Universidad de Salamanca
Grado en Matemáticas

REVISIÓN DE MÉTODOS MULTIVARIANTES SUPERVISADOS Y NO SUPERVISADOS

Trabajo Fin de Grado

VNiVERSiDAD D SALAMANCA

Alumno: Pedro Ángel Fraile Manzano

Tutoras: Ana Belén Nieto Librero y Nerea González García

Salamanca, Julio de 2023

Índice general

Introducción

El Análisis Multivariante es, según *Martínez Arias, R.*[21], el conjunto de técnicas y métodos que busca describir y extraer información de las relaciones entre variables medidas en una o varias muestras u observaciones. Dentro de esta definición, entran todos los procedimientos que analizan de manera simultánea más de una variable. Los métodos multivariantes se pueden clasificar según el objetivo a perseguir:

- ***Reducción de datos***: En estos procedimientos se busca analizar la estructura de los datos para buscar una simplificación de la misma. Dentro de este tipo de técnicas están el *Análisis Factorial*, *Análisis de Componentes Principales, Análisis Discriminante* etc.
- ***Clasificación y Agrupación:*** Es decir, buscar modos de clasificar las observaciones en grupos homogéneos ya sea de acuerdo a una variable categórica conocida o buscando esa homogeneidad dentro de los datos. Podemos incluir aquí el *Análisis de Conglomerados*
- ***Análisis de Relaciones y Dependencias.***
- ***Construcción de Modelos y pruebas de hipótesis***

También se pueden clasificar según el conocimiento de los datos o de la variable a analizar como métodos supervisados o no supervisados.

En esta memoria se revisarán métodos de todas las tipologías, tanto supervisados como no supervisados. Primero con un acercamiento teórico, otorgando una idea de la base sobre los que se construyen y luego se aplicarán sobre distintos ejemplos desarrollados en Python con distintas librerías especializadas en Análisis Multivariante como TensorFlow o Scikit-Learn.

Estas técnicas han visto un auge en los últimos años, debido a su utilidad en el análisis de grandes bases de datos y la creciente capacidad de recogida de datos que se tiene en la actualidad. Además, el llevar a buen término el estudio anteriormente era complejo debido a la gran cantidad de cálculos que se necesitan realizar. Actualmente esos cálculos son automatizados, en algunos casos, con una única línea de código es posible llevar a cabo ciertas técnicas de las descritas.

Además, ciertos autores como *Hastie T., Tibshirani R. y Friedman J.* [8] enfocan también estos problemas desde el punto del aprendizaje automático, ya sea también supervisado o no.

Capítulo 1

Métodos Supervisados

1.1. Introducción

Sea un conjunto de datos obtenidos de observar ciertas variables aleatorias de manera simultánea. De ese conjunto de variables se dividirán en:

- ***Variables de entrada o inputs***: Son las variables independientes que determinarán de manera aleatoria al segundo conjunto de variables. Al conjunto de variables aleatorias de entrada se la denotará como el vector aleatorio $\mathbf{x}^T = [X_1, \dots X_p]$.
  Tomando n observaciones de estas variables se obtiene la *matriz de datos* $\mathbf{X}$. Esta matriz es de tamaño $n \times p$ y contiene como filas los vectores de longitud p que representan los datos de cada observación, se denotan como $\mathbf{x}_i$. Por ejemplo, en el caso de que se recojan datos sobre distintos modelos de coches serían la potencia del motor, el tipo de combustible $etc\dots$

- ***Variables de salida u outputs***: Son las variables dependientes de las anteriores. A este conjunto de variables se les denota con el vector aleatorio $\mathbf{y}^T = [Y_1, \dots Y_k]$, en el caso de que se tenga una única variable respuesta se denotará como Y.
  Como en el caso de las variables de entrada, se tomarán n observaciones de las k variables, dando como resultado la matriz de respuestas $\mathbf{Y}$ de tamaño $n \times k$, donde cada observación es una fila y se denota como $\mathbf{y}_i$, en el caso de que $k > 1$ y como y_i cuando $k = 1$. Siguiendo el ejemplo anterior, se podrían tener variables respuesta como el tipo de etiqueta medioambiental siendo esta una variable cualitativa o el precio del vehículo que tiene naturaleza cuantitativa.

Por ende, se recogen observaciones simultáneas de las variables de entrada y de salida formando parejas $(\mathbf{y}_i, \mathbf{x}_i)$. El objetivo de estos métodos supervisados es encontrar una relación funcional o predictor de tal manera que para una nueva observación de las variables de entrada $\mathbf{x}_0$ se pueda hacer una predicción $\hat{\mathbf{y}}_0$ del valor real $\mathbf{y}_0$ de la variable respuesta.

1.1.1. Mínimos cuadrados y K-vecinos más cercanos

Supóngase el caso en el que únicamente tenemos una variable respuesta Y. Además supóngase que la relación que se utiliza para realizar predicciones $\hat{y}$ de los valores respuesta es una combinación lineal de los valores predictores, de manera que se tiene un vector $\hat{\beta} = [\hat{\beta}_0, \ldots, \hat{\beta}_p]$ de parámetros que permite hacer predicciones de la siguiente manera para una nueva observación $\mathbf{x}_0$:

$$\hat{y}_0 = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{0j} \tag{1.1.1}$$

Utilizando una primera componente de $\mathbf{x}_0$ igual a 1 se puede dar la expresión como producto matricial.

$$\hat{y}_0 = \mathbf{x}_0^T \hat{\beta} \tag{1.1.2}$$

Para este y muchos otros casos el modelo ha de ajustarse a los datos recogidos para las p variables de entrada y las variables respuesta que se tengan en ese momento. Para poder dar una medida de la calidad del ajuste se definen las funciones de pérdida.

Definición 1.1.1. Se llama *función de pérdida* a la función que cuantifica el error cometido al usar la predicción y se denota como $L(y, \hat{y})$

Si se toma como medida global la suma de las pérdidas de todas las observaciones, en el caso de que $L(y, \hat{y}) = (y - \hat{y})^2$ se obtiene el *Error Cuadrático Acumulado* o *RSS* por su siglas en inglés.

Definición 1.1.2. Se define el *Error Cuadrático Acumulado* a la suma de los cuadrados de las diferencias entre las predicciones y los valores reales de la variable respuesta. En el caso de una sola variable aleatoria respuesta y suponiendo que se han recogido n observaciones conjuntas se expresa de la siguiente manera:

$$RSS = \sum_{i=1}^{n} L(y_i, \hat{y}_i) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{1.1.3}$$

para el caso del modelo lineal:

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \beta)^2 \tag{1.1.4}$$

En esencia, este error cuadrático acumulado no es más que la pérdida total del conjunto de datos. Esto nos da una forma de la calidad global del ajuste. Además es generalizable a otras funciones de pérdida.

Para ajustar los parámetros β, se busca minimizar la cantidad de error que se comete al predecir, por tanto, establecemos el problema de optimización con la función $RSS(\beta)$ como función objetivo. Es por ello, que esta debe ser derivable y aunque sea posible elegir otras funciones de pérdida, como pudiera ser $L(y, \hat{y}) = |y - \hat{y}|$, estas resultan en funciones de pérdida globales no derivables.

La expresión de $RSS(\beta)$ en forma matricial teniendo en cuenta que $\mathbf{X}$ sea la matriz de observaciones de tamaño $(n \times p+1)$ que tiene la primera columna constante 1 e $\mathbf{y}$ es el vector columna o matriz que contiene las observaciones de la o las variables respuestas es la siguiente:

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}^T\beta)^T(\mathbf{y} - \mathbf{X}^T\beta) \tag{1.1.5}$$

Que al derivarla respecto de β e igualarlo a 0 se obtiene que:

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0 \tag{1.1.6}$$

Que tiene solución única si la matriz $\mathbf{X}^T\mathbf{X}$ es invertible resultando en que $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.

Este método de ajuste de los parámetros de la función de predicción se llama método de los *Mínimos Cuadrados*, que es de largo el más utilizado aunque tiene sus problemas.

Mientras que el modelo (la relación que se usa como predicción) que se ha utilizado asume una cierta estructura global, que tiene sus desventajas, también se pueden dar métodos que no se centren en la estructura a nivel global, sino local.

Con este objetivo se construirá uno de los predictores locales más simples y conocidos el método de k-vecinos.

Definición 1.1.3. Sea un conjunto de n observaciones $\mathbf{x}_i, i = 1, \dots n$, se define el conjunto $N_k(\mathbf{x}_0)$ al conjunto de las k observaciones más cercanas al punto $\mathbf{x}_0$.

Una vez definido ese conjunto se puede definir el siguiente predictor:

$$\hat{y}(\mathbf{x}_0) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x}_0)} y_i \tag{1.1.7}$$

Este método de predicción no requiere que se realice el ajuste de ningún parámetro (*es un predictor no paramétrico*) y además es bastante simple de utilizar, basta con promediar los valores respuesta de cada una de las observaciones del conjunto de datos.

En contraste, el método de los K-Vecinos y los métodos locales en general, sufren de lo que se llama la *maldición de la dimensionalidad*. Este suceso consiste en que a medida que aumentamos la dimensión, en este caso la cantidad de variables aleatorias a observar, la densidad de los datos es cada vez menor. Esto provoca que por ejemplo que los conjuntos N_k sean conjuntos muy dispersos, por tanto las observaciones que usamos como referencias y la observación de la cual se va a realizar la predicción están muy lejanas.
En particular, la densidad es proporcional a $n^{\frac{1}{p}}$, donde p es el número de variables y n el número de observaciones. Por ejemplo, si una muestra densa en 1 dimensión tiene $n = 100$, para mantener dicha densidad en $p = 10$ se necesitarían $n = 100^{10}$. De las características de los espacios de observaciones de altas dimensiones se habla en *Koppen,*[11]

1.1.2. Decisión Estadística

Sea un vector aleatorio real de entrada $\mathbf{x} \in \mathbb{R}^p$, una variable de salida $Y \in \mathbb{R}$ y sea la probabilidad conjunta $\mathbb{P}(Y, \mathbf{x})$ de ambas, entonces se busca una función $f(\mathbf{x})$ para predecir Y, es decir, $\hat{Y} = f(\mathbf{x})$. Para ello, se utilizan las funciones de pérdida, en lo sucesivo utilizaremos la pérdida cuadrática $L(Y, \hat{Y}) = L(Y, f(\mathbf{x})) = (Y - f(\mathbf{x}))^2$.

Definición 1.1.4. Teniendo en cuenta lo anterior, se define el *error de predicción esperado* como la esperanza de la nueva variable que define la función de pérdida.

$$\begin{aligned} EPE(f) &= E(Y - f(\mathbf{x}))^2 \\ &= \int [y - f(\mathbf{x})]^2 \mathbb{P}(dx, dy) \end{aligned} \tag{1.1.8}$$

Para ajustarlo a la situación de predecir el valor de Y para nuevos valores observados de $\mathbf{x}$, se puede condicionar respecto del valor de $\mathbf{x}$ obteniendo la siguiente expresión:

$$EPE(f) = E_{\mathbf{x}} E_{Y|\mathbf{x}}([Y - f(\mathbf{x})]^2 | \mathbf{x}) \tag{1.1.9}$$

Esta expresión otorga según *Hastie et.al* [8] un criterio para elegir la f. Basta con tomar f de manera que:

$$f(x) = \text{argmin}_c E_{Y|\mathbf{x}}([Y - c]^2 | \mathbf{x} = x) \tag{1.1.10}$$

Por tanto, el que minimiza en el caso de la pérdida establecida anteriormente es $f(x) = E(Y|\mathbf{x} = x)$.

Definición 1.1.5. Se llama *función de regresión* a la función $f(x) = E(Y|\mathbf{x} = x)$

Por ejemplo, para el caso del modelo de K-Vecinos, la función de regresión es la siguiente:

$$\hat{f}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i \tag{1.1.11}$$

Hasta ahora, solo se han contemplado los casos en los que la variable respuesta era cualitativa. Para el caso en el que esta sea categórica, la mayoría de conceptos son análogos con ciertas modificaciones.

Sea el caso de una variable de salida categórica G que toma k valores de manera que el predictor $\hat{G}$ tambien toma esos valores, entonces la única modificación que debemos hacer es la de la función de pérdida que pasa a ser una matriz $\mathbf{L}$ de tamaño $k \times k$ donde $L_{i,j}$ es la penalización por categorizar como $\mathcal{G}_j$ algo que en realidad es $\mathcal{G}_i$. Entonces tenemos que *el error de predicción esperado* es:

$$EPE = E[L(G, \hat{G})] = E_{\mathbf{x}} \sum_{i=1}^{k} L[\mathcal{G}_k, \hat{G}(\mathbf{x})] \mathbb{P}(\mathcal{G}_k | \mathbf{x}) \tag{1.1.12}$$

Esta definición otorga un criterio análogo para establecer el predictor $\hat{G}(x)$, que si se utiliza la pérdida 0-1, es decir, aquella que otorga una penalización de 1, se obtiene el *clasificador bayesiano*.

