Departamento de Ingeniería Matemática
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Curso: Aprendizaje de Máquinas, MA5204
Profesor: Felipe Tobar
Profesores Auxiliares: V. Faraggi, F. Fêtis, B. Moreno, F. Vásquez, A. Wortsman
Fecha de publicación: 12/04/21

Tarea Práctica #1

Dedicación recomendada: 10 horas

Fecha de entrega: 26 de abril

Instrucciones: La tarea es grupal, en grupos de 2 o 3 integrantes, ni más ni menos. En caso de ser un grupo de 2 personas, deben realizar P1 y P2, en el caso de 3 personas, deben realizar P1, P2 y P3. El formato de entrega es un reporte en `.pdf` de máximo 4 planas sin anexos, con doble columna, abstract, título e integrantes. Si hace su reporte en LaTeX, utilice el template de la conferencia ICML disponible en este enlace, o bien uno de formato similar. También debe entregar los códigos utilizados en formato `.html`, los cuales puede obtener descargando directamente desde *Jupyter* su notebook en ese formato.

P1. **[Modelo de Naïve Bayes[1] [30 %]]**

Naïve Bayes Classifier es un modelo de probabilidades condicionales para clasificación, donde se asume que las categorías o atributos son mutuamente independientes dado el valor de la clase. Este modelo utiliza fuertemente el teorema de Bayes. Formalmente, el modelo se desarrolla de la siguiente forma: Considere $C_i \in C$, donde C son las clases, y C_i es la i-ésima clase con $i \in \{1, \dots, K\}$, con K el número de clases. Además, considere $X = (x_1, \dots, x_n)^\top$ el vector de atributos para un dato con n atributos. Para un vector de atributos X y una clase C_i dada, tenemos que, por el teorema de Bayes, la probabilidad de C_i se puede escribir como:

$$\mathbb{P}(C_i|x_1, ..., x_n) = \frac{\mathbb{P}(C_i)\mathbb{P}(x_1, ..., x_n|C_i)}{\mathbb{P}(x_1, ..., x_n)}. \tag{1}$$

Sin embargo, la premisa de independencia condicional de Naïve Bayes establece que:

$$\mathbb{P}(x_j|C_i, x_j, ..., x_{j-1}, x_{j+1}, ..., x_n) = \mathbb{P}(x_j|C_i),$$

para $j \in \{1, ..., n\}$. Consecuentemente, la ecuación (1) se puede reescribir como:

$$\mathbb{P}(C_i|x_1, ..., x_n) = \frac{\mathbb{P}(C_i)\prod_{j=1}^{n}\mathbb{P}(x_j|C_i)}{\mathbb{P}(x_1, ..., x_n)}.$$

Recordando la Teoría de Bayes, tenemos que $\mathbb{P}(x_1, ..., x_n)$ es difícilmente computable, además de ser constante para cada clase C_i. Entonces, como en verdad queremos derivar el estimador MAP, proporcionalmente se obtiene que:

[1] Los modelos de Naïve Bayes hacen referencia a varios modelos que se derivan de forma similar a lo mostrado, que utilizan la condición o premisa de independencia condicional dado una clase. En esta pregunta se verá un caso puntual de estos.

$$\mathbb{P}(C_i|x_1,...,x_n) \propto \mathbb{P}(C_i)\prod_{j=1}^{n}\mathbb{P}(x_j|C_i),$$

y considerando el MAP obtenemos:

$$\hat{K} = \underset{i}{\text{argmax}}\ \mathbb{P}(C_i)\prod_{j=1}^{n}\mathbb{P}(x_j|C_i). \tag{2}$$

de donde la clase predicha será $\hat{C} = C_{\hat{K}}$.

El objetivo de esta pregunta es implementar el Naïve Bayes Classifier. Considere la base de datos de golf de `data_golf_train.csv`, donde la clase a predecir es *Play* y sus atributos son 2 variables categóricas (*Outlook* y *Windy*) y dos numéricas (*Temperature* y *Humidity*). Se pide lo siguiente:

a) Implemente el Naïve Bayes Classifier. Para implementar el algoritmo, debe encontrar las probabilidades $\mathbb{P}(x_j|C_i), \forall i \in \{1,2\}, \forall j \in \{1,...,n\}$. Para las variables categóricas, se deben encontrar las tablas de probabilidades condicionales, mientras que para las variables numéricas, debe asumir que $x_j|C_i \sim \mathcal{N}(\mu, \sigma^2)$, y estimar μ y σ para cada clase con los estimadores usuales.

