UNED

FACULTAD DE CIENCIAS

DEPARTAMENTO DE ESTADÍSTICA, INVESTIGACIÓN OPERATIVA Y CÁLCULO NUMÉRICO

PROYECTO DE FIN DE MASTER:

ESTIMACIÓN DE MATRICES DE COVARIANZAS: NUEVAS PERSPECTIVAS

TUTOR: HILARIO NAVARRO VEGUILLAS

REALIZADO POR: RAQUEL PASCUAL GARCÍA

Les dedico esta investigación a mis hijas Laura y Julia por lo mucho que las quiero.

Agradecimientos

Me gustaría mostrar mi agradecimiento a las personas que me han ayudado durante todo este tiempo.

A Marcos, por sus ánimos en los períodos de confusión y desanimo, por sus consejos y sobre todo por creer en mí siempre.

A mi padre por todo el apoyo brindado desde el primer momento hasta el último.

A Hilario Navarro, mi tutor, por su inmensa paciencia y por haber mostrado siempre disponibilidad para discutir dudas y la dirección correcta de la investigación. Le quiero agradecer muy especialmente su interés en que este trabajo fuese “un buen trabajo”.

A todos mis familiares y amigos que durante todo este tiempo me han ayudado y animado para llevar a cabo este proyecto.

A Jerome Friedman, Trevor Hastie, Robert Tibshirani e investigadores que como ellos comparten de forma altruista sus conocimientos. Les agradezco su generosidad y sus aportaciones que han sido la base fundamental de este trabajo.

INDICE

Capítulo 1

Introducción

1.1 Planteamiento del problema

El estudio de las matrices de covarianzas es fundamental en las técnicas del análisis multivariante. Por ejemplo, para el cálculo del estadístico *Wilks' Lambda,* para el método del *ratio* de verosimilitudes, en los análisis de componentes principales y factorial…

Uno de los métodos más estudiado y utilizado es el de máxima verosimilitud cuyo estimador es la matriz de covarianzas muestrales $S = \frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$. Este método es el más utilizado para encontrar un estimador de la matriz de covarianzas poblacional ($\sum$) cuando la población de partida es normal $N(\mu, \sum)$ y cuando el tamaño muestral, *n*, supera a la dimensión poblacional, *p*. La matriz de covarianzas muestrales es definida positiva por lo que se puede calcular su inversa, necesaria también en muchas técnicas del análisis multivariantes tales como el cálculo del estadístico *Hotelling's* T^2 o para la estimación de modelos gráficos Gaussianos.

En la actualidad los conjuntos de datos superan la capacidad de la estadística clásica debido a que la dimensión, *p*, es comparable (o incluso mayor) al tamaño de la muestra, *n.* Algunos ejemplos son*:*

- En estudios biológicos, donde, a veces se tiene un número de observaciones insuficientes, ya sea porque se tienen pocos datos experimentales disponibles o porque la recogida de muestras se encuentra bajo restricciones de presupuesto o/ y tiempo ([1] y [2]).

- En imágenes de satélite, donde se obtienen pocas observaciones (generalmente menos de 10 bandas espectrales) en más de 100.000 longitudes de onda de más de una red de píxeles [3].

- En quimiometría que para la determinación de las concentraciones de ciertos compuestos químicos se realizan estudios de calibración para analizar la intensidad de diferentes longitudes de onda espectrales (generalmente entre 500 y 1000 medidas de intensidad) utilizando un pequeño número de muestras químicas [3].

- En datos de expresión genética donde actualmente se utilizan métodos de *microarrays* para el estudio de tumores malignos humanos donde se vigilan los niveles de expresión de un gran número de genes (entre 5000 y 10000 o incluso más) en un pequeño número de muestras de tumor. Un ejemplo de ello se muestra en el estudio sobre cáncer [4] donde se estudian las correlaciones

parciales de 78 genes cancerígenos (3003 correlaciones parciales en total) de solamente 35 muestras de tumor disponibles (*n*=35).

Cuando esto sucede, la matriz de covarianzas muestrales, S, no es un buen estimador de $\sum$, muchos de sus autovalores llegan a tomar el valor de cero y como consecuencia se convierte en una matriz singular y no se puede calcular su inversa. Por ello, desde hace tiempo, en este contexto (p>n), que se denomina de alta dimensionalidad, se ha investigado sobre un estimador de $\sum$ que no tenga los defectos de S, se busca que la matriz sea más exacta, que esté bien preparada (*well-conditioned)*, es decir, que el *ratio* entre el mayor y el menor valor singular no sea muy grande y que sea siempre definida positiva para poder calcular su inversa.

El objetivo de este trabajo es buscar un estimador $\sum$ en el contexto de la alta dimensionalidad que cumpla con estas características deseadas.

En la actualidad los científicos que buscan un estimador adecuado de $\sum$ para datos de alta dimensionalidad se centran sobre todo en las técnicas de regularización y *shrinkage*, por esa razón, son en estas dos técnicas en las que se centra también el presente trabajo.

En la primera parte del capítulo 1, se hace un estudio del arte en el que se resumen los métodos clásicos de cálculo de la matriz de covarianzas. En la segunda parte del capítulo se presentan los resultados obtenidos de un análisis espectral de S en las situaciones donde *n*>*p*, *n*=*p* y *n*<*p*, se realiza este análisis para situar lo que sucede con el estimador de máxima verosimilitud de $\sum$, es decir con S, cuando *p*>>*n,* y para presentar una visión más clarificadora sobre el alcance del problema a resolver en el presente trabajo.

En el capítulo 2 se presenta, en el entorno de la denominada regularización *Lasso*, el algoritmo *Graphical Lasso* que es uno de los métodos que se utilizan actualmente para estimar $\sum$ y también $\sum^{-1}$ y su correspondiente modelo gráfico Gaussiano en el contexto de la alta dimensionalidad.

Para finalizar, en el capítulo 3, se explica otra de las técnicas más utilizadas: La solución *shrinkage*.

El entorno computacional que se ha utilizado en este trabajo de investigación ha sido el paquete estadístico de R, un *software* de gran difusión en los últimos tiempos debido fundamentalmente a su carácter gratuito, su alta capacidad de cálculo y el hecho de que esté en constante revisión. Como consecuencia se está produciendo una aparición continua de nuevas bibliotecas para resolver problemas estadísticos complejos, por lo que podríamos decir que R es actualmente el lenguaje de la estadística. R se considera un *software* clónico del paquete S-Plus, el cual utiliza el lenguaje S, un lenguaje diseñado por la compañía *AT&T's* en los Laboratorios Bell. Fueron dos profesores de la universidad de *Aucklan* (Nueva Zelanda), *Ross Ihaka* y *Robert Gentleman* los que elaboraron una versión reducida de S para tareas docentes. R (la inicial del nombre de ambos profesores) fue la denominación que le pusieron a este paquete estadístico. En 1995, *Martin Maechler* les convenció para su distribución gratuita. Las primeras versiones se mostraron en 1999. Hoy en día, se dan continuas aportaciones gratuitas y programadas en R.

En este trabajo de investigación se utilizarán varios paquetes estadísticos de R: `Huge` [5], `mvtnorm` [6], `corpcor` [7], *y* `glasso` [8]. En el trabajo, se investiga con una base de datos de alta dimensionalidad: *Breast cancer data* [9].

1.2 Resultados notables. Antecedentes.

1.2.1 Métodos clásicos de estimación de la matriz de covarianzas poblacional.

Los resultados clásicos resuelven el problema de la estimación de $\sum$ cuando la población de partida es normal, en algunos casos de no normalidad y cuando *n* supera a *p*.

La técnica más utilizada para llegar al objetivo de hallar el estimador de $\sum$ que mejor explique los datos observados de una población normal multivariante siendo *n* mucho más grande que *p* es la estimación de máxima verosimilitud, cuyo estimador es la matriz de covarianzas muestrales.

Sea $X_1,...,X_N$ una muestra aleatoria de una población normal multivariante de media μ y matriz de covarianzas $\sum$. Al ser cada uno de estos *p x 1* vectores $X_1,...,X_n$ independientes entre sí y cada uno con distribución $N_p(\mu,\Sigma)$, la función de densidad conjunta de todas las observaciones será el producto de las densidades marginales:

$$\mathrm{L}(\mu,\Sigma)=\prod_{j=1}^{n}\left\{\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}e^{-(x_j-\mu)'\Sigma^{-1}(x_j-\mu)/2}\right\}=\frac{1}{(2\pi)^{np/2}}\frac{1}{|\Sigma|^{n/2}}e^{-\sum_{j=1}^{n}(x_j-\mu)'\Sigma^{-1}(x_j-\mu)/2}.$$

Esta función de μ y $\sum$ es la función de verosimilitud.

Para hallar el estimador de $\sum$ (también de μ) que mejor explique los datos observados, hay que encontrar los valores de μ y $\sum$ que maximicen esta función. A esta técnica se le conoce con el nombre de estimación de máxima verosimilitud y es la técnica clásica utilizada para encontrar un estimador de $\sum$ cuando *n* > *p* y cuando la distribución de los datos de partida es normal multivariante. Los estimadores resultantes ($\hat{\mu}$ y $\hat{\Sigma}$) se llaman estimadores de máxima verosimilitud.

A continuación se muestra como, efectivamente, un estimador de $\sum$ aplicando esta técnica es $\hat{\Sigma}=S=\frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha-\bar{x})(x_\alpha-\bar{x})'$:

El exponente de la función de verosimilitud (aparte de por el factor $-\frac{1}{2}$) se puede escribir como:

$$tr\left[\Sigma^{-1}\left(\sum_{j=1}^{n}(x_j-\bar{x})(x_j-\bar{x})'\right)\right]+n(\bar{x}-\mu)'\Sigma^{-1}(\bar{x}-\mu).$$

Se sabe que $\sum^{-1}$ es una matriz definida positiva, por lo que, $(\bar{x}-\mu)'\Sigma^{-1}(\bar{x}-\mu)>0$, a no ser que $\mu=\bar{x}$. Entonces, la función de verosimilitud se maximiza con respecto a μ cuándo $\hat{\mu}=\bar{x}$.

Quedaría por maximizar: $L(\hat{\mu},\Sigma)=\frac{1}{(2\pi)^{np/2}}\frac{1}{|\Sigma|^{n/2}}e^{-tr\left[\Sigma^{-1}\left(\sum_{j=1}^{n}(x_j-\bar{x})(x_j-\bar{x})'\right)\right]/2}$ con respecto a $\sum$.

El máximo se obtiene en: $\sum=\frac{1}{n}\sum_{\alpha=1}^{n}(x_j-\bar{x})(x_j-\bar{x})'$. La demostración se puede ver en [10].

Podemos concluir que cuando una muestra aleatoria $X_1,...,X_N$ donde cada uno de estos *p x 1* vectores $X_1,...,X_n$ son independientes entre sí y cada uno con distribución $N_p(\mu,\sum)$, $\hat{\mu}=\bar{X}$ y $\hat{\sum}=\frac{1}{n}\sum_{\alpha=1}^{n}(x_j-\bar{x})(x_j-\bar{x})'=\frac{(n-1)}{n}S$ siendo $S=\frac{1}{n-1}\sum_{\alpha=1}^{n}(x_j-\bar{x})(x_j-\bar{x})'$ son los estimadores de máxima verosimilitud de μ y $\sum$.

Sobre $\hat{\Sigma}$ se han estudiado y demostrado las propiedades asintóticas (que se verán en el apartado siguiente).

También se ha analizado si $\hat{\sum}=\frac{(n-1)}{n}S$ es un estimador sin sesgo. Las conclusiones se desarrollan a continuación:

Se sabe que $\hat{\Sigma}$ será un estimador sin sesgo sí $E(\hat{\Sigma})=\Sigma$. Para su análisis se necesita el teorema que se enuncia a continuación.

Teorema 1

$n\hat{\Sigma}$ se distribuye como $\sum_{\alpha=1}^{n-1}Z_\alpha Z_\alpha'$ donde Z_α se distribuye como una $N(0,\sum)$ $\alpha=1,...,n-1$, siendo $Z_1,...,Z_{n-1}$ independientes.

La demostración se puede consultar en [11].

Por el teorema 1, $E(\hat{\Sigma})=\frac{1}{n}E\left[\sum_{\alpha=1}^{n-1}Z_\alpha Z_\alpha'\right]=\frac{n-1}{n}\sum$. Por lo que $\hat{\Sigma}$ es un estimador sesgado de $\sum$.

Por esta razón, se define: $S=\frac{1}{n-1}A=\frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha-\bar{x})(x_\alpha-\bar{x})'$ como la matriz de covarianzas muestral que es un estimador sin sesgo de $\sum$ ya que en este caso $E(S)=\frac{1}{n-1}\left[\sum_{\alpha=1}^{n-1}Z_\alpha Z_\alpha'\right]=\Sigma$. Entonces,

se puede concluir diciendo que el estimador de máxima verosimilitud de $\sum$ es la matriz de covarianzas muestral, es decir, $S = \frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$.

Tanto $\overline{X}$ como S son estadísticos suficientes, es decir, toda la información muestral de los datos de la matriz X está contenida en $\overline{X}$ y S. Esto generalmente no es cierto cuando la población es no normal por lo que, si no se puede considerar que los datos provienen de una distribución normal multivariante, las técnicas que dependen únicamente de $\overline{X}$ y S podrían ignorar otra información muestral útil.

1.2.2 Propiedades asintóticas de la matriz de covarianzas muestral

Evidentemente, las propiedades asintóticas del estimador de máxima verosimilitud de $\sum$, han sido estudiadas, es decir, se ha analizado el comportamiento del estimador cuando $n \to \infty$ y de esta forma conocer si es un estimador consistente y su distribución asintótica. El estimador de máxima verosimilitud de $\sum$ es consistente y asintóticamente normal como se demuestra a continuación.

Para comprobar la consistencia se debe comprobar que cuando $n \to \infty$, $\hat{\sum}$ se aproxima al auténtico valor del parámetro $\sum$.

Se considera un elemento de la matriz de covarianzas muestral, s_{ik}.Entonces:

$$s_{ik} = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ij} - \bar{x}_i)(x_{kj} - \bar{x}_k) = \frac{1}{n-1}\sum_{j=1}^{n}(x_{ij} - \mu_i + \mu_i - \bar{x}_i)(x_{kj} - \mu_k + \mu_k - \bar{x}_k) = \frac{1}{n-1}$$
$$\sum_{j=1}^{n}(x_{ij} - \mu_i)(x_{kj} - \mu_k) + \frac{n}{n-1}(\bar{x}_i - \mu_i)(\bar{x}_k - \mu_k)$$

Cuando $n \to \infty$ el segundo término converge a cero, ya que, por la ley de los grandes números, cada componente del vector muestral $\overline{X}$ es un estimador consistente de la componente asociada al vector de los valores poblacionales μ, si los vectores de las observaciones están distribuidos de forma independiente e idéntica. El primer término converge a σ_{ik}, haciendo $Y_j = (X_{ij} - \mu_i)(X_{kj} - \mu_k)$ con $E(Y_j) = \sigma_{ik}$.

Cuando se desconoce la distribución exacta de un estimador, podemos preguntarnos si cuando $n \to \infty$ el estimador sigue alguna distribución conocida, de ser así, esa será su distribución asintótica. La distribución asintótica de la matriz de covarianzas muestral es una distribución normal (la demostración se puede encontrar en [11]). Sobre la matriz de covarianzas muestral, también se conoce su distribución exacta que es la distribución de *Wishart* y que se explica a continuación en el siguiente apartado.

1.2.3 La distribución exacta de la matriz de covarianzas muestral: La distribución de *Wishart*

Gracias a las propiedades asintóticas, se conocen las características de la matriz de covarianzas muestrales cuando el tamaño muestral tiende a ∞. Sin embargo, la presente investigación se centra en tamaños muestrales pequeños, por lo que, nos interesaría más bien conocer su distribución exacta.

Se considera el estimador de máxima verosimilitud de $\sum$: $S = \frac{1}{n-1}A = \frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$,

entonces, $A = (N-1)S = \sum_{\alpha=1}^{n}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$, donde las variables $X_1,...,X_n$, $(n > p)$ se distribuyen de forma independiente según una $N(\mu,\Sigma)$. Entonces, por el teorema 1, se sabe que A se distribuye como $\sum_{i=1}^{n-1} Z_i Z_i'$ siendo $Z_1,...,Z_{n-1}$ independientes cada una con distribución $N(0,\sum)$.

La función de densidad de A cuando A es definida positiva es:

$$\frac{|A|^{\frac{1}{2}(n-p-1)} e^{-\frac{1}{2}tr\Sigma^{-1}A}}{2^{\frac{1}{2}np}\pi^{p(p-1)/4}|\Sigma|^{\frac{1}{2}n}\prod_{i=1}^{p}\Gamma\left[\frac{1}{2}(n+1-i)\right]}$$

A continuación se va a demostrar que la distribución de A o S cuando $\Sigma = I$ es en cierto sentido una generalización de la distribución χ^2 con $n-i+1$ grados de libertad.

