MA4402 - Simulación Estocástica: Teoría y Laboratorio Primavera 2021
Profesor: Joaquín Fontbona
Auxiliares: Bruno Hernández, Pablo Araya

Laboratorio 4.1: Descenso de gradiente estocástico y aplicaciones

Problema 1: Regresión logística con gradiente estocástico

En este problema aplicaremos el algoritmo de descenso de gradiente estocástico a un método de clasificación supervisada denominado **regresión logística**. Consideramos un conjunto de M datos correspondiente a vectores $\{x^{(m)}\}_{i=1,\dots,M} \subset \mathbb{R}^p$ de características, y sus respectivas etiquetas $\{y^{(m)}\}_{i=1,\dots,M} \subset \{0,1\}$ que indican si cada uno de ellos pertenece a una de dos "clases" posibles C_0 ó C_1. El objetivo es usar estos datos para aprender como clasificar correctamente cualquier vector nuevo x de características en una de las clases C_k, $k = 0, 1$ (sin ver su etiqueta).

Por simplicidad nos restringiremos a modelos de clasificación lineales, los que pueden escribirse de manera general como

$$\phi(x) = \sigma\left(\langle a, x\rangle + b\right),$$

donde $x \in \mathbb{R}^p$, $a \in \mathbb{R}^p$ y $b \in \mathbb{R}$ son parámetros fijos, y $\sigma : \mathbb{R} \to [0,1]$ es una función "de activación", creciente, no lineal, fija, que se usará para atribuir el vector x a la clase C_1 o C_0 según si el valor de ϕ está por encima o debajo de un cierto umbral (por ejemplo, etiquetar x como de clase C_1 si $\phi(x) = \sigma(a^{\mathrm{t}}x + b) \in (1/2, 1]$, y de clase C_0 en caso contrario). Para simplificar la notación introducimos la escritura

$$\sigma(w^{\mathrm{t}}x) = \sigma\left(\langle a, x\rangle + b\right),$$

donde $w = (a, b) \in \mathbb{R}^{p+1}$ y, con abuso de notación, x en el lado izquierdo corresponde al vector agrandado $(x, 1) \in \mathbb{R}^{p+1}$. Puesto que $\sigma(w^{\mathrm{t}}x) \in (0, 1)$ podemos pensar intuitivamente en esta cantidad como "probabilidad" de que un vector de características aleatorias x provenga de la clase C_1.

Escogeremos como σ la llamada función logística o sigmoide:

$$\phi(r) = \sigma(r) = \frac{1}{1 + e^{-r}}.$$

1. Dado un dato x, interprete geométricamente la condición $\sigma(w^{\mathrm{t}}x) \in (1/2, 1]$.

   Veamos ahora que la intepretación probabilista antes mecionada es rigurosa para el caso de un modelo Bayesiano en que los datos son generados por dos posibles leyes Gaussianas multivariadas. De manera más precisa supongamos que, condicionalmente a que el dato x proviene de la clase C_i, $i \in \{0, 1\}$, se tiene que $x \sim \mathcal{N}(\mu_i, \Sigma)$ normal multivariada en $\mathbb{R}^p$ de media μ_i y varianza-covarianza no singular Σ (igual para ambas clases). Supongamos además que cada dato proviene de una clase C_i, que tiene probabilidad a priori $p(C_i) \in (0, 1)$ de ser elegida para generar ese dato. Explicite $p(x|C_i)$ para $i = 0, 1$ y muestre que $p(C_1|x)$ esta dada por

   $$p(C_1|x) = \sigma(w^{\mathrm{t}}x) = \frac{1}{1 + \exp\left(-\langle a, x\rangle - b\right)},$$

   con

   $$a = \Sigma^{-1}(\mu_1 - \mu_0) \text{ y } b = \frac{1}{2}(\mu_0^{\mathrm{t}}\Sigma^{-1}\mu_0 - \mu_1^{\mathrm{t}}\Sigma^{-1}\mu_1) + \ln\left(\frac{p(C_1)}{p(C_0)}\right).$$