b) Obtenga la predicción de las clases para los datos de `data_golf_test.csv`, comente brevemente sobre los valores dentro del máximo de (2) y la diferencia de los atributos de los datos de testeo. Haga una predicción de su clasificador para los datos de entrenamiento y reporte los resultados obtenidos.

c) Implemente un modelo de Regresión Logit. Utilice la librería `sklearn`. Prediga las clases para los datos de entrenamiento y testeo. Compare los resultados obtenidos con el modelo de Naïve Bayes. Reporte el valor de los coeficientes obtenidos en la Regresión Logit y comente, complemente con los resultados observados de Naïve Bayes. ¿Cuál modelo es mejor?, ¿según cuál métrica?, ¿Por qué utilizó tal métrica?, ¿Se puede decir algo sobre la cantidad de los datos y el rendimiento de los modelos?.

P2. **MCO, LASSO y RR [40 %]**

Se considera el modelo lineal con una variable dependiente $y \in \mathbb{R}$ con una independiente $\widetilde{x} \in \mathbb{R}^{d+1}$ mediante la relación $y = \theta^\top \widetilde{x}$ en donde $\widetilde{x} = (1,x)^\top$ con $x = (x_1,...,x_d)$ donde d es la cantidad de atributos y $\theta \in \mathbb{R}^{d+1}$. En base a un conjunto de observaciones de la forma $\{(\widetilde{x}_i, y_i)\}_{i=1}^n$, existen distintos estimadores puntuales de θ los cuales se derivan de aplicar distintas penalizaciones en los problemas de regresión, los cuales son codificadas en la función de costo. Un criterio estándar de penalización es el basado en la norma de los parámetros, el cual está dado por:

$$J_\rho = ||Y - \widetilde{X}\theta||_2^2 + \rho||\theta||_p^p, \quad p \geq 0, \quad \rho \geq 0, \tag{3}$$

donde $||\cdot||_p$ denota la norma ℓ_p. Se estudiarán 3 estimadores puntuales, los cuales son: Mínimos Cuadrados Ordinarios (MCO), Regresión Ridge (RR) y Regresión LASSO (LASSO). Cada estimador se puede obtener de la ecuación (3): cuando $\rho = 0$ se tiene MCO, $p = 2$ es RR y $p = 1$ recupera LASSO. A cada parámetro θ obtenido desde estos problemas de optimización les denominará, respectivamente, θ_{MCO}, θ_{RR} y θ_{LASSO}.

El objetivo de esta pregunta es interpretar las regresiones desde un punto de vista bayesiano. Considerando entonces que la relación entre la variable dependiente e independiente viene dada por:

$$y|\theta, \widetilde{x} \sim \mathcal{N}(y; \theta^\top \widetilde{x}, \sigma^2) \tag{4}$$
$$\theta \sim \pi(\theta), \tag{5}$$

donde σ es conocido y $\pi(\theta)$ denota la distribución a priori de θ. Trabajaremos con la estimación puntual definida *máximo a posteriori* (MAP), definida mediante:

$$\theta^{MAP} = \arg\max_{\theta} p(Y|\widetilde{X}, \theta)\pi(\theta). \tag{6}$$

Observe que θ^{MAP} es el estimador puntual de cada uno de los casos descritos arriba: Para MCO, se debe considerar una distribución prior uniforme, la cual es impropia (por ejemplo $\pi(\theta) = 1$). En el caso de RR, se debe considerar que $\pi(\theta) \sim \mathcal{N}(0, \frac{\sigma^2}{\rho} I_{d+1})$. Por último, para recuperar el estimador LASSO, se debe considerar que $\theta_i \sim \text{Laplace}(0, \frac{2\sigma^2}{\rho})$.

a) Se pide que implemente MCO, LASSO y RR en la base de datos *California Housing*. Utilice `sklearn`, el cual le permitirá implementar todos los modelos en unas pocas líneas, su código debería empezar de la siguiente forma:

```
from sklearn.datasets import fetch_california_housing
data = fetch_california_housing()
print(data.keys()) #1era es input y 2da es output
print(data.feature_names) #variables de entrada
# importar modelos
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
```

b) Analice los resultados obtenidos basados en las métricas R^2 y RMSE. Complemente sus análisis en base a los valores de θ encontrados por cada método, grafique los parámetros obtenidos para cada atributo con su nombre. En particular, ¿Qué puede decir de la magnitud de los elementos de θ en cada método? Interprete los modelos obtenidos, en base a su derivación por el enfoque bayesiano y los criterios estándar de penalización, discuta sobre las características de θ que promueve cada prior y cómo se realiza la penalización en la ecuación (3).

