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

A Loreto, que me inspiró a no abandonar
mis sueños aunque las tormentas fueran duras.
A Paula, la que me iluminó
en los momentos más oscuros.
A mi familia, que empujó mis aspiraciones
hasta que se convirtieron en realidades.
A Moloka'i, Titania y Pangea que me lo
dieron todo cuando el mundo se paró.

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:*** Con fines predictivos o descriptivos, este tipo de estudios pueden ser usados en aplicaciones médicas, como pueden ser los diagnósticos precoces, aunque también pueden ser utilizados en campos como la sociología, la psicología etc...
- ***Construcción de Modelos y pruebas de hipótesis***: Ya sean modelos predictivos, como lo son en su mayoría las *Redes Neuronales.* También permite a los investigadores contrastar hipótesis de manera simple, dando mecanismos inferenciales.

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] o *James G, Witten D, Hastie T, y Tibshirani R.* [20] enfocan también estos problemas desde el punto del aprendizaje automático, ya sea también supervisado o no. El crecimiento de la disciplina llamada *Ciencia de Datos* ha conseguido que unidos al aprendizaje automático, estos modelos consigan grandes aplicaciones, por ejemplo, las redes neuronales se han conseguido aplicar exitosamente en varios campos.

De manera regular, estos métodos se han usado en campos en los que hay que estudiar fenómenos de gran complejidad como la psicología y sociología en los que se busca tener mejor entendimiento de un suceso y tomar medidas efectivas y todo lo óptimas posibles, *(Véanse [24], [23] y [25])*, o también para detectar factores de riesgo de conductas dañinas como el consumo de drogas etc...

El Análisis Multivariante permite a distintos tipos de investigadores sacar conclusiones de manera sistemática y simple. Además ahora con la cantidad de librerías y paquetes que están disponibles en los distintos lenguajes, es más accesible que nunca el poder hacer un análisis multivariante.

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.

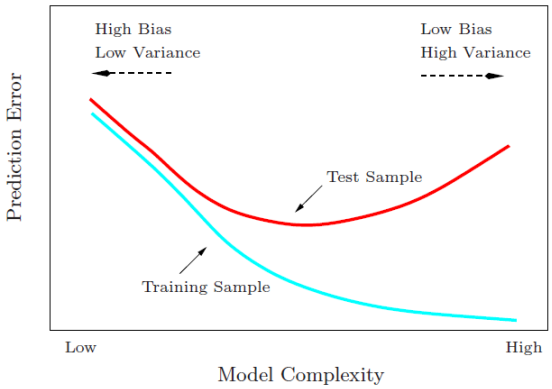
Supóngase que la variable respuesta sigue la siguiente expresión $Y = f(\mathbf{x}) + \varepsilon$ donde $\mathbb{E}(\varepsilon) = 0$ y $Var(\varepsilon) = \sigma^2$, supongamos además que las observaciones $\mathbf{x}_i$ son ya conocidas. Entonces, se puede expresar el error de predicción de una nueva 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 encontrar como un conjunto de variables respuesta se ven afectadas por otro conjunto de variables predictoras. En este caso, se estudia como una combinación lineal de las variables puede afectar a las variables respuesta. Esto se puede hallar con fines predictivos o con el fin de analizar cómo cada una de las variables predictivas afectan a las variables respuesta.

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. Clasificación supervisada

Supóngase un conjunto de observaciones de un vector aleatorio $\mathbf{x}$ de tamaño p las cuales pertenecen a dos o más poblaciones conocidas a priori. El objetivo es dada una nueva observación $\mathbf{x}_0$ poder asignarla a una de las poblaciones. Otro objetivo que se puede buscar con estas técnicas es determinar que variables son determinantes y tienen mayor importancia a la hora de discriminar entre varias categorías.

Este problema se le conoce como *Clasificación Supervisada*, ya que se conoce un conjunto de observaciones de las cuales se sabe a que población pertenece. También se le conoce como *Análisis Discriminante*, ya que se pueden establecer funciones que permitan discriminar observaciones unas de otras.

Alguna de las aplicaciones que utilizan las técnicas que se describen en esta sección son el reconocimiento de patrones, la concesión de créditos, o asignación de textos a autores.

Las primeras aproximaciones que se dieron de estas técnicas fueron dadas por *Fisher R.* en 1936 introduciendo el conjunto de datos *Iris de Fisher* para la clasificación.

Durante esta sección se seguirán *Hastie T. et.al.*[8], *Cuadras C.M.*[7] y *Peña D.* [22]

1.3.1. Formalización

Supóngase dadas dos poblaciones P_1, P_2 de las cuales son conocidas sus funciones de densidad f_1, f_2. Si además se conocen las probabilidades a priori de pertenencia a cada una de las poblaciones π_1, π_2 Entonces la probabilidad conociendo el valor de $\mathbf{x}_0$ de pertencia a $i = 1, 2$ es :

$$P(i|\mathbf{x}_0) = \frac{P(\mathbf{x}_0|i)\pi_i}{\pi_1 P(\mathbf{x}_0|1) + \pi_2 P(\mathbf{x}_0|2)} = \frac{f_i(\mathbf{x}_0)\pi_i}{f_1(\mathbf{x}_0)\pi_1 + f_2(\mathbf{x}_0)\pi_2} \tag{1.3.1}$$

De esta manera se puede clasificar como una población dependiendo de que $f_i(\mathbf{x}_0)\pi_i$ sea mayor *(Esto es generalizable a más poblaciones.)*

Definición 1.3.1. Se llama *función discriminante* aquella que:

$$f_i : \mathbf{\Omega} \longrightarrow \mathbb{R} \tag{1.3.2}$$

Donde Ω es el espacio de observaciones posibles. Definida de tal manera que si $f_i(\mathbf{x}_0) > 0 \Rightarrow \mathbf{x}_0 \in P_i$ y en caso contrario $\mathbf{x}_0 \notin P_i$.

En el caso anterior de dos poblaciones, $f_1(\mathbf{x}_0)\pi_1 - f_2(\mathbf{x}_0)\pi_2$ sería la función discriminante para discernir si $\mathbf{x}_0$ pertenece a P_1. Ya que representa la probabilidad de ser de dicha clase conociendo el valor de las variables predictoras.

Definición 1.3.2. Se llama *Error de clasificación* al coste de clasificar de manera errónea una observación y se denota como $c(i|j) =$ *"Error de clasificar como P_i una observación perteneciente a P_j"*

De esta manera, si el coste de mala clasificación en una y otra población son distintas podemos trasladarlo a la función discriminante de la siguiente manera:

$$\frac{f_1(\mathbf{x}_0)\pi_1}{c(1|2)} - \frac{f_2(\mathbf{x}_0)\pi_2}{c(2|1)} \tag{1.3.3}$$

Por consiguiente, a igualdad de los otros términos escogeremos el que menos coste, el que mayor verosimilitud o el que mayor probabilidad a priori tenga.