2. Supondremos en lo que sigue que cada observación $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$ se obtiene de manera independientemente, sampleando, *para cada una*, primero una clase $y^{(m)} \in \{0, 1\}$ y luego un vector de características $x^{(m)}$ aleatorias de datos de clase $C_{y^{(m)}}$. Para encontrar el parámetro $w = (a, b) \in \mathbb{R}^{p+1}$ que

mejor explica las etiquetas de los datos observados $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$, se propone maximizar la función de "verosimilitud "

$$\prod_{m=1}^{M} \sigma\big(w^{\mathrm{t}} x^{(m)}\big)^{y^{(m)}} \Big(1 - \sigma\big(w^{\mathrm{t}} x^{(m)}\big)\Big)^{(1-y^{(m)})},$$

lo cual es equivalente a minimizar la función de pérdida

$$L(w) = -\frac{1}{M} \sum_{m=1}^{M} y^{(m)} \log \sigma\big(w^{\mathrm{t}} x^{(m)}\big) + (1 - y^{(m)}) \log\Big(1 - \sigma\big(w^{\mathrm{t}} x^{(m)}\big)\Big). \tag{1}$$

En el caso de la parte 1, dé una interpretación de esta función en términos de verosimilitudes "a posteriori" y explique por qué tiene sentido buscar maximizarla. Encuentre además (en general) una expresión simple para la derivada con respecto a w de cada uno de los términos de la suma.

3. Proponga e implemente un algoritmo de descenso de gradiente estocástico, para resolver numéricamente el problema de minimización

$$\min_{w} L(w).$$

Genere 4000 datos "de entrenamiento" etiquetados de dos clases Gaussianas en $\mathbb{R}^2$ como en la parte 1, equiprobables, para cada una de 2 elecciones fijas de parámetros (μ_0, μ_1) (correspondiente a distintos grados de separación de las Gaussianas).

Utilice el algoritmo para encontrar el hiperplano que mejor separa las dos Gaussianas en cada caso. Para el paso considere varias sucesiones del estilo $\gamma_t = \frac{\gamma_0}{1+t/t_0}$, con distintos valores de parámetros (por ejemplo $t_0 = 100$ y $\gamma_0 = 0{,}05$). Para cada conjunto de datos y grado de separación, corra el algoritmo varias veces desde varias condiciones iniciales distintas ¿Cómo influyen estas en el resultado? Compare los resultados obtenidos con el hiperplano w "teórico " para esas Gaussianas. Grafique los hiperplanos obtenidos junto con (parte de) las nubes de datos. En cada caso, guarde la trayectoria de puntos w_t generados por el algoritmo.

4. Considere ahora un caso de Gaussianas "poco separadas". Fije una elección de condición inicial entre las consideradas en la parte anterior. Corra el algoritmo con *mini-batchs* de tamaños distintos. Grafique en cada caso la evolución de los valores de la función que se minimiza, y comente las diferencias observadas con distintos pasos y tamaños.

Discuta la rapidez de convergencia del algoritmo, la influencia del paso elegido y del tamaño del *mini-batch*.

Finalmente, para cada uno los conjuntos de datos considerados en esta parte, genere ahora 1000 datos adicionales "de prueba". Cuantifique el error de generalización (o "fuera de muestra") de la estimación realizada, usando la log-verosimilitud como función de w, calculada con estos nuevos datos. Muestre gráficamente como mejora ese error de generalización a medida que w_t evoluciona hasta el valor final obtenido con el algoritmo.

5. Aplique ahora regresión logística para clasificar tumores de mama, en las clases "benigno" o "maligno", usando los datos provistos en Material Docente (`data.csv`[1]). Para ello estandarice cada una de las columnas de datos y entrene el algoritmo antes implementado con el $80\,\%$ de los datos. Use después el $20\,\%$ restante para evaluar como generaliza el clasificador encontrado con regresión logística .