Sea $(Z_1,...,Z_n) = \begin{pmatrix} v_1' \\ \vdots \\ v_p' \end{pmatrix}$, entonces, los elementos de $A = (a_{ij})$ son el producto escalar de estos $v_1,...,v_p$, es decir, $a_{ij} = v_i' v_j$. Estos $v_1,...,v_p$ vectores son independientes cada uno con una distribución $N(0,I_n)$. Por conveniencia se transforman a unas nuevas coordenadas según la ortogonalización de Gram-Schmidt:

$$w_1 = v_1$$

$$w_2 = v_2 - \frac{v_2' w_1}{\|w_1\|^2} w_1$$

$$w_i = v_i - \sum_{i=1}^{i-1} \frac{v_i' w_i}{\|w_i\|^2} w_i .$$

Donde w_k es ortogonal a w_i siendo $k < i$.

Se define $t_{ii} = \|w_i\| = \sqrt{w_i' w_i}$ y $t_{ij} = \frac{v_i' w_j}{\|w_j\|}$. Entonces, $v_i = \sum_{j=1}^{i}\left(\frac{t_{ij}}{\|w_j\|}\right) w_j$ y $a_{hi} = v_h' v_i = \sum_{j=1}^{\min(h,i)} t_{hj} t_{ij}$ siendo t_{ij} las primeras $i-1$ coordenadas de v_i en el sistema de coordenadas de $w_1,...,w_{i-1}$

Si se define la matriz triangular inferior: $T = (t_{ij})$ con $t_{ij} = 0 \forall i < j$, entonces, A se puede escribir como $A = TT'$.

Se sabe que las coordinadas de v_i referidas a las nuevas coordenadas ortogonales, siendo $v_1,...v_{i-1}$ los primeros ejes de coordenadas, se distribuyen de forma independiente con una distribución $N(0,1)$ y t_{ii}^2 tiene una distribución χ^2 con $n-i+1$ grados de libertad. Además como la distribución condicional de $t_{i1},...,t_{ii}$ no depende de $v_1,...,v_{i-1}$, se distribuyen de forma independiente a $t_{11}, t_{21}, t_{22}..., t_{i-1i-1}$. Entonces, si se consideran $Z_1,...,Z_n$ $(n \geq p)$ variables que se distribuyen de forma independiente cada una de acuerdo a una $N(0,\Sigma)$, donde $\Sigma = I$ y se considera $A = \sum_{i=1}^{n} Z_i Z_i' = TT'$ dónde $t_{ij} = 0 \forall i < j$ y $t_{ii} > 0$. Entonces $t_{11}, t_{21},..., t_{pp}$ están independientemente distribuidas cada una con una distribución $N(0,1)$ y t_{ii}^2 tiene una distribución χ^2 con $n-i+1$ grados de libertad. Por todo esto, cuando $\Sigma = I$, la distribución de A (o S) es en cierto sentido una generalización de la distribución χ^2 con $n-i+1$.

Si generalizamos al caso de cualquier Σ, la distribución muestral de la matriz de covarianzas muestral es la distribución de *Wishart* y se define como la suma de los productos de vectores aleatorios normales multivariantes independientes: $\sum_{i=1}^{n-1} Z_i Z_i'$. Es decir, la distribución de *Wishart* con *n* grados de libertad, $W_m(\bullet/\Sigma)$, es la distribución de $\sum_{i=1}^{n} Z_i Z_i'$, donde cada Z_i se distribuye de forma independiente como una $N(0,\Sigma)$.

La distribución de *Wishart* se utiliza mucho por ejemplo en análisis multivariantes de inferencia sobre $\sum$.

1.3 Análisis espectral

Con el fin de analizar cómo funciona la matriz de covarianzas muestral en el contexto de la alta dimensionalidad se realiza un análisis espectral de este estimador en diferentes situaciones. Se realizan 100 simulaciones de una distribución normal multivariante: $\mu = 0$ y $\Sigma = s_{ij}$ donde $s_{ij} = \rho^{|i-j|}$ siendo $0 < \rho < 1$ (matriz de *Toeplitz*) para los casos: *n* > *p*, *n* = *p*, *n* < *p*. La dimensión se fija en 10 variables y el proceso se realiza para los valores de $\rho = 0.2$ y $\rho = 0.5$. En cada uno de estos casos se analiza la variabilidad del autovalor inferior y del autovalor superior y se comparan sus valores con los valores poblacionales (la programación se puede consultar en el apéndice I). Los valores poblacionales son: Para el caso de $\rho = 0.2$, el valor del menor y del mayor autovalor son 0.67 y 1.47 respectivamente y en el caso de $\rho = 0.5$ el valor del menor y del mayor autovalor son 0.34 y 2.68 respectivamente.

En los gráficos que se muestran a continuación se observa como para el caso $\rho = 0.2$ el autovalor inferior cuando *n* = *p* y cuando *n* < *p* alcanza el valor cero y el autovalor superior aumenta apreciablemente su valor y su variabilidad cuando *n* < *p*, al contrario de lo que ocurre en el caso clásico de *n* > *p*.

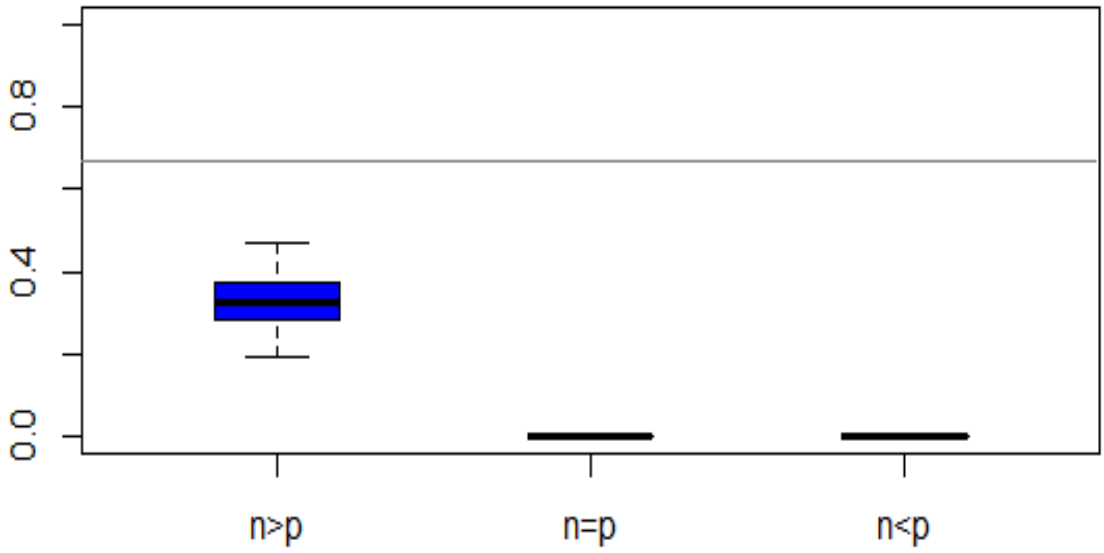

Figura 1 Variabilidad del menor autovalor para los casos: *n* > *p*, *n* = *p*, *n* < *p* y comparación con su valor poblacional

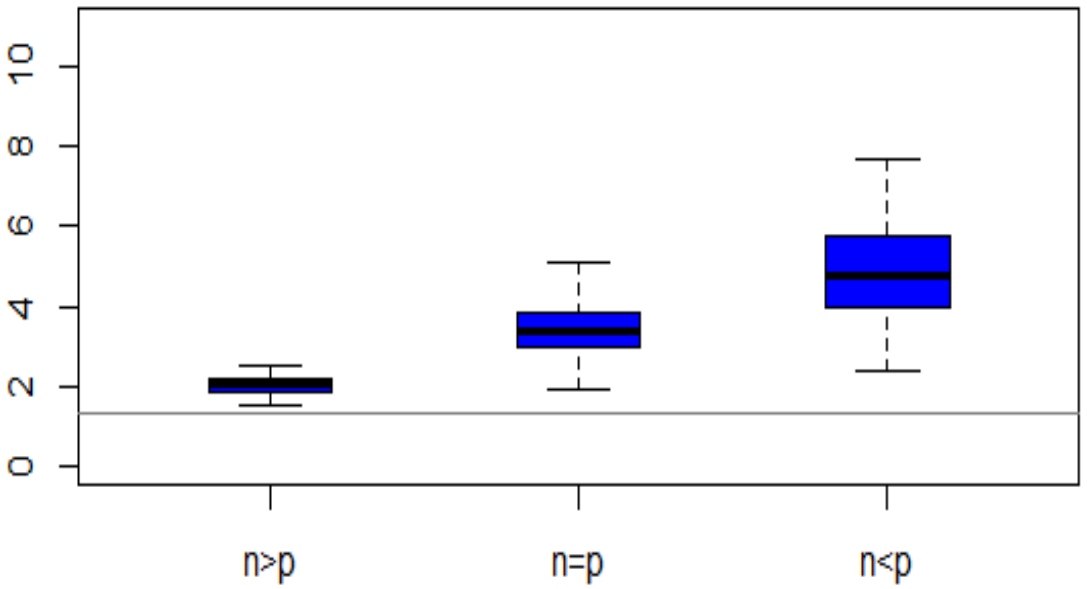

Figura 2 Variabilidad del autovalor superior para los casos: *n* > *p*, *n* = *p*, *n* < *p* y comparación con su valor poblacional

Para el caso con $\rho = 0.5$ los resultados son similares como muestran las siguientes figuras.

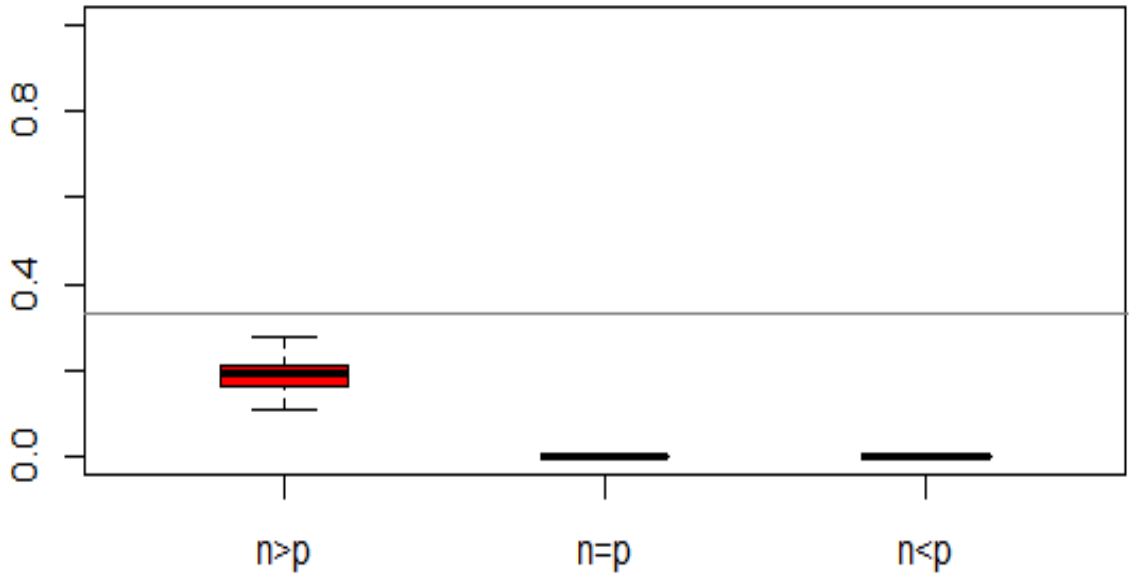

Figura 3 Variabilidad del menor autovalor para los casos: *n* > *p*, *n* = *p*, *n* < *p* y comparación con su valor poblacional

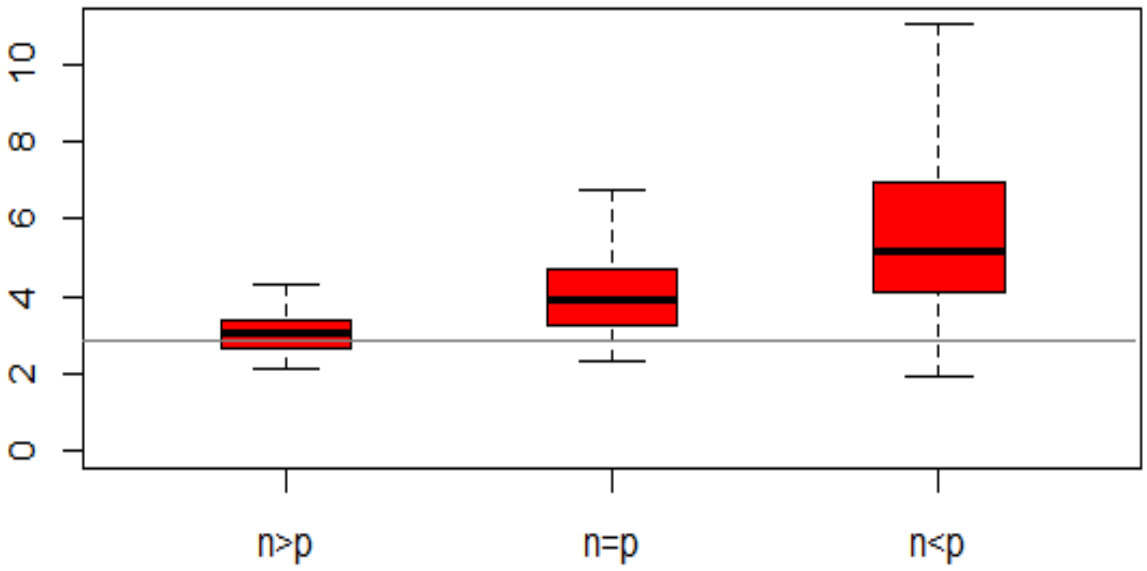

Figura 4 Variabilidad del autovalor superior para los casos: *n* > *p*, *n* = *p*, *n* < *p* y comparación con su valor poblacional

Como se observa en ambos casos, cuando *n* es mayor que *p*, los valores que toman el autovalor superior e inferior de la matriz de covarianzas muestral son cercanos a sus valores poblacionales. Sin embargo, en los casos donde *n* es igual a *p* o menor, se ve que esta característica ya no se cumple, la distribución del autovalor superior se dispersa mucho y el autovalor inferior llega a tomar el valor cero y como consecuencia, la matriz deja de ser de rango completo y no se puede invertir. Los experimentos aquí presentados están de acuerdo con los resultados de los trabajos [12], [13], entre otros. Este análisis espectral manifiesta que efectivamente, la matriz de covarianzas muestrales es un estimador débil en los casos en los que *n* no supera a *p*.

Capítulo 2.

Regularización. El algoritmo *Graphical Lasso*

2.1 Introducción

En este capítulo se muestra que la estimación de los modelos gráficos Gaussianos y las matrices de covarianzas en el contexto de la alta dimensión se puede reducir a resolver una serie de problemas de regresión lineal regularizados.

En el análisis de los datos de alta dimensión es fundamental la reducción de la dimensión, por eso, un principio clave es el principio de *sparsity,* que asume que solamente un pequeño número de predictores contribuyen a la respuesta. Uno de los métodos más populares que se basan en este principio es el método *Lasso (Least absolute shrinkage and selection operator)* introducido por *Tibshirani* en 1996 [14] para la estimación de modelos de regresión lineal de alta dimensión. Este método consiste en inducir *sparsity* en el modelo, es decir, reduce a cero a los coeficientes de regresión, β, con valores más pequeños, manteniendo el valor de los coeficientes con valores más grandes, de esta manera consigue reducir la alta dimensión.

El problema empieza a plantearse, entre otras razones, cuando se observa que el modelo de regresión lineal clásico: $Y = X\beta + \varepsilon$ y la estimación de sus coeficientes de regresión por el método de mínimos cuadrados no funciona para la alta dimensionalidad. Este método trabaja bien cuando el tamaño muestral es grande y el número de variables es pequeño (p<50) por lo que no es necesario preocuparse del número de variables predictoras. A pesar de su buen funcionamiento para estos casos, tiene una serie de debilidades. Por ejemplo, la dificultad en la interpretación del modelo, ya que ningún $\hat{\beta}_j$ se iguala a cero, y por lo tanto se tienen que tener en cuenta todas las variables en la interpretación. También aparece una debilidad cuando existen muchas variables, X_j, correlacionadas, sus coeficientes estimados $\hat{\beta}_j$ tienen una alta varianza, $\mathrm{var}\left(\hat{\beta}_j\right)$, y como consecuencia, un coeficiente con valor positivo grande para una variable podría ser cancelado por un coeficiente con valor negativo grande similar. Dejando a un lado las debilidades y centrándonos en su aplicación para datos de alta dimensión, se comprueba que el método de mínimos cuadrados no funciona bien ya que la matriz $X^{'}X$ no es invertible y por lo tanto no se pueden estimar los coeficientes de regresión ($\hat{\beta} = \left(X^{'}X\right)^{-1}X^{'}Y$).

Tradicionalmente este problema se resuelve con la regresión lineal *Ridge* (se verá en el apartado 2.2.2) la cual reemplaza la suma de los residuos al cuadrado por su versión penalizada, $SRCP(\beta) = \|Y - X\beta\|^2 + \lambda\sum_{j=1}^{p}\beta_j^2$, y cuya única solución, $\hat{\beta} = \left(X^{'}X + \lambda I\right)^{-1}X^{'}Y$, incorpora una constante

a la diagonal de la matriz $X^{'}X$ lo que hace que se pueda invertir siempre. Además, la solución *Ridge* funciona bastante bien en presencia de multicolinealidad y cuando *p* no es muy grande, sin embargo, en general, no induce *sparsity* en el modelo.