Definición 1.1.6. Se llama *clasificador bayesiano* al predictor

$$\hat{G}(x) = \mathcal{G}_k \text{ si } \mathbb{P}(\mathcal{G}_k|\mathbf{x} = x) = \max_{g \in \mathcal{G}} \mathbb{P}(g|\mathbf{x} = x) \tag{1.1.13}$$

Es decir, se otorga la categoría que más probabilidad de ser tiene conociendo el valor de $\mathbf{x}$.

Una vez definidos los principales predictores, hay que tener en cuenta que este tipo de métodos se pueden ver desde el prisma de un problema de aprendizaje automático, en particular, son métodos de aprendizaje supervisado.

En este caso, no se detallará como un problema de aprendizaje automático aunque varios métodos de los detallados frecuentemente se interpreten de esta manera.

Otra manera es interpretarlo cómo un problema de aproximación de funciones, de manera que se aproximen teniendo en cuenta los puntos $\mathbf{x}_i$ en $\mathbb{R}^p$, por ejemplo, en el caso del modelo lineal se intenta ajustar un hiperplano afín.

Aún así, a estos métodos se les llama *métodos supervisados* ya que se parte del conocimiento de las observaciones $(\mathbf{x}_i, \mathbf{y}_i)$ y se adapta el modelo y los parámetros conociendo el valor real de las observaciones de la variable respuesta.

1.1.3. Sesgo y Varianza del Modelo

A la hora de elegir un modelo, hay que tener en cuenta la complejidad del mismo, ya que un modelo demasiado complejo puede ajustarse perfectamente a los datos de entrenamiento, pero tener una capacidad de generalización nula. Por otro lado, un modelo demasiado simple puede no captar las características básicas de los datos.

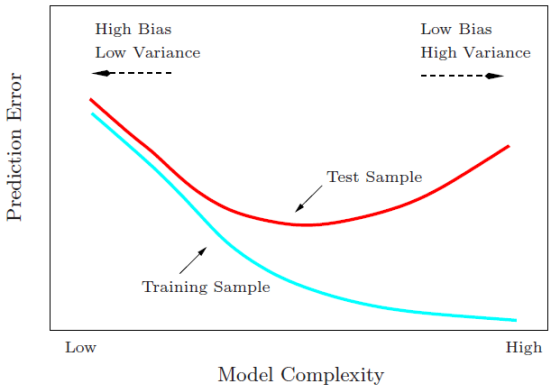
Supóngase que la variable respuesta sigue la siguiente expresión $Y = f(\mathbf{x}) + \varepsilon$ donde $\mathbb{E}(\varepsilon) = 0$ y $Var(\varepsilon) = \sigma^2$, supongamos además que las observaciones $\mathbf{x}_i$ son ya conocidas. Entonces, se puede expresar el error de predicción de una nueva observación $\mathbf{x}_0$ de la siguiente manera:

$$\begin{aligned} EPE(\mathbf{x}_0) &= \mathbb{E}[(Y - \hat{f}(x))^2 | X = \mathbf{x}_0] = \\ &= \sigma^2 + B^2(\hat{f}(\mathbf{x}_0)) + Var(\hat{f}(\mathbf{x}_0)) \end{aligned} \tag{1.1.14}$$

Esta expresión está compuesta de dos términos que dependen de la estimación, $\hat{f}$, que son el sesgo y la varianza de esta. Ahora bien, hay un término que no depende de la estimación que es σ^2, el término de error inevitable.

Por ejemplo, el modelo de K-Vecinos más cercanos, la complejidad está influida por el parámetro K. Cuántos más vecinos tenga el vecindario, mayor es el sesgo del predictor, ya que entran en juego más observaciones para calcular el valor de y_i y menor es la varianza.

La siguiente gráfica ha sido extraída de *Hastie et. al.* [8] y representa el error de predicción en función de la complejidad del modelo. La línea azul es el error de predicción sobre muestras del conjunto de entrenamiento y la roja sobre muestras nuevas que no han sido utilizadas para el ajuste.



Hay un punto de equilibrio, en el cual una complejidad media otorga un error de predicción aceptable sobre muestras de entrenamiento y un error de predicción no muy alto sobre muestras que no lo son. Cuando se toma modelos demasiado complejos el modelo puede sufrir de *sobreajuste*

Definición 1.1.7. Se dice que un predictor o modelo sufre de *overfitting* o *sobreajuste* en los casos en los que el predictor tiene un bajo sesgo pero una varianza alta. Esto se ve reflejado en un muy buen ajuste en el conjunto de entrenamiento pero una calidad de predicción de nuevos datos mala.

Estos casos de overfitting se dan por ejemplo, cuando tenemos un conjunto de datos con más variables observadas que observaciones. Hay distintos métodos para evitar el sobreajuste en conjuntos de datos con pocas observaciones, estos se detallan en el capítulo 7 de *Hastie et.al.*[8]

1.2. Métodos Lineales para regresión

El objetivo de la regresión es conseguir una relación funcional que permita predecir los valores de una variable respuesta cuantitativa en función de los valores de unas variables de entrada. En el caso de la regresión lineal, ya sea simple o múltiple, se asume que esa relación funcional es lineal.

Para ello, se tendrá en cuenta que las observaciones de la variable respuesta siguen la relación que se detallaba anteriormente:

$$Y = f(\mathbf{x}) + \varepsilon \tag{1.2.1}$$

Donde ε es una variable aleatoria con $\mathbb{E}(\varepsilon) = 0$ y $Var(\varepsilon) = \sigma^2$.

En este tipo de métodos se supone que f es una función lineal de las variables de entrada del vector aleatorio $\mathbf{x}^T = [X_1 \dots X_p]$. De esta manera, se tiene que:

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{1.2.2}$$

Por tanto, la predicción de la variable de salida $\hat{y}_i$ para una observación $\mathbf{x}_i$ del vector aleatorio se puede expresar de la siguiente manera:

$$\hat{y}_i = f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j \tag{1.2.3}$$

De esta manera, se tiene un vector de p parámetros $\beta_1 \dots \beta_p$ además de otro β_0. Esto se puede simplificar añadiendo al vector aleatorio una variable aleatoria que sea constantemente 1. De esta manera, sea $\beta^T = [\beta_0, \beta_1 \dots \beta_p]$ y una observación $\mathbf{x}_i^T = [1, x_{i1}, \dots x_{ip}]$ la expresión anterior se simplifica de la siguiente manera:

$$\hat{y}_i = f(\mathbf{x}_i) = \mathbf{x}_i^T \beta \tag{1.2.4}$$

Y en general, para una matriz de datos $\mathbf{X}$ de tamaño $n \times p$ al que se le añade una primera columna de 1's por tanto, acaba siendo una matriz de tamaño $n \times (p+1)$ y sea $\hat{\mathbf{y}} = [\hat{y}_1, \dots \hat{y}_p]$ el vector de predicciones. Entonces, se obtiene la siguiente expresión simplificada:

$$\hat{\mathbf{y}} = \mathbf{X}\beta \tag{1.2.5}$$

Los parámetros β son parámetros poblacionales y desconocidos, por ende, se deben estimar conociendo únicamente una muestra de la población general. En las siguientes secciones se detallan la obtención de dichas estimaciones mediante el método de mínimos cuadrados y las propiedades inferenciales que estos estimadores $\hat{\beta}$ tienen.

1.2.1. Ajuste de los parámetros por mínimos cuadrados

Sea una matriz de datos $\mathbf{X}$ de tamaño $n \times p+1$ resultado de hacer n observaciones de p variables aleatorias y añadir a la primera componente de cada observación un 1. Sea también un vector de respuestas $\mathbf{y}^T = [y_1, \dots y_n]$ de tamaño n, resultado de observar la variable respuesta Y.

Tomando la suma de los errores cuadrados y minimizando se puede obtener una estimación de máxima verosimilitud del vector de parámetros β. Utilizando las expresiones anteriores:

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\beta) = \mathbf{y} - \mathbf{X}\beta \tag{1.2.6}$$

Derivando respecto al vector de parámetros β:

$$\frac{\partial RSS(\beta)}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) \tag{1.2.7}$$

Y la segunda derivada respecto de β^T:

$$\frac{\partial^2 RSS(\beta)}{\partial \beta \partial \beta^T} = -2\mathbf{X}^T\mathbf{X} \tag{1.2.8}$$

Asumiendo que $\mathbf{X}$ es una matriz de rango máximo, de lo contrario, habría dos columnas o más que son combinación lineal la una de la otra, la matriz $\mathbf{X}^T\mathbf{X}$ es definida positiva, por tanto la solución de la siguiente ecuación:

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0 \tag{1.2.9}$$

Es un mínimo local de $RSS(\beta)$, si además, $\mathbf{X}^T\mathbf{X}$ es una matriz invertible, entonces tiene solución única y es :

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1.2.10}$$

Proposición 1.2.1. Esta estimación de los parámetros β por mínimos cuadrados es equivalente a la estimación de estos mediante el método de máxima verosimilitud.

Demostración. Sea una muestra de tamaño n, $y_i, i = 1, \dots n$ de una variable aleatoria Y, de manera que su función de probabilidad, $\mathbb{P}_\theta(y)$ depende de los parámetros θ. Entonces el método de máxima verosimilitud busca maximizar la siguiente función.

$$L(\theta) = \sum_{i=1}^{n} log(\mathbb{P}_\theta(y_i)) \tag{1.2.11}$$

Suponiendo que la variable respuesta cumple como antes que $Y = f_\theta(\mathbf{x}) + \varepsilon$, donde $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, provoca que si se suponen conocidos a priori el parámetro θ y el vector aleatorio $\mathbf{x}$ entonces :

$$\mathbb{P}(Y|\mathbf{x}, \theta) = \mathcal{N}(f_\theta(\mathbf{x}), \sigma^2) \tag{1.2.12}$$

Teniendo esto en cuenta, la función $L(\theta)$ tiene la siguiente expresión:

$$L(\theta) = -\frac{n}{2}log(2\pi) - nlog(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2 \tag{1.2.13}$$

Por tanto, teniendo en cuenta que el último término es $RSS(\theta)$, entonces maximizar $L(\theta)$ es equivalente a minimizar $RSS(\theta)$ □

Corolario 1.2.1. Las estimaciones para cualquier f_θ obtenidas por el método de los mínimos cuadrados son equivalentes a las del método de máxima verosimilitud.

Por tanto, los valores predichos $\hat{\mathbf{y}}$ se calculan de la siguiente manera:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1.2.14}$$

Si se toma el espacio $\mathbb{R}^n$ generado por las columnas de la matriz de datos $\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_p$, entonces $\hat{\mathbf{y}}$ es tal que $\mathbf{y} - \hat{\mathbf{y}}$ es ortogonal a ese subespacio.

Inferencias Estadísticas sobre $\hat{\beta}$

Sean las observaciones y_i no correlacionadas y con varianza constante σ^2, sean los $\mathbf{x}_i$ no aleatorios ya conocidos, entonces:

$$Var(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2 \tag{1.2.15}$$

Supóngase además que se sabe que el modelo lineal es correcto, de esta manera se tiene que la media condicional es lineal y tendremos que los datos son de la forma:

$$Y = \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \varepsilon \tag{1.2.16}$$

Teniendo en cuenta las propiedades de ε que se asumían al tomar un modelo aditivo.

Un estimador insesgado que se puede dar de la varianza σ^2 es el siguiente:

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.2.17}$$

Por tanto la distribución de este estimador es:

$$(n-p-1)\hat{\sigma}^2 \sim \sigma^2\chi^2_{n-p-1} \tag{1.2.18}$$

Teniendo que el vector de parámetros estimados es un estimador insesgado tendrá una distribución $\hat{\beta} \sim \mathcal{N}(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2)$.
Teniendo en cuenta lo anterior, la construcción del siguiente estadígrafo de contraste para el parámetro estimado $\hat{\beta}_j$ es relativamente sencilla:

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}} \sim t_{n-p-1} \tag{1.2.19}$$

Donde v_j es el elemento j-ésimo de la matriz $(\mathbf{X}^T\mathbf{X})^{-1}$. Por tanto, podemos generar un intervalo de confianza a un nivel de confianza de $1-2\alpha$

$$(\hat{\beta}_j - z^{(1-\alpha)}v_j\hat{\sigma}, \hat{\beta}_j + z^{(1-\alpha)}v_j\hat{\sigma}) \tag{1.2.20}$$

Esto permite hacer un conjunto de confianza para el vector de parámetros β

$$C_\beta = \{\beta|(\hat{\beta}-\beta)^T\mathbf{X}^T\mathbf{X}(\hat{\beta}-\beta) \leq \hat{\sigma}^2{\chi^2_{p+1}}^{(1-\alpha)}\} \tag{1.2.21}$$

Otro contraste que se puede realizar es comprobar si un subconjunto de variables es más significativo estadísticamente que otro, de manera que se pueda eliminar variables que no aporten información en pos de la sencillez del modelo y su posterior interpretación.

Sea p_1+1 el número del conjunto más grande de parámetros y RSS_1 su error cuadrático respectivo, sean respectivamente RSS_0 y p_0+1 para el segundo conjunto o subconjunto menor entonces el estadigráfo:

$$F = \frac{\dfrac{(RSS_0 - RSS_1)}{p_1 - p_0}}{\dfrac{RSS_1}{N-p_1-1}} \sim F_{(p_1-p_0),(N-p_1-1)} \tag{1.2.22}$$

Este estadígrafo mide si la diferencia entre los errores cuadráticos es lo suficientemente grande para que hacer el cambio de p_1 a p_0 parámetros sea útil.