[1] Fuente: `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29`

Problema 2: Regresión Lineal

Dado un conjunto de datos $(x_i, y_i)_{i=1}^n \subset \mathbb{R}^m \times \mathbb{R}$, se propone la siguiente relación entre sus componentes

$$y = \theta^T x + \varepsilon, \tag{2}$$

en donde ε es una variable aleatoria con valor esperado 0 y distribución desconocida. El problema de regresión lineal consiste en encontrar el parámetro θ tal que el conjunto de datos satisfaga (2) de manera tal que $\mathrm{Var}(\varepsilon)$ sea lo más pequeña posible. Esto conlleva al problema de optimización

$$\hat{\theta} = \arg\min_{\theta} \mathrm{Var}(\varepsilon) = \arg\min_{\theta} \mathbb{E}\left((y - \theta^T x)^2\right).$$

El objetivo de esta pregunta es aplicar el modelo anterior sobre el conjunto de datos **Boston Housing Prices** y estimar el mejor parámetro posible utilizando el método de descenso de gradiente estocástico.

1. Cargue el conjunto de datos utilizando el siguiente código:

```
from sklearn.datasets import load_boston
import pandas as pd

data = load_boston()
df = pd.DataFrame(data.data, columns = data.feature_names)
df['PRICE'] = data.target
```

   Observe la cantidad de variables y el tipo de datos que posee. Normalice los datos para que el modelo a trabajar funcione. Extienda la base de datos para obtener un modelo de regresión lineal, esta vez representado por una función afín de la forma $y_i = \theta^T x_i + b + \varepsilon_i$, con $\theta \in \mathbb{R}^m$, $b \in \mathbb{R}$.

2. Separe los datos en un conjunto de entrenamiento y otro de prueba según la proporción 80 % y 20 %, para esto le será útil la función `train_test_split` de la librería `sklearn`. Justifique brevemente por qué esto es necesario.

En lo que sigue justifique sus respuestas graficando la función de costos cada cierta cantidad de iteraciones. Cuando se pida comparar diferentes implementaciones debe realizarlo en base al conjunto de datos de prueba y el error cuadrático medio incurrido con la estimación obtenida. Tanto la cantidad de iteraciones como los parámetros pueden ser escogidos libremente.

3. Implemente el algoritmo de descenso de gradiente estocástico para un modelo de regresión lineal, especificando cuál es la función de costos y su gradiente. Considere los siguientes casos

   a) *Learning rates* constantes.

   b) *Learning rates* variables (proponga al menos 2).

   Proponga al menos dos en cada caso. Compare los resultados obtenidos para cada elección ¿Cuál es la mejor elección?

4. Modifique el algoritmo anterior para trabajar con *mini-batch*. Pruebe el desempeño (nuevamente en términos del error cuadrático medio para el conjunto de prueba) del algoritmo para distintos tamaños de *mini-batch* y *learning rates* y justifique cual es el mejor ¿Existe alguna relación entre ambos parámetros?

5. Implemente las siguientes variantes de descenso de gradiente estocástico y compare el desempeño de estos con los algoritmos implementados en las partes anteriores

   a) **Momentum:** El método consiste en ir generando los pasos de descenso como

   $$m_i = \beta m_{i-1} + (1-\beta)\nabla_\theta f(\theta_i, x_i), \quad m_0 = 0,$$

   tal que

   $$\theta_{i+1} = \theta_i - \eta\, m_i,$$

   donde $\beta \in (0,1)$ (debe ser elegido).

   b) **Adagrad:** El *learning rate* es variable y se genera de la siguiente manera:

   $$\eta_i = \frac{\eta}{\sqrt{v_i + c}},$$

   donde $c > 0$, $\eta > 0$ y

   $$v_i = \sum_{j=1}^{i} \|\nabla_\theta f(\theta_j, x_j)\|^2, \quad v_0 = 0.$$