Para corregir esta desventaja de la solución *Ridge se* propone la regresión lineal *Lasso* (se verá en el apartado 2.2.3) que al igual que en la regresión Ridge minimiza la suma de los residuos al cuadrado penalizada, pero en vez de con la penalización *Ridge:* $L_2 = \sum_{j=1}^{p} \beta_j^2$, con la penalización *Lasso* $L_1 = \sum_{j=1}^{p} \left|\beta_j\right|$. De esta forma se consigue que más β_j con valores pequeños sean estimados por cero. El éxito de la solución *Lasso* en el contexto de la regresión lineal, ha derivado en su utilización para la estimación de $\sum$ y $\sum^{-1}$ cuando *p* supera a *n*.

Para mostrar los valores de $\sum^{-1}$ se suelen utilizar los denominados modelos gráficos Gaussianos. Los modelos gráficos consisten en un conjunto de nodos (donde cada nodo representa una variable aleatoria) y un conjunto de aristas que juntan pares de nodos. Si el modelo gráfico es no dirigido, entonces, las aristas no tendrán flechas direccionales, estos casos también se conocen con el nombre de campos aleatorios de *Markov* o redes de *Markov.* Los modelos gráficos Gaussianos (MGG), también conocidos como *covariance selection models* o *concentration graph models,* son modelos gráficos no dirigidos donde se asume que las variables aleatorias que lo componen son continúas con una distribución Gaussiana multivariante de media μ y matriz de covarianzas $\sum$. El desarrollo inicial de los MGG es solamente aplicable a los casos donde *n* >> *p,* pero debido al creciente interés de los casos donde *p* supera a *n* se proponen una serie de modificaciones con el fin de poder utilizarlos en el entorno de la alta dimensionalidad. Los MGG exploran y visualizan las relaciones de dependencia condicional entre variables aleatorias Gaussianas. Es usual utilizarlos para mostrar la estructura de las covarianzas parciales entre las variables, es decir, para mostrar los valores de la inversa de la matriz de covarianzas [11]. En estos gráficos la ausencia de una arista entre dos vértices tiene el significado de que las correspondientes variables aleatorias son condicionalmente independientes dadas las otras variables, es decir, que el valor de $\sum^{-1}$ correspondiente a las dos variables es cero. Así, el problema de estimar un MGG es equivalente a estimar $\sum^{-1}$.

Para formular el problema de la estimación de $\sum$, $\sum^{-1}$ y su correspondiente MGG en el lenguaje de la regresión lineal se muestra la relación que existe entre la estructura de las covarianzas parciales y la regresión lineal múltiple (*p* > 1), en el apartado 2.3.1. En los siguientes apartados, se explican dos algoritmos que utilizan esta relación para estimar $\sum$, $\sum^{-1}$ y su correspondiente MGG: Un algoritmo de regresión modificado para la estimación de un modelo gráfico Gaussiano con estructura del grafo conocida, que se verá en el apartado 2.3.2, y *e*l algoritmo *Graphical Lasso* que se verá en el apartado 2.3.3. En ambos casos, se asume que las variables aleatorias son continuas con una distribución Gaussiana multivariante de media μ y matriz de covarianzas $\sum$. Ambos algoritmos se basan en maximizar, con respecto a $\sum^{-1}$, la función log-verosimilitud parcialmente maximizada por el parámetro μ y penalizada. La diferencia entre los dos algoritmos radica en que en el caso del algoritmo de regresión modificado no existe regularización, mientras que en el caso del algoritmo *Graphical Lasso* si, en concreto una regularización *Lasso.* Además, en el algoritmo de regresión modificado se necesita conocer la estructura del grafo a priori (conocer que aristas están ausentes, es decir, que valores de $\sum^{-1}$ son cero) pero esto, en la mayoría de los casos, es desconocido

a priori. El algoritmo *Graphical Lasso* se puede ver como una solución a esta limitación del algoritmo de regresión modificado.

2.2 Métodos lineales de regresión

El modelo de regresión lineal clásico y el método de mínimos cuadrados

Para estimar los parámetros de interés, β_j, en el modelo de regresión lineal clásico :

$$Y = f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \varepsilon = X\beta + \varepsilon,$$

cuando *n* > *p,* se aplica el método de mínimos cuadrados que consiste en hallar aquellos β_j que minimicen la suma de los residuos al cuadrado:

$$SRC(\beta) = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 = (Y - X\beta)'(Y - X\beta)$$

y cuya solución es: $\hat{\beta} = (X'X)^{-1} X'Y$.

La naturaleza de la regresión lineal múltiple se interpreta más fácilmente si estudiamos su representación geométrica. A partir del vector de respuesta medio,

$$E(Y) = X\beta = \beta_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \beta_1 \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} + \ldots + \beta_p \begin{bmatrix} x_{1p} \\ x_{2p} \\ \vdots \\ x_{np} \end{bmatrix},$$

se observa que $E(Y)$ es una combinación lineal de las columnas de la matriz X . Generalmente, el vector respuesta Y no se halla en el subespacio generado por $X_1, \ldots, X_p$, debido al residuo aleatorio ε, lo que significa que Y no es exactamente una combinación lineal de las columnas de X . La igualdad de la cual se obtiene la estimación del vector de los parámetros desconocidos, β: $X'(Y - X\beta) = 0$, expresa, que el vector de residuos $\varepsilon = Y - \hat{Y}$ es ortogonal al subespacio generado por $X_1, \ldots, X_p$ y el vector respuesta estimado, $\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y$, es la proyección ortogonal de Y dentro del subespacio generado por $X_1, \ldots, X_p$ tal y como se observa en la siguiente figura.

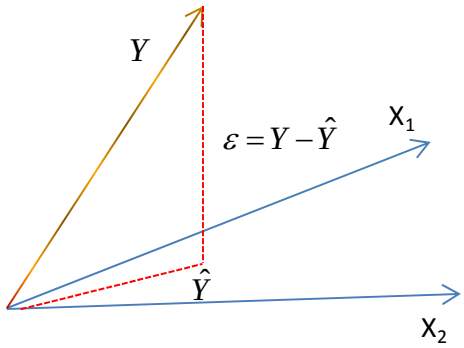

Figura 5 Significado geométrico de la regresión lineal clásica

La regresión lineal Ridge

La regresión *Ridge* disminuye el valor de los coeficientes de regresión, β_j, imponiendo una penalización en su tamaño. Para estimar estos coeficientes de interés, la solución *Ridge* minimiza la suma de los residuos al cuadrado penalizada:

$$SRCP(\beta) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \left\| Y - X\beta \right\|^2 + \lambda \sum_{j=1}^{p} \beta_j^2 ,$$

donde $\lambda > 0$ es el parámetro de regularización que controla la longitud del vector de los parámetros de regresión, es decir, controla la cantidad de encogimiento de β_j. A valores más grandes de λ, mayor encogimiento de los coeficientes.

Otra forma equivalente de expresar el problema *Ridge* es:

$$\hat{\beta}^{RIDGE} = MIN \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \text{ sujeto a: } \sum_{j=1}^{p} \beta_j^2 \le t ,$$

donde 't' es un umbral elegido adecuadamente para la convergencia. Esta expresión representa claramente la naturaleza de la solución *Ridge:* Minimizar la suma de los residuos al cuadrado imponiendo una restricción de tamaño a los parámetros β_j.

Para hallar la solución *Ridge,* se debe restar a cada observación la media de la variable correspondiente: $x_{ij}^{RIDGE} = x_{ij} - \bar{x}_j$. El parámetro β_0 se estima de la forma: $\hat{\beta}_0 = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ y los restantes parámetros β_j se estiman mediante la regresión *Ridge* sin tener en cuenta el parámetro β_0.

La única solución *Ridge* es: $\hat{\beta}^{RIDGE} = \left(X^{'}X + \lambda I \right)^{-1} X^{'}Y$, donde I es la matriz identidad *p* x *p*. En este caso, lo que se hace es añadir la cantidad λ a la diagonal de la matriz $X^{'}X$ y conseguir de esta forma

que sea siempre no singular incluso cuando $X^{'}X$ es singular. Como se observa, la solución *Ridge* sigue siendo una función lineal de Y.

A partir de la descomposición del valor singular de la matriz X, se puede profundizar en la naturaleza de la regresión lineal *Ridge*. La descomposición del valor singular de la matriz X es una factorización del tipo: $X = UDV^{'}$, donde U y V son *n* x *p* y *p* x *p* matrices ortogonales y D es una matriz diagonal formada por los valores singulares de X en su diagonal principal tal que $d_1 \geq d_2 \geq ... \geq d_p \geq 0$. Evidentemente, si algún $d_i = 0$, la matriz X sería singular.

A continuación, para entender el funcionamiento de la regresión *Ridge,* aplicamos la descomposición del valor singular a la matriz X: $X = UDV^{'}$ para seguidamente comparar el vector respuesta estimado mediante la regresión *Ridge* con el vector respuesta estimado mediante el método clásico de mínimos cuadrados.

En el caso del método clásico de mínimos cuadrados, el vector respuesta estimado quedaría de la forma, $\hat{Y} = X\hat{\beta} = X\left(X^{'}X\right)^{-1}X^{'}Y = UU^{'}Y = \sum_{j=1}^{p} u_j u_j^{'} y$, donde $u_j^{'}y$ son las coordenadas de y con respecto a la base ortonormal U.

En el caso de la regresión *Ridge*, se obtiene:

$$\hat{Y}_{RIDGE} = X\hat{\beta}_{RIDGE} = X\left(X^{'}X + \lambda I\right)^{-1}X^{'}Y = UD\left(D^2 + \lambda I\right)^{-1}DU^{'}Y = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^{'} y$$

Donde $\sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda} u_j^{'} y$, son las coordenadas de y con respecto a la base ortonormal U. Puesto que $\lambda > 0$ se tiene que $\frac{d_j^2}{d_j^2 + \lambda} < 1$. Por lo tanto, para este caso se reducen (*shrinkage*) estas coordenadas debido a los factores $\frac{d_j^2}{d_j^2 + \lambda}$. Se conseguirá una mayor cantidad de reducción aplicada a las coordenadas de y cuanto más pequeño sea d_j^2.

Debido a lo expuesto en el párrafo anterior, se puede ver que cuando X es una matriz ortonormal los coeficientes de regresión de la solución *Ridge*: $\hat{\beta}^{RIDGE} = \left(X^{'}X + \lambda I\right)^{-1}X^{'}Y$, tienen una solución explícita que se consigue al aplicar una transformación a la estimación de los coeficientes por el método de mínimos cuadrados $\left(\hat{\beta} = \left(X^{'}X\right)^{-1}X^{'}Y\right)$ que consiste en un encogimiento (*shrinkage*) proporcional: $\frac{\hat{\beta}_j}{(1+\lambda)}$.

La regresión lineal *Lasso*

La regresión lineal *Lasso* induce más *sparsity* al modelo que la regresión lineal *Ridge.* Esta solución se fundamenta en minimizar la suma de los residuos al cuadrado penalizada, pero en vez de con la penalización *Ridge:* $L_2 = \sum_{j=1}^{p} \beta_j^2$, con la penalización *Lasso* $L_1 = \sum_{j=1}^{p} |\beta_j|$.

El objetivo es minimizar:

$$SRCP(\beta) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \|Y - X\beta\|^2 + \lambda\sum_{j=1}^{p}|\beta_j|,$$

donde $\lambda > 0$ es, de nuevo, el parámetro de regularización que controla la *sparsity* del modelo. Cuanto mayor es el valor de λ , a más coeficientes de regresión se les obliga a tomar el valor cero. Nótese que, como el parámetro de regularización controla la *sparsity* del modelo, la selección adecuada de su valor es de suma importancia.

Otra forma de expresar el problema *Lasso* es:

$$\hat{\beta}^{LASSO} = MIN\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \text{ Sujeto a: } \sum_{j=1}^{p}|\beta_j| \le t\,.$$

Se observa que, al igual que en la regresión lineal *Ridge,* se trata de minimizar la suma de los residuos al cuadrado sujeta a una restricción de desigualdad en los parámetros β_j .

Como sucede en la regresión lineal *Ridge*, la constante β_0 , estimada es $\hat{\beta}_0 = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Sin embargo, la nueva restricción: $L_1 = \sum_{j=1}^{p}|\beta_j|$ hace que la solución de la regresión lineal *Lasso* no sea una función lineal de Y .

Para profundizar en el significado y la naturaleza de la solución de la regresión lineal *Lasso* pasaremos a explicar la función denominada 'operador *soft-threshold'.* Esta función es fundamental para poder entender y calcular la solución de la regresión lineal *Lasso.*

Supongamos el problema, *n*=*p*=1 y $X = 1$. En estas circunstancias, el problema de la regresión lineal *Lasso* consiste en minimizar la función: $SRCP(\beta^{Lasso}) = \frac{1}{2}(y-\beta)^2 + \lambda|\beta|$ para un valor fijo de λ . Su derivada con respecto a β es: $SRCP'(\beta^{Lasso}) = -y + \beta + \lambda \cdot sign(\beta)$, donde $sign(\beta)$ se define como: $|\beta| = \beta \cdot sign(\beta)$ y por conveniencia se ha decidido que su valor en cero sea cero ya que la función $SRCP(\beta^{Lasso}) = \frac{1}{2}(y-\beta)^2 + \lambda|\beta|$ no es diferenciable cuando $\beta = 0$. Igualando la derivada a cero se obtiene la solución explícita de los coeficientes de regresión: $\hat{\beta}^{Lasso}(\lambda) = sign(y)(|y| - \lambda)_+$ donde

$(|y|-\lambda)_+ = |y|-\lambda$ sí $|y|-\lambda > 0$ y cero en otro caso. A partir de este resultado se observan dos características fundamentales de la solución de la regresión lineal *Lasso:* La primera ya se ha comentado, a mayores valores de parámetro λ se encogen los coeficientes de la regresión lineal Lasso hacia cero, se observa que en el momento en el que λ exceda a $|y|$, $\hat{\beta}^{Lasso}(\lambda)$ llega a ser cero. La segunda característica es que la solución de la regresión lineal *Lasso* es una función lineal a trozos con respecto al parámetro de regularización λ, tal y como se puede observar en la siguiente figura.

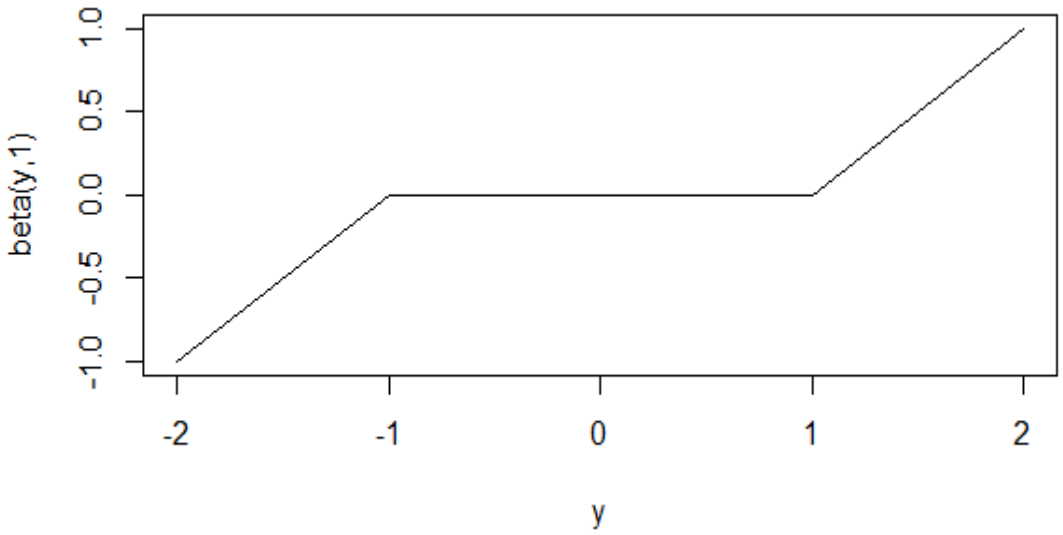

Figura 6 Gráfico del operador *soft-thresholding* para lambda=1

La función obtenida de igualar la derivada a cero para obtener la solución de los coeficientes de regresión: $\hat{\beta}^{Lasso}(\lambda) = sign(y)(|y|-\lambda)_+$, es lo que se define como operador *soft-threshold* [15].