1.2.2. Regresión Múltiple mediante Ortogonalización sucesiva

Otra manera de afrontar el problema de ajuste es hacerlo de una manera geométrica, se busca un subespacio que se ajuste lo mejor posible a un vector de respuestas, en particular, se buscan proyecciones ortogonales del vector respuesta o del subespacio respuesta en el subespacio de los datos.

Teniendo en cuenta que el vector de respuestas $\mathbf{y}$ sigue el modelo $\mathbf{y} = \mathbf{X}\beta + \varepsilon$, entonces, $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$ pertenece al subespacio generado por las columnas de la matriz de datos $\mathbf{X}$. En caso de que $\mathbf{y} \in Col(\mathbf{X})$, se tiene que existe un β tal que $\mathbf{y} = \mathbf{X}\beta$.

Definición 1.2.1. Se llama vector residuo $\mathbf{r}$ al vector cuyas componentes son $r_i = y_i - \mathbf{x}_i\hat{\beta}$, es decir, $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\beta}$

Para que la distancia entre el vector respuesta y el espacio de datos sea lo menor posible el vector residuo debe ser ortogonal al espacio que generan las columnas de la matriz de datos y esto provoca lo siguiente:

$$\begin{aligned} &r \perp \mathbf{X} \Rightarrow \mathbf{X}^T\mathbf{r} = \mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\beta}) = 0 \\ &\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\hat{\beta} = 0 \Rightarrow \hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \end{aligned} \tag{1.2.23}$$

Por tanto, el vector de estimadores $\hat{\beta}$ es equivalente al cálculo por el método de los mínimos cuadrados. Por lo tanto, teniendo en cuenta que $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. De

esta manera, lo que se está haciendo es tomar la proyección de $\mathbf{y}$ sobre el espacio $Col(\mathbf{X})$.

Teniendo en cuenta esto en *Hastie et.al.* [8] define un algoritmo para calcular el vector de parámetros β. Teniendo en cuenta esto y que $\mathbf{y} = (y_1, \ldots, y_n)^T$ el vector de observaciones de la variable respuesta y $\mathbf{x} = (x_1, \ldots x_n)^T$ el vector de observaciones de una única variable predictora. Entonces se puede expresar el vector de parámetros con el producto escalar:

$$\hat{\beta} = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle} \tag{1.2.24}$$

En el caso general, en el que se tienen p variables predictoras, esto cambia, si todas las columnas de la matriz fuesen ortogonales entre sí cada componente se calcularía de la siguiente manera:

$$\hat{\beta}_j = \frac{\langle \mathbf{x}_j, \mathbf{y} \rangle}{\langle \mathbf{x}_j, \mathbf{x}_j \rangle} \tag{1.2.25}$$

En caso contrario, se deben ortogonalizar los datos, para ello, se utiliza el procedimiento de ortogonalización de Gram-Schmidt

1.2.3. Regresión sobre varias variables

Hasta ahora, se ha considerado una única variable aleatoria respuesta, sea $\mathbf{y}$ el vector aleatorio de variables respuesta $\mathbf{y}^T = [Y_1, \ldots, Y_K]$ y un vector aleatorio de variables de entrada $\mathbf{x}^T = [X_0, X_1, \ldots, X_p]$. Se puede establecer el siguiente modelo análogo:

$$Y_k = \beta_{0k} + \sum_{j=1}^{p} X_j \beta_{jk} + \varepsilon_k = f_k(\mathbf{x}) + \varepsilon_k \tag{1.2.26}$$

Recogidas n muestras tendremos la siguiente expresión matricial:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \tag{1.2.27}$$

Dentro de esa expresión matricial se tiene que:

- $\mathbf{Y}$ es la matriz de tamaño $n \times K$ que contiene los valores observados de las variables respuesta
- $\mathbf{X}$ matriz de datos habitual de tamaño $n \times p + 1$
- $\mathbf{B}$ matriz de tamaño $p + 1 \times K$ que contiene los parámetros de la regresión
- $\mathbf{E}$ es la matriz de tamaño $n \times K$ que contiene los errores.

El proceso de ajuste, en el caso de que los errores $\varepsilon^T = [\varepsilon_1, \ldots \varepsilon_K]$ no estén correlados, de la matriz de parámetros es análogo al de una sola variable respuesta de tal manera que minimizando el error cuadrático acumulado se obtiene

$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. En caso de que los errores tengan una matriz de covarianzas conocida, Σ, entonces es necesario hacer la siguiente modificación en el RSS:

$$RSS(\mathbf{B}, \Sigma) = \sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^T\Sigma^{-1}(y_i - f(\mathbf{x}_i)) \tag{1.2.28}$$

En la última expresión se usa la *distancia de Mahalanobis*.

Definición 1.2.2. Según *Cuadras C.M.*[7] la *distancia de Mahalanobis* entre dos observaciones $\mathbf{x}_i, \mathbf{x}_j$ extraídas de una misma población con matriz de covarianzas Σ se define de la siguiente manera:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T\Sigma^{-1}(\mathbf{x}_i - \mathbf{x}_j)} \tag{1.2.29}$$

En el caso anterior, $(y_i - f(\mathbf{x}_i)) = \varepsilon_i$ y se está haciendo la distancia de Mahalanobis respecto de su media, el 0.

1.2.4. Selección de subconjuntos y métodos penalizados

Anteriormente, se ha detallado un estadígrafo que permitía hacer un contraste sobre la cantidad de variables a considerar en la regresión. Esta reducción de características a considerar permite hacer mucho más interpretable el modelo obtenido durante todo el proceso de ajuste y en algunos casos incluso aumentar la precisión del modelo ya que puede ocurrir que se eliminen variables que introduzcan ruido en el modelo.

Se podría seguir un método exhaustivo que calcule cada uno de los subconjuntos posibles para cada número de variables que creamos necesarias y calcular el error cuadrático acumulado de cada uno de los subconjuntos posibles. Pero este método es de una complejidad computacional alta.

Otra posibilidad sería utilizar el estadígrafo de contraste F definido en la Ecuación (1.2.22) empezando con una sola variable e ir añadiendo cada variable que mejore el ajuste. También se puede hacer al revés, empezando con el modelo con todas las variables e ir reduciendo la cantidad de estas.

El problema de estos métodos de selección de variables es que es un proceso discreto, una variable es o no considerada para el modelo siguiente y esto puede generar sobreajuste o infraajuste, no habiendo un término medio.

Para ello, existen los métodos penalizados o de encogimiento, que añaden un término de penalización para que los parámetros β no sean muy grandes, en el caso del ajuste por mínimos cuadrados del modelo lineal.

$$PRSS(\beta) = \sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{x}_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{1.2.30}$$

El último término hace que parámetros β_j grandes sean considerados perjudiciales, y el parámetro λ es una forma de regular cuanta importancia tiene dicha penalización, cuanto mayor sea, este provocará un encogimiento mayor de los parámetros. De esta manera, se tiene una forma continua de considerar los pesos no eliminando en de manera total las variables o haciendo el β_j correspondiente cercano a 0.

1.3. Métodos de Clasificación y Discriminación

Los métodos de clasificación buscan separar los elementos de un espacio de observaciones en grupos conocidos previamente.

Definición 1.3.1. Llamaremos *discriminación o análisis discriminante* a aquel que busca describir las características principales de cada uno de los grupos o clases mediante *funciones discriminantes*

Definición 1.3.2. Llamaremos *métodos de clasificación* a aquellos que dada una nueva observación, $\mathbf{x}$, buscan predecir con la máxima precisión a que clase pertenecen mediante *reglas de clasificación*

Hay que señalar que no siempre hay una diferencia clara entre ambas disciplinas y que puede haber veces que ambas se solapen.

1.3.1. Análisis Discriminante

Según *Lebart L., Morineau, A. y Warwick K.M.* [16] el análisis discriminante es un conjunto de técnicas que permiten describir y clasificar un gran número de observaciones de las cuales se han medido una gran cantidad de variables.

El método que se describe aquí es un método supervisado ya que se conocen las clases a las que pertenecen cada una de las observaciones.

1.3.2. Formalización del Análisis Discriminante

Sea $\mathbf{X}$ la matriz de datos de tamaño $n \times p$ donde las filas $\mathbf{x}_i$ son cada una de las observaciones de las p variables. Dichas observaciones están particionadas en general por q grupos, sea I_k el conjunto de observaciones pertenecientes al k-ésimo grupo, sea también n_k el número de observaciones que pertenecen al k-ésimo grupo.

Se definen las medias muestrales $\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$ de la j-ésima variable en la población en total. También se define la media muestral dentro de cada grupo que es $\overline{x}_{jk} = \frac{1}{n_k}\sum_{i\in I_k} x_{ij}$

Por ende podemos dar la distancia entre dos variables como:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ij'} - \overline{x}'_j) \tag{1.3.1}$$

Esto se puede particionar por grupos de la siguiente manera:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in I_k}(x_{ij} - \overline{x}_j)(x_{ij'} - \overline{x}'_j) \tag{1.3.2}$$

y a su vez cada uno de los $(x_{ij} - \overline{x}_j)$ se pueden dividir en la parte intergrupos e intragrupos:

$$(x_{ij} - \overline{x}_j) = (x_{ij} - \overline{x}_{jk}) + (\overline{x}_{jk} - \overline{x}_j) \tag{1.3.3}$$

Sustituyendo y simplificando lo necesario:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in I_k}(x_{ij}-\overline{x}_{jk})(x_{ij'}-\overline{x}_{j'k})+\sum_{k=1}^{q}\frac{n_k}{n}(\overline{x}_{jk}-\overline{x}_j)(\overline{x}_{j'k}-\overline{x}_{j'}) \quad (1.3.4)$$

Esto nos permite dar una descomposición de la matriz de covarianzas total de la siguiente forma :

$$\mathbf{T} = \mathbf{B} + \mathbf{W} \quad (1.3.5)$$

Donde:

- $\mathbf{T}$ es la matriz que expresa la covarianza total y sus coeficientes $t_{jj'} = Cov(X_j, X'_j)$
- $\mathbf{B}$ es la matriz que expresa la covarianza entre los grupos y sus coeficientes son $b_{jj'} = \sum_{k=1}^{q}\frac{n_k}{n}(\overline{x}_{jk} - \overline{x}_j)(\overline{x}_{j'k} - \overline{x}_{j'})$
- $\mathbf{W}$ es la matriz que expresa la covarianza dentro de los grupos y sus coeficientes son $w_{jj'} = \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in I_k}(x_{ij} - \overline{x}_{jk})(x_{ij'} - \overline{x}_{j'k})$

Para cualquier combinación lineal que se quiera hacer de las variables de entrada de la forma $\mathbf{a}^T\mathbf{x}$, donde el vector $\mathbf{a}$ es un vector de p constantes, entonces la varianza se transforma de la siguiente manera:

$$Var(\mathbf{a}^T\mathbf{x}) = \mathbf{a}^T\Sigma\mathbf{a} \quad (1.3.6)$$

Entonces transformando por el vector $\mathbf{a}$ tenemos que la Ecuación (1.3.5) se transforma de la siguiente manera:

$$\mathbf{a}^T\mathbf{T}\mathbf{a} = \mathbf{a}^T\mathbf{B}\mathbf{a} + \mathbf{a}^T\mathbf{W}\mathbf{a} \quad (1.3.7)$$

Recopilando, el objetivo del análisis discriminante lineal es encontrar combinaciones lineales que maximicen la varianza entre grupos y minimicen la varianza dentro de los grupos. Eso es equivalente a encontrar el vector $\mathbf{a}$ tal que:

$$f(\mathbf{a}) = \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{T}\mathbf{a}} \quad (1.3.8)$$

Es máxima. Si además utilizamos la restricción $\mathbf{a}^T\mathbf{T}\mathbf{a} = 1$. En principio la función objetiva es homogénea, es decir, $f(\mu\mathbf{a}) = f(\mathbf{a})$ Utilizando el método de los multiplicadores de Lagrange derivamos respecto del vector $\mathbf{a}$ tendremos que:

$$L(\mathbf{a}) = \mathbf{a}^T\mathbf{B}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{T}\mathbf{a} - 1) \quad (1.3.9)$$

al derivarla respecto de $\mathbf{a}$ se obtiene que:

$$\frac{\partial L(\mathbf{a})}{\partial \mathbf{a}} = 2\mathbf{B}\mathbf{a} - 2\lambda\mathbf{T}\mathbf{a} \quad (1.3.10)$$

En consecuencia:

$$\mathbf{Ba} = \lambda \mathbf{Ta} \tag{1.3.11}$$

Si además $\mathbf{T}$ es no singular

$$\mathbf{T}^{-1}\mathbf{Ba} = \lambda \mathbf{a} \tag{1.3.12}$$

Es decir, el vector $\mathbf{a}$ es el vector de valor propio λ, tomando el valor propio máximo de la matriz $\mathbf{T}^{-1}\mathbf{B}$.

Definición 1.3.3. Al valor λ se le conoce como *potencia discriminante* de la combinación $\mathbf{a}$.