Supóngase ahora que tanto f_1, f_2 son densidades normales con medias distintas μ_1, μ_2 pero con una matriz de covarianzas común $\mathbf{\Sigma}$:

$$f_i(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{p}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}} exp\left\{\frac{-1}{2}(\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)\right\} \tag{1.3.4}$$

De esta manera la función discriminante se puede transformar sustituyendo las densidades por la expresión anterior y tomando logaritmos :

$$(\mathbf{x}-\mu_1)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_1)+log\left(\frac{\pi_1}{c(1|2)}\right)-(\mathbf{x}-\mu_2)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_2)-log\left(\frac{\pi_2}{c(2|1)}\right) \tag{1.3.5}$$

Hay que tener en cuenta que el término $D_i^2 = (\mathbf{x}-\mu_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu_i)$ se puede interpretar como la *distancia de Mahalanobis* de la observación $\mathbf{x}$ a la $i-$ésima población. Por otro lado, se tiene un término que relaciona probabilidad de pertenencia con el coste de clasificación errónea. En el caso de que tanto las probabilidades como el coste fueran iguales, entonces la función discriminante para la población 1 sería:

$$D_2^2 > D_1^2 \tag{1.3.6}$$

Definición 1.3.3. Llamaremos *frontera de decisión* al hiperplano que divide el espacio de observaciones en tantas regiones como poblaciones haya. En el caso de una función discriminante como las desarrolladas la frontera son las observaciones tales que $f(\mathbf{x}) = 0$

Para obtener la frontera de decisión se igualan ambos términos y desarrollando ambas distancias podríamos desechar el término $\mathbf{x^T\Sigma^{-1}x}$ ya que es común a ambas, obteniendo la siguiente expresión:

. (1.3.7)

1.3.2. Análisis mediante Variables Canónicas discriminantes

Hasta ahora, se han supuesto conocidas las distribuciones de las poblaciones estudiadas de manera que utilizando las probabilidades de pertenencia y los costes se desarrollan funciones discriminantes de manera sencilla. Pero la realidad pocas veces se ajusta a estos supuestos, es por ello que surgen las técnicas que descomponen la varianza en dos términos la varianza dentro de los grupos y fuera de ellos de tal manera que se busca minimizar la primera y maximizar la segunda.

Es por ello, que a continuación se detallará el método por el cual se va a calcular una nueva base del espacio de observaciones que cumpla lo anterior, minimizar la

variación entre poblaciones y maximizar la variabilidad entre grupos siguiendo el desarrollo dado por *Lebart L., Morineau, A. y Warwick K.M.* [16]

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:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ij'} - \overline{x}'_j) \tag{1.3.8}$$

Esto se puede particionar por grupos de la siguiente manera:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in I_k}(x_{ij} - \overline{x}_j)(x_{ij'} - \overline{x}'_j) \tag{1.3.9}$$

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

$$(x_{ij} - \overline{x}_j) = (x_{ij} - \overline{x}_{jk}) + (\overline{x}_{jk} - \overline{x}_j) \tag{1.3.10}$$

Sustituyendo y simplificando lo necesario:

$$Cov(X_j, X_{j'}) = \frac{1}{n}\sum_{k=1}^{q}\sum_{i\in I_k}(x_{ij} - \overline{x}_{jk})(x_{ij'} - \overline{x}_{j'k}) + \sum_{k=1}^{q}\frac{n_k}{n}(\overline{x}_{jk} - \overline{x}_j)(\overline{x}_{j'k} - \overline{x}_{j'}) \tag{1.3.11}$$

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

$$\mathbf{T} = \mathbf{B} + \mathbf{W} \tag{1.3.12}$$

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:

$$Var(\mathbf{a}^T\mathbf{x}) = \mathbf{a}^T\Sigma\mathbf{a} \tag{1.3.13}$$

Entonces transformando por el vector $\mathbf{a}$ tenemos que la Ecuación (1.3.12) se transforma de la siguiente manera:

$$\mathbf{a}^T\mathbf{T}\mathbf{a} = \mathbf{a}^T\mathbf{B}\mathbf{a} + \mathbf{a}^T\mathbf{W}\mathbf{a} \tag{1.3.14}$$

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:

$$f(\mathbf{a}) = \frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{T}\mathbf{a}} \tag{1.3.15}$$

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:

$$L(\mathbf{a}) = \mathbf{a}^T\mathbf{B}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{T}\mathbf{a} - 1) \tag{1.3.16}$$

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

$$\frac{\partial L(\mathbf{a})}{\partial \mathbf{a}} = 2\mathbf{B}\mathbf{a} - 2\lambda\mathbf{T}\mathbf{a} \tag{1.3.17}$$

En consecuencia:

$$\mathbf{B}\mathbf{a} = \lambda\mathbf{T}\mathbf{a} \tag{1.3.18}$$

Si además $\mathbf{T}$ es no singular

$$\mathbf{T}^{-1}\mathbf{B}\mathbf{a} = \lambda\mathbf{a} \tag{1.3.19}$$

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

Proposición 1.3.1. El desarrollo antes detallado es análogo para $f(\mathbf{a}) = \dfrac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}}$

En el caso de que tomamos el desarrollo para obtener los vectores propios de $\mathbf{W^{-1}B}$ es una matriz con rango $r = \text{mín}(p, q-1)$ entonces podemos utilizar la matriz de cambio de base $\mathbf{U_r}$ que tiene como columnas los vectores propios de la matriz $\mathbf{W^{-1}B}$ y $\mathbf{x}$ el vector a transformar, de esta manera, tenemos que $\mathbf{z} = \mathbf{U_r^T x}$.

A continuación para determinar si una nueva observación $\mathbf{x}_0$ basta con calcular la distancia con las medias de los grupos transformadas, es decir, $||\mathbf{z_0} - \mathbf{U_r}\mu_\mathbf{i}||^2$ y comprobar cual es la menor.

Para determinar cuantas variables canónicas se puede utilizar la varianza entre grupos que explica cada variable canónica.

Proposición 1.3.2. La varianza explicada por cada variable canónica es el valor propio de la matriz $\mathbf{W^{-1}B}$ de su vector propio asociado, su *potencia discriminante*.

Demostración. Sea el vector propio $\mathbf{a}_i$ con valor propio λ_i entonces tendremos que por construcción $\mathbf{a}_i^T\mathbf{W}\mathbf{a}_i = 1$, entonces los vectores son unitarios respecto a la métrica que induce la matriz de covarianzas intra-grupos, entonces la varianza explicada por cada variable es $VE(\mathbf{a}_i) = \mathbf{a}_i^T\mathbf{B}\mathbf{a}_i$, por ser valores propios de $\mathbf{W^{-1}B}$ se cumple que $\mathbf{B}\mathbf{a}_i = \lambda_i\mathbf{W}\mathbf{a}_i$, entonces

$$VE(\mathbf{a}_i)\mathbf{a}_i^T\mathbf{B}\mathbf{a}_i = \lambda_i\mathbf{a}_i^T\mathbf{W}\mathbf{a}_i = \lambda_i \tag{1.3.20}$$

□

Para elegir un número de variables canónicas a usar se puede fijar un umbral por el cual tomemos las m variables cuya varianza explicada acumulada sea superior. De esta manera, se puede llevar a cabo una reducción de la dimensionalidad de los datos de manera parecida a la que se desarrollará en el *Análisis de Componentes Principales.* La principal diferencia con este último es que mientras uno busca representar de una manera simple la variabilidad de los datos, esta busca las direcciones en las cuales los grupos o clases se distinguen de manera más significativa

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.

El algoritmo de aprendizaje que oculta una red neuronal es la optimización de algún funcional, de manera que estas se pueden aplicar a problemas que consistan en la resolución de dichos problemas. La ventaja de este tipo de algoritmos es que en esencia, son un conjunto de parámetros que pueden ser ajustados para cualquier tarea y cualquier tipo de función a aproximar solo hace falta la complejidad del modelo adecuada según *Hornik, K., Stinchcombe, M., y White, H.* [28].

Una neurona artificial es mucho más simple que una neurona, *López R. E. Balsa-Canto y E. Oñate* [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.