A través del operador *soft-threshold* se pueden resolver muchos problemas de regresión penalizados. Como ya se ha comentado, la solución de la regresión lineal *Lasso* no es una función lineal de Y por lo que los coeficientes de regresión no se pueden calcular como en los casos de la regresión lineal clásica o la regresión lineal *Ridge.* Para calcular la solución de la regresión lineal *Lasso* existen algoritmos eficientes (con el mismo coste computacional que para calcular la solución *Ridge*). Estos algoritmos son: El algoritmo LAR (*Least Angle Regression)*, el algoritmo *Least Angle Regression: Lasso Modification* y el método *Pathwise Coordinate Optimization,* el cual utiliza el operador *soft-threshold* para su resolución. A continuación se describen y se explican estos tres algoritmos:

Algoritmo Least Angle regression:

1. Se estandarizan los valores de las variables predictoras. Se comienza con el valor residual: $r = y - \bar{y}$ y $\hat{\beta}_1, \ldots, \hat{\beta}_p = 0$.
2. Se busca la variable predictora X_j que esté más correlacionada con el valor residual r.
3. Se cambia el valor de β_j de cero al valor estimado por el método de mínimos cuadrados, hasta que se encuentre alguna otra variable, X_k, que tenga tanta correlación con el residuo actual ($r = y - \beta_j X_j$) como X_j.
4. Se cambia el valor de β_j y de β_k a los valores estimados por el método de mínimos cuadrados, hasta que se encuentre alguna otra variable, X_l, que tenga tanta correlación con el residuo actual ($r = y - \beta_j X_j - \beta_k X_k$).
5. Se continúa así hasta que todas las *p* variables predictoras se han introducido. Después de min (*n*-1, *p*) pasos, se llega a la solución final.

Algoritmo 1 Algoritmo *Least Angle regression*

El algoritmo 1 describe *LAR,* que es un algoritmo relativamente nuevo [16] que construye el modelo añadiendo variable a variable secuencialmente. En cada paso identifica a la variable adecuada, la incluye en el modelo y actualiza el modelo con un ajuste de mínimos cuadrado.

Se utiliza una modificación del algoritmo *LAR: Lasso Modification* cuando los valores de los β_j que se van estimando, no alcancen nunca el valor cero. El *algoritmo LAR: Lasso Modification* consta de los mismo pasos que el algoritmo *LAR* sólo que se añade uno más entre el paso 4 y 5:

> 4bis. Si un coeficiente β_i no toma nunca el valor cero, se elimina la variable correspondiente, X_i, y se vuelve a ajustar el modelo por el método de mínimos cuadrados.

El método *Pathwise Coordinate Optimization* consiste en un algoritmo de descenso coordinado para hallar la solución de la regresión lineal *Lasso*. En general, para hallar la solución *Lasso* se utiliza el algoritmo LAR. Sin embargo, en [17] se muestra que el algoritmo del descenso coordinado es un procedimiento muy competitivo en comparación con el algoritmo LAR ya que ofrece un conjunto de soluciones de forma eficiente (denominado en la literatura como *path of solutions*).

En el método del descenso coordinado se considera, en principio, una sola variable predictora: x_{i1} y la variable respuesta $y_i, i = 1,\ldots,n$. Se asume que x_{i1} está estandarizada: $\sum_{i=1}^{n} \frac{x_{i1}}{n} = 0$ y $\sum_{i=1}^{n} x_{i1}^2 = 1$. Entonces, para un λ fijo, $\hat{\beta}^{Lasso}(\lambda) = S(\hat{\beta}, \lambda) = sign(\hat{\beta})(|\hat{\beta}| - \lambda)_+$.

El operador *soft-threshold* tendrá como solución:

$$\hat{\beta} - \lambda \ si \ \hat{\beta} > 0 \ y \ \lambda < \left|\hat{\beta}\right|$$

$$\hat{\beta} + \lambda \ si \ \hat{\beta} < 0 \ y \ \lambda < \left|\hat{\beta}\right|$$

$$0 \ si \ \lambda \geq \left|\hat{\beta}\right|$$

Se consideran ahora varias variables predictoras no correlacionadas. La idea es optimizar $f(\beta) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|$ sucesivamente sobre cada parámetro, manteniendo los demás parámetros fijos en sus valores actuales.

Se denota $\hat{\beta}_k(\lambda)$ a la estimación β_k para un valor de λ. Se puede reordenar la función $f(\beta)$ para aislar β_j: $R\left(\hat{\beta}(\lambda), \beta_j\right) = \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{k\neq j} x_{ik}\hat{\beta}_k(\lambda) - x_{ij}\beta_j\right)^2 + \lambda\sum_{k\neq j}\left|\hat{\beta}_k(\lambda)\right| + \lambda\left|\beta_j\right|$, donde se ha eliminado el término independiente β_0 y dejado el factor $\frac{1}{2}$ por conveniencia. Esto se puede ver como un problema de regresión lineal *Lasso* de una sola variable X_j cuya variable respuesta es el residuo parcial: $y_i - \hat{y}_i^{(j)} = y_i - \sum_{k\neq j} x_{ik}\hat{\beta}_k(\lambda)$ (el valor del residuo después de haber eliminado el efecto de la variable j). Esto tiene una solución explícita:

$$\hat{\beta}_j(\lambda) \leftarrow S\left(\hat{\beta}_j(\lambda) + \sum_{i=1}^{n} x_{ij}\left(y_i - \hat{y}_i^{(j)}\right), \lambda\right),$$

Donde $S(t,\lambda) = sign(t)\left(|t| - \lambda\right)_+$ es, de nuevo, el operador *soft-threshold*. Para su resolución se puede empezar con cualquier valor para β_j como por ejemplo los coeficientes de regresión univariados de mínimos cuadrados. Se puede observar que los valores de $\hat{\beta}_j(\lambda)$ convergen a $\hat{\beta}^{Lasso}$.

El método descenso coordinado proporciona un algoritmo simple y rápido de resolución de problemas Lasso especialmente para valores de *p* grandes [17].

2.3 Un algoritmo de regresión modificado y el algoritmo *Graphical Lasso*

Relación entre las covarianzas parciales y la regresión lineal múltiple

En este apartado se muestra que existe una relación entre el vector de los coeficientes del modelo de regresión lineal, β, y los valores de $\sum^{-1}$, donde se recoge la información de las covarianzas parciales.

Consideramos el modelo de regresión lineal en un contexto diferente al definido en el apartado 2.2.1. En vez de estimar el vector de los parámetros desconocidos, β, aplicando el método de mínimos cuadrados, se estima minimizando el error cuadrático medio. Supongamos que las variables: $Y, Z_1,...,Z_p$ son aleatorias y tienen una distribución normal multivariante, de media μ y matriz de covarianzas $\sum$. Si se considera el problema de predecir Y utilizando un predictor lineal de la forma: $\beta_0 + \sum_{j=1}^{p} Z_j \beta_j$, se deberá elegir aquellos valores de β_0 y β que minimicen el error cuadrático medio: $E\left(Y - \beta_0 - \beta^{'} Z\right)^2$. Cuando los coeficientes toman los valores de: $\beta = \sum^{zz} \sigma_{zy}$ y $\beta_0 = \mu_y - \beta^{'} \mu_z$, el predictor lineal tiene el menor error cuadrático medio (la demostración se puede ver en [10]). Donde se han realizado las particiones de μ, $\sum$ y $\sum^{-1}$:

$\sum = \begin{pmatrix} \sum_{zz} & \sigma_{zy} \\ \sigma_{yz} & \sigma_{yy} \end{pmatrix}$, $\sum^{-1} = \begin{pmatrix} \sum^{zz} & \sigma^{zy} \\ \sigma^{yz} & \sigma^{yy} \end{pmatrix}$ y $\mu = \begin{pmatrix} \mu_y \\ \mu_z \end{pmatrix}$ con $\sigma_{zy} = \left\lfloor \sigma_{yz_1},...,\sigma_{yz_p} \right\rfloor$ y $\sum_{zz}$ de rango completo.

En la expresión de $\beta = \sum^{zz} \sigma_{zy}$ se observa que efectivamente se puede relacionar los valores de los coeficientes del modelo de regresión lineal con la estructura de los valores de $\sum^{-1}$.

Otra forma de expresar el coeficiente de regresión es: $\beta = \sum^{zz} \sigma_{zy} = -\frac{\sigma^{zy}}{\sigma^{yy}}$, como se demostrará más adelante. Al poder realizar la igualdad, $\beta = -\frac{\sigma^{zy}}{\sigma^{yy}}$, se consigue que si un elemento del vector β es cero, lo sea en σ^{zy} y lógicamente en σ^{yz} también aumentando de esta forma el patrón de ceros de $\sum^{-1}$.

Para demostrar que $\beta = \sum^{zz} \sigma_{zy} = -\frac{\sigma^{zy}}{\sigma^{yy}}$ se parte de la igualdad $\sum . \sum^{-1} = I$, de donde se obtiene, entre otras, la ecuación: $\sum_{zz} \sigma^{zy} + \sigma_{zy} \sigma^{yy} = 0$ y, se deduce, que $\sigma^{zy} = -\sigma^{yy} \sum^{zz} \sigma_{zy}$. Sustituyendo $\sigma^{zy} = -\sigma^{yy} \sum^{zz} \sigma_{zy}$ en $\beta = -\frac{\sigma^{zy}}{\sigma^{yy}}$, se tiene: $\beta = -\frac{\sigma^{zy}}{\sigma^{yy}} = \frac{\sigma^{yy} \sum^{zz} \sigma_{zy}}{\sigma^{yy}} = \sum^{zz} \sigma_{zy}$. Por lo que queda demostrado.

Con todo esto, se puede observar que efectivamente las covarianzas parciales y la regresión lineal múltiple están relacionadas y se puede conocer la estructura de las covarianzas parciales, la estructura de $\sum^{-1}$, a partir de la regresión lineal múltiple.

Un algoritmo de regresión modificado para la estimación de un modelo gráfico Gaussiano con estructura del grafo conocida

El algoritmo que se explica a continuación, estima $\sum$ y $\sum^{-1}$ en el contexto de la alta dimensionalidad. Para aplicar este algoritmo se deben conocer de antemano las aristas que están ausentes en el MGG o lo que es lo mismo que valores de $\sum^{-1}$ que son nulos.

Sea: $S = \frac{1}{N}\sum_{\alpha=1}^{N}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$ la matriz de covarianzas muestrales. Se sabe que la función log-verosimilitud de los datos (ignorando las constantes) es:

$$\mathrm{L}(\mu, \Sigma) = -\log|\Sigma| - tr\left(\Sigma^{-1}A\right) - (\bar{x} - \mu)'\Sigma^{-1}(\bar{x} - \mu)$$

Siendo $A = \sum_{\alpha=1}^{N}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$. Maximizando esta función parcialmente con respecto al parámetro μ, tomará la forma:

$$\mathrm{L}(\Sigma) = -\log|\Sigma| - tr\left(\Sigma^{-1}A\right)$$

Haciendo $\Theta = \Sigma^{-1}$, la función log-verosimilitud maximizada parcialmente por el parámetro μ es:

$$\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta)$$

Para hallar el estimador de máxima verosimilitud de $\Theta = \Sigma^{-1}$, habría que maximizar la función $\mathrm{L}(\Theta)$ con respecto a Θ. En este caso se busca hacerlo con la restricción de que un conjunto de parámetros de $\Theta = \Sigma^{-1}$ sean ceros, o lo que es lo mismo, que exista un conjunto de aristas ausentes en el MGG. En el siguiente gráfico (Figura 7) se muestra un ejemplo donde solamente el vértice k y el h se unen al vértice i, con lo que el problema tendría la restricción de que $\sigma^{ij} = 0$.

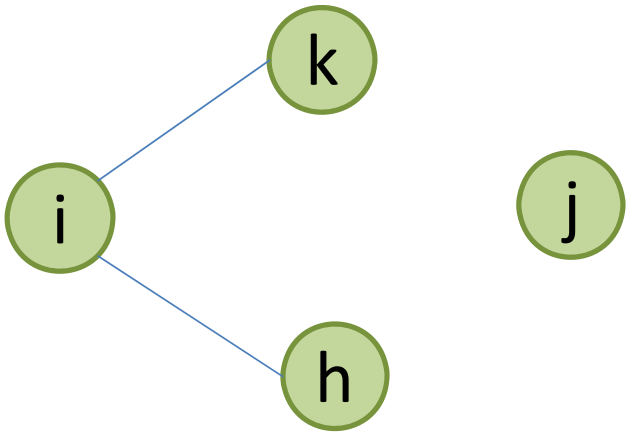

Figura 7 Ejemplo de la estructura de un MGG

Maximizar la función: $\mathrm{L}(\Theta)=\log|\Theta|-tr(S\Theta)$ sujeta a la restricción de que un conjunto de elementos de $\Theta=\Sigma^{-1}$ sean cero equivale a resolver un problema de optimización convexo de Θ con restricciones de igualdad. Para resolver este problema a la función $\mathrm{L}(\Theta)$ se le añaden los multiplicadores de *Lagrange* para todas las aristas ausentes (puesto que son las restricciones del problema). Así, la función queda de la forma:

$$\mathrm{L}(\Theta)=\log|\Theta|-tr(S\Theta)-\sum\gamma_{ik}\sigma^{jk}.$$

Como se busca el máximo de esta función, se deriva con respecto a Θ y se iguala a cero:

$\Theta^{-1}-S-\Gamma=0$ (Ecuación gradiente), donde Γ es la matriz de parámetros de *Lagrange* con valores distintos de cero para todos los pares de vértices con arista ausente.

Resolveremos este problema de optimización a partir de la regresión lineal múltiple. En [18], se muestra que la derivada de $\log|\Theta|$ es igual a Θ^{-1}. Haciendo $W=\Theta^{-1}$ y realizando unas particiones de W, S y Γ, la ecuación gradiente se puede escribir de la forma:

$$\begin{pmatrix}W_{11} & w_{12}\\ w_{21} & w_{22}\end{pmatrix}-\begin{pmatrix}S_{11} & s_{12}\\ s_{21} & s_{22}\end{pmatrix}-\begin{pmatrix}\mathrm{Y}_{11} & \gamma_{12}\\ \gamma_{21} & \gamma_{22}\end{pmatrix}=0,$$

Donde la ecuación del bloque superior derecho es: $w_{12}-s_{12}-\gamma_{12}=0$.

Por otro lado, se sabe que: $WW^{-1}=W\,\Sigma^{-1}=I$. Aplicando a esta igualdad la misma partición queda:

$$\begin{pmatrix}W_{11} & w_{12}\\ w_{21} & w_{22}\end{pmatrix}\begin{pmatrix}\mathrm{T}^{11} & \sigma^{12}\\ \sigma^{21} & \sigma^{22}\end{pmatrix}=\begin{pmatrix}I & 0\\ 0 & 1\end{pmatrix}.$$

De la ecuación del bloque superior derecho, se obtiene el valor de w_{12}: $W_{11}\sigma^{12}+w_{12}\sigma^{22}=0$, de donde se deduce que $w_{12}=-\dfrac{W_{11}\sigma^{12}}{\sigma^{22}}$. En el apartado anterior, se demostró que $\beta=-\dfrac{\sigma^{12}}{\sigma^{22}}$, por lo tanto, se puede expresar w_{12} en función de β: $w_{12}=W_{11}\beta$. Luego, la ecuación de la función gradiente, $w_{12}-s_{12}-\gamma_{12}=0$,

también se puede escribir en función de β: $W_{11}\beta - s_{12} - \gamma_{12} = 0$ que se interpreta como las p-1 ecuaciones de regresión con restricción para la estimación de X_p en función de las restantes $X_1,...,X_{p-1}$ variables excepto que en vez de S_{11} se utiliza W_{11}, que es la actual matriz de covarianzas del modelo. Se resuelve la ecuación: $W_{11}\beta - s_{12} - \gamma_{12} = 0$ la cual nos permitirá estimar los w_{12} que dan la solución al objetivo de hallar: $\hat{\Sigma} = \hat{W}$. Se suponen *p-q* elementos distintos de cero en γ_{12}, es decir, *p-q* aristas restringidas a tener valor cero. Esas *p-q* filas, carecen entonces de información y se pueden eliminar de W_{11} dando lugar a W_{11}^*. De la misma forma se reduce β a β^*. Nos queda, entonces, un sistema de ecuaciones *q* x *q*: $W_{11}^*\beta^* - s_{12}^* = 0$.

El vector final $\hat{\beta}$ se conseguirá rellenando con *p-q* ceros el vector $\hat{\beta}^*$ en las posiciones correspondientes.

Teniendo todo esto en cuenta, se construye un algoritmo iterativo que estima $\hat{W}$ y $\hat{\Theta}$ sujetos a las restricciones de las aristas ausentes:

Algoritmo de regresión modificado para la estimación de un modelo gráfico Gaussiano con estructura del grafo conocida:

1. Se inicializa $W = S$
2. Se repite para j=1, 2, …, *p* hasta que converja:
   (a) Realizamos la partición de W en dos partes: Parte 1: Todas las filas y columnas que no sean la j-ésima posición. Parte dos: La j-ésima fila y columna.
   (b) Se resuelve $W_{11}^*\beta^* - s_{12}^* = 0$ para las aristas no ausentes del gráfico mediante la regresión lineal múltiple. Obteniendo $\hat{\beta}$ rellenando el vector $\hat{\beta}^*$ con ceros en las posiciones correspondientes.
   (c) Actualizamos de la forma: $w_{12} = W_{11}\hat{\beta}$
3. En el ciclo final, para cada j, resolvemos $\hat{\sigma}^{12} = -\hat{\beta}\hat{\sigma}^{22}$ sabiendo que $\frac{1}{\hat{\sigma}^{22}} = s_{22} - w_{12}^{'}\hat{\beta}$.

