n)}$. ¿Por qué se divide por $\sqrt{N}$?

   b) Analizamos dos aproximaciones para la estimación:

      (I) Dado el mejor modelo $\mathcal{H}^*$, la aproximación conservadora de 1-sigma selecciona el modelo más simple a una distancia $\sigma_{\mathrm{cv}}(\mathcal{H}^*)$ del mejor.

      (II) La aproximación que minimiza una cota selecciona el modelo que minimiza $E_{\mathrm{out}}(\mathcal{H}) + \sigma_{\mathrm{cv}}(\mathcal{H})$

   Usar el diseño experimental del ejercicio sección.2.1 (Sobreajuste) para comparar estas aproximaciones con la estimación "no-regularizada" de validación cruzada. Para ello hacer lo siguiente

      1) Fijar $Q_f = 15$, $N = 20$ y $\sigma^2 = 1$.

      2) Aplicar cada una de las dos aproximaciones propuestas así como el estimador estandar de validación cruzada para seleccionar el valor óptimo del parámetro de regularización λ en el conjunto $\{0{,}05; 0{,}10; 0{,}15; \ldots; 5\}$ usando regularización por "weight decay", $\Omega(\mathbf{w}) = \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$.

      3) Dibujar el `error fuera de la muestra`, para cada una de las técnicas, usando cada una de ellas como una función de N, con $N \in \{2 \times Q, 3 \times Q, \ldots, 10 \times Q\}$. ¿Cuales son sus conclusiones?