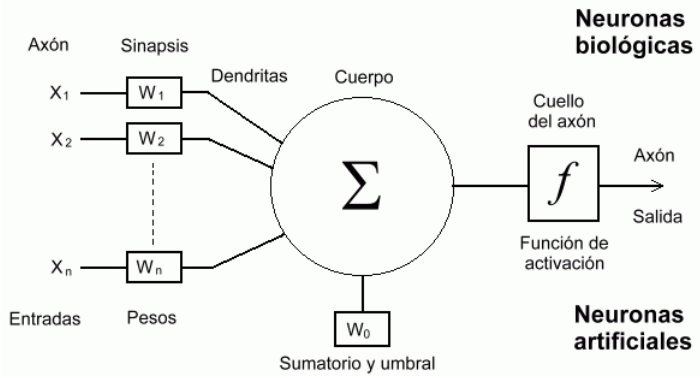
Una vez definidos los elementos que forman una neurona artificial estos se utilizan de manera que se utiliza la función:

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

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

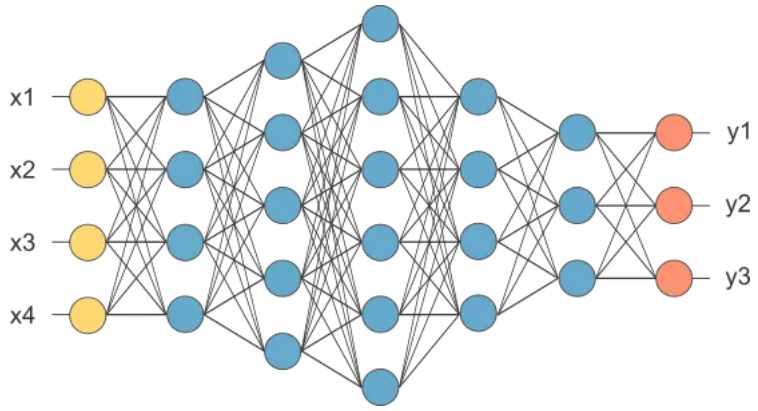


Figura 1.1: Imagen extraída directamente de www.neuraldesigner.com

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.

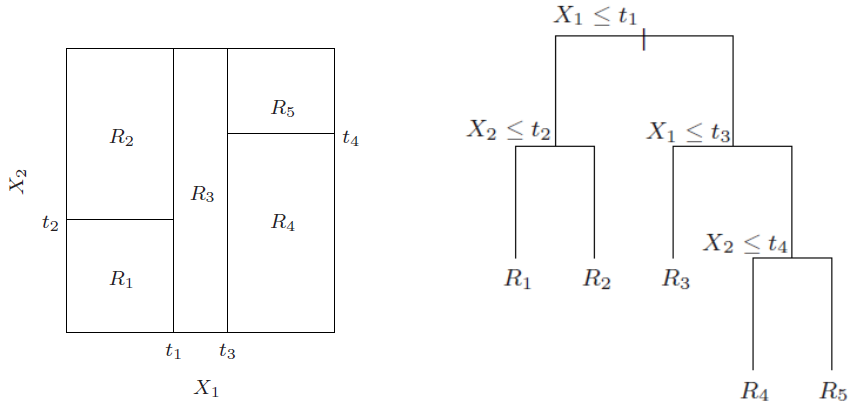
En esta memoria se han detallado los tipos más simples de redes neuronales y de neuronas, con las cuales ya se obtiene una gran capacidad de predicción. Sin embargo, hay estructuras neuronales como las redes convolucionales o las capas LSTM cuya estructura *(Llamadas así por sus siglas en inglés Long-Short 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

Los métodos no supervisados son aquellos en los que no se conoce la variable a estudiar. Es decir, no se busca predecir una variable Y en función de los predictores o detallar la relación entre dos grupos de variables. En los métodos no supervisados se busca hallar información sobre la estructura de los propios datos, unas veces para simplificar dicha estructura o para formar grupos homogéneos de observaciones.

Los métodos no supervisados aunque muy útiles y con un gran abanico de aplicaciones posibles, no permiten dar una medida de cuanto de bien funcionan, donde en los supervisados se tiene la función de pérdida de una manera u otra, en los métodos no supervisados hay que utilizar herramientas más complejas y algunas como el Análisis Factorial, incluso hay que utilizar restricciones para que los métodos de obtención de los estimadores resulten únicos.

Sin embargo esa necesidad de estudiar las estructuras de los datos más allá de la dependencia funcional de unos grupos de variables con otros como pueden ser la mayoría de métodos de regresión que se han expuesto anteriormente es lo que dan a este tipo de métodos multivariantes la importancia que tienen.

En el caso del Análisis de Componentes Principales y el Análisis Factorial, se busca extraer la máxima información de la matriz de covarianzas o de correlaciones para poder dar una simplificación de la estructura de los datos.Sin embargo, en el caso del Análisis de Clusters el objetivo no es simplificar la estructura de los datos si no ver como estos están agrupados y en algunos casos como de homogéneos son. Otros métodos multivariantes no supervisados serían las *Reglas de Asociación* que detallan *Hastie et.al.*[8] en el capítulo 14 que busca asociaciones entre variables.

También hay variaciones de las técnicas que se verán en esta memoria, como Curvas o Superficies Principales o las técnicas de escalado multidimensional, que por falta de tiempo no han sido consideradas.

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.

El Análisis de Componentes Principales, es una de las técnicas que exploran la matriz de covarianzas $\mathbf{\Sigma}$. De esta manera, se persiguen dos objetivos, la reducción del tamaño de los datos con la mínima pérdida de información y la mejora de la interpretabilidad de los mismos.

Normalmente, el *Análisis de Componentes Principales* es un método intermedio que ayuda a facilitar la aplicación de otros muchos como puede ser el *Análisis de Clusters* o la *Regresión Múltiple*.

En esta sección se siguen los cálculos de las componentes de *Chatfield,C y Collins A.J* [5], se toman definiciones de *Cuadras C.M.*[7] y algunos resultados de *Jollife I.T* [10]. Por último, para facilitar la comprensión se ha utilizado *González García, N., y Taborda Londoño, A.*[27]

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 $\mathbf{\Sigma}$.

Definición 2.2.1. *Cuadras C.M.* [7]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

En *Chatfield C. y Collins A.J* [5], el cálculo de la primera componente principal se lleva a cabo mediante un proceso de optimización de la función $Var(\mathbf{z}_1)$ sujeto a la restricción $\mathbf{a}_1^T\mathbf{a}_1 = 1$.

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, $f(\mathbf{x}) = \mathbf{x}^T\mathbf{\Sigma}\mathbf{x}$ y la restricción aplicada es $g(\mathbf{x}) = \mathbf{x}^T\mathbf{x} = 1$.

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

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

Igualando a 0 tenemos la siguiente ecuación:

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

Para que $\mathbf{a}_1$ sea un vector no trivial, se elige λ de tal manera que $|\mathbf{\Sigma} - \lambda I| = 0$, es decir, λ es un vector propio de la matriz de covarianzas, $\mathbf{\Sigma}$. 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 $Var(\mathbf{z}_1) = Var(\mathbf{a}_1^T\mathbf{x}) = \mathbf{a}_1^T\mathbf{\Sigma}\mathbf{a}_1 = \mathbf{a}_1^T\lambda\mathbf{a}_1$, 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 :

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

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:

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

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

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

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 $\mathbf{\Sigma}$. 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 $\mathbf{\Sigma}$ 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)*:

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

Donde:

- $\mathbf{U}$ matriz ortogonal y de tamaño $n \times n$
- $\mathbf{\Sigma}$ 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:

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

Donde la matriz $\mathbf{\Sigma}^T\mathbf{\Sigma}$ 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:

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

Se ha elegido la norma de Frobenius 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:

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

Donde:

- $\mathbf{U}_m$ matriz ortogonal de tamaño $n \times m$, resultado de tomar de $\mathbf{U}$ únicamente la matriz las m primeras columnas.
- $\mathbf{\Sigma}_m$ 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. En el caso del Ánalisis Factorial, se explora la correlación que se da entre las distintas variables, por ejemplo, un conjunto de éstas están muy correladas entre sí y no con el resto, estas pueden ser resumidas por un único factor sin mucha pérdida de información.