Algoritmo 2 Algoritmo de regresión modificado para la estimación de un MGG con estructura del grafo conocida

La igualdad del paso 3 del algoritmo 2 se deriva de la ecuación $W.\sum^{-1} = I$ a través de las ecuaciones que nos proporciona: $W_{11}\sigma^{12} + w_{12}\sigma^{22} = 0$ y $w_{21}\sigma^{12} + w_{22}\sigma^{22} = 1$. De la primera ecuación se deduce: $\sigma^{12} = -\frac{w_{12}\sigma^{22}}{W_{11}} = -\beta\sigma^{22}$ que sustituyendo en la segunda ecuación queda: $w_{21}\left(-\beta\sigma^{22}\right) + w_{22}\sigma^{22} = 1 \Rightarrow \sigma^{22}\left(-\beta w_{21} + w_{22}\right) = 1 \Rightarrow \frac{1}{\sigma^{22}} = w_{22} - \beta w_{21} = w_{22} - \beta w_{12}$. Por otro lado se sabe que $w_{22} = s_{22}$ puesto que la diagonal principal de Γ es cero. De todo esto se deduce que:

$\frac{1}{\hat{\sigma}^{22}} = s_{22} - w_{12}^{'}\hat{\beta}$, que es la ecuación del paso 3.Este último paso nos permite llegar al objetivo de estimar $\sum^{-1}$.

2.3.1.1 Resolución del ejemplo de Whittaker (1990)

En este apartado se resuelve el problema propuesto por [19] aplicando el algoritmo explicado en el apartado 2.3.2, de regresión modificado para la estimación de un modelo gráfico Gaussiano con estructura del grafo conocida. El modelo gráfico Gaussiano es el siguiente: V_3

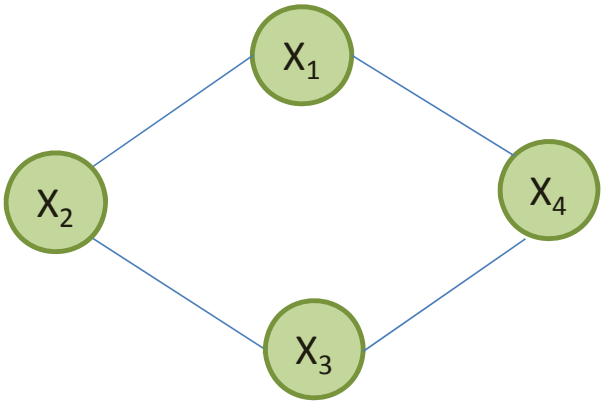

Figura 8 MGG del ejemplo de Whittaker (1990)

La matriz de covarianzas muestrales es:

$$S = \begin{pmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

Se conoce de antemano que posiciones de la matriz $\hat{\sum}^{-1}$ son ceros: σ^{13}, σ^{31}, σ^{24}, σ^{42}. El resto de los valores de la matriz $\hat{\sum}^{-1}$ se obtendrán del algoritmo. Así mismo, los valores de $\hat{\sum}$ serán iguales a los de S exceptuando las posiciones: w_{13}, w_{31}, w_{24}, w_{42} que también se obtendrán a partir del algoritmo.

A continuación se detallan los pasos:

Paso 1

Inicializamos $W = S$:

$$W = \begin{pmatrix} 10 & 1 & 5 & 4 \\ 1 & 10 & 2 & 6 \\ 5 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

Paso 2

Caso de j=1.

La partición de W es:

$$W_{11} = \begin{pmatrix} 10 & 2 & 6 \\ 2 & 10 & 3 \\ 6 & 3 & 10 \end{pmatrix}, w_{12} = w_{21} = \begin{pmatrix} 1 \\ 5 \\ 4 \end{pmatrix}, \; w_{22} = 10$$

Se busca β_2 y β_4 tal que $X_1 = \beta_0 + \beta_2 X_2 + \beta_4 X_4$. En este caso, la variable 3 es excluida (arista ausente).

$$\begin{pmatrix} 10 & 6 \\ 6 & 10 \end{pmatrix}\begin{pmatrix} \beta_2 \\ \beta_4 \end{pmatrix} - \begin{pmatrix} 1 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \beta_4 = 0.53 y \beta_2 = -0.22$$

Se actualiza w_{12}: $w_{12} = W_{11}\hat{\beta}$

$$\begin{pmatrix} 10 & 2 & 6 \\ 2 & 10 & 3 \\ 6 & 3 & 10 \end{pmatrix}\begin{pmatrix} -0.22 \\ 0 \\ 0.53 \end{pmatrix} = \begin{pmatrix} 1 \\ 1.16 \\ 4 \end{pmatrix}$$

Por lo que la nueva W es:

$$W = \begin{pmatrix} 10 & 1 & 1.16 & 4 \\ 1 & 10 & 2 & 6 \\ 1.16 & 2 & 10 & 3 \\ 4 & 6 & 3 & 10 \end{pmatrix}$$

Se repite el procedimiento para j=2, j=3 y j=4

El algoritmo converge rápidamente. Colocando cada w_{ij} en su lugar correspondiente se obtiene $\hat{\Sigma}$:

$$W = \hat{\Sigma} = \begin{pmatrix} 10 & 1 & 1.31 & 4 \\ 1 & 10 & 2 & 0.87 \\ 1.31 & 2 & 10 & 3 \\ 4 & 0.87 & 3 & 10 \end{pmatrix}$$

A partir del paso nº 3 del algoritmo, se calcula $\hat{\Sigma}^{-1}$:

Paso 3

Caso de j=1.

$\frac{1}{\hat{\sigma}^{22}} = s_{22} - w_{12}^{'}\hat{\beta} = 10 - \begin{pmatrix}1 & 1.16 & 4\end{pmatrix}\begin{pmatrix}-0.22\\0\\0.53\end{pmatrix} = 10 - 1.9 = 8.1 \rightarrow \hat{\sigma}^{22} = 0.12$. Es decir, la posición $\sigma^{11} = 0.12$

$$\hat{\sigma}^{12} = -0.12\begin{pmatrix}-0.22\\0\\0.53\end{pmatrix} = \begin{pmatrix}0.03\\0\\-0.05\end{pmatrix} = \begin{pmatrix}\hat{\sigma}^{12} = \hat{\sigma}^{21}\\\hat{\sigma}^{13} = \hat{\sigma}^{31}\\\hat{\sigma}^{14} = \hat{\sigma}^{41}\end{pmatrix}$$

Se repite el procedimiento para j=2, j=3 y j=4

Recopilando toda la información del paso 3, se obtiene $\hat{\sum}^{-1}$:

$$\hat{\sum}^{-1} = \begin{pmatrix}0.12 & -0.01 & 0 & -0.05\\-0.01 & 0.11 & -0.02 & 0\\0 & -0.02 & 0.11 & -0.03\\-0.05 & 0 & -0.03 & 0.13\end{pmatrix}$$

El algoritmo de regresión modificado se puede resolver con el programa estadístico R y el paquete *glasso* utilizando el argumento opcional: *zero,* que es un argumento que permite indicar que valores σ^{ij} son cero por lo que es un argumento útil cuando se sabe de antemano que aristas del grafo están ausentes (la programación se puede consultar en el anexo I).

La salida obtenida con el programa estadístico R es la siguiente:

```
$w
          [,1]       [,2]       [,3]       [,4]
[1,] 10.0000000  0.9999919  1.314205  3.9999991
[2,]  0.9999919 10.0000000  2.000001  0.8704694
[3,]  1.3142047  2.0000015 10.000000  3.0000000
[4,]  3.9999991  0.8704694  3.000000 10.0000000
$wi
           [,1]         [,2]          [,3]         [,4]
[1,]  0.11965735 -0.007858889  0.00000000 -0.04717890
[2,] -0.00785844  0.104770128 -0.01992122  0.00000000
[3,]  0.00000000 -0.019921213  0.11369673 -0.03237498
[4,] -0.04717880  0.000000000 -0.03237494  0.12858405
```

La primera matriz es la estimación de $\hat{\sum} = W$ y la segunda la de $\hat{\sum}^{-1}$.

Como ya se ha comentado, conocer de antemano la estructura del modelo gráfico no dirigido no suele suceder. Como solución a esta cuestión está el algoritmo *Graphical Lasso* el cual se explica en el apartado siguiente.

El algoritmo Graphical Lasso

2.3.1.2 El método propuesto

Como ya se ha visto, el objetivo fundamental de la penalización *Lasso* es inducir *sparsity* en el modelo de regresión lineal y por lo tanto conseguir el mayor número de coeficientes de regresión lineal con valor de cero. Además, en el apartado 2.3.1 se explicó que si un elemento del vector β era cero, por ejemplo el coeficiente de regresión de la variable independiente Z_1, lo es también el valor de las correspondientes componentes de $\sum^{-1}$: $\sigma^{Z_1 Y}$ y $\sigma^{Y Z_1}$. Por todo ello se puede decir que el objetivo de la penalización *Lasso* es también incrementar el nº de ceros de la matriz $\sum^{-1}$(estimación dispersa de $\sum^{-1}$). A partir de este razonamiento, muchos autores han propuesto el uso de la regularización *Lasso* para estimar $\sum$, $\sum^{-1}$ y su correspondiente MGG cuando *p* supera a *n*.

Una de las propuestas se presenta en [20] donde se estima un modelo gráfico no dirigido ajustando un modelo de regresión lineal *Lasso* a cada una de las *p* variables, usando las restantes como variables predictoras, en este caso, $\hat{\sigma}_{ij} \neq 0$ si $\hat{\beta}_{ij} \neq 0 y \hat{\beta}_{ji} \neq 0$, es decir, $\hat{\beta}$ tiene el mismo patrón de ceros que $\hat{\sum}^{-1}$, además se muestra que esta estimación del patrón de ceros de $\sum^{-1}$ es asintóticamente consistente.

En la propuesta [21], que es la que se desarrolla en este trabajo, se propone encontrar el máximo de la función log-verosimilitud parcialmente maximizada por el parámetro μ con una penalización *Lasso* con respecto a $\Theta = \Sigma^{-1}$.

$$\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$$

A este problema se le denomina problema *Graphical Lasso* [22]. Usando la función log-verosimilitud parcialmente maximizada por el parámetro μ con una penalización *Lasso,* se consigue que la solución $\hat{\Theta}$ sea una matriz definida positiva para todo $\lambda > 0$ incluso si S es singular y además, cuando λ es lo suficientemente grande, la estimación de Θ será dispersa debido a la penalización *Lasso* en los elementos de Θ [14]. El algoritmo que se explica en este apartado para resolver este problema *Graphical Lasso* es el que se propone en [22] el cual utiliza la propuesta del método del descenso coordinado del trabajo de [23] para la estimación de $\sum$ (el objetivo de este trabajo), Una vez estimada $\sum$ se puede hallar la estimación de $\sum^{-1}$ y su correspondiente MGG.

Se considera una penalización *Lasso* en la función log-verosimilitud parcialmente maximizada por el parámetro μ :

$$\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1,$$

Donde $\|\Theta\|_1$ es la norma L_1 (la suma de los valores absolutos de los elementos de $\sum^{-1}$). Se busca encontrar el máximo de esta función, $\mathrm{L}(\Theta)$, y de esta forma hallar el estimador de máxima verosimilitud de $\Theta = \Sigma^{-1}$ penalizado. Una condición necesaria y suficiente para que $\Theta = \Sigma^{-1}$ maximice nuestro problema *Graphical Lasso* es que satisfaga la ecuación:

$$\Theta^{-1} - S - \lambda\Gamma(\Theta) = 0,$$

Esta es la ecuación sub-gradiente puesto que la función en $\Theta = 0$ no es diferenciable (ver [18]). El elemento $\Gamma(\Theta)$ es el subgradiente de $|\Theta|$ y su resultado es $sign(\sigma^{ik}) = sign(\sigma^{jk})$ sí $\sigma^{jk} \neq 0$ y, si $\sigma^{jk} = 0$, $sign(\sigma^{ik}) \in [-1,1]$.

La ecuación del bloque superior derecho, en este caso es: $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$ (ya que β y σ^{12} tienen signos opuestos: $\beta = -\frac{\sigma^{12}}{\sigma^{22}}$).

A continuación se resuelve este problema de optimización a partir de la regresión lineal múltiple ya que las ecuaciones del sistema, $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$, son equivalentes a las ecuaciones estimadas para una regresión *Lasso.* Se explica a continuación.

Se considera una regresión lineal *Lasso* con Y la variable dependiente y $X_1,\ldots,X_p$ el vector de las variables aleatorias predictoras. La solución *Lasso* para los β_j consiste, como se ha visto en el apartado 1.2.3, en minimizar la función:

$$SRCP(\beta) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \frac{1}{2}\|Y - X\beta\|^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \frac{1}{2}(Y - X\beta)'(Y - X\beta) + \lambda\|\beta\|_1$$

Derivando con respecto a β e igualando a cero queda la expresión: $X'X\beta - X'Y + \lambda Sign(\beta) = 0$ (ecuación gradiente). Entonces, por analogía con: $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$, hasta un factor $\frac{1}{n}$, $X'Y$ es el análogo de s_{12} y $X'X$ es W_{11}, la actual matriz de covarianzas del modelo. De esta forma se ve que se puede resolver el sistema $W_{11}\beta - s_{12} + \lambda Sign(\beta) = 0$ a partir de la regresión lineal *Lasso.*

Teniendo todo esto en cuenta, se observa que reemplazando el paso 2b del algoritmo de regresión modificado por un paso de regresión lineal *Lasso* modificado, se obtiene un procedimiento para resolver el actual problema de encontrar el máximo de la actual función $L(\Theta)$. Además, el problema *Lasso* del paso 2b se puede resolver utilizando el método del descenso coordinado ([22], [24]) visto en el apartado 2.2.3: Para este caso se toma $V = W_{11}$ y $u = s_{12}$, la solución será de la forma:

$$\hat{\beta}_j \leftarrow S\left(u_j - \sum_{k \neq j} V_{kj}\hat{\beta}_k, \lambda\right)/V_{jj}$$

La ecuación anterior para j=1, 2,…,*p*, 1, 2, …, *p*, …, donde $S(t,\lambda) = sign(t)(|t| - \lambda)_+$ es el operador *soft-threshold*.

A todo este procedimiento se le denomina: Algoritmo *Graphical Lasso* [22] y se resume en el algoritmo 3.

Algoritmo Graphical Lasso

1. Se inicializa $W = S + \lambda I$
2. Se repite para j=1, 2, …, *p* hasta que converja:
   (a) Realizamos la partición de W en dos partes: Parte 1: Todas las filas y columnas que no sean la j-ésima posición. Parte dos: La j-ésima fila y columna.
   (b) Se resuelve el problema de regresión lineal *Lasso*: $W_{11}\beta - s_{12} + \lambda sign(\beta) = 0$ mediante el método del descenso coordinado.
   (c) Actualizamos de la forma: $w_{12} = W_{11}\hat{\beta}$
   (d) En el ciclo final, para cada j, resolvemos $\hat{\sigma}^{12} = -\hat{\beta}\hat{\sigma}^{22}$ sabiendo que $\frac{1}{\hat{\sigma}^{22}} = s_{22} - w_{12}^{'}\hat{\beta}$.

Algoritmo 3 Algoritmo *Graphical Lasso*

Podríamos considerar en el algoritmo presentado que $W = S$, obteniendo para este caso la solución $\hat{\beta}$ de la estimación *Lasso* de la p_ésima variable a partir de las *p*-1 variables restantes que es justo la propuesta de [20] pero, tal y como señala [23], $W \neq S$ en general y por lo tanto no obtendríamos como resultado el estimador de máxima verosimilitud (penalizado). La propuesta de [20] puede verse entonces como una aproximación a la solución del problema de maximizar $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$. (ver [21] y [22]) mientras que el algoritmo aquí presentado, que no utiliza la restricción $W = S$ sino $W = S + \lambda I$, es más potente puesto que si calcula la solución exacta, es decir, el estimador de máxima verosimilitud (penalizado). En [22], en el apartado de las comparaciones de tiempo, se comenta, que a pesar de que [20] no proporcionar la solución exacta, es bastante más rápido que el algoritmo *Graphical Lasso*.

2.3.1.3 Resolución de simulación

Para ilustrar el procedimiento, se muestra la resolución paso a paso del algoritmo *Graphical Lasso* aplicado a una simulación de 3 variables normales con $\mu = 0$ y $\Sigma = 1$, para un valor del parámetro de regularización de 0.5 y del umbral fijo *t*=0.001:

```
#tres var con cinco obs de una distribución normal multivariante de media cero y matriz de cov 1

n <- 5
mu <- c(0,0,0)
sigma <- matrix(c(1,0,0,0,1,0,0,0,1),3)
datos <- mvrnorm(n,mu,sigma)
```

Los datos obtenidos se muestran en la tabla 1:

	1	**2**	**3**
1	-1.52	1.09	1.56
2	0.21	0.80	-1.87
3	-1.51	-1.54	0.89
4	0.15	-0.88	1.45
5	0.11	-0.76	0.04

Tabla 1 Datos de una simulación de una distribución normal de media $\mu = 0$ y matriz de covarianzas $\Sigma = 1$

La matriz de covarianzas muestrales:

```
cdatos<-cov(datos)
```

```
          [,1]        [,2]        [,3]
[1,]  0.839620 -0.009870 -0.706515
[2,] -0.009870  1.304720 -0.484635
[3,] -0.706515 -0.484635  1.992430
```

A continuación se detallan los pasos del algoritmo *Graphical Lasso* para un valor del parámetro de regularización de 0.5 y del umbral fijo *t*=0.001:

Paso 1

Inicializamos W

```
LI<-matrix(c(0.5,0,0,0,0.5,0,0,0,0.5),3)
W<-cdatos+LI
```

$$W = \begin{pmatrix} 1.34 & -0.01 & -0.71 \\ -0.01 & 1.80 & -0.48 \\ -0.71 & -0.48 & 2.50 \end{pmatrix}$$

Paso 2

Caso de j=1.