Observación Esta técnica se diferencia del Análisis de Componentes Principales en que en el Análisis Discriminante se maximiza la distancia o variación entre grupos conocidos, mientras que el Análisis de Componentes Principales únicamente busca las direcciones en las que los datos varían más.

Pero, en caso de ser aplicada, el análisis discriminante desarrollado aquí se puede utilizar como método de reducción de la dimensionalidad para el caso de la clasificación. De esta manera, utilizando mecanismos similares a los que se describirán en la parte del Análisis de Componentes Principales la matriz de datos puede ser reducida.

1.3.3. Métodos de Clasificación

En la introducción de este capítulo se detalló que el clasificador Bayesiano utilizaba las probabilidades conocido el valor de la observación $\mathbf{x}_0$. El caso de la regresión logística intenta modelizar el cociente de las probabilidades como una función lineal.

Hastie et.al.[8] detallan el caso en profundidad cuando la variable Y tiene dos posibles valores.

$$log\frac{P(Y=1|\mathbf{x}=\mathbf{x}_0)}{P(Y=2|\mathbf{x}=\mathbf{x}_0)} = \beta^T\mathbf{x}_0 \tag{1.3.13}$$

Donde β es un vector de longitud $p+1$ y a $\mathbf{x}_0$ es una observación del vector aleatorio y se le ha añadido en la primera componente el valor constante 1.

Esto sería el caso para 2 valores posibles pero en el caso de que se tuviesen K valores posibles, se definen las siguientes funciones lineales:

$$log\frac{P(Y=j|\mathbf{x}=\mathbf{x}_0)}{P(Y=K|\mathbf{x}=\mathbf{x}_0)} = \beta_j^T\mathbf{x}_0 \quad j = 1\dots K-1 \tag{1.3.14}$$

Por tanto, el modelo queda definido por $K-1$ funciones lineales. Es decir estamos asumiendo que la probabilidad $P(Y=j|\mathbf{x}=\mathbf{x}_0)$ se puede expresar como una función de la siguiente manera:

$$P(Y=j|\mathbf{x}=\mathbf{x}_0) = \frac{e^{\beta_j^T\mathbf{x}_0}}{1+\sum_{k=1}^{K-1} e^{\beta_k^T\mathbf{x}_0}} \quad j = 1\dots K-1 \tag{1.3.15}$$

Además, $P(Y = K|\mathbf{x} = \mathbf{x}_0) = \dfrac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k^T \mathbf{x}_0}}$.

El ajuste de estos modelos se lleva a cabo por el método de máxima verosimilitud. Donde tendríamos que

$$l(\theta) = \sum_{i=1}^{N} log(p_{g_i}(x_i; \theta)) \tag{1.3.16}$$

En el caso de la regresión logística $p_j = P(Y = j|\mathbf{x} = \mathbf{x}_0)$ además supóngase que se tienen únicamente dos clases o dos valores para la variable Y por tanto, la verosimilitud puede ser expresada de la siguiente manera

$$l(\beta) \tag{1.3.17}$$

1.4. Redes Neuronales

Las redes neuronales artificiales, son un algoritmo basado en el funcionamiento de las propias neuronas del cerebro que reciben señales de entradas de las neuronas con las cuales están conectadas, las procesan y envían el resultado a las neuronas con las que estén conectadas.

Una neurona artificial es mucho más simple que una neurona, López R. [15] lo define de la siguiente manera:

Definición 1.4.1. Una neurona procesa una entrada $\mathbf{x}$ de acuerdo con unos pesos sinápticos (b, ω) que luego es transformada por una función de activación $g(u)$, las más habituales son la función sigmoide, una función lineal, la tangente hiperbólica o directamente la identidad.

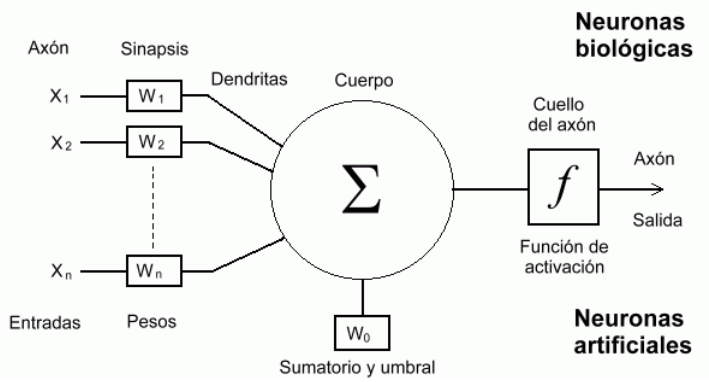
Una vez definidos los elementos que forman una neurona artificial estos se utilizan de manera que se utiliza la función:

$$\begin{aligned} f : \mathbb{R}^p &\longrightarrow \mathbb{R} \\ f(\mathbf{x}) &\longrightarrow f(\mathbf{x}; b, \omega) \end{aligned} \tag{1.4.1}$$

Donde

$$f(\mathbf{x}) = g\left(b + \sum_{i=1}^{p} \omega_i x_i\right) \tag{1.4.2}$$

El siguiente diagrama proporciona una forma sencilla de entender el funcionamiento de dicho modelo, incluyendo la analogía de las neuronas biológicas.



El potencial de este tipo de algoritmos es poder conectar neuronas unas con otras, de manera que la salida que produce una funcione de entrada para otra.

Definición 1.4.2. Se llama capa de neuronas a un conjunto de neuronas que tienen en común las señales de entrada y producen un vector de salidas que pueden o no servir como señal de entrada para otra.

La siguiente imagen es un esquema de una red neuronal con 7 capas de neuronas interconectadas donde la primera capa es de escalado y la última de desescalado. Se puede observar que se predicen las variables y_1, y_2, y_3 usando como entrada las variables x_1, x_2, x_3, x_4

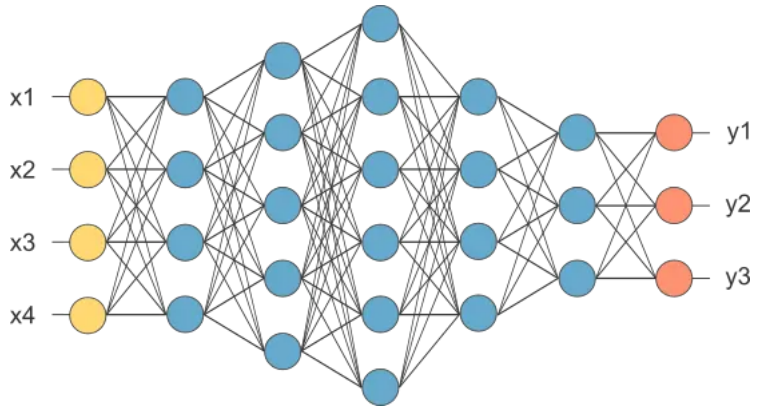


Figura 1.1: Imagen extraída directamente de www.neuraldesigner.com

1.4.1. Proyection Pursuit Regression

Sean un vector aleatorio $\mathbf{x}$ de longitud p, una variable objetivo Y y una familia de vectores de parámetros de longitud p, $\{\omega_m\}_{m=1}^{M}$. Entonces el modelo de Regresión por Búsqueda de Proyecciones *(PPR en inglés)* es de la forma :

$$f(\mathbf{x}) = \sum_{m=1}^{M} g_m(\omega_m^T \mathbf{x}) \tag{1.4.3}$$

En este modelo las funciones g_m no son especificadas y son estimaciones a lo largo de las direcciones de los vectores $\omega_m^T \mathbf{x}$. Según *Hastie, Tibshirani y Friedman* [8] para un M lo suficiente grande y utilizando las g_m apropiadas este modelo puede ayudar a la predicción de cualquier función continua real.

Para ajustar este tipo de modelos tenemos un ajuste en dos pasos, primero se estiman las funciones g_m en función de los parámetros ω_m del paso anterior y luego con las g_m estimadas se optimiza respecto de los parámetros ω_m. Teniendo en cuenta lo siguiente :

- Los métodos de obtención de las g_m deben producir funciones derivables.
- Se pueden ir actualizando tanto las g_m como los ω_m. Aún así, ω_m no se suele actualizar para evitar un gasto computacional excesivo.
- Se puede usar validación cruzada o ir comprobando si hay mejora para estimar la cantidad total de sumandos M.

El uso de este tipo de modelos suele estar restringido así como el de las redes neuronales a la predicción, ya que suelen producir modelos de alta complejidad y difícilmente interpretables.

Sobre este modelo se sustentan las redes neuronales, que establecen previamente las funciones de activación g_m.

En el caso de las redes neuronales, el método de ajuste varía al modelo previamente descrito ya que las funciones están prefijadas.

1.4.2. Red Neuronal de 2 capas

En pos de la sencillez de los resultados, se detallará la de una red neuronal de dos capas. El resto de casos $m \geq 2$, el proceso es análogo. Añadir que la situación es aquella en la que se quiere predecir K variables objetivo a partir de observaciones de p variables.

La primera capa tiene los siguientes elementos:

- Como datos de entrada cada una de las observaciones $\mathbf{x}$ de tamaño p e incluimos en cada uno el término inicial $x_0 = 1$ de tal manera que hay $p + 1$ datos de entrada.
- Un total de M unidades lo que provocará un conjunto $\{z_m\}$ de datos de salida.
- Cada unidad de las M tiene unos pesos α_m de dimensión $p + 1$, donde la primera componente α_{m0} se denomina **sesgo**.
- Una función de activación *(Que puede o no ser lineal)* $g^{(1)}$.

De esta manera, tenemos que los datos de salida de esta primera capa son de la forma:

$$z_m = g^{(1)}(\alpha_m^T \mathbf{x}) \tag{1.4.4}$$

De manera análoga tenemos una segunda capa de neuronas con los siguientes elementos:

- Como datos de entrada los M resultados de la capa anterior, z_m al que añadimos $z_0 = 1$ y denotaremos como el vector $\mathbf{z}$ de longitud $M + 1$.
- Un total de K unidades lo que provocará un conjunto $\{t_k\}$ de datos de salida.
- Cada unidad de las K, tiene unos pesos β_m de dimensión $M + 1$ igual que en la anterior.
- Una función de activación *(Que puede o no ser lineal)* $g^{(2)}$ que puede ser la misma que en la anterior o no.

De esta manera, tenemos que los datos de salida de esta primera capa son de la forma:

$$t_k = g^{(2)}(\beta_k^T \mathbf{z}) \tag{1.4.5}$$

Hay que destacar que los datos deben estar escalados, o pueden serlo mediante una capa que los escale per sé.

Definición 1.4.3. Sea una matriz de datos $\mathbf{X}$ de tamaño $n \times p$, resultado de observar las variables $X_1 \dots X_p$, se llama *capa de escalado* a la capa de neuronas con los siguientes elementos:

- Un vector de pesos con una sola componente igual a 1 y el resto nulas.
- Una función de activación $g(z) = \frac{z - \mu_i}{\sigma_i}$ donde μ_i, σ_i son las media y desviación típica muestrales de cada variable.

El resultado de que la matriz de datos sea procesada por este tipo de capa es una matriz $\mathbf{X}'$ centrada y estandarizada.

1.4.3. Ajuste y uso de una Red Neuronal

Una vez formulado el funcionamiento de una red neuronal, el ajuste de los pesos y sesgos de esta se puede plantear como un problema de minimización de la pérdida en función de los propios pesos, utilizando métodos de optimización numérica como el método del gradiente o el método de Quasi-Newton.

Hay que tener en cuenta para seleccionar el modelo, es decir, el número de neuronas y capas, que las redes neuronales son proclives al sobreajuste. Esto es debido a la gran cantidad de parámetros que se pueden tener en una red con una complejidad media. Teniendo en cuenta esta problemática, se introduce una penalización a términos muy grandes y hace que los pesos sean parecidos a 0.

Una vez ajustados los pesos el modelo resultante puede ser complejo de interpretar, pero proporciona predicciones que potencialmente pueden ser todo lo precisas que se requieran según el número de capas y neuronas que utilicemos.

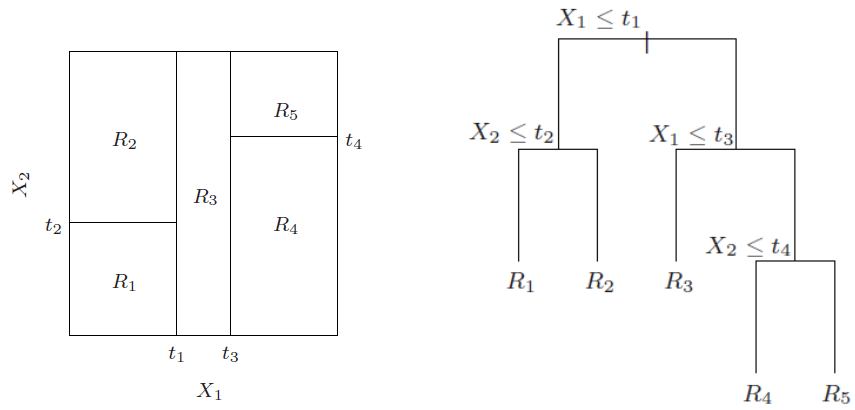
En esta memoria se han detallado los tipos más simples de redes neuronales y de neuronas, con las cuales ya se obtiene una gran capacidad de predicción. Sin embargo, hay estructuras neuronales como las redes convolucionales o las capas LSTM cuya estructura *(Llamadas así por sus siglas en inglés Long-Short Term Memory)* que permiten modelizar sistemas en que los estados futuros son afectados por los estados anteriores como pudiera ser el precio de una acción bursátil o el tiempo atmosférico.