De esta manera, tanto el Análisis de Componentes Principales y el Análisis Factorial buscan lo mismo una aproximación de la matriz de covarianzas, aunque este último da un acercamiento más elaborado

El caso que inició este tipo de análisis fue la comprensión de la inteligencia humana donde Pearson y Spearman intentaron dimensionarla con un único factor *"g"*. También otros casos clásicos son la estructura de la personalidad medida con tests y escalas, pero resumida a dos factores.

Como se ha podido observar, el Análisis Factorial tiene su mayor predicación en la psicología y sociología donde los factores ayudan a los investigadores a resumir la información de manera sencilla y concisa. Este Análisis Factorial puede ser confirmatorio o descriptivo como se detallará más adelante.

A lo largo de esta sección se seguirán *Peña, D.;*[22] y *Cuadras C.M*[7], y se tomarán ciertas definiciones de *Chatfield, C. y A.J.;* [5].

2.3.1. Formalización

Según *Peña, D.* [22], la relación que establece el modelo factorial dado un conjunto de n observaciones sobre p variables aleatorias representadas por la matriz $\mathbf{X}$ es la siguiente:

$$\mathbf{X} = \mathbf{1}\mu^T + \mathbf{F}\mathbf{\Lambda}^T + \mathbf{U} \tag{2.3.1}$$

Donde:

- La matriz $\mathbf{F}$ de tamaño $(n \times m)$ obtenida de realizar n observaciones del vector aleatorio $\mathbf{f}$ formado por m variables aleatorias que siguen una distribución normal multivariante $\mathcal{N}_m(\mathbf{0}, \mathbf{I})$, a este vector aleatorio se le llama *vector de factores* que son las llamadas variables latentes.
- $\mathbf{\Lambda}$ es la *matriz de carga* de tamaño $(p \times m)$ que contiene la información de como $\mathbf{f}$ se relaciona con $\mathbf{x}$.
- La matriz $\mathbf{U}$ es la matriz de perturbaciones de tamaño $(n \times p)$ que contiene las n observaciones del vector aleatorio $\mathbf{u}$ cuya información no está recopilada por el vector de factores.

Observación: De esta manera, se establecen los factores como ortogonales o incorrelados, esta condición no es precisamente necesaria pero no se detallará el caso

contrario. A los modelos que no cumplen dicha condición se les llama factores oblicuos y presentan ciertas particularidades a la hora de la estimación.

Por tanto, se puede expresar cada observación $\mathbf{x}_0$, como la suma de $m+2$ términos de la siguiente manera :

$$x_{0j} = \mu_j + \mathbf{\Lambda}_{j1} f_{10} + \ldots \mathbf{\Lambda}_{jm} f_{m0} + u_{0j} \quad j = 1 \ldots p \tag{2.3.2}$$

Para calcular la varianza teniendo en cuenta las distribuciones que se han supuesto. Entonces, se tiene que:

$$E((\mathbf{x} - \mu)\mathbf{f}^T) = \mathbf{\Lambda} E(\mathbf{f}\ \mathbf{f}^T) + E(\mathbf{u}\ \mathbf{f}^T) = \mathbf{\Lambda} \tag{2.3.3}$$

Ya que se ha establecido que los factores son variables incorreladas y con matriz de covarianzas $\mathbf{I}$ y además, los factores son independientes de las perturbaciones. En esta linea, tomando ψ, la matriz de covarianzas del vector aleatorio de perturbaciones, se puede dar la siguiente expresión de la matriz de covarianzas de los datos:

$$\mathbf{\Sigma} = E[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T] = \mathbf{\Lambda} E[\mathbf{f}\ \mathbf{f}^T]\mathbf{\Lambda}^T + E[\mathbf{u}\ \mathbf{u}^T] = \mathbf{\Lambda}\mathbf{\Lambda}^T + \psi \tag{2.3.4}$$

Se detallan en *Chatfield, C y Collins, A.J.* [5] algunos conceptos que ayudan a simplificar la discusión

Definición 2.3.1. Llamaremos *comunalidad* a los efectos de los factores en cada una de las variables

$$h_i^2 = \sum_{j=1}^{m} \lambda_{ij}^2 \tag{2.3.5}$$

Donde $\mathbf{\Lambda} = \{\lambda_{ij}\}$. De esta manera, se puede dar una expresión simplificada de la varianza de cada variable: $\sigma_i^2 = h_i^2 + \psi_i^2$. Es decir, la varianza se divide en lo que explican los factores y lo que no.

Ahora, se plantean varios problemas, entre ellos, ¿Es único este modelo?, ¿ Cuántos factores se deben elegir?, ¿ Cuáles son los efectos de los factores y cómo se calculan ? ¿ Cómo se calculan los factores ?

2.3.2. Unicidad del modelo

Sea el modelo con los vectores aleatorios, $\mathbf{x} = \mu + \mathbf{\Lambda}\mathbf{f} + \mathbf{u}$, entonces tomando una matriz $\mathbf{A}$ no singular el modelo anterior es equivalente a $\mathbf{x} = \mu + \mathbf{\Lambda}\mathbf{A}\mathbf{A}^{-1}\mathbf{f} + \mathbf{u}$, entonces la matriz $\mathbf{\Lambda}' = \mathbf{\Lambda}\mathbf{A}$ y $\mathbf{f}' = \mathbf{A}^{-1}\mathbf{f}$ en este caso tenemos un modelo equivalente con factores que pueden ser correlados y con la misma precisión.

Además si se toma una matriz de transformación ortogonal, $\mathbf{H}$ se obtiene un modelo equivalente en el que los factores son incorrelados, por tanto los modelos son invariantes por rotaciones.

Para resolver este problema de indeterminación se pueden dar dos restricciones:

$$\mathbf{\Lambda}^T\mathbf{\Lambda} = \mathbf{D} \tag{2.3.6}$$

Donde D es una matriz diagonal.

En el caso de que la matriz $\mathbf{\Lambda}^T\mathbf{\Lambda}$ no fuese diagonal, se puede tomar la transformación ortogonal $\mathbf{H} \Rightarrow \mathbf{H}^T = \mathbf{H}^{-1}$ donde las columnas aparte de ser ortogonales, son vectores propios de $\mathbf{\Lambda}^T\mathbf{\Lambda}$ entonces, la matriz $\mathbf{\Lambda H}$ si cumple la condición, ya que :

$$(\mathbf{\Lambda H})^T\mathbf{\Lambda H} = \mathbf{H}^T\mathbf{\Lambda}^T\mathbf{\Lambda H} = \mathbf{H}^{-1}\mathbf{\Lambda}^T\mathbf{\Lambda H} = \mathbf{D} \tag{2.3.7}$$

Corolario 2.3.1. La matriz $\mathbf{H}$ es la única que hace que $\mathbf{\Lambda}$ cumpla la condición

Proposición 2.3.1. Si la matriz $\mathbf{\Lambda}$ cumple lo anterior, $\mathbf{\Lambda}^T\mathbf{\Lambda} = \mathbf{D}$, o en su defecto tomamos su transformada, entonces $\mathbf{\Lambda}$ es la matriz formada por los vectores propios de $\mathbf{V} - \psi$

Demostración. Es decir, en la ecuación de la varianza podemos tener que

$$\begin{aligned} \mathbf{V} &= \mathbf{\Lambda}^T\mathbf{\Lambda} + \psi \\ (\mathbf{\Sigma} - \psi) &= \mathbf{\Lambda}^T\mathbf{\Lambda} \\ (\mathbf{\Sigma} - \psi)\mathbf{\Lambda} &= \mathbf{\Lambda D} \end{aligned}$$

□

De esta manera, se restringen, las posibles transformaciones de la matriz, quedando una única transformación a realizar.