La partición de W es:

$$W_{11} = \begin{pmatrix} 1.80 & -0.48 \\ -0.48 & 2.50 \end{pmatrix}, w_{12} = w_{21} = \begin{pmatrix} -0.01 \\ -0.71 \end{pmatrix}, \; w_{22} = 1.34$$

Se busca β_2^{Lasso} y β_3^{Lasso} tal que $X_1 = \beta_0 + \beta_2^{Lasso} X_2 + \beta_3^{Lasso} X_3$. Para ello se aplica el método del descenso coordinado:

Se estiman los coeficientes de regresión univariados por el método de mínimos cuadrados:

$$X_1 = \beta_0 + \beta_2 X_2$$

```
lm12<-lm(datos[,1]~datos[,2])

Coefficients:
(Intercept)   datos[, 2]
  -0.513952    -0.007565
```

$$X_1 = \beta_0 + \beta_3 X_3$$

```
lm13<-lm(datos[,1]~datos[,3])

Coefficients:
(Intercept)   datos[, 3]
   -0.3652      -0.3546
```

Los datos necesarios para la aplicación del método del descenso coordinado:

$$V = W_{11} = \begin{pmatrix} 1.80 & -0.48 \\ -0.48 & 2.50 \end{pmatrix},\ u = s_{21} = \begin{pmatrix} -0.01 \\ -0.71 \end{pmatrix},\ \hat{\beta}_2^{MC} = -0.008,\ \hat{\beta}_3^{MC} = -0.35$$

$$\hat{\beta}_2 \leftarrow S(u_2 - V_{32}\hat{\beta}_3, 0.5)/1.80 = S\left(-0.01 - 0.48 * 0.35, 0.5\right)/1.80 = S(-0.18, 0.5)/1.80 = 0$$

Ya que $\lambda \geq |\hat{t}|$

$$\hat{\beta}_3 \leftarrow S(u_3 - V_{23}\hat{\beta}_2, 0.5)/2.50 = S\left(-0.71 - 0.48 * 0.008, 0.5\right)/2.50 = S(-0.71, 0.5)/2.5 = \frac{-0.21}{2.5} = -0.08$$

ya que $\hat{t} < 0$ y $\lambda < |\hat{t}|$

Se actualiza w_{12}: $w_{12} = W_{11}\hat{\beta}$

$$\begin{pmatrix} 1.80 & -0.48 \\ -0.48 & 2.50 \end{pmatrix}\begin{pmatrix} 0 \\ -0.08 \end{pmatrix} = \begin{pmatrix} 0.04 \\ -0.2 \end{pmatrix}$$

La nueva W es:

$$W = \begin{pmatrix} 1.34 & -0.04 & -0.2 \\ -0.04 & 1.80 & -0.48 \\ -0.2 & -0.48 & 2.50 \end{pmatrix}$$

Caso j=2.

La partición de W es:

$$W_{11} = \begin{pmatrix} 1.34 & -0.2 \\ -0.2 & 2.50 \end{pmatrix},\ w_{12} = w_{21} = \begin{pmatrix} -0.04 \\ -0.48 \end{pmatrix},\ w_{22} = 1.80$$

Se busca β_1^{Lasso} y β_3^{Lasso} tal que $X_2 = \beta_0 + \beta_1^{Lasso} X_1 + \beta_3^{Lasso} X_3$. Para ello se aplica el método del descenso coordinado:

Los coeficientes de mínimos cuadrados son:

```
lm21<-lm(datos[,2]~datos[,1])

Coefficients:
(Intercept)   datos[, 1]
  -0.26402     -0.01176

lm23<-lm(datos[,2]~datos[,3])

Coefficients:
(Intercept)   datos[, 3]
   -0.1573      -0.2432
```

$$V = W_{11} = \begin{pmatrix} 1.34 & -0.2 \\ -0.2 & 2.50 \end{pmatrix},\ u = s_{21} = \begin{pmatrix} -0.01 \\ -0.48 \end{pmatrix},\ \hat{\beta}_1^{MC} = -0.01,\ \hat{\beta}_3^{MC} = -0.24$$

$\hat{\beta}_1 \leftarrow S(-0.01 - 0.2 * 0.24, 0.5)/1.34 = S(-0.18, 0.5)/1.34 = 0$ ya que $\lambda \geq |\hat{t}|$

$\hat{\beta}_3 \leftarrow S(-0.48 + 0.2\hat{\beta}_1, 0.5)/2.50 = S(-0.48, 0.5)/2.5 = \dfrac{-0.21}{2.5} = -0.08$ ya que $\hat{t} < 0$ y $\lambda < |\hat{t}|$

Se actualiza w_{12}: $w_{12} = W_{11}\hat{\beta}$

$$\begin{pmatrix} 1.34 & -0.2 \\ -0.2 & 2.50 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

El algoritmo converge enseguida a la nueva W:

$$W = \hat{\Sigma} = \begin{pmatrix} 1.34 & 0 & -0.2 \\ 0 & 1.80 & 0 \\ -0.2 & 0 & 2.50 \end{pmatrix}$$

A partir del paso nº 3 del algoritmo, se calcula $\hat{\Sigma}^{-1}$:

Paso 3

Caso j=1.

$\dfrac{1}{\hat{\sigma}^{22}} = w_{22} - w_{12}^{'}\hat{\beta} = 1.34 - \begin{pmatrix} 0.04 & -0.2 \end{pmatrix}\begin{pmatrix} 0 \\ -0.08 \end{pmatrix} = 1.34 - 0.016 = 1.32 \rightarrow \hat{\sigma}^{22} = 0.76$. Es decir, la posición $\sigma^{11} = 0.76$

$$\hat{\sigma}^{12} = -0.76\begin{pmatrix} 0 \\ -0.08 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.06 \end{pmatrix} = \begin{pmatrix} \hat{\sigma}^{12} = \hat{\sigma}^{21} \\ \hat{\sigma}^{13} = \hat{\sigma}^{31} \end{pmatrix}$$

Caso j=2.

$\frac{1}{\hat{\sigma}^{22}} = w_{22} - w_{12}^{'}\hat{\beta} = 1.80 - \begin{pmatrix}0 & 0\end{pmatrix}\begin{pmatrix}0\\0\end{pmatrix} = 1.80 \rightarrow \hat{\sigma}^{22} = 0.56$. Es decir, la posición $\sigma^{22} = 0.56$

$$\hat{\sigma}^{12} = -0.56\begin{pmatrix}0\\0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} = \begin{pmatrix}\hat{\sigma}^{21} = \hat{\sigma}^{12}\\ \hat{\sigma}^{23} = \hat{\sigma}^{32}\end{pmatrix}$$

$$\sigma^{33} = \frac{1}{w_{33}} = \frac{1}{2.5} = 0.4$$

Recopilando toda la información del paso 3, se obtiene $\hat{\sum}^{-1}$:

$$\hat{\sum}^{-1} = \begin{pmatrix}0.76 & 0 & 0.06\\ 0 & 0.56 & 0\\ 0.06 & 0 & 0.4\end{pmatrix}$$

Se resuelve con R:

```
glasso(cdatos,rho=0.5) #por defecto t=0.001

$w
      [,1] [,2]  [,3]
[1,]  1.34  0.0 -0.21
[2,]  0.00  1.8  0.00
[3,] -0.21  0.0  2.50

$wi
        [,1]         [,2]         [,3]
[1,] 0.75622372 0.0000000 0.06352279
[2,] 0.00000000 0.5555556 0.00000000
[3,] 0.06352279 0.0000000 0.40533591
```

2.3.1.4 Estimación de λ y t. Método de la validación cruzada

Para resolver el algoritmo *Graphical Lasso*, se necesita estimar los valores de '*t*' y 'λ'. En el ejemplo anterior estaban fijados a 0.5 el parámetro de regularización y el umbral fijo *t* a 0.001. Como valor de '*t*' se toma por defecto $t = 0.001$ [22]. En este apartado estudiaremos en detalle el método a seguir para estimar el valor de λ. Uno de los métodos más utilizado para su estimación, es el método de la validación cruzada [25] el cual se explica a continuación.

Dado un estimador, $\hat{f}$, construido a partir de un conjunto de datos: (x_i, y_i) i=1, …, *n*. Se considera el error cuadrático medio de este estimador: $EP(\hat{f}) = E\left[\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{'} - \hat{f}(x_i)\right)^2\right]$, donde $y_i^{'} = f(x_i) + \varepsilon_i^{'}$, i=1, …, *n* son otro conjunto de observaciones independientes a las observaciones de partida: $y_i, \ldots, y_n$.

Se supone que $\hat{f} = \hat{f}_\theta$, es decir, que el estimador actual depende de un parámetro desconocido θ. El objetivo es elegir aquel θ que minimice la función: $EP(\hat{f}_\theta)$.

Sea $y_1,...,y_n$ un conjunto de datos llamados *training data* e $y_1^{'},...,y_n^{'}$ un conjunto de datos llamados *test data* (datos que no son los que se utilizan para construir $\hat{f}_\theta$). Entonces, se utiliza $TestError\left(\hat{f}_\theta\right)=\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{'}-\hat{f}_\theta(x_i)\right)^2$ como estimación de $EP\left(\hat{f}_\theta\right)$. Pero, en la mayoría de los casos, no se tiene un conjunto de *test data* y por lo tanto no se podría estimar este *test error*.

Con el método de la validación cruzada se puede estimar el error de predicción, $EP\left(\hat{f}_\theta\right)$: Se parte de un conjunto de datos: $y_i, x_{1i},...,x_{pi}$, i=1, …, *n* y un estimador dependiente de un parámetro desconocido: $\hat{f}_\theta$. Cuando θ sea un parámetro continuo no será posible considerar todos sus valores ya que se trata al parámetro como un parámetro discreto: $\{\theta_1,...,\theta_m\}$. El algoritmo consiste en ir probando con cada uno de estos valores a ver cuál es el que consigue el menor error de predicción.

Para poder hacer la validación cruzada se fija un determinado número, *K*, y se dividen los datos en *K* partes iguales (a este método se le conoce como *K-fold cross validation*), generalmente *K* es 5 o 10. El método tiene en cuenta todos los datos a excepción de los del k-ésimo grupo (conjunto *training),* y a continuación valida el estimador en el k-ésimo grupo (conjunto *validation)* iterando de esta forma *K* veces k=1,…,K. Entonces, para cada uno de los posibles valores de los parámetros desconocidos $\theta\in\{\theta_1,...,\theta_m\}$ se calcula el estimador $\hat{f}_\theta^{-k}$ con el conjunto '*training'* y se calcula el error total de predicción sobre el conjunto *validation.* Se calcula: $e_k(\theta)=\sum_{i\in training}\left(y_i-\hat{f}_\theta^{-k}(x_i)\right)^2$ y luego, para cada uno de los valores del parámetro desconocido θ, se calcula la media de todos los errores de predicción calculados:

$$CV(\theta)=\frac{1}{n}\sum_{k=1}^{K}e_k(\theta)=\frac{1}{n}\sum_{k=1}^{K}\sum_{i\in training}\left(y_i-\hat{f}_\theta^{-k}(x_i)\right)^2$$

El valor del parámetro desconocido óptimo será aquel minimice $CV(\theta)$.

En el caso del algoritmo *Graphical Lasso*, para el caso del parámetro λ, se trata de estimar el valor que maximiza la función $\mathrm{L}(\Theta)=\log|\Theta|-tr(S\Theta)-\lambda\|\Theta\|_1$ para diferentes valores de λ y para cada uno de estos valores de λ se calcula el máximo de la función $\mathrm{L}(\Theta)$ con el conjunto *training* y se calcula el error total de predición sobre el conjunto *validation.* El algoritmo consiste en iterar de esta forma k veces k=1,…,K y luego, para cada uno de los valores de λ se calcula la media de todos los errores de predicción calculados:

$$CV(\lambda)=\frac{1}{n}\sum_{k=1}^{K}e_k(\lambda)=\frac{1}{n}\sum_{k=1}^{K}\sum_{i\in trainingdata}\left(y_i-\hat{\mathrm{L}}(\Theta)_\lambda^{-k}(x_i)\right)^2.$$

Y se elige como valor óptimo el que minimice $CV(\lambda)$.

Este método se utiliza mucho para estimar el valor del parámetro de regularización, aunque se considera un método computacionalmente intenso [17].

En la actualidad, el paquete de R huge proporciona otros métodos de selección del parámetro de regularización basados en criterios de estabilidad.

2.3.1.5 Ejemplo final con los paquetes de R 'huge' y 'glasso'

En el apartado 2.3.3.2 se realizaba una simulación paso a paso del algoritmo *Graphical Lasso.* Sin embargo, el programa estadístico R tiene los paquetes huge y `glasso` que pueden realizar el algoritmo *Graphical Lasso* y estimar grafos no dirigidos para bases de datos de alta dimensión, como veremos en este apartado.

Con ambos paquetes aparte de realizar el algoritmo *Graphical Lasso* de la propuesta de [22] se puede obtener la estimación de un grafo no dirigido mediante la propuesta de [20] y con ambos paquetes se puede construir un *path of values* para diferentes valores de λ, esto es, la estimación de un MGG para un conjunto de diferentes valores de λ.

Algunas de las diferencias entre los dos paquetes son que la programación del paquete `glasso` es en *fortan* y la de huge en *C* lo que hace que este último sea más portable y fácil de modificar. El paquete huge ofrece una serie de opciones adicionales como la función `huge.generator()` que genera datos multivariantes de una población Gaussiana o el argumento `scr` en la función `huge()` que permite filtrar las variables y reducir la alta dimensionalidad antes de estimar el grafo. El paquete huge propone tres procedimientos para seleccionar el parámetro de regularización: El procedimiento *stars* [26] que consiste en elegir el valor óptimo de λ a partir de sub muestreos y según la variabilidad observada (nivel de dispersión), el procedimiento *ric* [27], un método eficiente desarrollado recientemente que se basa en estimar λ a partir de rotaciones aleatorias y por último el criterio de información bayesiano ampliado [28].

A continuación se muestra un ejemplo de estimación de $\sum$, $\sum^{-1}$ y del correspondiente MGG aplicado a una base de datos de alta dimensionalidad utilizando ambos paquetes. Se usa el paquete huge para estimar el *path of values* que nos proporcionará un primer acercamiento a la relación entre los valores de λ y el grado de dispersión que se alcanza y, más tarde, a partir del *path of values* , se estima el valor óptimo de λ y el MGG asociado habiendo reducido la dimensión de la base de datos ya que, como se verá, no es posible hallar el valor óptimo de λ con una dimensión tan alta (se utiliza una base de datos con 3883 variables). Una vez estimado el valor óptimo de λ y su MGG asociado, se procede a estimar $\sum$ y $\sum^{-1}$ a partir del paquete `glasso`. Al final y con el objetivo de mostrar la eficacia del algoritmo *Graphical Lasso* se utiliza la función `rank.condition` del paquete de R `corpcor` para conocer el *ratio* entre el mayor y el menor valor singular y comprobar de esta forma, que en el caso de $\hat{\sum}$ mediante el algoritmo *Graphical lasso* el valor de este ratio es menor en comparación al caso de S.

La base de datos de alta dimensión que se ha utilizado es: *Breast cancer data* la cual contiene 49 observaciones y 3883 variables y se utiliza ampliamente en la literatura especializada como un conjunto de datos óptimos para procedimientos que sirven para el caso Gaussiano.

Con la función huge del paquete huge se estima un *path of values* para diferentes valores de λ:

```
huge(x, lambda = NULL, nlambda = NULL, lambda.min.ratio = NULL, method = "mb",
scr = NULL, scr.num = NULL, cov.output = FALSE, sym = "or", verbose = TRUE)
```

x: Puede ser una matriz de los datos *n* x *p* o la matriz de covarianzas muestrales de los datos.

`lambda`: Se puede dar un conjunto de valores positivos para λ en orden decreciente. Generalmente se utiliza *lambda = NULL* y se deja al programa que seleccione su propio conjunto de valores.

`nlambda`: Se indica el número de valores de λ que se quiere para construir el *path of values,* por defecto es 30 para los métodos del *Graphical Lasso* y de *Meiushausen & Bühlmann.*

`lambda.min.ratio`: Se indica el valor más pequeño para *lambda.* Por defecto es 0.1 para los métodos del *Graphical Lasso* y de *Meiushausen & Bühlmann.*

`method`: Se utiliza la opción *"mb"* para el caso de *Meiushausen & Bühlmann* y la opción "glasso" para la estimación del algoritmo *Graphical Lasso.*

`scr`: Se utiliza *scr = TRUE* si se quiere preseleccionar variables antes de realizar el algoritmo y reducir la alta dimesionalidad.