Para evaluar el rendimiento de una red neuronal se pueden utilizar técnicas de validación cruzada o separar el conjunto de entrenamiento en entrenamiento, validación y test. En este proceso, se puede también hacer comparaciones entre varios modelos para elegir el número de neuronas y de capas.

Para mejorar el rendimiento de la propia red neuronal se pueden aplicar técnicas previamente a los datos como el Análisis de Componentes Principales o seleccionar las variables más significativas como se detalla en la sección de regresión que trata de ello.

1.5. Árboles de Decisión y Bosques Aleatorios

Sea un vector aleatorio $\mathbf{x}$ con p los datos de entrada, e Y la variable respuesta. Supóngase también que se toman n observaciones obteniéndose parejas $(\mathbf{x}_i, y_i)$. De esta manera, tenemos que las $\mathbf{x}_i \in \mathbb{R}^p$.
Los métodos de árboles son un método divisivo, ya que tras aplicarlos, se obtiene una partición del espacio de observaciones $\mathbb{R}^p$ y luego en cada región del espacio se ajusta un modelo más simple, incluso una constante.
La ventaja de este tipo de métodos es que son fácilmente interpretables, ya que pueden ser representados mediante un diagrama de tipo árbol. De esta manera, pueden ser utilizados según *Brown et.al.*[4] tanto para análisis exploratorio, detectando características de grandes cantidades de datos, como para tareas de regresión y clasificación.
Las siguientes imágenes procedentes de *Hastie et. al.*[8] muestran el diagrama resultante tras dividir el espacio de observaciones mediante un árbol.



(a) División de $\mathbb{R}^p$ (b) Diagrama resultante

Figura 1.2: Representación de la división de $\mathbb{R}^p$ y el diagrama de árbol resultante

Otra ventaja de este tipo de métodos es que permite trabajar con conjuntos de datos en los que se estudian más variables en comparación de las observaciones, es decir, $p > n$. Por ejemplo, el ejemplo que desarrollan *Díaz-Uriarte y De Andrés* [6], en el que trabajan con un microarray genético.

Otro de los problemas que surgen del propio planteamiento del modelo en sí es que el algoritmo para obtener las divisiones están diseñados para sobreajustarse a los datos por tanto, surgen métodos como la "poda". Se desarrollará más adelante, pero el idea es poder modelizar toda la variación del conjunto de entrenamiento.

1.5.1. Árboles de Regresión

Sea un vector aleatorio $\mathbf{x}$ de variables predictoras como antes y una variable Y de respuesta. Supóngase que queremos dividir el espacio de observaciones en M regiones, $R_1, \dots R_M$, se puede modelar en cada una de las regiones como la constante c_m. Por tanto teniendo en cuenta, la siguiente definición.

$$\mathbf{1}_m(\mathbf{x}) = \begin{cases} 0 & \text{si } x \notin R_m \\ 1 & \text{si } x \in R_m \end{cases} \tag{1.5.1}$$

Es decir, la función $\mathbf{1}(\mathbf{x})$ es la función característica de la región R_m, por tanto puede construirse el siguiente predictor:

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathbf{1}_m(\mathbf{x}) \tag{1.5.2}$$

Un problema que surge es que en principio cualquier tipo de región sería aceptable. Esta libertad llevaría a tener que explorar cualquier región posible, lo cual no es viable computacionalmente hablando. Es por ello, que se toma un tipo de división en particular, las divisiones binarias en semihiperplanos. De esta manera, se generan regiones que son rectángulos de altas dimensiones.

Si se toma como criterio de ajuste los mínimos cuadrados obtendremos que $\hat{c}_m = ave(y_i | x_i \in R_m)$, ya que es la media condicional de la y sabiendo el valor de x.

Para obtener las regiones se toma una variable X_j y un valor de separación s, es decir, cada separación se puede identificar con una pareja (j, s). Por ejemplo, supóngase una separación (j, s) entonces se generan dos regiones del espacio de las observaciones R_1, R_2.

$$R_1 = \{\mathbf{x} | X_j \leq s\} \quad R_2 = \{\mathbf{x} | X_j > s\} \tag{1.5.3}$$

Hay que establecer un criterio para elegir que separaciones se deben hacer en particular, para ello se utilizan los mínimos cuadrados:

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \right] \tag{1.5.4}$$

Es decir, teniendo en cuenta que se va a predecir el valor de la variable respuesta y_i como una constante, la función anterior es la función RSS

Las mejores estimaciones de las constantes según el criterio de los mínimos cuadrados sería la media de las respuestas dentro de las regiones, es decir:

$$\hat{c}_m = \frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} y_i \tag{1.5.5}$$

De esta manera, con estas estimaciones de las constantes de cada región, es posible encontrar la mejor división del espacio de observaciones.

Se puede conseguir la mejor pareja (j, s) fijando la j-ésima variable, encontrar el mejor valor s para cada variable, de manera sencilla.

Definición 1.5.1. Diremos que un algoritmo es *voraz* o *greedy* si en cada paso local busca la solución óptima.

En cada paso, el árbol busca en cada paso la división que mayor disminución produce en el RSS, por tanto, se utiliza un algoritmo voraz.

El siguiente problema es controlar la complejidad del modelo, en particular, cuanto debe crecer el árbol, ya que un árbol demasiado grande puede pecar de problemas de sobre ajuste.

La estrategia más común es tener un criterio de parada, en particular hacer crecer un árbol grande T_0 y "podarlo" cuando en los nodos terminales tenga menos de 5 individuos según *Díaz-Uriarte y De Andrés* [6], además este es el valor por defecto que otorga la librería de `R, randomForest`. Si se hace más grande esto provoca que los árboles sean más pequeños.

Definición 1.5.2. Llamaremos tamaño de un árbol T y se denota por $|T|$ al número de nodos terminales, es decir, al número de regiones en las que se particiona el espacio de observaciones.

Se puede entonces definir el error cuadrático medio de un nodo de un árbol de la siguiente manera:

$$Q_m(T) = \frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \hat{c}_m)^2 \tag{1.5.6}$$

Con la anterior función y el concepto del tamaño del árbol, el cual está directamente relacionado con la complejidad del modelo, se puede establecer un criterio para un árbol que penalice la complejidad:

$$C_\alpha(T) = \sum_{m=1}^{|T|} n_m Q_m(T) + \alpha|T| \tag{1.5.7}$$

El valor de α es un parámetro que es elegido y su valor hace que se penalice más o menos la complejidad del modelo. A valores más altos, tendremos un modelo menos complejo y con $\alpha = 0$ es el modelo obtenido por mínimos cuadrados.

El objetivo es encontrar el árbol de menor tamaño para cada α que minimice la función (1.5.7). Para ello se empieza con el árbol completo o con un árbol inicial T_0 y se van juntando regiones de tal manera que el aumento provoquen sobre la función $\sum_{m=1}^{|T|} n_m Q_m(T)$, sea el menor posible, resultando en una secuencia de árboles entre los cuales se encuentra el que minimiza la función $C_\alpha(T)$ para un α determinado.

Para llevar a cabo el proceso completo, se deben aplicar procesos de validación cruzada, por ejemplo en *Divakaran S.*[19] o en *James G. et.al.* [20] se detalla el algoritmo de Breiman. En este algoritmo se usa la validación cruzada de K-folds, *(Para más detalles sobre dicho proceso, léase la sección 7.10 de Hastie et.al [8] o la sección 5.1.3 James et.al. [20])* para poder elegir el α que minimice el error.

1.5.2. Árboles de Clasificación

Gran parte de lo anterior se puede detallar de manera análoga a los árboles de regresión. En esencia, un árbol de clasificación es un conjunto de funciones discriminantes utilizadas de manera recurrente obteniéndose en el caso óptimo una partición del espacio de observaciones en el que cada región contiene observaciones que pertenecen únicamente a una categoría.

A partir de este punto, la variable respuesta será una variable categórica con valores posibles $1, \ldots, K$.

Sea un nodo m que representa una de las regiones R_m en las que se han particionado el espacio con n_m observaciones en ella. Entonces, la proporción de observaciones que son de la clase k-ésima en el nodo m es:

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} \mathbf{1}(y_i = k) \tag{1.5.8}$$

En cada región R_m se tomará como predicción la respuesta cuya $\hat{p}_{mk}$ sea máxima.

Una vez definido esto se ha de definir el concepto de pureza de un nodo

Definición 1.5.3. Diremos que un nodo es *puro u homogéneo* si todas las observaciones que pertenecen a la región establecida por dicho nodo pertenecen a la misma clase. En caso contrario se dice que el nodo es *impuro*

Es este concepto de la impureza lo que permite establecer un criterio para realizar las divisiones del espacio. Es decir, en cada paso se elegirá la división (j, s) que produzca los nodos más puros.

Teniendo en cuenta la definición que dimos de $\hat{p}_{mk}$ se pueden dar distintas medidas de la impureza de un nodo. Como detallan tanto *Hastie et.al*[8] como *Brown et.al.*[4]

Definición 1.5.4. Sea un nodo y su región correspondiente, R_m, sea además k el valor para el cual $\hat{p}_{mk}$ es mayor que el resto, se define el *Error de Clasificación errónea* como:

$$\frac{1}{n_m} \sum_{\mathbf{x}_i \in R_m} \mathbf{1}(y_i \neq k) = 1 - \hat{p}_{mk} \tag{1.5.9}$$

Es la proporción de observaciones en el nodo que no tienen la misma categoría que el más frecuente. Esta medida no es la más usada debido a que no es derivable y no es muy sensible a los cambios.

Por otro lado, el *índice Gini* no solo tiene en cuenta la proporción máxima o la categoría máxima sino que da una medida de la varianza total entre las K categorías posibles. Lo cual arroja una mejor visión de la pureza de cada nodo.

Definición 1.5.5. Se llama *índice Gini* a la siguiente medida:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{1.5.10}$$

Otra medida alternativa sería la *entropía* del nodo que se define

Definición 1.5.6. Se llama *entropía* de un nodo m o de una región R_m a:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} log(\hat{p}_{mk}) \tag{1.5.11}$$

Ambas medidas son bastante similares.

El algoritmo para construirlos es análogo al de los árboles de regresión. Tomando la medida de impureza o pureza que se desee se puede construir un árbol de clasificación teniendo en cuenta que el error de clasificación produce de manera general según *James et.al.* [20] árboles con una precisión mayor.

Aunque en la mayoría de casos según *Brown et.al*[4] lo que ocurre al utilizar distintas medidas es que los modelos no obtienen errores de predicción muy alejados, sin embargo, pueden provocar que los modelos tengan complejidades muy diferentes.

1.5.3. Bosques Aleatorios

Los bosques aleatorios y otros métodos de juntar árboles de decisión son maneras de solventar ciertos problemas que tienen los árboles de decisión individuales como puede ser la precisión o la dependencia del conjunto de datos, ya que una pequeña variación en estos puede provocar grandes cambios.

Durante esta sección no se detallarán los métodos de construcción de cada árbol individual ya que se han descrito anteriormente y se hará referencia a ellos como árboles simplemente.

Estos métodos se basan en el Bagging y el Bootstrap, ambos métodos se pueden utilizar cuando la cantidad de datos es pequeña o cuando el modelo sufre de *sobreajuste*.

El proceso de Bagging tiene como base el hecho de que si se tienen n observaciones independientes de una población con varianza σ^2 entonces la media de dichas observaciones tendrá varianza $\frac{\sigma^2}{n}$, es decir, tomar la media de distintas observaciones provoca una disminución en la varianza.

De esta manera, partiendo de un conjunto inicial de datos, se toman muestras aleatorias de los mismos y se entrenan distintos modelos para luego hacer la media entre todos los modelos. Es decir, se hace un muestreo aleatorio entre las n observaciones obteniendo B conjuntos de entrenamiento distintos y se estiman los predictores $\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_B(\mathbf{x})$ y el modelo final es:

$$\hat{f}_{bag}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(\mathbf{x}) \tag{1.5.12}$$

Esto se puede aplicar perfectamente al caso de la modelización mediante árboles de decisión aunque esta técnica puede ser útil para cualquier tipo de modelos.

En este caso, estamos utilizando en cada división de los árboles todos los predictores, sin embargo, cuando una variable tiene demasiada importancia a la hora de predecir la variable de respuesta, la mayoría de divisiones en casi todos los árboles

van a utilizar ese predictor, ya que va a provocar mejoras sustanciales en el *RSS* o en el *índice Gini* dependiendo del tipo de árbol.

Para evitar dicho problema, en el algoritmo del bosque aleatorio se toma en cada división no el total de predictores, si no una muestra aleatoria de tamaño m que habitualmente es $m = \sqrt{p}$ de esta manera, se puede tener modelos con mayor variedad.