Otra forma de restringir el modelo sería estableciendo la siguiente restricción:

$$\mathbf{\Lambda}^T\psi^{-1}\mathbf{\Lambda} = \mathbf{D} \tag{2.3.8}$$

Proposición 2.3.2. La restricción anterior es válida y hace que el modelo sea único.

Demostración. En el caso que $\mathbf{\Lambda}^T\psi^{-1}\mathbf{\Lambda}$ no sea diagonal, tomando la matriz ortogonal $\mathbf{H}$ que tenga por columnas los vectores propios de la matriz $\mathbf{\Lambda}^T\psi^{-1}\mathbf{\Lambda}$ y transformando de forma que $\mathbf{\Lambda}' = \mathbf{\Lambda H}$ se obtiene un modelo equivalente y que además cumple la restricción anterior y de manera análoga a la otra restricción, $\mathbf{\Lambda}^T\mathbf{\Lambda} = \mathbf{D}$, se puede demostrar que es la única ya que ninguna otra resultaría en una matriz diagonal. □

Proposición 2.3.3. La matriz $\psi^{-\frac{1}{2}}\mathbf{\Sigma}\psi^{-\frac{1}{2}}$ tiene como vectores propios $\psi^{-\frac{1}{2}}\mathbf{\Lambda}$ y los valores propios $\mathbf{D} + \mathbf{I}$

Demostración. Basta con a la ecuación $\mathbf{\Sigma} = \mathbf{\Lambda}^T\mathbf{\Lambda} + \psi$ multiplicarla de la siguiente manera por $\psi^{-1}\mathbf{\Lambda}$:

$$\mathbf{\Sigma}\psi^{-1}\mathbf{\Lambda} + \mathbf{\Lambda} = \mathbf{\Lambda D} \tag{2.3.9}$$

multiplicando por la izquierda por $\psi^{-\frac{1}{2}}$:

$$\psi^{-\frac{1}{2}}\mathbf{\Sigma}\psi^{-\frac{1}{2}}\psi^{-\frac{1}{2}}\mathbf{\Lambda} = \psi^{-\frac{1}{2}}\mathbf{\Lambda(D+I)} \tag{2.3.10}$$

Por tanto, $\psi^{-\frac{1}{2}}\mathbf{\Sigma}\psi^{-\frac{1}{2}}$ es la matriz cuyos valores propios son los elementos de $\mathbf{D+I}$ y los vectores propios son las columnas de la matriz $\psi^{-\frac{1}{2}}\mathbf{\Lambda}$. □

Esta restricción se utiliza en el método máximo verosímil de estimación de los factores.

En estos casos lo que se quiere hallar son los términos de la matriz $\mathbf{\Lambda}$ que es un total de pm y los p términos de la matriz ψ, suponiendo que se da la restricción de que $\mathbf{\Lambda}^T\psi^{-1}\mathbf{\Lambda}$, entonces hay $\frac{m(m-1)}{2}$ restricciones. Además usando la matriz de covarianzas muestrales $\mathbf{S}$, obtenemos $\frac{p(p+1)}{2}$ ecuaciones de manera que si se desea que el sistema quede determinado la condición necesaria es que se de la siguiente condición:

$$\begin{aligned} p + pm - \frac{m(m-1)}{2} &\leq \frac{p(p+1)}{2} \\ (p-m)^2 &\geq p + m \end{aligned} \tag{2.3.11}$$

Teniendo en cuenta estos conceptos, hay que desarrollar métodos para estimar los parámetros desconocidos, los más comunes son el método del Factor Principal y el método de Máxima Verosimilitud.

2.3.3. Método del Factor Principal

En el Análisis Factorial hay un principio que se debe seguir:

Definición 2.3.2. El *Principio de Parsimonia* es aquel por el cual se toma la solución con menos factores comunes con el fin de obtener el modelo menos complejo.

En el caso del método del Factor Principal se busca lo mismo que en el caso de una Componente Principal, que este tenga una varianza máxima y que esté incorrelada. En *Cuadras C.M.*[7] se detalla el proceso de obtención de la descomposición que detallaremos a continuación, esta es similar a la realizada en la sección de las Componentes Principales.

Sea $\mathbf{S}$ la matriz de covarianzas muestrales, de manera que si tenemos en cuenta estimaciones de la matriz de covarianzas de las perturbaciones $\hat{\psi}$, entonces:

$$\mathbf{S} - \hat{\psi} = \mathbf{\Lambda}^T\mathbf{\Lambda} \tag{2.3.12}$$

Entonces la matriz $\mathbf{S} - \psi$, es simétrica ya que la propia $\mathbf{S}$ lo es y la matriz $\hat{\psi}$ es diagonal por hipótesis del modelo, *Cuadras C.M.* [7]. Teniendo en cuenta estas consideraciones se obtiene la descomposición:

$$\mathbf{S} - \psi = (\mathbf{H}\mathbf{G}^{\frac{1}{2}})(\mathbf{H}\mathbf{G}^{\frac{1}{2}})^T \tag{2.3.13}$$

Donde $\mathbf{H}$ es una matriz de tamaño $p \times p$ cuadrada y ortogonal y $\mathbf{G}$ es una matriz con la siguiente distribución:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{1m\times m} & \mathbf{0}_{m\times(p-m)} \\ \mathbf{0}_{(p-m)\times m} & \mathbf{0}_{(p-m)\times(p-m)} \end{pmatrix} \tag{2.3.14}$$

Donde m es el rango de la matriz $\mathbf{S} - \psi$. De manera análoga al caso de las Componentes Principales podemos tener una reducción de la matriz de rango m con

la mínima pérdida de variabilidad *(Teorema de Eckart-Young)*. Tomando $\mathbf{H}_1$ la matriz ortogonal de tamaño $p \times m$ con los vectores propios con valores propios no nulos de $\mathbf{S} - \psi$ y $\mathbf{G}_1$ es la matriz diagonal cuadrada de orden m cuyos elementos de la diagonal son los valores propios no nulos. Entonces, tomando $\hat{\mathbf{\Lambda}} = \mathbf{H}_1\mathbf{G}_1^{\frac{1}{2}}$

Proposición 2.3.4. La matriz $\hat{\mathbf{\Lambda}}$ cumple la condición (2.3.6)

Demostración. Tomando la definición dada de la matriz $\hat{\mathbf{\Lambda}} = \mathbf{H}_1\mathbf{G}_1^{\frac{1}{2}}$. Se tiene:

$$\hat{\mathbf{\Lambda}}^T\hat{\mathbf{\Lambda}} = (\mathbf{H}_1\mathbf{G}_1^{\frac{1}{2}})^T(\mathbf{H}_1\mathbf{G}_1^{\frac{1}{2}}) = \mathbf{G}_1^{\frac{1}{2}}\mathbf{H}_1^T\mathbf{H}_1\mathbf{G}_1^{\frac{1}{2}} = \mathbf{G}_1 \tag{2.3.15}$$

Donde al ser $\mathbf{H}_1$ ortogonal, $\mathbf{H}_1^T\mathbf{H}_1 = \mathbf{I}$. Y como $\mathbf{G}_1$ es una matriz diagonal, se cumple (2.3.6) □

Ahora hay que ser capaz de ofrecer una estimación de las comunalidades.

Las comunalidades son la suma de la variación de los datos que explicaban los factores comunes para cada una de las variables, es decir, se sabe que $s_i^2 = h_i^2 + \hat{\psi}_i^2$

A la hora de tomar una estimación, se pueden asumir de manera inicial como $\hat{\psi}_i^2 = 0$, es decir, se asume que $s_i^2 = \hat{h}_i^2$ de manera que toda la variabilidad de los datos está explicada por la factores, lo cual no siempre es adecuado por que se puede obtener una estimación sesgada.