`scr.num`: Solamente se utiliza cuando *scr = TRUE* y *method="mb"* para indicar el nº de nodos vecinos que se quiere por nodo.

`cov.output`: Si se utiliza *cov.output = TRUE* la salida incluye un *path* de matrices de covarianzas estimadas aunque, puesto que no suelen ser dispersas, puede utilizar mucha memoria.

`sym`: Opción solo aplicable al caso de *method="mb"*.

`verbose`: Si no se quiere imprimir la información se pone *verbose = FALSE.*

A continuación se muestra la estimación de un *path of values* de la base de datos: *Breast cancer data:*

```
is.matrix(west.mat.clean)#true
isSymmetric(west.mat.clean)#false
covar<-cov(west.mat.clean)
Path<-huge(covar,nlambda=30,method ="glasso")
plot(Path)
```

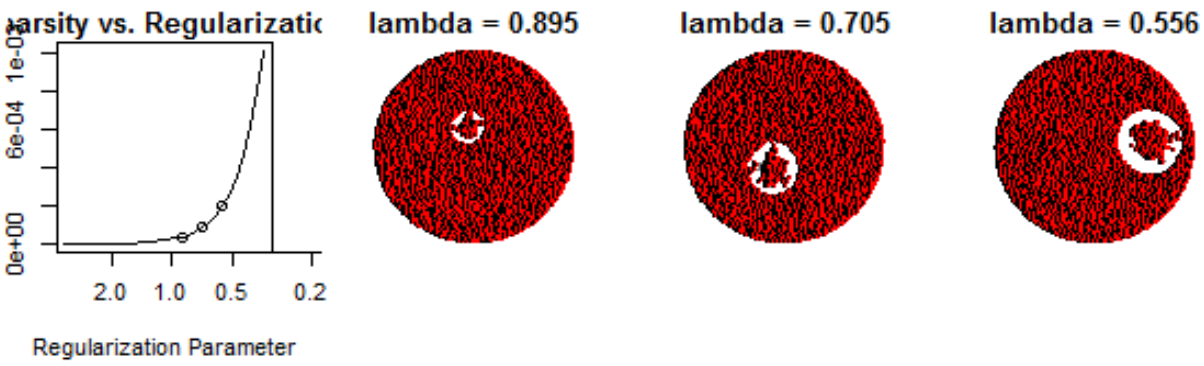

Figura **9** ***Path of values*** **de la base de datos** ***breast cancer data***

En este caso, el programa se ha centrado en los parámetros de regulación: $\lambda = 0.9$, $\lambda = 0.71$ y $\lambda = 0.56$. Para cada valor ha estimado su correspondiente modelo gráfico Gaussiano. A la izquierda se

muestra un gráfico que relaciona el valor del parámetro de regularización y el grado de dispersión que adquiere $\sum^{-1}$. Se observa que a medida que $\lambda \to \infty$, disminuye el nivel de dispersión de $\sum^{-1}$, es decir, a medida que $\lambda \to \infty$, más valores de $\sum^{-1}$ son cero.

El *path of values* ofrece varios valores de λ, pero para estimar la matriz de covarianzas se requiere un valor concreto de λ. La función huge.select del paquete huge estima, a partir del *path of values,* un valor óptimo de λ para utilizarlo como parámetro de regularización en la estimación de $\sum$, $\sum^{-1}$*:*

```
huge.select(est, criterion = NULL, ebic.gamma = 0.5, stars.thresh = 0.1,
stars.subsample.ratio = NULL, rep.num = 20, verbose = TRUE)
```

est: Se tiene que introducir un *path of values* estimado a partir de la matriz de los datos y no a partir de la matriz de covarianzas muestrales, en esos caso, la función no se ejecuta, el mensaje es: '*Model selection is not available when using the covariance matrix as input*'.

criterion: Se puede elegir entre los tres siguientes criterios: *Rotation information criterion (ric)*, *stability approach to regularization selection (stars)* y *extended Bayesian information criterion (ebic)*.

ebic.gamma: Es un parámetro de ajuste cuando el criterio es *ebic* y el método ha sido *glasso*. Por defecto es 0.5.

stars.thresh: El umbral de la variabilidad cuando el criterio es *stars*. Por defecto es 0.1.

stars.subsample.ratio: Es el *ratio* del submuestreo cuando el criterio es *stars*. Por defecto es 10*sqrt(n)/n cuando n<144 y 0.8 cuando n<=144.

rep.num: El número de submuestreos o rotaciones cuando los criterios son *stars* o *ric* respectivamente. Por defecto es 20.

verbose: Si no se quiere imprimir la información se pone *verbose = FALSE.*

Para poder aplicar la función huge.select, uno de los argumentos necesarios es un *path of values* estimado a partir de la matriz de los datos y no a partir de la matriz de covarianzas muestrales, por lo que se procede a estimar el *path of values* con la matriz de los datos .

```
Path<-huge(west.mat.clean,nlambda=30,method ="glasso")
```

Sin embargo, de esta forma, se esperó a que el algoritmo terminase más de 12 horas, mientras que la estimación del *path of values* con la matriz de covarianzas muestral como argumento acabó su carga de trabajo en 30 minutos aproximadamente.

Puesto que la función huge.select requiere un *path of values* que se calcule a partir de la matriz de los datos y se comprobó que calcularlo de esta forma requiere mucho tiempo, para ilustrar el funcionamiento de la función huge.select se procedió a disminuir la dimensión de la base de datos pero manteniendo la condición de que *p > n.*

Se seleccionaron las 100 primeras variables de la base de datos, es decir, se trabaja con una base de datos de 49 observaciones y 100 variables. El resultado de la estimación del *path of values* con la matriz de los datos es*:*

```
mibbdd<-west.mat.clean[,1:100]
Path<-huge(mibbdd,nlambda=30,method ="glasso")
Path
plot(Path)
```

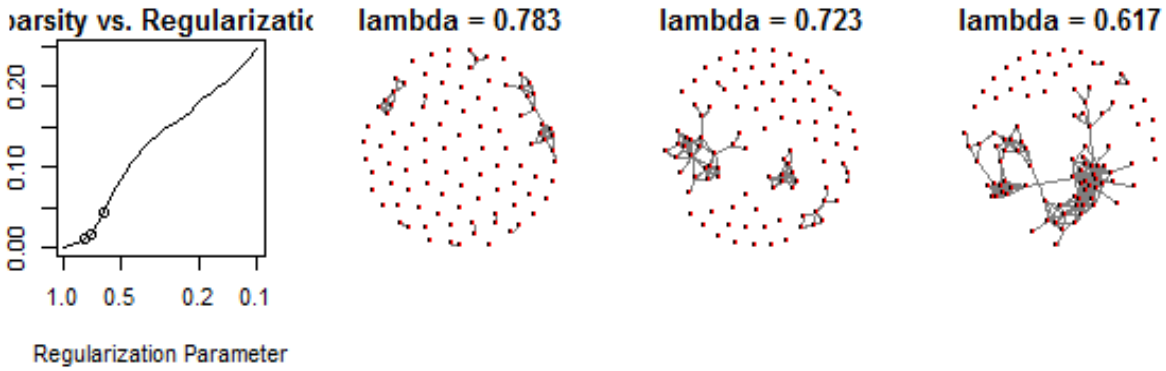

Figura 10 *Path of values* de las 100 primeras variables de la base de datos *breast cancer data*

A continuación se calcula el valor del parámetro de regularización λ a partir del criterio *stars* para un umbral de grado de dispersión de 0.1.

```
lambda<-huge.select(Path,criterion="stars")
lambda$opt.lambda
plot(lambda)

> lambda$opt.lambda
[1] 0.485982
```

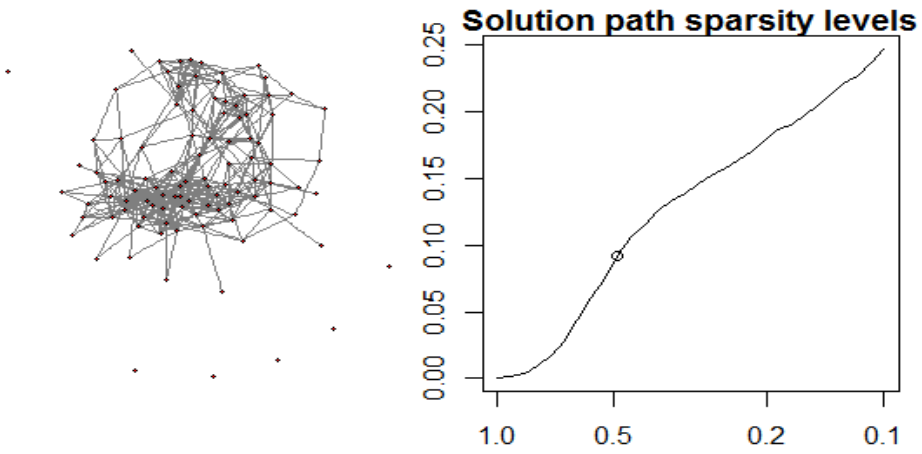

Figura 11 Modelo gráfico Gaussiano para el valor de λ calculado mediante el criterio *stars* (umbral=0.1)

El valor del parámetro de regularización estimado en este caso es: $\lambda = 0.49$ y el gráfico Gaussiano es el que se presenta en la figura 11.

A continuación se calcula el valor del parámetro de regularización λ a partir del criterio *stars* para un umbral del grado de dispersión a 0.05.

```
lambda<-huge.select(Path,criterion="stars",stars.thresh = 0.05)
lambda$opt.lambda
plot(lambda)

> lambda$opt.lambda
[1] 0.6166922
```

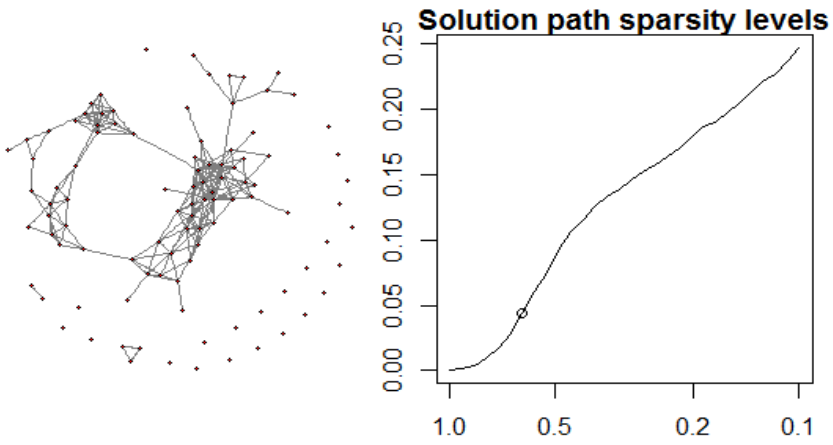

Figura 12 Modelo gráfico Gaussiano para el valor de λ calculado mediante el criterio *stars* (umbral=0.05)

En este caso el valor del parámetro de regularización es $\lambda = 0.62$, que ha aumentado con respecto al caso anterior, $\lambda = 0.49$, sin embargo, el nivel de dispersión ha disminuido (existen menos aristas que unan pares de vértices y por lo tanto el nº de valores de $\sum^{-1}$ que son cero ha aumentado), el nivel de dispersión ha pasado de 0.1 aproximadamente a tenerlo de 0.05 tal y como se observa en los gráficos 11 y 12.

También se estima el valor de λ a partir del criterio *ric.* Este criterio rota aleatoriamente las variables para cada muestra varias veces y selecciona la regularización mínima (el valor de λ mínimo) que hace que todos los valores de $\sum^{-1}$ sean ceros (nivel de dispersión=0):

```
lambda2<-huge.select(Path,criterion="ric")
lambda2$opt.lambda
plot(lambda2)

> lambda2$opt.lambda
[1] 200.9497
```

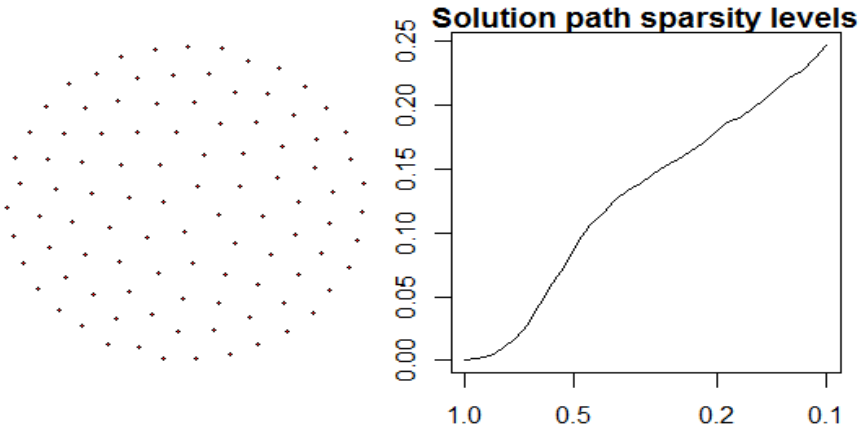

Figura 13 Modelo gráfico Gaussiano para el valor de λ calculado mediante el criterio *ric*

Otro criterio es aquel que estima el valor de λ a partir del *ebic* que solamente es aplicable para el algoritmo *Graphical Lasso:*

```
lambda3<-huge.select(Path,criterion="ebic")
lambda3$opt.lambda
plot(lambda3)

> lambda3$opt.lambda
[1] 0.9930361
```

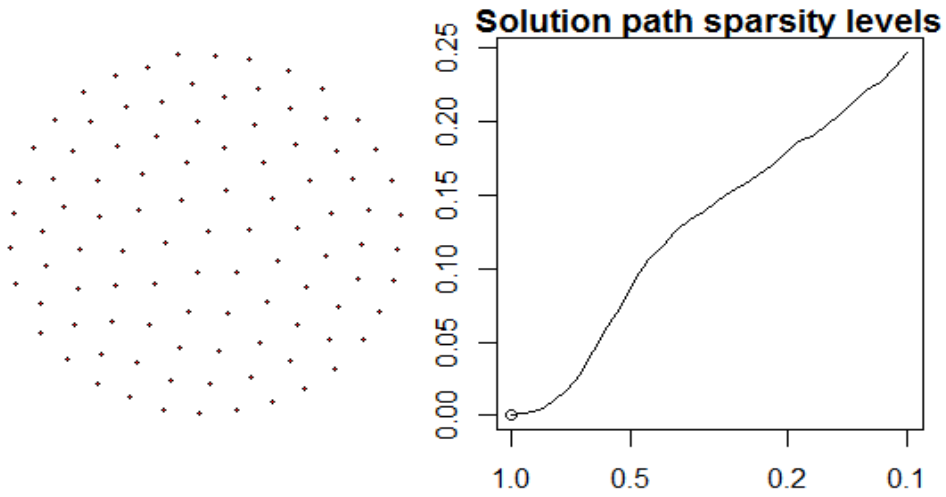

Figura 14 Modelo gráfico Gaussiano para el valor de λ calculado mediante el criterio *ebic*

Aplicando este criterio, se observa que con un valor de $\lambda = 1$ ya se alcanza un nivel de dispersión de 0. Sin embargo, según el criterio *ric* para alcanzar este mismo nivel de dispersión se necesita un valor de $\lambda = 201$.

Se considera el valor del parámetro de regularización estimado mediante el criterio *stars* de $\lambda = 0.62$ ya que tiene un nivel de dispersión más bajo que el caso de $\lambda = 0.49$ sin llegar a la situación de que todos los valores de $\sum^{-1}$ sean ceros (lo que sucede con los criterios de *ric* y *ebic*).

A continuación se estima $\sum$, $\sum^{-1}$ mediante el algoritmo *grafical lasso* a partir del paquete de R `glasso`*:*

```
glasso(s, rho, zero=NULL, thr=1.0e-4, maxit=1e4, approx=FALSE,
penalize.diagonal=TRUE, start=c("cold","warm"), w.init=NULL,wi.init=NULL,
trace=FALSE)
```

`s`: Se tiene que introducir la matriz de covarianzas

`rho`: Se tiene que indicar el valor del parámetro de regularización λ

`zero`*:* Se puede indicar que valores de $\sum^{-1}$ son ceros.

`thr`: Se tiene que indicar el valor del umbral fijo '*t*'. Por defecto es 0.001.

`maxit`: Se puede indicar el número máximo de iteraciones que se le permite realizar al algoritmo. Por defecto son 10000.

`approx`: Es un indicador de aproximación. Si se le da el valor *TRUE* realiza la aproximación de [20] en vez del algoritmo *grafical lasso* [22].

`penalize.diagonal`*:* Se especifica si se debe penalizar la diagonal principal de $\sum^{-1}$. Por defecto es *TRUE.*

`start`*:* Se permite dar valores iniciales para $\sum$ y $\sum^{-1}$ con la opción "*warm*"

`w.init`*:* Se tiene que introducir una matriz *pxp* con los valores iniciales de $\hat{\sum}$ (si se ha utilizado *start*="*warm*").