En *Biau G. y Scornet E.*[2] se detalla el algoritmo para la creación de un bosque aleatorio. En este algoritmo se deben definir los siguientes parámetros:

- $\mathcal{M}$: Número de árboles que se incluirán en el bosque aleatorio.
- $\mathtt{m_{try}}$, es el número de predictores que entre los cuales se pueden usar en cada división del espacio.
- a_n: Número de observaciones del conjunto de entrenamiento que se utilizarán para entrenar cada árbol.

El algoritmo en sí es el siguiente:

---

for $j = 1 \dots \mathcal{M}$ **do:**

- Se extrae de manera uniforme un subconjunto de tamaño a_n del conjunto de observaciones de entrenamiento.
- Se extrae aleatoriamente un conjunto de predictores posibles $\mathcal{M}_{try}$ de tamaño $\mathtt{m_{try}}$ y se construye el árbol, utilizando como posibles predictores únicamente los que están en el conjunto $\mathcal{M}_{try}$

Para hacer la predicción para una nueva observación $\mathbf{x}_0$, hay que calcular la predicción para cada uno de los $\mathcal{M}$ árboles. Una vez calculadas en el caso de que la variable respuesta sea numérica se hace la media de todas las predicciones. Si por el contrario, Y es una variable categórica, se toma como predicción aquella que haya salido con mayor frecuencia, lo que se llama predicción por *voto de mayoría*.

En el artículo de *Biau, G y Scornet, E.* [2] se detallan más en profudidad las distintas características de estos modelos. En particular, se detalla la necesidad de utilizar modelos simplificados para poder analizar las características de los bosques aleatorios. Estos pueden ser los bosques puramente aleatorios que hace particiones totalmente aleatorias sin tener un criterio de acuerdo a los datos y otras modificaciones.

En particular, *Breiman L.* [3], utiliza los siguientes resultados empíricos para poder analizar la consistencia y el ratio de convergencia de los bosques aleatorios utilizando un modelo más simplificado de los mismos.

- El muestreo aleatorio que se hace del conjunto de entrenamiento no tiene efecto en el ratio de error.
- Si para conseguir una partición en la que únicamente en cada nodo haya una sola observación se utiliza la mediana de las variables aleatorias esto no tiene efecto en el ratio de error.

- Hacer divisiones sin tener en cuenta las clases que conocemos, sí aumenta el error.

De esta manera, establece un modelo simplificado en el las variables predictoras se pueden dividir en dos grupos, las variables fuertes $\mathcal{S}$ y las variables débiles $\mathcal{W}$. Las variables fuertes son las que tienen una influencia directa sobre las predicciones y las débiles se pueden interpretar como ruido.

Teniendo en cuenta lo anterior, las variables aleatorias fuertes se dividen en la mediana, mientras que si es débil se toma un valor aleatorio para la división.

Como resultado de dichas modificaciones *Breiman, L.* [3] concluye que el ratio de convergencia es $\mathcal{O}\left(n^{\left(\frac{-0,75}{|\mathcal{S}|log2+0,75}\right)}\right)$.
Por tanto, el ratio de error depende del número de variables fuertes.

Capítulo 2

Métodos no supervisados

2.1. Introducción

2.2. Análisis de Componentes Principales

El análisis de componentes principales fue en primera instancia desarrollado a principios del siglo XX por el estadístico Pearson (1901). Fue una de las primeras técnicas de análisis multivariante. Como muchos otros métodos de este tipo no tuvo un verdadero desarrollo y expansión hasta que la capacidad de computación fue suficiente para manejar cantidades de datos considerables.

2.2.1. Definición y cálculo de las Componentes

Sea un vector aleatorio $\mathbf{x}^T = [X_1, \dots X_p]$ con vector de medias μ y matriz de covarianzas Σ.

Definición 2.2.1. Las componentes principales son combinaciones lineales de las variables $X_1 \dots X_p$

$$\mathbf{z}_j = a_{1j}X_1 + \dots a_{pj}X_p = \mathbf{a}_j^T\mathbf{x} \tag{2.2.1}$$

Donde $\mathbf{a}_j$ es un vector de constantes y la variable $\mathbf{z}_j$ cumple lo siguiente:

- Si $j = 1$ $Var(\mathbf{z}_1)$ es máxima restringido a $\mathbf{a}_1^T\mathbf{a}_1 = 1$
- Si $j > 1$ debe cumplir:
  - $Cov(\mathbf{z}_j, \mathbf{z}_i) = 0 \quad \forall i \neq j$
  - $\mathbf{a}_j^T\mathbf{a}_j = 1$
  - $Var(\mathbf{z}_j)$ es máxima.

De esta manera lo que se busca es una nueva base que reúna las direcciones de máxima variación

El cálculo de la primera componente principal se lleva a cabo con un proceso de optimización de la función $Var(\mathbf{z}_1)$ sujeto a la restricción de que $\mathbf{a}_1^T\mathbf{a}_1 = 1$.

Aplicando el método de los multiplicadores de Lagrange, dada una función $f(\mathbf{x}) = f(x_1, \ldots, x_p)$ diferenciable con una restricción $g(\mathbf{x}) = g(x_1, \ldots, x_p) = c$, existe una constante λ de manera que la ecuación:

$$\frac{\partial f}{\partial x_i} - \lambda \frac{\partial g}{\partial x_i} = 0 \quad i = 1, \ldots p \tag{2.2.2}$$

Tiene como solución los puntos estacionarios de $f(\mathbf{x})$. Además, si se define la función $L(\mathbf{x}) = f(\mathbf{x}) - \lambda[g(\mathbf{x}) - c]$ es posible simplificar la expresión anterior a:

$$\frac{\partial L}{\partial \mathbf{x}} = 0 \tag{2.2.3}$$

Para el caso de las componentes principales, la función objetivo es la varianza de la combinación lineal, es decir, $f(\mathbf{x}) = \mathbf{x}^T\Sigma\mathbf{x}$ y la restricción aplicada es $g(\mathbf{x}) = \mathbf{x}^T\mathbf{x} = 1$.

Tomando $\mathbf{x} = \mathbf{a}_1$ se puede establecer $L(\mathbf{a}_1) = \mathbf{a}_1^T\Sigma\mathbf{a}_1 - \lambda[\mathbf{a}_1^T\mathbf{a}_1 - 1]$. Que al derivarla se obtiene:

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{a}_1} &= 2\Sigma\mathbf{a}_1 - 2\lambda\mathbf{a}_1 \\ &= 2(\Sigma - \lambda)\mathbf{a}_1\end{aligned}$$

Igualando a 0 tenemos la siguiente ecuación:

$$(\Sigma - \lambda I)\mathbf{a}_1 = 0 \tag{2.2.4}$$

Para que $\mathbf{a}_1$ sea un vector no trivial, se elige λ de tal manera que $|\Sigma - \lambda I| = 0$, es decir, λ es un vector propio de la matriz de covarianzas, Σ. Al ser ésta una matriz semidefinido positiva y simétrica, los valores propios son reales y positivos. Por tanto, $\mathbf{a}_1$ es un vector propio de la matriz de covarianza.

La función a maximizar es $Var(\mathbf{z}_1) = Var(\mathbf{a}_1^T\mathbf{x}) = \mathbf{a}_1^T\Sigma\mathbf{a}_1 = \mathbf{a}_1^T\lambda\mathbf{a}_1$, y para maximizarla basta tomar $\lambda = \text{máx}\{\lambda_1 \ldots \lambda_p\}$. reordenando si es necesario, se tiene que $\lambda = \lambda_1$

Una vez calculada la primera componente principal $\mathbf{z}_1$, la segunda componente se calcula de manera análoga, maximizando $Var(\mathbf{z}_2) = Var(\mathbf{a}_2^T\mathbf{x})$ condicionada por $\mathbf{a}_2^T\mathbf{a}_2 = 1$. A esta restricción tenemos que añadir la restricción $Cov(\mathbf{z}_1, \mathbf{z}_2) = 0$

Proposición 2.2.1. La condición $Cov(\mathbf{z}_1, \mathbf{z}_2) = 0$ equivale a la condición $\mathbf{a}_2^T\mathbf{a}_1 = 0$.

Demostración. Utilizando que $\mathbf{z}_j = \mathbf{a}_j^T\mathbf{x} \quad \forall j$, se tiene entonces que :

$$\begin{aligned}Cov(\mathbf{z}_2, \mathbf{z}_1) &= Cov(\mathbf{a}_2^T\mathbf{x}, \mathbf{a}_1^T\mathbf{x}) \\ &= \mathbb{E}(\mathbf{a}_2^T(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\mathbf{a}_1) \\ &= \mathbf{a}_2^T\mathbb{E}((\mathbf{x} - \mu)(\mathbf{x} - \mu)^T)\mathbf{a}_1 \\ &= \mathbf{a}_2^T\Sigma\mathbf{a}_1 \\ &= \mathbf{a}_2^T\lambda_1\mathbf{a}_1\end{aligned}$$

De manera que, si $a_2^T\lambda_1 a_1 = 0 \Rightarrow a_2^T a_1 = 0$, luego son vectores ortogonales entre sí. □

Observación: Esta proposición se puede extender de manera simple al caso de tener que calcular la i-ésima componente principal habiendo calculado las anteriores de las cuales se sepan los valores propios asociados.

Corolario 2.2.1. Las componentes principales son todas ortogonales entre sí.

Para $k = 2$, se dan dos restricciones, $\mathbf{a}_2^T\mathbf{a}_2 = 1$ y además $\mathbf{a}_1^T\mathbf{a}_2 = 0$. Para este caso existen λ, ϕ de manera que la función a maximizar es:

$$L(\mathbf{a}_2) = \mathbf{a}_2^T\Sigma\mathbf{a}_2 - \lambda[\mathbf{a}_2^T\mathbf{a}_2 - 1] - \phi(\mathbf{a}_1^T\mathbf{a}_2) \tag{2.2.5}$$

Que al ser derivado respecto $\mathbf{a}_2$ obtenemos:

$$2\Sigma\mathbf{a}_2 - 2\lambda\mathbf{a}_2 - \phi\mathbf{a}_1 = 0 \tag{2.2.6}$$

Que al multiplicar todo por $\mathbf{a}_1^T$ resulta que $\phi = 0$. De esta manera, se obtiene en la ecuación lo mismo que en el cálculo de la primera.

Por tanto, $\lambda = \lambda_2$ que es el segundo valor propio más grande, y $\mathbf{a}_2$ es el vector propio de valor propio λ_2.

Un proceso similar se puede seguir para calcular el resto de componentes principales.

Por ende, se obtiene que las componentes principales vienen dadas por los vectores propios de la matriz de covarianzas Σ. Además sabemos que $Var(\mathbf{a}_k^T\mathbf{x}) = \lambda_k$ donde λ_k es el k-ésimo valor propio más grande.

Sea ahora la matriz $\mathbf{A}$ cuyas columnas son los $\mathbf{a}_k$. Entonces el vector $\mathbf{z}$ que contiene a las componentes principales viene dado por la transformación:

$$\mathbf{z} = \mathbf{A}^T\mathbf{x} \tag{2.2.7}$$

Se deduce rápidamente que la matriz $\mathbf{A}$ es ortonormal, de manera que $\mathbf{A}^T\mathbf{A} = \mathbf{I}$

2.2.2. PCA en matrices de datos

Sea $\mathbf{x}$ el vector aleatorio de longitud p, tomando n observaciones de ese vector se obtienen $\mathbf{x}_1 \dots \mathbf{x}_n$. Esta recopilación de observaciones nos permite construir la matriz de datos $\mathbf{X}$ cuyas filas son cada una de las observaciones. Esta matriz $\mathbf{X}$ es de tamaño $n \times p$. Para este caso, las componentes principales se definen de manera análoga:

Definición 2.2.2. Dado un un vector aleatorio de longitud p del cual hemos extraído n observaciones se definen las componentes principales como:

$$\widetilde{z}_{ij} = \mathbf{a}_j^T \mathbf{x}_i \tag{2.2.8}$$

Donde $\mathbf{a}_j$ es un vector de constantes de longitud p que cumple lo siguiente:

- Si $j = 1$, entonces $\mathbf{a}_1$ maximiza la varianza de la muestra, es decir máximiza $\frac{1}{n-1}\sum_{i=1}^{n}(\widetilde{z}_{i1} - \overline{z}_1)^2$. Además debe cumplir que $\mathbf{a}_1^T\mathbf{a}_1 = 1$
- Si $j > 1$ debe cumplir:
  - Los vectores $\mathbf{a}_j$ son ortogonales entre sí.
  - $\mathbf{a}_j^T\mathbf{a}_j = 1$
  - La varianza muestral es máxima.

Es decir el factor $\widetilde{z}_{ij}$ es la transformación de la observación j-ésima por la i-ésima componente principal.

Por tanto, el proceso que se detalla para un vector aleatorio $\mathbf{x}$ con matriz de covarianzas Σ se puede extender a este caso en el conocemos la matriz de covarianzas muestrales $\mathbf{S}$.

Con el objetivo de hacer las demostraciones más sencillas y compactas tomaremos la matriz $\mathbf{X}$ como la matriz centrada $\overline{\mathbf{X}}$, es decir:

$$\overline{x}_{ij} = x_{ij} - \overline{x}_j \tag{2.2.9}$$

Donde $\overline{x}_j$ es la media muestral de la j-ésima variable. Esto hace que $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$. Esto permite hablar de los valores y vectores propios de $\mathbf{S}$ y de $\mathbf{X}^T\mathbf{X}$ indistintamente, ya que los vectores propios son los mismos y los valores propios son proporcionales.