Otra opción sería tomar $\hat{\psi}_j^2 = \dfrac{1}{s_{jj}^*}$ donde s_{jj}^* es el j-ésimo elemento de la diagonal de la matriz $\mathbf{S}^{-1}$. De esta manera lo que se busca es que la comunalidad $\hat{h}_j^2$ sea la variabilidad que no es común al resto de variables, es decir:

$$\hat{h}_j^2 = s_j^2 - s_j^2(1 - R_j^2) = s_j^2R_j^2 \tag{2.3.16}$$

De esta manera se tienen dos maneras de estimar de manera inicial las comunalidades para poder realizar el siguiente método:

1. Se toma una estimación inicial de las comunalidades como se ha visto anteriormente, en el caso de que sea la primera iteración, $\hat{\psi}_0$ se toma cualquiera de las alternativas, en caso contrario $\hat{\psi}_i = diag(\mathbf{S} - \hat{\mathbf{\Lambda}}_i)$

2. Se calcula la matriz simétrica $\mathbf{Q}_i = \mathbf{S} - \hat{\psi}_i$

3. Se obtiene la descomposición espectral de $\mathbf{Q}_i = \mathbf{H}_{1i}\mathbf{G}_{1i}\mathbf{H}_{1i}^T + \mathbf{H}_{2i}\mathbf{G}_{2i}\mathbf{H}_{2i}^T$. Donde la matriz $\mathbf{H}_{1i}$ son los m vectores propios con los mayores valores propios y $\mathbf{G}_{1i}$ es la matriz que contiene en la diagonal estos m mayores valores propios. Puede ocurrir que $\mathbf{Q}_i$ sea no definida positiva y por tanto, haya valores propios negativos.

4. Tomar $\hat{\mathbf{\Lambda}}_{i+1}\mathbf{H}_{1i}\mathbf{G}_{1i}^{\frac{1}{2}}$ e iterar hasta que se de el criterio de parada o de convergencia $||\hat{\mathbf{\Lambda}}_{i+1} - \hat{\mathbf{\Lambda}}_i|| \leq \varepsilon$.

En *Peña, D*[22] se muestran ejemplos de aplicación simples en el que se utiliza el método del factor principal para realizar la estimación de la matriz de cargas de un solo factor.

En esencia lo que se realiza con este algoritmo es minimizar la siguiente función:

$$F = tr(\mathbf{S} - \mathbf{\Lambda^T\Lambda} - \psi)^2 \tag{2.3.17}$$

A su vez es equivalente a minimizar lo siguiente:

$$F = \sum_{i=1}^{p}\sum_{j=1}^{p}(s_{ij} - v_{ij})^2 \tag{2.3.18}$$

Donde $\mathbf{V} = \mathbf{\Lambda\Lambda^T} + \psi$, de manera que la mejor aproximación a la matriz de covarianzas de rango m es como en el caso análisis de Componentes Principales, $\mathbf{A} = \mathbf{HD}^{\frac{1}{2}}$, donde como en el método anterior, donde $\mathbf{H}$ contiene los m vectores propios con mayores valores propios de $\mathbf{A}^T\mathbf{A}$ y $\mathbf{D}^{\frac{1}{2}}$ la matriz diagonal con las raíces de los m valores propios más altos.

2.3.4. Método Máximo Verosímil

Otra forma de estimar los parámetros es utilizando ecuaciones de Máxima Verosimilitud, en este caso lo que se busca estimar es la matriz de covarianzas de una distribución de datos que en principio es $\mathcal{N}_p(\mu, \mathbf{\Sigma})$ que tomando el estimador de la media por la media muestral obtenemos que la función logarítmica es:

$$log(\mathbf{\Sigma}|\mathbf{X}) = -\frac{n}{2}log|\mathbf{\Sigma}| - \frac{n}{2}tr(\mathbf{S\Sigma}^{-1}) \tag{2.3.19}$$

Sustituyendo $\mathbf{\Sigma}$ por el modelo en el que $\mathbf{\Sigma} = \mathbf{\Lambda^T\Lambda} + \psi$

$$L(\mathbf{\Lambda}, \psi) = -\frac{n}{2}(log|\mathbf{\Lambda^T\Lambda} + \psi| + tr(\mathbf{S}(\mathbf{\Lambda^T\Lambda} + \psi)^{-1})) \tag{2.3.20}$$

Ahora el objetivo es maximizar esta función respecto de $\mathbf{\Lambda}$ y de ψ como en el método univariante de Máxima Verosimilitud. *Peña D.*[22] obtiene las siguientes expresiones una vez se ha derivado respecto a las matrices $\mathbf{\Lambda}, \psi$

$$\hat{\psi} = diag(\mathbf{S} - \hat{\mathbf{\Lambda}}\hat{\mathbf{\Lambda}}^T) \tag{2.3.21}$$

$$(\hat{\psi}^{\frac{-1}{2}}(\mathbf{S} - \mathbf{I})\hat{\psi}^{\frac{-1}{2}})(\hat{\psi}^{\frac{-1}{2}}\hat{\mathbf{\Lambda}}) = (\hat{\psi}^{\frac{-1}{2}}\hat{\mathbf{\Lambda}})\mathbf{D} \tag{2.3.22}$$

De esta manera la matriz $(\hat{\psi}^{\frac{-1}{2}}\hat{\mathbf{\Lambda}})$ es la matriz que tiene como columnas los vectores propios de la matriz $(\hat{\psi}^{\frac{-1}{2}}(\mathbf{S}-\mathbf{I})\hat{\psi}^{\frac{-1}{2}})$ y plantea ecuaciones que pueden resolverse de manera iterativa, para ello, *Peña, D*[22] propone el siguiente algoritmo iterativo:

Se calcula una estimación inicial de la matriz de cargas $\hat{\mathbf{\Lambda}}_0$ que puede ser calculada mediante el método del factor principal

1. Una vez se tiene $\hat{\mathbf{\Lambda}}_i$ se calcula $\hat{\psi}_i$ mediante la ecuación $\hat{\psi}_i = diag(\mathbf{S}_i - \hat{\Lambda}_i\hat{\Lambda}_i^T)$
2. Se calcula la matriz $\mathbf{A}_i = (\hat{\psi}^{\frac{-1}{2}}(\mathbf{S} - \hat{\psi}_i)\hat{\psi}^{\frac{-1}{2}}) = \hat{\psi}^{\frac{-1}{2}}\mathbf{S}\hat{\psi}^{\frac{-1}{2}} - \mathbf{I}$ de manera que se ponderan las covarianzas en función de su componente específica.
3. Se calcula la descomposición espectral de $\mathbf{A}_i = \mathbf{H}_{1i}\mathbf{G}_{1i}\mathbf{H}_{1i}^T + \mathbf{H}_{2i}\mathbf{G}_{2i}\mathbf{H}_{2i}^T$ de manera análoga al método del factor principal. Y por tanto, la matriz $\mathbf{G}_{1i}$ contiene los m valores propios mayores, de la matriz.
4. Se toma $\hat{\mathbf{\Lambda}}_{i+1} = \hat{\psi}^{\frac{1}{2}}\mathbf{H}_{1i}\mathbf{G}_{1i}^{\frac{1}{2}}$ se sustituye en la función de verosimilitud y se maximiza respecto a ψ, después se vuelve al segundo paso hasta llegar al criterio de parada.

Como se puede observar, el método de máxima verosimilitud se desarrolla de manera bastante similar al del factor principal, pero tiene una complejidad mucho más alta, aunque tiene varias propiedades interesantes que se detallan en la siguiente proposición.

