Curso de aprendizaje automatizado

PCIC, UNAM

Tarea 2: Regresión y clasificación lineal

Fecha límite: 6 de abril.

Formato: Libretas de Jupyter de manera independiente y reproducible.

Forma de entrega: Enviar tarea por Google Classroom.

Predicción de precios de automóviles

A partir del conjunto de datos *Automobile Dataset*[1], realiza la regresión de los precios de automóviles con las siguientes variantes:

a. Mínimos cuadrados con expansión polinomial de diferentes grados.

b. Mínimos cuadrados con expansión polinomial de grado 20 y penalización por norma ℓ_1 y ℓ_2 con diferentes valores de λ.

c. Mínimos cuadrados con expansión polinomial de grado 2 y selección de atributos[2].

Grafica el error cuadrático medio en entrenamiento y validación con respecto al grado del polinomio, valor de λ y número de atributos. Todos los modelos deberán ser evaluados con 10 repeticiones de validación cruzada de 5 particiones. Selecciona uno de los modelos y reporta su desempeño en el conjunto de prueba[3].

Regresión con datos sintéticos

Aplica regresión lineal a los datos sintéticos que se encuentran divididos en los archivos de entrenamiento `x_entrenamiento.csv` y `y_entrenamiento.csv` y los de validación `x_validacion.csv` y `y_validacion.csv`[4]. Estos datos fueron contaminados con ruido gaussiano con media igual a 0 y desviación estándar igual a 0.05. Realiza lo siguiente:

[1] Disponible en `https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data` y la descripción en `https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.names`

[2] Puedes usar cualquier estrategia para la selección de atributos

[3] Es posible hacer un reajuste del modelo con los hiperparámetros seleccionados.

[4] Estos archivos se encuentran disponibles en `http://turing.iimas.unam.mx/~gibranfp/cursos/aprendizaje_automatizado/data/regl_data.zip`

a. Grafica los datos de entrenamiento y de validación y comenta brevemente acerca de cómo están distribuidos.

b. Considera un modelo de la forma $f(\mathbf{x}) = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2$ y realiza la regresión lineal. Reporta los parámetros que encontraste usando el estimador de máxima verosimilitud y el valor del error cuadrático medio para los datos de entrenamiento y de validación.

c. Usa una expansión de base polinomial y entrena un modelo de regresión lineal con regularización por norma ℓ_2. Reporta los parámetros obtenidos y el error cuadrático medio para los datos de entrenamiento y de validación.

Opcional, +1 Considera que los parámetros θ_0, θ_1 y θ_2 están distribuidos de acuerdo a una distribución normal multivariante $\mathcal{N}(0, I)$ (media igual a 0 y matriz de covarianza igual a la identidad). Aplica regresión lineal bayesiana y obtén la distribución predictiva, reportando la media y la varianza para el vector $\tilde{\mathbf{x}} = [1, 2]$. Grafica la distribución a posteriori de los parámetros con 5, 10, 30 y 60 datos. Adicionalmente, genera 20 muestras de la distribución a posteriori de los parámetros con 5, 10, 30 y 60 datos y gráfica las curvas de los modelos para estas muestras.

Predicción de juegos

Un club del juego de Go recopiló los resultados de varias partidas entre diferentes jugadores, almacenados en el archivo `partidas_entrenamiento.txt`, con el objetivo de predecir el resultado de partidas futuras, ejemplos de las cuales se encuentran en el archivo `partidas_prueba.txt`. Los archivos `partidas_entrenamiento.txt` y `partidas_prueba.txt`[5] contienen 3 columnas: la primera corresponde al identificador del jugador A, la segunda al identificador del jugador B y la tercera es el resultado de la partida (1 si ganó el jugador A o 0 si ganó el jugador B). En el club hay un total de D jugadores, por lo que cada identificador es un número entero entre 1 y D. La predicción del resultado de un juego se puede plantear como un problema de clasificación: dados 2 jugadores (A y B) se requiere predecir si A ganó ($y = 1$) o si fue B ($y = 0$). Realice los siguientes ejercicios:

- Entrena y evalúa un clasificador bayesiano ingenuo. Al ser un modelo generativo (modela la probabilidad conjunta $P(\mathbf{x}, y)$), es posible generar partidas artificiales con los parámetros calculados. Genera nuevas partidas que sigan la distribución modelada.

- Entrena y evalúa un clasificador de regresión logística[6]. Debido a que ambos atributos son categóricos, es necesario cambiar la codificación. Explica el procedimiento y la lógica de la codificación que realizaste. Visualiza los valores de los parámetros del modelo de regresión logística y discute qué interpretación tendrían de acuerdo a la codificación realizada. Grafica las curvas ROC y de precisión-exhaustividad y reporta sus áreas bajo la curva.

- Compara el clasificador bayesiano ingenuo y regresión logística en este problema. ¿Qué ventajas y desventajas tienen los modelos entrenados? ¿Qué pasaría si se entrena el clasificador bayesiano ingenuo con los vectores recodificados o si se entrena un modelo de regresión logística usando los vectores de entrada originales? ¿Consideras que las presuposiciones de cada clasificador son apropiadas para los datos del problema? ¿Para este tipo de problemas cuál de los dos recomendarías y por qué?

[5] Estos archivos se encuentran disponibles en `https://github.com/gibranfp/CursoAprendizajeAutomatizado/blob/master/data/`

[6] Se espera que el clasificador por regresión logística se programe usando únicamente las bibliotecas NumPy y SciPy de Python.

- Deriva la regla de actualización para el algoritmo del descenso por gradiente de un clasificador donde $\hat{y} = sigm(\boldsymbol{\theta}^\top \mathbf{x})$ y la función de pérdida sea

$$E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} \left(\hat{y}^{(i)} - y^{(i)} \right)^2 .$$

  Discute las diferencias entre este clasificador y el de regresión logística y el efecto que tendría en el comportamiento del algoritmo del descenso por gradiente cambiar la función de pérdida por

$$E(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(\hat{y}^{(i)} - y^{(i)} \right)^2 .$$

Regresión logística *vs* clasificador bayesiano ingenuo

Compara los métodos de regresión logística[7] y el clasificador bayesiano ingenuo en las siguientes tareas:

- *Clasificación de spam*[8]
- *Clasificación de tumores de seno*[9]

Discute qué modelo seleccionarías y por qué. Reporta el desempeño del modelo seleccionado en el conjunto de prueba[10]. Todos los modelos deberán ser evaluados con 10 repeticiones de validación cruzada estratificada de 5 particiones.

[7] Se espera que el clasificador de regresión logística se programe usando únicamente las bibliotecas NumPy y SciPy de Python.

[8] Con el conjunto de datos disponible en `http://turing.iimas.unam.mx/~gibranfp/cursos/aprendizaje_automatizado/data/spam.csv`

[9] Con el conjunto de datos disponible en `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data` y la descripción en `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names`

[10] Es posible hacer un reajuste del modelo con los hiperparámetros seleccionados.