2.2.3. Reducción de la dimensionalidad

Uno de los objetivos de las componentes principales es reducir la dimensionalidad de la matriz de datos de tamaño $n \times p$, $\mathbf{X}$. Esta matriz de datos se puede interpretar como un conjunto de puntos del espacio $\mathbb{R}^p$.

La intención final es buscar una proyección sobre una subvariedad de dimensión $m < p$ que reduzca la pérdida de información de la matriz y que brinde una mayor capacidad de interpretación de los datos, ya que en el caso de que $m = 2$ o $m = 3$ se podrán hacer representaciones gráficas de manera sencilla . En virtud de conseguir esto se deben definir los siguientes conceptos:

Definición 2.2.3. Dada una matriz $\mathbf{X} \in \mathbb{M}_{n\times p}(\mathbb{R})$ existe la descomposición en valores singulares *(SVD en inglés)*:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.2.10}$$

Donde:

- $\mathbf{U}$ matriz ortogonal y de tamaño $n \times n$
- Σ matriz de tamaño $n \times p$ diagonal, cuyos elementos no nulos son los valores singulares $\sigma_1 \geq \ldots \geq \sigma_r \geq 0$ que son los valores propios de la matriz $\mathbf{X}^T\mathbf{X}$ y $r = rg(\mathbf{X})$
- $\mathbf{V}$ matriz ortogonal y de tamaño $p \times p$

Proposición 2.2.2. La matriz $\mathbf{V}$ de tamaño $(p \times p)$ es la matriz que contiene los vectores para hacer la combinación lineal que definen las componentes principales.

Demostración. La matriz $\mathbf{X}^T\mathbf{X}$ es la matriz de covarianzas $(p \times p)$ por la descomposición en valores singulares tenemos que:

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= (\mathbf{U}\Sigma\mathbf{V}^T)^T(\mathbf{U}\Sigma\mathbf{V}^T)\\ &= \mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T\\ &= \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T\end{aligned}$$

Donde la matriz $\Sigma^T\Sigma$ es una matriz diagonal de tamaño $p \times p$ cuyos elementos son los cuadrados de los valores singulares de $\mathbf{X}$, que son a su vez los valores propios de $\mathbf{X}^T\mathbf{X}$.

Añadiendo la condición de ortogonalidad de $\mathbf{V} \Rightarrow \mathbf{V}^{-1} = V^T$ es fácil ver que la matriz $\mathbf{V}$ es la matriz cuyas columnas son los vectores propios de $\mathbf{X}^T\mathbf{X}$ □

Corolario 2.2.2. El cálculo de las componentes principales de la matriz de datos $\mathbf{X}$ es equivalente a calcular la descomposición en valores singulares de la misma.

Definición 2.2.4. Sea $\mathbf{A} \in \mathbb{M}_{n\times p}(\mathbb{R})$ definimos la *norma de Frobenius* de la matriz $\mathbf{A}$ como :

$$||\mathbf{A}||_F = (tr(\mathbf{A}^T \cdot \mathbf{A}))^{\frac{1}{2}} = \left(\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2\right)^{\frac{1}{2}} \tag{2.2.11}$$

Proposición 2.2.3. La norma de Frobenius es invariante a transformaciones ortogonales

Demostración. Sea $\mathbf{U}$ una matriz ortogonal, que cumple $\mathbf{U}^T \cdot \mathbf{U} = \mathbf{U} \cdot \mathbf{U}^T = \mathbf{I}$, sea una matriz cualquiera $\mathbf{A}$, entonces:

$$\begin{aligned} ||\mathbf{U} \cdot \mathbf{A}||_F^2 &= tr((\mathbf{UA})^T \cdot (\mathbf{UA})) \\ &= tr((\mathbf{A}^T\mathbf{U}^T) \cdot \mathbf{UA})) \\ &= tr(\mathbf{A}^T\mathbf{A}) \\ &= ||\mathbf{A}||_F^2 \end{aligned}$$

□

Se ha elegido la norma de frobenius se ha elegido por la siguiente propiedad.

Proposición 2.2.4. Dada una matriz de datos $\mathbf{X}$ de tamaño $n \times p$ entonces

$$||\mathbf{X}||_F^2 = (n-1)\sum_{i=1}^{p} s_{ii}^2 \tag{2.2.12}$$

Donde las s_{ii}^2 son las varianzas muestrales.

Demostración. Debido a la centralidad impuesta a la matriz $\mathbf{X}$, sabemos que la matriz de covarianzas es $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$ por tanto,se tiene que utilizar la definición de la norma:

$$\begin{aligned} ||\mathbf{X}||_F^2 &= tr(\mathbf{X}^T\mathbf{X}) \\ &= (n-1)tr(\mathbf{S}) \\ &= (n-1)\sum_{i=1}^{p} s_{ii}^2 \end{aligned}$$

□

Por tanto, la norma de Frobenius da una imagen del tamaño de la matriz de datos en función de la varianza total de los datos, lo que concuerda con la idea de buscar una matriz que aproxime la matriz de datos con la mínima pérdida de variación de los datos.

Definición 2.2.5. Se llama matriz reducida de orden $m \leq p$ de $\mathbf{X}$ y se denota como $\mathbf{X}_m$, a la matriz $n \times p$ resultado de:

$$\mathbf{X}_m = \mathbf{U}_m\Sigma_m\mathbf{V}_m^T \tag{2.2.13}$$

Donde:

- $\mathbf{U}_m$ matriz ortogonal de tamaño $n \times m$, resultado de tomar de $\mathbf{U}$ únicamente la matriz las m primeras columnas.

- Σ_m matriz cuadrada de tamaño m diagonal con los m primeros valores singulares.
- $\mathbf{V}_m$ matriz ortogonal de tamaño $p \times m$ obtenida al tomar las m primeras columnas de $\mathbf{V}$.

Teorema 2.2.1 (De Eckart-Young)**.** Sea $\mathbf{A}$ una matriz de coeficientes reales de tamaño $n \times p$ y rango r entonces se cumple que:

$$||\mathbf{A} - \mathbf{B}||_F \leq ||\mathbf{A} - \mathbf{A}_m||_F \quad \forall \mathbf{B}/rg(\mathbf{B}) = m \leq r \tag{2.2.14}$$

Por tanto, la matriz reducida brinda la mejor aproximación de la matriz de datos teniendo un criterio de aproximación basado en la variación de los datos. En consecuencia se puede

Como conclusión, se puede definir un criterio para elegir el orden de la matriz reducida m. Se puede entonces definir la variación acumulada de la siguiente manera:

$$t_m = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{2.2.15}$$

Donde los λ_i son los valores propios de la matriz $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$.

Cuanto más cercana sea t_m a 1 manteniendo m lo más pequeña posible mejor por que implicaría que con pocas componentes principales podemos "explicar" la mayor parte de la variación de los datos. Según Jollife [10] entre un 0,8 y 0,9 es lo más habitual.

2.3. Análisis Factorial

El Ánalisis de componentes principales no era más que una descomposición de la varianza de las variables observadas, y luego una transformación de las variables mediante transformaciones ortogonales.

De esta manera, no se explora más allá de la varianza la relación de las variables. En el caso del Análisis Factorial el objetivo es encontrar un conjunto de variables nuevas (*que pueden ser combinaciones o transformaciones no lineales*) de las variables iniciales

2.4. Análisis de Clusters

Según [12] el análisis de clusters tiene como objetivo conseguir una clasificación de las observaciones en subconjuntos que tengan sentido de acuerdo al contexto de la investigación. El clustering es un tipo de clasificación de los datos que tiene las siguientes características:

- *Exclusividad:* Una observación no puede pertenecer a más de un cluster a la vez.
- *Es intrínseco:* No hay ninguna etiqueta que permita clasificar las observaciones, son las características de las propias muestras las que servirán como diferenciadores. Lo que hace que sea un método *no supervisado*.

Además de estas características el clustering puede ser jerárquico *(concepto que se desarrollará más adelante)* o particional.
Por otro lado, los algoritmos se pueden clasificar según sean *aglomerativos*, en los que se empieza teniendo cada una de las observaciones y se van formando los clusters hasta tener un solo cluster con todas las muestras. Por el contrario, los *algoritmos divisivos*, empiezas con un solo cluster y se van haciendo subconjuntos del mismo.

Otra clasificación puede venir dada por el modo en el que se fijan en las variables, si es *Monotético*, en cada paso se centra en una variable, mientras que si es *Politético* se utilizan todas las variables a la vez.

Con el objetivo de formalizar lo anterior, de ahora en adelante, sea el caso en el que se han tomado n observaciones $\omega_1, \ldots, \omega_n$, se denota como $\Omega = \{\omega_1, \ldots, \omega_n\} = \{1, \ldots, n\}$ para abreviar.

Crear una clasificación entre las observaciones de Ω es establecer una relación de equivalencia $\mathcal{R}$ sobre Ω. De esta manera, podemos establecer que:

$$\Omega = \bigsqcup_{i=1}^{m} c_i \tag{2.4.1}$$

Donde c_i son las clases de equivalencia de la relación $\mathcal{R}$.

Definición 2.4.1. Se llama *clustering* a la partición que provoca la relación $\mathcal{R}$ y se definen los *clusters* como las clases de equivalencia de dicha relación, $\{c_i\}$.

2.4.1. Estructuración jerárquica de Ω

En un proceso de clustering jerárquico se crea una sucesión de particiones sobre el mismo conjunto donde los sucesivos clusterings se obtienen agrupando clusters.

Para poder formalizar el clustering jerárquico debemos definir el concepto de *jerarquía indexada*

Definición 2.4.2. Una *jerarquía indexada* (C, α) sobre un conjunto Ω, es una colección de clusters $C \subset \mathcal{P}(\Omega)$ y un índice que cumplen:

- *Intersección.* Si $c, c' \in C$ entonces $c \cap c' \in \{c, c', \emptyset\}$. Es decir, dos clusters o están contenido el uno en el otro o son disjuntos.
- *Reunión* Cada cluster se puede caracterizar como la unión de los clusters que contiene. Es decir, $c \in C \Rightarrow c = \cup\{c'/c' \in C, \quad c' \subset c\}$
- El conjunto de todos los clusters es el conjunto total.

El índice es una aplicación $\alpha : C \to \mathbb{R}$ que cumple lo siguiente :

$$\alpha(i) = 0, \forall i \in \Omega \qquad \alpha(c) \leq \alpha(c') \text{ si } c \subset c' \tag{2.4.2}$$

De esta manera, el índice puede medir como de heterogéneos son los clusters, es decir, puede otorgar una medida de lo similares o no que son las observaciones dentro de un mismo cluster.

Proposición 2.4.1. Para todo $x \geq 0$ la relación $\mathcal{R}_x$ sobre Ω:

$$i\mathcal{R}_x j \quad \text{si} \quad i, j \in c \quad \text{siendo} \quad \alpha(c) \leq x \tag{2.4.3}$$

Es una relación de equivalencia.

Es decir, con esta relación se establece que dos elementos son "iguales" cuando hay un cluster de un nivel de heterogeneidad menor que un umbral que los contiene. Es decir, se fija un criterio por el cual dos observaciones son lo suficientemente similares.

Definición 2.4.3. Un espacio ultramétrico es una pareja (Ω, u) donde Ω es un conjunto finito y u una función distancia sobre $\Omega \times \Omega$ que verifica para cada elemento de $i, j, k \in \Omega$

- $u(i, j) \geq u(i, i) = 0$
- $u(i, j) = u(j, i)$
- *Propiedad ultramétrica:* $u(i, j) \leq \sup\{u(i, k), u(j, k)\}$

Se puede ver que toda distancia ultramétrica cumple la desigualdad triangular. Por tanto, todo espacio ultramétrico es métrico. Cómo añadido, juntando los elementos próximos de Ω se mantiene la propiedad ultramétrica.

Teorema 2.4.1. Supongamos un clustering con m clusters de $\Omega = \bigsqcup_{i=1}^{m} c_i$ sobre el que se tiene una distancia ultramétrica u.
Tomando c_i, c_j los dos clusters más cercanos y uniéndolos podemos definir una nueva distancia u' sobre los $m-1$ clusters nuevos.

Demostración. Sea $k \neq i, j$ por la propiedad ultramétrica se tiene que:

$$\begin{aligned} u(i,k) &\leq sup\{u(i,j), u(j,k)\} = u(j,k) \\ u(j,k) &\leq sup\{u(i,j), u(i,k)\} = u(i,k) \end{aligned} \tag{2.4.4}$$

En consecuencia, $u(i,k) = u(j,k)$ y por tanto, $u(c_k, c_i) = u(c_k, c_j)$.
Definiendo de la distancia ultramétrica u':

$$\begin{aligned} u'(c_k, c_i \cup c_j) = u(c_k, c_i) = u(c_k, c_j) \quad k \neq i, j \\ u'(c_a, c_b) = u(c_a, c_b) \quad a, b \neq i, j \end{aligned} \tag{2.4.5}$$

Con esta nueva distancia se cumple la propiedad ultramétrica. Para elementos c_a, c_b con $a, b \neq i, j$ es evidente, pues no hemos cambiado la definición de la distancia para esos clusters. Ahora sea el siguiente caso:

$$\begin{aligned} u'(c_a, c_i \cup c_j) = u(c_a, c_i) &\leq \sup\{u(c_a, c_b), u(c_b, c_i)\} = \\ &= \sup\{u'(c_a, c_b), u'(c_b, c_i \cup c_j)\} \end{aligned} \tag{2.4.6}$$

Por tanto, haciendo uniones entre los elementos más cercanos no modifica la estructura de espacio ultramétrico.