Proposición 2.3.5. La estimación Máximo Verosímil es invariante por transformaciones lineales.

Demostración. Sea $\mathbf{D}$ una matriz diagonal cualquiera, de esta manera, se tiene la transformación lineal $\mathbf{y} = \mathbf{DX}$ de los datos, entonces la matriz de covarianzas muestrales $\mathbf{S}_y = \mathbf{DS}_x\mathbf{D}$, por ejemplo, podría ser la matriz que contiene las desviaciones estándar, por lo que se transformaría $\mathbf{S}$ en la matriz de correlaciones. De manera análoga la parte específica de la varianza se transforma $\psi_y = \mathbf{D}\psi_x\mathbf{D}$

De esta manera, la matriz de la cual se calcula la descomposición espectral

$$\hat{\psi}_y^{\frac{-1}{2}}\mathbf{S}_y\hat{\psi}_y^{\frac{-1}{2}} = (\mathbf{D}\psi_x\mathbf{D})^{\frac{-1}{2}}\mathbf{DS}_y\mathbf{D}(\mathbf{D}\psi_x\mathbf{D})^{\frac{-1}{2}} = \hat{\psi}_x^{\frac{-1}{2}}\mathbf{S}_x\hat{\psi}_x^{\frac{-1}{2}} \tag{2.3.23}$$

□

De esta manera, la ventaja de estos métodos de máxima verosimilitud sale a la luz, que es poder trabajar con variables estandarizadas o no obteniendo los mismos resultados. Lo que facilita el trabajo a la hora de preprocesar los datos y evita fallos a la hora de la interpretación.

Aún así se han buscado métodos alternativos para calcular la matriz de cargas, por ejemplo, el método de Mínimos Cuadrados Generalizados que aplica regresiones múltiples para estimar la matriz de cargas *Veasé la sección 12.4.2 de Peña D.*[22]

Otras consideraciones

Sobre este modelo se puede realizar contrastes de hipótesis para poder analizar la conveniencia del propio modelo obtenido. Además se pueden aplicar transformaciones ortogonales que nos permitan tener las máximas variaciones entre los factores, el llamado método *Varimax*. Incluso existen maneras de dar modelos factoriales los cuales estén correlados, algo que aquí ha sido obviado de manera deliberada por falta de tiempo.

2.4. Análisis de Clusters

Según *Jain,A.K. y Richard, C.D;* [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*.

Una vez clarificado las características esenciales del clustering, este se puede dar de manera *jerárquica* o *particional*.

Definición 2.4.1. Se dice que un método de clustering es *jerárquico* si genera una secuencia de particiones del espacio de observaciones las cuales están anidadas

Definición 2.4.2. Un método de clustering *particional* genera una única partición del espacio.

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*, se empieza con un solo cluster y se van haciendo subconjuntos del mismo.

Otra clasificación puede venir dada por el modo en el que se consideran 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.

También hay que distinguir métodos que usan como fundamento el álgebra matricial, centrándose en los datos que nos puede dar la matriz de distancias entre observaciones o en la Teoría de Grafos.

En primer lugar, es necesario definir los conceptos principales de esta sección que es un clustering y un cluster:

Definición 2.4.3. 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, $\{c_i\}$.

Por tanto, el análisis de clusters es equivalente a establecer una relación de equivalencia entre las observaciones.

2.4.1. Algoritmos Jerárquicos

Los algoritmos jerárquicos son aquellos que utilizan como entrada no la matriz de los datos per sé, en su lugar, estas utilizan como entrada la matriz de distancias o similaridad.

Uno de los inconvenientes mayores de este tipo de algoritmos es como calcular la matriz de similaridades o distancias. En el caso de que las variables sean numéricas y continuas entonces se puede utilizar la distancia euclídea entre elementos de $\mathbb{R}^p$. Además hay que tener en cuenta si se va a estandarizar ya que en el caso de que si se estandarice se dará el mismo peso a todas las variables, dando igual su escala, en cambio si no se estandariza, la escala provoca que las características con mayores valores tengan mayor peso en las distancias. Por otro lado, si se tienen variables binarias o categóricas hay que tener en cuenta cómo medir sus distancias, todo esto se soluciona con las siguiente definición y consideraciones:

Definición 2.4.4. Llamaremos similaridad entre dos muestras i, h según la variable j a una función s_{jih} la cual cumpla que:

- La similaridad de una muestra consigo misma es igual a la unidad $s_{jii} = 1 \forall j, i$
- $0 \leq s_{jih} \leq 1 \quad \forall j, i, h$
- Simetría: $s_{jih} = s_{jhi}$

Una vez considerado esto, se puede crear un coeficiente de similaridad entre dos mediante el coeficiente de *Gower*:

$$s_{ih} = \frac{\sum_{j=1}^{p} w_{jih} s_{jih}}{\sum_{j=1}^{p} w_{jih}} \tag{2.4.1}$$

Donde la variable w_{jih} puede ser 0 ó 1, dependiendo de si se quiere tomar o no dicha variable o no en la comparación de dichas muestras.

Lo más habitual para formalizar las similaridades es tomar lo siguiente:

- Para continuas se puede tomar el valor $s_{jih} = 1 - \frac{|x_{ji} - x_{hi}|}{rango(X_j)}$
- En cambio para variables binarias en el caso de que coincida el atributo $x_{ji} = x_{jh}$ la similaridad es 1 y 0 en el caso contrario. También se puede decir que si hay presencia del atributo y coinciden es 1 y en caso contrario 0.

Una vez decidida la forma de decir de inicialmente dar la matriz de distancias o similaridades, se da el algoritmo aglomerativo de generación de la jerarquía:

- Se comienza con un *clustering* que tiene n *clusters* con una observación cada uno y se calcula la matriz de distancias
- Se toman los elementos que más cercanos están y se forman una nueva clase.
- Se sustituyen los dos elementos anteriores por la clase y se calcula la distancia entre la clase y el resto de observaciones de acuerdo con uno de los criterios preestablecidos.
- Repetir los dos pasos anteriores hasta obtener una única clase.

Dependiendo de la forma que se de unir los grupos se obtendrá un algoritmo u otro de clustering:

- Si $f(x, y) = min(x, y)$ es decir, se toma como distancia entre la clase y el resto de observaciones como el mínimo de las que se den, a este algoritmo se le llama *single linkage*, este tipo de enlace tiende a formar.
- Si $f(x, y) = max(x, y)$ es análogo al anterior, pero tomando la distancia máxima, a este método se le llama *complete linkage*.
- Si $f(x, y) = \frac{x + y}{2}$ se hace la media de las distancias de los dos grupos antes de unirlos, a este método se le llama *average linkage*.

Este tipo de métodos aporta una jerarquía en la agrupación de los datos en función de su similitud, de manera que a distintos grados de similitud se pueden establecer distintas reparticiones del espacio de observaciones. Además esto se puede representar fácilmente con un dendograma o diagrama de árboles otorgando una manera sencilla de establecer un criterio de similitud máxima.

2.4.2. Algoritmos Particionales

El clustering particional busca dado un conjunto de datos heterogéneos dividirlos en grupos homogéneos, como por ejemplo la segmentación de clientes o votantes. , buscaremos un criterio por el cual podamos afirmar que una partición del espacio es mejor que la anterior o no. En estos casos la variabilidad total de todas las observaciones se puede descomponer como la variabilidad intra-cluster y la inter-cluster como se detalla en la sección del análisis discriminante.

