Estadística

Anexos de Estadística

Intro a Estadística Multivariada

Esperanza y Media

La esperanza es el valor esperado de una variable aleatoria, si se busca el valor esperado X, esta operación se denota como $E[X]$. Se calcula como el producto punto entre el vector de los posibles valores de la variable aleatoria y la probabilidad de que ocurran dichos valores.

La media que se denota como μ se calcula en Inferencia Estadística a partir de los datos que se tienen de cierta variable aleatoria. El cómo se utiliza la media sobre vectores, columnas o renglones de matrices se encuentra en el ANEXO 1.

Varianza y Desviación Estándar

La varianza mide qué tan dispersos son los valores posibles de una variable aleatoria. La operación se denota como $Var[X]$ y se calcula de la siguiente manera:

$$Var[X] = E[(X - E[X])^2)$$

La muestra de cómo se calcula esta variable se muestra en el ANEXO 2.

La desviación estándar se calcula simplemente como la raíz cuadrada de la varianza.

Covarianza y Correlación

La covarianza en probabilidad es la medida de probabilidad conjunta de dos variables aleatorias, describe cómo cambian dos variables respecto a la otra. Se le denota como $cov(X, Y)$ y se calcula como:

$$cov(X, Y) = E[(X - E[X]) \times (Y - E[Y])]$$

Lo importante de esta operación es observar el signo resultante, si ambas crecen en la misma dirección, será positivo y en el otro caso, negativo. Además, si se busca normalizar este valor entre −1 y 1, se aplica lo que se conoce como el coeficiente de correlación de Pearson. Este se calcula de la siguiente manera:

$$r = \frac{cov(X, Y)}{\sigma_X \times \sigma_Y}$$

La manera de realizar todas estas operaciones en Python se encuentra en el ANEXO 3.

Matriz de Covarianza

La matriz de covarianza es una matriz en donde se compara la covarianza entre dos o más variables aleatorias. Es decir, se cumple que:

$$\Sigma_{i,j} = cov(X_i, X_j)$$

En donde X es una matriz en la cual todos los renglones son una variable aleatoria. Cabe mencionar que la diagonal de la matriz de covarianza es la varianza de la variable aleatoria y además, la matriz es cuadrada y simétrica. Los cálculos en Python de estas operaciones se encuentran en el ANEXO 4.

Análisis de Componentes Principales

Qué es el PCA

El PCA es un método de reducción de dimensión, se puede pensar que es una proyección de los datos, es decir, si se cuenta con una matriz que tiene muchas características, al aplicarle el PCA se terminará con una proyección de ésta, en una matriz de menor dimensión. Veamos a continuación los pasos a seguir para realizar este método:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

$$B = PCA(A)$$

El primer paso sería calcular el valor medio de cada columna.

$$M = mean(A)$$

Y luego centramos todos los valores al restar el valor medio de cada columna.

$$C = A - M$$

El siguiente paso es calcular la matriz de covarianza de la matriz centrada en C. La correlación se hará respecto a las columnas, incluyéndose a sí misma. Finalmente se calcula la matriz de eigendescomposición de la matriz de covarianza $V = cov(C)$ que resulta en una lista de eigenvalores y eigenvectores. $values, vectors = eig(V)$. Los eigenvectores representan las direcciones o componentes del subespacio reducido de B y los eigenvalores representan las magnitudes para las direcciones. Los eigenvectores se ordenan por los eigenvalores en orden descendiente para aplicarle un rango a los componentes del nuevo subespacio. Si un eigenvalor está cercano a 0, significa que no tiene mucha información relevante y se puede eliminar la variable asociada. La manera de hacer ese cálculo en Python es a través de la librería Scikit-Learn, la muestra de ello se encuentra en el ANEXO 5.

Regresión Lineal

La regresión lineal es un método para modelar la relación entre una o más variables independientes y una variable dependiente. Se le considera un buen método introductorio de Machine Learning. El método puede ser reformulado con la notación y operaciones matriciales.

Qué es la Regresión Lineal

Es un método para modelar la relación entre dos valores escalares, la variable de entrada x y la variable de salida y. El modelo asume que y es una función lineal de la variable de entrada. Es decir:

$$y = b_0 + b_1 \times x_1$$

El modelo también puede ser utilizado con varias variables de entrada. El objetivo es encontrar los valores de los coeficientes b que minimicen el error en la predicción del valor de salida.

Formulación Matricial de Regresión Lineal

La regresión lineal puede denotarse como:

$$y = X \cdot b$$

En donde X es la matriz con las variables de entrada y b es el vector con los coeficientes a modificarse. El problema por resolverse es un sistema de ecuaciones que está sobredeterminado, lo cual lo hace un poco complejo para resolverse, pues hay muchos posibles valores para los coeficientes. Además, todas

las soluciones tendrán un error, pues en realidad se está tomando únicamente una aproximación. Por tanto, se toma la solución a partir de los coeficientes que minimicen el error. A este método se le conoce como el método de los mínimos cuadrados. La fórmula por minimizar es la siguiente:

$$||X \cdot b - y||^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j} \cdot (b_j - y_i)^2$$

La realización de este método en Python se encuentra en el ANEXO 6.

Resolución a través de la Inversa

El primer intento, será resolver esto por medio de la inversa utilizando la matriz inversa. Así, dado X, qué son los valores que multiplicados a los de la entrada, dan los de la salida. Como vimos ya en la sección pasada, las ecuaciones normales definen cómo calcular b directamente.

$$b = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

Esto se puede calcular rápidamente utilizando una función en Python. La muestra de cómo se hace en Python se encuentra en el ANEXO 7.

Resolución a través de Descomposición QR

La descomposición QR parte la matriz en sus elementos constitutivos. $A = Q \cdot R$. Este método es popular para resolver la ecuación de mínimos cuadrados. Tras derivar, se puede encontrar los coeficientes de la manera:

$$b = R^{-1} \cdot Q^T \cdot y$$

En este método también se involucra la inversa, pero se aplica sobre una matriz mucho más simple. La muestra de cómo se hace en Python se encuentra en el ANEXO 8.

Resolución vía SVD y Pseudoinversa

Vamos a calcular $b = X^+ \cdot y$ como ya sabemos utilizar la Pseudoinversa obteniéndola de la manera

$$X^+ = U \cdot D^+ \cdot V^T$$

La muestra de cómo se hace en Python se encuentra en el ANEXO 9.

Resolución vía la Función Conveniencia

La Pseudoinversa vía SVD para resolver por mínimos cuadrados es la manera estándar de hacer las cosas. Esto se debe a que es estable y funciona con muchos Datasets. NumPy da la opción de utilizar la función lstsq() que resuelve utilizando mínimos cuadrados. La función toma a X como input y el vector y y devuelve b que son los coeficientes, además de los valores residuales. La muestra de cómo se hace en Python se encuentra en el ANEXO 10.