□

Dado estos resultados se puede seguir el siguiente procedimiento para obtener una jerarquía indexada.

Algoritmo fundamental de clasificación

Dado un espacio ultramétrico (Ω, u) utilizando el teorema anterior se pueden ir juntando los clusters más próximos conservamos la distancia ultramétrica.

---

1. Se empieza con el clustering $\Omega = \{1\} \sqcup \ldots \sqcup \{n\}$.
2. Calculamos todas las distancias $u(c_a, c_b)$ y unimos los clusters que más cercanos estén utilizando la misma distancia ultramétrica como en el teorema anterior.
3. Se considera la nueva partición

$$\Omega = \{1\} \sqcup \ldots \sqcup \{i, j\} \sqcup \ldots \sqcup \{n\} \tag{2.4.7}$$

   Cada vez que se une c_i con c_j se define el índice $\alpha(c_i \cup c_j) = u(c_i, c_j)$.

Este procedimiento nos brinda una forma sencilla de dada una distancia ultramétrica construir una jerarquía indexada.

En el caso contrario en el que se disponga de una jerarquía indexada, se puede crear un espacio ultramétrico con la siguiente proposición.

Proposición 2.4.2. Dado un conjunto Ω finito sobre el que tenemos una jerarquía indexada (C, α) podemos establecer sobre Ω una estructura de espacio ultramétrico.

Demostración. Definiendo la distancia

$$u(i,j) = \alpha(c_{ij}) \tag{2.4.8}$$

Donde c_{ij} es el mínimo cluster que contiene a ambos i, j. Sean ahora los clusters mínimos que contienen a $\{i,k\}, \{j,k\}, c_{ik}, c_{jk}$ respectivamente, entonces:

$$c_{ik} \cap c_{jk} \neq \emptyset \tag{2.4.9}$$

Por ser una jerarquía indexada tenemos dos posibilidades de inclusión uno en el otro:

$$\begin{cases} c_{ik} \subset c_{jk} \Rightarrow i,j,k \in c_{jk} \Rightarrow c_{ij} \subset c_{jk} \Rightarrow u(i,j) = \alpha(c_{ij}) \leq u(j,k) = \alpha(c_{jk}) \\ c_{jk} \subset c_{ik} \Rightarrow i,j,k \in c_{ik} \Rightarrow c_{ij} \subset c_{ik} \Rightarrow u(i,j) = \alpha(c_{ij}) \leq u(i,k) = \alpha(c_{ik}) \end{cases} \tag{2.4.10}$$

Por tanto, se obtiene que $u(i,j) \leq \sup\{u(i,k), u(j,k)\}$

□

2.4.2. Algoritmos basados en jerarquías

Sea ahora una matriz de datos $\mathbf{X}$ resultante de hacer n observaciones sobre p variables aleatorias. Sea una distancia δ *(Se detallará más adelante como elegirla en función del tipo de datos que tengamos.)* sobre el espacio de observaciones y Δ la matriz $\Delta = (\delta(i,j))$ de tamaño $n \times n$, la matriz que contiene todas las distancias entre observaciones.

En el caso de que δ sea ultramétrica se puede utilizar el algoritmo fundamental, en el caso contrario hay que realizar una modificación.

Algoritmo de clasificación de observaciones

Lo que se busca en este tipo de algoritmos es convertir δ en una distancia últramétrica, por ende dado el espacio métrico (Ω, δ) el algoritmo es el siguiente:

---

1. Se empieza con el clustering $\Omega = \{1\} \sqcup \ldots \sqcup \{n\}$.
2. Calculamos todas las distancias $u(c_a, c_b)$ y unimos los clusters que más cercanos estén y se define la distancia de un elemento k al cluster de la siguiente manera:
$$\delta(k, \{i,j\}) = f(\delta(i,k), \delta(j,k)) \tag{2.4.11}$$
Donde la función f hace que δ cumpla la propiedad ultramétrica
3. Se considera la nueva partición
$$\Omega = \{1\} \sqcup \ldots \sqcup \{i,j\} \sqcup \ldots \sqcup \{n\} \tag{2.4.12}$$
Cada vez que se une c_i con c_j se define el índice $\alpha(c_i \cup c_j) = \delta'(c_i, c_j)$.

De esta manera se obtiene también una jerarquía indexada (C, α)

Dependiendo de la función f que se elija, se tienen distintos algoritmos:

- Si $f(x, y) = min(x, y)$ se obtiene el algoritmo de *single linkage*.
- Si $f(x, y) = min(x, y)$ se obtiene el algoritmo de *complete linkage*.
- Si $f(x, y) = \frac{x+y}{2}$ se obtiene el algoritmo de *average linkage*.

Es decir, tomando distintas f obtenemos distintos algoritmos. En estos casos, los algoritmos todos son algoritmos aglomerativos.

2.4.3. Elección de las diferencias

Añadir a esto que el mayor problema que conllevan este tipo de algoritmos es elegir y definir una distancia. Sea un vector aleatorio $\mathbf{x}$ de longitud p sobre el que se realizan n observaciones, entonces la forma más habitual de definir la distancia entre dos observaciones es:

$$D(x_i, x_j) = \sum_{k=1}^{p} \omega_k \cdot d_k(x_{ik}, x_{jk}); \quad \sum_{k=1}^{p} \omega_k = 1 \tag{2.4.13}$$

Donde las d_k son las distancias adaptadas al tipo de variable que sea la variable aleatoria X_k. En particular:

***Variables Cuantitativas*:** Basta con que sea una función monótona creciente del valor absoluto de la diferencia entre las dos observaciones. De manera general, se utilizaría la distancia euclídea, pero esta tiene el problema de que no tiene en cuenta las escalas de los datos es por eso que tenemos que estandarizar y centrar los datos para que no haya problemas de escalas. Una medida de distancia que suple esto es la distancia de Mahalanobis.

Definición 2.4.4. Dadas dos muestras u observaciones $\mathbf{x}_i, \mathbf{x}_j$ sacadas de la misma población de un total de n de las cuales se observan p variables aleatorias, con matriz de covarianzas muestrales $\mathbf{S}$. Se define la distancia de Mahalanobis de la siguiente manera:

$$\mathbf{M}^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i, \mathbf{x}_j)^T \mathbf{S}^{-1} (\mathbf{x}_i, \mathbf{x}_j) \tag{2.4.14}$$

Variables Ordinales: Para una variable categórica cuyos M valores tienen un orden se puede hacer la siguiente transformación para tratarlas como una variable cuantitativa:

$$\frac{i - \frac{1}{2}}{M} \quad i = 1 \dots M \tag{2.4.15}$$

Variables Categóricas: En este caso, suponiendo que la variable tiene M posibles valores, lo habitual es definir una matriz $\mathbf{L}$ de tamaño $M \times M$ simétrica , de manera que el elemento $L_{r,r'} = L_{r',r}$ es la distancia entre la categoría r y la r' que habitualmente se suele tomar como 1 y la diagonal igual a 0.

De esta manera, podemos definir de manera general una distancia entre las observaciones.

2.4.4. Algoritmos no jerárquicos

Teniendo en cuenta como deben ser los clusterings el objetivo es conseguir g clusters que provocan una partición exhaustiva del conjunto de observaciones. Sea el espacio métrico (Ω, d) se puede considerar que la *dispersión total de las muestras* dada un clustering con g clusters tiene la siguiente expresión:

$$\mathbf{T} = \mathbf{B} + \mathbf{W} \tag{2.4.16}$$

Donde B es la distancia inter-clusters y W es la distancia intra-cluster del clustering realizado.
Por tanto, de acuerdo a nuestro objetivo un clustering en g clusters debe ser aquel que minimice W y maximice B

El número de posibles clusterings de un mismo conjunto de n observaciones es de $\frac{g^n}{g!}$. Hay muchas de esas posibles particiones que no son óptimas y hay algunas que son mínimos locales. El algoritmo más habitual es el de K-medias o K-medoides.

Algoritmo de K-medias y K-medoides
Este algoritmo requiere que se prefije el número de clusters que se van a utilizar, además de que las variables sean cuantitativas,*(El algoritmo de K-medoides es una modificación de este que soluciona dicha restricción)* .

---

1. Se inicializa con g puntos aleatorios del espacio $\mathbb{R}^p$. Se asigna a cada observación un cluster de acuerdo a cual de los g puntos es más cercano.
2. Se calculan los puntos medios de cada cluster y se reasignan las observaciones de acuerdo a cual está más próximo
3. Se repite el paso anterior, calculando W en cada paso y se detiene el proceso cuando no haya mejora ninguna o se de un criterio de parada.

En este algoritmo ocurre que la distancia al cuadrado de los puntos de cada cluster al centroide disminuye en cada paso debido a como está formulado.

Además del problema de elegir el número de clusters, se da el problema que sólo se puede utilizar con variables cuantitativas. Haciendo una modificación en el algoritmo de K-medias obtenemos el algoritmo de K-medoides.

---

1. Para una asignación de clusters se calculan las distancias entre los puntos de cada cluster. Y se toma de cada cluster el que minimice la suma las distancias del cluster. Que serán los medoides de cada cluster.
2. Se asigna a cada punto el cluster cuyo medoide sea más cercano a dicha observación
3. Se repiten los pasos anteriores hasta que las asignaciones no cambien.

Por tanto, los puntos de referencia de este algoritmo son observaciones, no como las medias que pueden no serlo. Además es mucho más costoso a nivel computacional.

Adicionalmente, este es uno de los algoritmos afectados por la maldición de la dimensionalidad en la que al aumentar la dimensión del espacio de posibles observaciones cada vez es menos denso en información y necesitamos más muestras.

Bibliografía

[1] **Abdi, H., and Williams, L. J. (2010).** *Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics* 2(4), 433–459.

[2] **Biau, G., and Scornet, E. (2016)**. *A random forest guided tour*. Test, 25, 197-227.

[3] **Breiman, L. (2004).** *Consistency for a simple model of random forests.* University of California at Berkeley. Technical Report, 670.

[4] **Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., and Brown, S. D. (2004).** *An introduction to decision tree modeling.* Journal of Chemometrics: A Journal of the Chemometrics Society, 18(6), 275-285.

[5] **Chatfield,C and Collins A.J (1989)**. *Introduction to multivariate analysis*, Chapman and Hall.

[6] **Díaz-Uriarte, R., and Alvarez de Andrés, S. (2006).** *Gene selection and classification of microarray data using random forest.* BMC bioinformatics, 7, 1-13.

[7] **Cuadras, C.M. (2014)**, *Nuevos métodos de Análisis Multivariante*, CMC Editions, Barcelona.

[8] **Hastie, T., Tibshirani, R. and Friedman J. (2001)**, *The Elements of Statistical Learning, Data Mining, Inference and Prediction* Springer

[9] **Hair, J. F., Black, W. C., Tatham, R. L., and Anderson, R. E. (1995).** *Multivariate data analysis with readings.* Prentice-Hall International, Englewood Cliffs, New Yersey.

[10] **Jollife I.T.(1986)**. *Principal Component Analysis*, Springer-Verlag.

[11] **Köppen, M. (2000)** The curse of dimensionality. *5th online world conference on soft computing in industrial applications (WSC5)* (Vol. 1, pp. 4-8).

[12] **Jain, A.K., Richard, C.D., (1988)** *Algorithms for clustering data.* Prentice-Hall, Inc.

[13] *Scikit Learn Documentation, Clustering Methods.* `https://scikit-learn.org/stable/modules/clustering.html#clustering`

[14] Neural Designer Blog `https://www.neuraldesigner.com/`

[15] **Lopez, R., Balsa-Canto, E. and Oñate, E. (2008)** *Neural networks for variational problems in engineering* Int. J. Numer. Meth. Engng., 75 https://doi.org/10.1002/nme.2304 1341-1360.

[16] **Lebart, L., Morineau, A., Warwick, K. M. (1984)**. *Multivariate Descriptive Analysis: Correspondence analysis and related techniques for large matrices*. Wiley.

[17] **Morrison. D.(1976).** *Multivariate statistical methods (2nd ed)*. McGraw-Hill Kogakusha.

[18] **Batista Foguet, J.M. y Martínez Arias, R. (1989)** *Análisis Multivariante: Análisis en Componentes Principales.* Editorial Hispano Europea.

[19] **Divakaran, S. (2022).** *Data Science: Principles and Concepts in Modeling Decision Trees.* Data Science in Agriculture and Natural Resource Management, 55-74.

[20] **James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013)**. *An introduction to statistical learning* New York: Springer.

[21] **Martínez Arias, Rosario.** *El análisis multivariante en la investigación científica.* Madrid: La Muralla, 1999. Print.