c) Separe el dataset en entrenamiento (80 %) y testeo (20 %) con la función `train_test_split` de `sklearn.model_selection`. Entrene los 3 modelos con los datos de entrenamiento y prediga los output de los datos de testeo. Reporte las métricas R^2 y RMSE. ¿Qué tan distinto fue el ajuste de los parámetros con respecto a la parte anterior?, ¿A qué se puede deber lo anterior?.

d) **(Bonus, para esta pregunta).** Estudie la variable de respuesta, identifique un potencial problema de esta variable y plantee una forma de resolverlo. Realice una eliminación de outliers (en base a percentiles) de los atributos y escale los datos (con el método que le parezca mejor). Ajuste nuevamente los 3 modelos, estudie las métricas obtenidas en sus predicciones y diga qué regresión obtiene el mejor rendimiento. Haga un nexo entre el rendimiento obtenido y el enfoque del modelo.

P3. **Más sobre regresiones [30 %]**

En el archivo `data_p3.txt` están los datos de la cantidad de pasajeros de una aerolínea medidos de forma mensual. Los datos son de la forma $\{(x_i, y_i)\}_{i=1}^n$ donde x_i representa un mes, e y_i la cantidad de pasajeros transportados en el mes correspondiente.

El objetivo de esta pregunta es modelar la cantidad de pasajeros (y) respecto al tiempo (x). Para esto, se asumirá el siguiente modelo:

$$y = f_\theta(x) + \eta,$$

donde θ son los parámetros de la función f que se deben ajustar, y $\eta \sim N(0, \sigma_n^2)$ es ruido gaussiano.

Separe los datos en un conjunto de entrenamiento y otro de test. Considere que el 75 % de los datos son para entrenar (primeros 9 años) y el 25 % es de test.

a) Cargue y grafique los datos de forma que los conjuntos de entrenamiento y de test sean fácilmente identificables.

b) Utilice la función `polyfit` de `numpy` para ajustar un modelo polinomial de grado 1,2,3 y 4. Encuentre el modelo que mejor se ajuste a los datos.

c) Como podrá observar, no es posible capturar el movimiento periódico de los datos usando sólo un polinomio. Denotemos por f^{pol} al polinomio que encontró en la parte anterior. Modelaremos la señal con el siguiente modelo:

$$y = f^{pol} + \theta_1 \sin(\theta_2 x + \theta_3) + \eta,$$

con $\eta \sim N(0, \sigma_n^2)$. Ajuste $\theta_{1:4}$ usando máxima verosimilitud. Note que este modelo no es lineal en los parámetros, por lo que no se puede escribir una solución de forma exacta. Para encontrar los parámetros de máxima verosimilitud, construya la función de verosimilitud y optimice utilizando la función `minimize` de `scipy.optimize` usando el método BFGS.
¿Qué tan bueno es el ajuste encontrado?

Indicación: Puede ser útil ocupar $\theta_4 = 0{,}01$ como condición inicial.

d) El comportamiento de los datos no es sólo sinusoidal, por lo que agregaremos una última componente. Denotemos por $f^{pol-sin}$ a la función f_θ encontrada en la parte anterior. Agregue una segunda componente sinusoidal modulada por una exponencial al modelo, es decir:

$$y = f_\theta + \eta = f^{pol-sin} + \theta_1 \sin(\theta_2 x + \theta_3) e^{\theta_4 x} + \eta.$$

Considere los parámetros de $f^{pol-sin}$ fijos e iguales a su resultado de la parte anterior. Nuevamente use máxima verosimilitud para encontrar $\theta_{1:4}$, optimizando con `minimize` y BFGS. Evalúe su solución prediciendo el 25 % restante de los datos.

Indiciación: Nuevamente, puede ser útil ocupar $\theta_4 = 0{,}01$ como condición inicial.

Presente sus resultados y discuta el método en que los obtuvo, en particular, explique el rol de cada una de las componentes del modelo final. ¿Habría sido posible entrenar el modelo final de una vez? Evalúe los modelos ajustados en función de su verosimilitud y del error de predicción en el conjunto de evaluación.