Optimización

Primer Cuatrimestre 2025

Práctica de Laboratorio N° 1

Ejercicio 1 (Máxima verosimilitud para la distribución Gamma) Sea $\{x_1, x_2, \ldots, x_n\}$ una muestra independiente de una variable aleatoria con distribución Gamma, con función de densidad

$$f(x; \alpha, \beta) = \frac{1}{\Gamma(\alpha)\,\beta^{\alpha}} x^{\alpha-1}\, e^{-x/\beta}, \quad x > 0,$$

donde $\alpha > 0$ es el parámetro de forma y $\beta > 0$ el parámetro de escala. El objetivo es estimar ambos parámetros a partir de la muestra mediante el método de máxima verosimilitud. Para ello, definimos la verosimilitud de una muestra, como:

$$L(\alpha, \beta) = \prod_{i=1}^{n} f(x_i; \alpha, \beta),$$

(a) Obtener una expresión para el logarítmo de la verosimilitud. ¿Por qué tomar el logaritmo preserva el máximo?

(b) Obtener las condiciones necesarias de optimalidad de primer orden respecto a α y β.

(c) Sustituyendo la solución obtenida para β, obtener una única ecuación no-lineal para estimar α a partir de los datos. Formular la solución mediante el método de Newton-Raphson.

(d) Utillizando la librería Distributions.jl, generar 1000 muestras de una variable gamma con parámetros conocidos y estimarlos utilizando el método propuesto.

Ejercicio 2 (Regresión logística) El dataset Titanic contiene información sobre los pasajeros del RMS Titanic, incluyendo variables predictoras (como edad, sexo, clase, etc.) y la variable respuesta $y_i \in \{0, 1\}$ que indica si el pasajero sobrevivió (1) o no (0). Se supone que las observaciones son independientes y se modela la probabilidad de supervivencia mediante un modelo de regresión logística.

Sea $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ el vector de variables explicativas del i-ésimo pasajero y se define

$$z_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} = \beta^T \tilde{x}_i,$$

donde $\tilde{x}_i$ es el vector extendido (con un 1 para el intercepto) y $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ es el vector de parámetros. La probabilidad de supervivencia se modela mediante la función sigmoide:

$$p_i = \sigma(z_i) = \frac{1}{1 + \exp(-z_i)}.$$

(a) Suponiendo que la variable respuesta sigue una distribución Bernoulli, verifique que la log-verosimilitud viene dada por la siguiente expresión:

$$\ell(\beta) = \sum_{i=1}^{n} \Big[y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \Big],$$

(b) Cargue y preprocese el dataset Titanic, imputando datos faltantes por la mediana, normalizando los datos, y transformando la variable respuesta en 0 y 1.

(c) Implemente la función log-verosimilitud $\ell(\beta)$ y calcule su gradiente utilizando la librería `ForwardDiff.jl`.

(d) Implemente el método de descenso por gradiente con paso constante y experimente sobre el valor necesario del paso para obtener convergencia con un dato inicial aleatorio.

Ejercicio 3 (Clasificación multiclase) El dataset Iris consta de 150 observaciones, cada una con 4 variables predictoras (por ejemplo, largo y ancho de sépalo y pétalo) y una etiqueta que indica la especie de Iris, la cual puede tomar 3 valores distintos (setosa, versicolor, virginica). Se supone que las observaciones son independientes y se modela la probabilidad de pertenecer a cada clase mediante un modelo de regresión logística multiclase utilizando la función softmax.

Sea $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$ el vector de variables predictoras de la i-ésima observación y sea $y_i \in \{1, 2, 3\}$ la etiqueta correspondiente. Para modelar la clasificación multiclase, se definen los logits como

$$z_i = Wx_i + b,$$

donde $W \in \mathbb{R}^{3\times 4}$ y $b \in \mathbb{R}^3$ es el vector de interceptos (o sesgos). La función softmax asigna a cada clase una probabilidad según

$$p_{ij} = \frac{\exp(z_{ij})}{\sum_{k=1}^{3} \exp(z_{ik})}, \quad j = 1, 2, 3,$$

donde z_{ij} es el j-ésimo componente del vector z_i, y p_{ij} es la probabilidad asignada a la clase j para la observación i. Además, las etiquetas y_i se transforman a un formato one-hot, de modo que para cada observación i, $y_{ij} = 1$ si la clase es j y 0 en caso contrario.

(a) Suponiendo que las observaciones siguen una distribución multinomial (o categórica), verifique que la log-verosimilitud viene dada por la expresión:

$$\ell(W, b) = \sum_{i=1}^{n} \sum_{j=1}^{3} y_{ij} \ln(p_{ij}),$$

(b) Importe el dataset Iris, divídalo en conjuntos de entrenamiento y prueba, normalice las variables predictoras y convierta las etiquetas a formato one-hot (o numérico) para su adecuada utilización en el modelo.

(c) Implemente la función log-verosimilitud $\ell(W, b)$ en Julia y, utilizando la librería `ForwardDiff.jl`, calcule su gradiente respecto a los parámetros W y b.

(d) Implemente el método de descenso por gradiente con paso constante, iniciando desde una condición inicial aleatoria para W y b. Experimente con diferentes valores del paso para determinar cuál permite una convergencia estable del algoritmo.

Ejercicio 4 (Redes Neuronales Multicapa) El dataset MNIST consta de 70,000 imágenes de dígitos escritos a mano (28×28 píxeles) con etiquetas en $\{0, 1, 2, \ldots, 9\}$. Se desea entrenar una red neuronal multicapa para clasificar los dígitos en sus respectivas categorías. Para ello, se considerará una red neuronal con una capa oculta, donde la entrada x_i es el vector de características de la i-ésima imagen (aplanada en un vector de dimensión 784) y la salida es un vector de probabilidades sobre las 10 clases, calculado mediante la función softmax. La arquitectura de la red se define mediante los siguientes parámetros:

1. Capa oculta con h neuronas y activación sigmoidal:

$$a_i^{(1)} = \sigma(W^{(1)} x_i + b^{(1)}),$$

donde $W^{(1)}$ es una matriz de pesos de dimensión $h \times 784$ y $b^{(1)}$ es el vector de sesgos de dimensión h.

2. Capa de salida con 10 neuronas y activación softmax:

$$p_i = \text{softmax}(W^{(2)} a_i^{(1)} + b^{(2)}),$$

donde $W^{(2)}$ es una matriz de pesos de dimensión $10 \times h$ y $b^{(2)}$ es el vector de sesgos de dimensión 10.

Se definen las etiquetas y_i en formato one-hot, de modo que $y_{ij} = 1$ si la imagen i pertenece a la clase j, y 0 en caso contrario. Como antes, uponiendo una distribución categórica para la variable respuesta, la log-verosimilitud viene dada por:

$$\ell(W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}) = \sum_{i=1}^{n} \sum_{j=1}^{10} y_{ij} \ln(p_{ij}).$$

(a) Importe el dataset MNIST, normalice los valores de los píxeles en el rango $[0, 1]$, divida el conjunto en datos de entrenamiento y prueba, y convierta las etiquetas a formato one-hot.

(b) Escriba la función log-verosimilitud en Julia y utilice la librería `ForwardDiff.jl` para calcular automáticamente el gradiente respecto a los parámetros $W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}$.

(c) Implemente el método de descenso por gradiente con paso constante, iniciando desde valores aleatorios para los parámetros. Experimente con diferentes valores del paso para determinar cuál permite una convergencia estable del algoritmo.