Definición 2.4.5. Se llama *centroide* de un cluster, C_k, $k = 1 \dots K$ al elemento que resulta de hacer la media de todos los elementos del cluster C_k:

$$\overline{x}_{jk} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij} \tag{2.4.2}$$

Definición 2.4.6. Se llama variabilidad o variación intra-cluster del cluster C_k $k = 1 \dots K$ a la siguiente expresión:

$$\mathbf{W}(C_k) = \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \overline{x}_{jk})^2 \tag{2.4.3}$$

Esta medida también se podría dar como la media de las diferencias cuadradas entre las observaciones del cluster, pero se seleccionará esta expresión.

Una vez dadas estas definiciones y criterios se puede construir un algoritmo iterativo como es el método de K-means que se puede resumir de la siguiente manera, sabiendo el número inicial de clusters, por ejemplo K:

1. Se inicializa la asignación de manera aleatoria, agrupando en los K clusters observaciones aleatoriamente.
2. Se itera lo siguiente hasta que no haya cambio en los clusters:
   a) Se calculan los centroides de cada uno de los K clusters

b) Se reasignan las observaciones de acuerdo a qué centroide es más cercano según la distancia euclidiana.

Proposición 2.4.1. El algoritmo de K-means sólo puede mejorar el criterio de la variación dentro de clusters.

Demostración. El paso 2a minimiza la suma de las desviaciones cuadradas y en el paso 2b la reasignación solo puede mejorar eso ya que se busca que la distancia euclídea al *centroide* sea la mínima. □

Esto provoca que el algoritmo no para hasta que no se da una mejora en el criterio. Sin embargo, este minimo que alcanza la función no tiene por qué ser global, ya que esta no es convexa y por tanto la consecución de este mínimo puede estar condicionada por la asignación inicial. Para solucionar estos problemas, se deben hacer varias ejecuciones del algoritmo para poder compararlas entre sí, por ejemplo *Scikit-Learn* implementa el método *k-means++* que no es más que realizar multiples ejecuciones con asignaciones iniciales distintas y luego las compara y otorga la mejor. Para poder compararlas se puede utilizar el parámetro inercia.

Definición 2.4.7. Se llama *inercia* de un clustering a la suma de las variaciones intra-clusters:

$$inercia = \sum_{k=1}^{K} \mathbf{W}(C_k) \tag{2.4.4}$$

La inercia representa como de compacta es nuestra partición a menor inercia más compacta es y por tanto más homogéneos son.

Otro de los problemas de este tipo de algoritmos es que se deben conocer a priori el número de clusters en los que se quieren dividir las observaciones. Para ello *Peña D.*[22] propone un contraste de hipótesis utilizando la inercia haciéndola depender del número de clusters

$$F = \frac{inercia(K) - inercia(K+1)}{\frac{inercia(K+1)}{n-K-1}} \tag{2.4.5}$$

Este cociente compara la disminución de la variación entre una partición con K y $K+1$ clústers, de esta manera, puede compararse con una F con $p, p(n-K-1)$ grados de libertad, pero que en la práctica no se puede asumir las hipótesis, por tanto, en la práctica si se da un valor mayor que 10 se asume que es conveniente añadir un clúster más.

Capítulo 3

Aplicación sobre datos

A lo largo de este capítulo se aplicarán algunas de las técnicas desarrolladas anteriormente a ejemplos prácticos con bases de datos reales obtenidas de distintos repositorios.

La principal herramienta usada ha sido el lenguaje de programación Python, utilizando librerías como Pandas para el manejo de las bases de datos, Scikit-Learn para la implementación de los modelos y otras librerias de visualización como Matplotlib para la inclusión de gráficos sencillos.

3.1. Clasificación de Iris

Uno de los ejemplos básicos del análisis discriminante es el conjunto de datos de distintas clases de flores Iris, en particular *Iris Setosa,Iris virginica, e Iris versicolor*. Nuestro objetivo con este conjunto de datos es encontrar variables latentes o factores que pueden darse debido a la alta correlación que se observan en un estudio inicial y luego realizar un análisis discriminante para obtener un modelo predictivo, usando las variables transformadas por los factores y las originales para comprobar como esta transformación puede afectar a las predicciones.

El *dataset* que se va a utilizar recoge datos de 150 muestras y toma las siguientes 5 medidas, ancho y largo del pétalo y ancho y largo del sépalo, además de su clasificación dentro de 1 las 3 especies antes detalladas.

El primer paso es intentar encontrar los factores latentes que puedan explicar la mayor parte de la variabilidad de forma común. En caso de encontrar esa variable común, que pudiera interpretarse en este caso como un *factor tamaño de la flor*, se podría simplificar el conjunto de datos.

Tras aplicar el análisis factorial con la funcionalidad implementada en la biblioteca *Scikit-Learn*, se ha podido obtener un modelo de un sólo factor común que acumula casi el 92 % de la variabilidad de las variables, siendo el resto repartido como variabilidad especifica de cada variable.

Por otro lado, tras ajustar las funciones discriminantes con los datos sin transformar, se consigue un `acuracy_score` del 100 %, sin embargo, utilizando los datos proyectados sobre el factor común, se puede observar que el modelo entrenado tiene una precisión del 95,5 % lo cual no es una pérdida significativa de precisión. Por

tanto, en este caso, se podría utilizar tanto el conjunto de datos original como el transformado para realizar las predicciones.

3.2. Alubias secas

3.3. Comparación regresión lineal, árboles de regresión y redes neuronales.

El objetivo de esta aplicación es comparar el rendimiento de los distintos algoritmos para la obtención de modelos predictivos que se han visto durante la memoria para ello, se utilizarán varias aproximaciones. Entre las aproximaciones usadas están el modelo lineal, que es la más sencilla de todas, los árboles de decisión para regresión que son los que un mayor *sobreajuste* pueden tener y las redes neuronales que son las que mayor adaptabilidad tienen a la hora de crear un modelo predictivo debido a la gran cantidad de parámetros que se manejan.

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, Tibshirani R.** *An Introduction to Statistical Learning: With Applications in R.* New York, NY: Springer; 2021.

[21] **Martínez Arias R.** *El análisis multivariante en la investigación científica.* Madrid: La Muralla; 1999.

[22] **Peña D.** *Análisis de datos multivariantes.* Madrid: McGraw-Hill; 2002.

[23] **Andrade-Sánchez, A. I., Galindo-Villardón, M. P., y Cuevas Romo, J. (2015).** *Análisis multivariante del perfil psicológico de los deportistas universitarios: Aplicación del CPRD en México.* Educación Física y Ciencia, 17(2), 00-00.

[24] **Hernandis, S. P., Diez, J. P., y Rouma, A. C. (2002).** *El consumo de inhalables y cannabis en la preadolescencia: Análisis multivariado de factores predisponentes.* Anales de Psicología/Annals of Psychology, 18(1), 77-93.

[25] **Machado-Duque, M. E., Echeverri Chabur, J. E., y Machado-Alba, J. E. (2015).** *Somnolencia diurna excesiva, mala calidad del sueño y bajo rendimiento académico en estudiantes de Medicina.* Revista colombiana de Psiquiatría, 44(3), 137-142.

[26] **Johnson, R. A., and Wichern, D. W.** *Applied multivariate statistical analysis.* New Jersey, Prentice Hall; 2002.

[27] **González García, N., y Taborda Londoño, A.** *Análisis de componentes principales Sparse: formulación, algoritmos e implicaciones en el análisis de datos.* Salamanca, 2015

[28] **Hornik, K., Stinchcombe, M., and White, H.** *Multilayer feedforward networks are universal approximators.* Neural networks, 2(5), 359-366, 1989. https://doi.org/10.1016/0893-6080(89)90020-8

[29] **Fisher,R. A.. (1988).** Iris. *UCI Machine Learning Repository.* https://doi.org/10.24432/C56C76.

[30] **Dry Bean Dataset. (2020)**. *UCI Machine Learning Repository.* `https://doi.org/10.24432/C50S4B`.