`wi.init`*:* Se tiene que introducir una matriz *pxp* con los valores iniciales de $\hat{\sum}^{-1}$ (si se ha utilizado *start*="*warm*").

`Trace`: Se puede imprimir la información de cada iteración. Por defecto es *FALSE*.

La programación y el resultado obtenido para los valores de $\lambda = 0.62$ y $t = 0.001$ se muestran a continuación:

```
covar<-cov(mibbdd)
a<-glasso(covar,0.62)
```

Por ejemplo, algunos de los valores de $\hat{\sum}^{-1}$ son:

```
          [,94]       [,95]       [,96]       [,97]       [,98]       [,99]       [,100]
 [94,]  0.623479014 0.000000 -0.24194974 0.000000 0.000000 0.0000000  0.000000000
 [95,]  0.000000000 1.235928  0.00000000 0.000000 0.000000 0.0000000  0.000000000
 [96,] -0.241950225 0.000000  0.50502637 0.000000 0.000000 0.0000000 -0.010626347
 [97,]  0.000000000 0.000000  0.00000000 1.293818 0.000000 0.0000000  0.000000000
 [98,]  0.000000000 0.000000  0.00000000 0.000000 1.301783 0.0000000  0.000000000
 [99,]  0.000000000 0.000000  0.00000000 0.000000 0.000000 0.8922396  0.000000000
[100,]  0.00000000  0.000000  0.00000000 0.000000 0.000000 0.8922396  0.000000000
```

Es interesante utilizar el paquete de R `corpcor` ya que tiene una función que calcula el *ratio* entre el mayor y el menor valor singular, por lo tanto, se puede utilizar para determinar si la matriz de covarianzas estimada esta *well-conditioned*. La función es la siguiente:

```
rank.condition(m, tol)
```

`m`*:* Se tiene que introducir una matriz real y simétrica.

`tol`*:* Se utiliza si se quiere introducir un valor determinado de tolerancia para que los valores sean considerados no ceros.

Para comprobar que el algoritmo grafical lasso proporciona una estimación de la matriz de covarianzas poblacional que está mejor *well-conditioned* que la clásica matriz de covarianzas muestral, se aplicó la función `rank.condition` a ambas matrices obteniendo los siguientes resultados.

Para la matriz de covarianzas muestral:

```
mibbdd<-west.mat.clean[,1:100]
covar<-cov(mibbdd)
rank.condition(covar)
```

El resultado es:

```
$condition
[1] Inf
```

Para la matriz de covarianzas estimada mediante el algoritmo del G*raphical lasso*:

```
mibbdd<-west.mat.clean[,1:100]
covar<-cov(mibbdd)
a<-glasso(covar,0.62)
rank.condition(a$w)
```

El resultado es:

```
$condition
[1] 6.522866
```

El valor del *ratio* en el caso de la matriz de covarianzas muestral es infinito mientras que el valor del *ratio* en el caso de la estimación mediante el algoritmo *Graphical Lasso* es 6.52, es decir, la matriz de covarianzas estimada mediante el algoritmo *Graphical lasso* está mejor mejor *well-conditioned* que la matriz de covarianzas estimada mediante el método clásico de máxima verosimilitud y podemos confirmar por lo tanto que el algoritmo G*raphical Lasso* da respuesta al objetivo del presente trabajo de encontrar una estimación de $\sum$ cuya matriz esté *well-conditioned* y que sea siempre definida positiva.

2.4 Nuevos conocimientos del algoritmo *Graphical Lasso* y dos nuevos algoritmos

Como ya se ha visto en el ejemplo final, cuando la dimension es muy alta, el tiempo que requiere el algoritmo G*raphical Lasso* propuesto en [22] puede ser un problema. En los ultimos años, se han escrito algunos artículos científicos que proponen variantes de este algoritmo G*raphical Lasso* "estándar" cuyo objetivo es ahorrar carga de trabajo y, de esta forma, obtener la solución G*raphical Lasso* en menos tiempo. Uno de los más destacables es el artículo [29] el cual propone dos nuevos algoritmos con el objetivo de mejorar la velocidad de resolución del problema G*raphical Lasso.*

Estos dos nuevos algoritmos se apoyan en una condicion necesaria y suficiente que se puede usar para identificar las componentes conectadas en el MGG, o dicho de otra forma, esta condición se puede utilizar para saber si la inversa de la matriz de covarianzas estimada será una matriz diagonal por bloques. La ventaja de esto es que la solución del problema *Graphical Lasso* se podría resolver en vez de aplicando directamente el algoritmo "estándar", aplicandolo a cada bloque y así conseguir reducir en gran medida los cálculos.

Esta condición necesaria y suficiente se expone en este trabajo más adelante en el teorema nº 2. Fue descubierta por [30] y es el teorema principal en el que se apoya el trabajo de [29].

En base a la condición necesaria y suficiente que se explica en el apartado 2.3.3.1 de que para que $\Theta = \Sigma^{-1}$ maximice $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$ es necesario y suficiente que satisfaga la ecuación subgradiente: $\Theta^{-1} - S - \lambda\Gamma(\Theta) = 0$ se enuncia el teorema nº 1, que establece que si se conoce a priori que la solución del problema G*raphical Lasso,* $\Theta = \Sigma^{-1}$, es una matriz diagonal por bloques, entonces, la solución se puede obtener aplicando el algoritmo G*raphical Lasso* 'estandar' a cada uno de los bloques.

Teorema 1

Si la solución de $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$ toma la forma: $\Theta = \begin{pmatrix} \Theta_1 & \\ & \Theta_2 \end{pmatrix}$, entonces, $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$ se puede resolver por separado maximizando por un lado:

$$\mathrm{L}(\Theta_1) = \log|\Theta_1| - tr(S_1\Theta_1) - \lambda\|\Theta_1\|_1$$

con respecto a Θ_1, y maximizando por otro lado:

$$\mathrm{L}(\Theta_2) = \log|\Theta_2| - tr(S_2\Theta_2) - \lambda\|\Theta_2\|_1$$

con respecto a Θ_2 y donde S_1 y S_2 son submatrices de S en correspondencia con Θ_1 y Θ_2.

Evidentemente, el teorema es extensible a cualquier nº de bloques.

En el teorema nº 2 se presenta la condición suficiente y necesaria de [29], que como ya se ha comentado, se puede utilizar para saber si la inversa de la matriz de covarianzas estimada será una matriz diagonal por bloques.

Teorema 2

Una condición necesaria y suficiente para la solución del problema *Graphical Lasso* sea una matriz diagonal por bloques: $C_1, C_2, ..., C_K$ es que $\left|S_{ii'}\right| \leq \lambda \forall i \in C_k, i' \in C_{k'}, k \neq k'$.

La demostración se puede consultar en [29].

A continuación se explica el corolario 1 que es una consecuencia directa del teorema nº 2 aplicado a los casos: $C_1 = \{i\}$ y $C_2 = \{1,2,...,i-1,i+1,...,p\}$. Este corolario implica que a partir de los elementos de fuera de la diagonal principal de una columna de S se puede conocer si el correspondiente nodo en el MGG está desconectado de todos los demás nodos.

Corolario 1

Una condición necesaria y suficiente para que el i-ésimo nodo del grafo esté completamente desconectado de todos los demás nodos en la solución de $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$ es que $\left|S_{ii'}\right| \leq \lambda \forall i \neq i'$.

Se ilustra el alcance de este corolario con un ejemplo. Se retoma la simulación del apartado 2.3.3.2 cuya matriz de covarianzas muestral era:

$S = \begin{pmatrix} 0.84 & -0.01 & -0.71 \\ -0.01 & 1.30 & -0.48 \\ -0.71 & -0.48 & 2.00 \end{pmatrix}$ y se buscaba la solución G*raphical Lasso* para un $\lambda = 0.5$. Se calcula para cada variable $\left|S_{ii'}\right| \forall i \neq i'$:

Para la variable 1

$\left|S_{21}\right| = 0.01 \leq 0.5$

$\left|S_{31}\right| = 0.71 \geq 0.5$

Para la variable 2

$\left|S_{12}\right| = 0.01 \leq 0.5$

$\left|S_{32}\right| = 0.48 \leq 0.5$

Para la variable 3

$$|S_{13}| = 0.71 \geq 0.5$$
$$|S_{23}| = 0.48 \leq 0.5$$

A la vista de los resultados y según el corolario, el nodo correspondiente a la variable nº 2 está desconectado de todos los demás nodos. Esto no sucede con los nodos de las variables 1 y 3 que se encuentran conectados entre sí. Recordamos que el resultado de maximizar $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$ fue:

$\hat{\Sigma}^{-1} = \begin{pmatrix} 0.76 & 0 & 0.06 \\ 0 & 0.56 & 0 \\ 0.06 & 0 & 0.4 \end{pmatrix}$, que da lugar al MGG que se muestra en la figura 15:

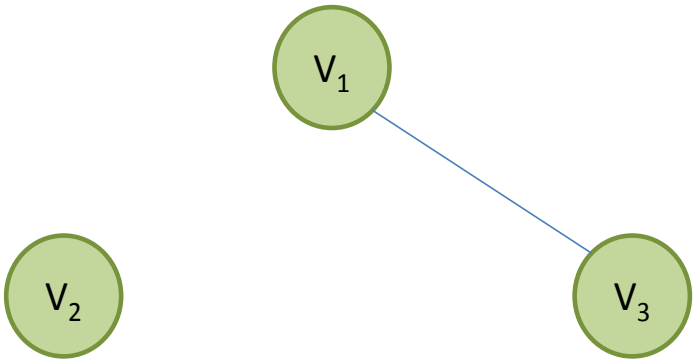

Figura 15 MGG de la simulación del apartado 2.3.3.2

Es decir, con la aplicación del corolario se hubiese podido conocer la forma del correspondiente MGG antes de aplicar el algoritmo.

En este caso, en vez del algoritmo *Graphical Lasso* estándar, se podría haber aplicado el algoritmo nº4 que se detalla a continuación, fue propuesto por [29] y se basa en el corolario que acabamos de enunciar.

Un algoritmo rápido basado en el corolario 1

1. Identificar los nodos que están totalmente desconectados de todos los demás nodos. Es decir, aplicar el corolario 1.
2. Sin pérdida de generalidad, se ordenan de tal manera que los q nodos totalmente desconectados del resto de los nodos preceden a las otros p-q nodos.
3. La solución del problema *Graphical Lasso* es la forma:

$$\Theta = \begin{pmatrix} \frac{1}{S_{11}+\lambda} & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \frac{1}{S_{qq}+\lambda} & \\ & & & & \Theta_{q+1} \end{pmatrix},$$

donde Θ_{q+1} resuelve el problema *Graphical Lasso* aplicado solamente a la submatriz de S $(p-q)\times(p-q)$ correspondiente a los nodos que no están completamente desconectados de los demás nodos.

Algoritmo 4 Un algoritmo rápido basado en el corolario 1

Al aplicar el algoritmo 4 en vez del algoritmo 3 se consigue reducir el nº de operaciones del algoritmo y por lo tanto, se emplea menos tiempo en conseguir la solución. En concreto, con el algoritmo 4, el nº de operaciones se reduce de p^3, que son el número de operaciones que se emplean para resolver el algoritmo 3 (ver [22]) a $(p^2+(p-q)^3)$, donde q representa el nº de nodos que están desconectados delos demás nodos. En la simulación del apartado 2.3.3.2 se hubiese pasado de 27 operaciones a 17. La solución tomaría la forma:

$$\hat{\Theta} = \begin{pmatrix} \frac{1}{1.30+0.5} = 0.56 & 0 & 0 \\ 0 & 0.76 & 0.06 \\ 0 & 0.06 & 0.4 \end{pmatrix}$$

Donde la 1ªcolumna representa a la variable nº 2, la segunda a la variable nº 1y la tercera a la variable nº 3.

A continuación se explica el otro algoritmo propuesto en [29] y que se basa en el teorema nº 2 (Algoritmo 5).

<u>Un algoritmo rápido basado en el teorema 2</u>

4. Sea A una matriz *pxp* cuyos elementos de fuera de la diagonal principal son de la forma $A_{ii'} = 1_{|S_{ii'}|>\lambda}$ y cuyos elementos de la diagonal principal son uno.

5. Identificar las $K \geq 1$ componentes conectadas del grafo, cada una con su correspondiente matriz de adyacencia A. Para cada $k = 1,...,K$, sea C_k el conjunto de índices en correspondencia con las variables conectadas en la k-ésima componente.

6. Sin pérdida de generalidad, se ordena de la siguiente forma: Si $i \in C_k, i' \in C_{k'}$ y $k < k'$, entonces $i < i'$.

7. La solución del problema G*raphical Lasso* es la forma:

$$\Theta = \begin{pmatrix} \Theta_1 & & & & \\ & \Theta_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & \Theta_K \end{pmatrix},$$

donde Θ_K resuelve el problema G*raphical Lasso* aplicado solamente a la submatriz de S que consiste en los valores correspondientes a las posiciones indicadas en C_k. Evidentemente, y en relación al algoritmo 4, si $C_k = \{i\}$, es decir, el i-ésimo nodo está completamente desconectado de todos los otros nodos, Θ_k será de la forma: $\frac{1}{(S_{ii} + \lambda)}$

Algoritmo 5 Un algoritmo rápido basado en el teorema 2

El paso nº 1 se puede calcular en p^2 operaciones por lo tanto el paso nº 2 se necesitará como mucho p^2 operaciones. En el paso 4, el problema G*raphical Lasso* debe ser resuelto K veces (uno para cada bloque) y esto requiere $\sum_{k=1}^{K} |C_k|^3$ operaciones por lo que, el algoritmo 5 consigue reducir el número de operaciones de p^3 a $p^2 + \sum_{k=1}^{K} |C_k|^3$.

En general se espera que el algoritmo 5 sea más rápido que el algoritmo 4 ya que el algoritmo 5 explota toda la estructura de la matriz diagonal por bloques y no solamente los bloques referentes a los nodos que están desconectados de los demás nodos. Sin embargo, podría darse el caso en que el bloque más grande de todos fuese el compuesto por los nodos desconectados, en ese caso, el algoritmo 4 debería ser más rápido que el algoritmo 5 ya que el paso nº 1 del algoritmo 4 es más rápido que los pasos nº 1 y 2 del algoritmo nº 5 y los restantes pasos requieren el mismo número de operaciones.

Si se desea profundizar más sobre los tiempos empleados por estos algoritmos se puede consultar el apartado de: '*Timing results*' de [29] que muestra los resultados de un estudio de simulación de una distribución normal $N(0, \Sigma)$ para tres casos distintos de Σ y para diferentes valores de p y de λ y se comparan los tiempos empleados en realizar el algoritmo 3, el algoritmo 4, el algoritmo 5 y el algoritmo de la aproximación de [20].

Como resumen de todo lo explicado en el apartado, se puede decir, que debido al teorema 2 (la condición necesaria y suficiente clave en el trabajo de [29]), con solamente realizar una comprobación inicial

en la matriz de covarianzas muestral antes de aplicar el algoritmo G*raphical lasso* 'estandar' se puede reducir mucho el número de operaciones a realizar por el algoritmo y por lo tanto ahorrar en tiempo. Además esta condición necesaria y suficiente nos da nuevos conocimientos de la solución del problema G*raphical Lasso* en el sentido de que dados dos parámetros de regularización λ_1 y λ_2 y tal que $\lambda_1 < \lambda_2$ el conjunto de todos los nodos desconectados de los demás nodos cuyo parámetro de regularización es λ_1 es un subconjunto del conjunto de nodos desconectados con parámetro de regularización λ_2.

Los paquetes estadísticos de R que implementan estos algoritmos son: Para el algoritmo 4 el paquete de R `glasso1.6` (versión 1.6 del paquete de `glasso` disponible en CRAN) y para el algoritmo 5 el paquete de R `glasso1.7` (versión 1.7 del paquete de `glasso` disponible en CRAN).

Capítulo 3.

Estimación *shrinkage*

3.1 Introducción

Otro de los métodos utilizados para la estimación de $\sum$ y $\sum^{-1}$ es el método *shrinkage*. La ventaja de este método son que la matriz de covarianzas estimada mediante este método es siempre definida positiva y está bien preparada *(well-conditioned)*, además, su cálculo es muy sencillo, no es computacionalmente costoso, no requiere ningún parámetro de ajuste, sólo el de la intensidad *shrinkage* que se estima directamente de los datos y no se necesita conocer la distribución subyacente, solo verificar que existen los dos primeros momentos.

El objetivo del capítulo es la explicación de la estimación de $\sum$ y $\sum^{-1}$ en el contexto de la alta dimensionalidad mediante el método *shrinkage.*

La estimación *shrinkage* de la matriz de covarianzas fue introducida por *Ledoit and Wolf:* [31], [32] y [33] en finanzas. El método no pretendía ser una solución para los casos donde *p* >> *n*. Es en [34], donde basándose en el método de *Ledoit and Wolf,* se muestra un método de estimación *shrinkage* de $\sum$ cuando *p* >> *n.* En este trabajo para hallar la solución lineal *shrinkage,* se considera una media ponderada entre la matriz de covarianzas muestrales, la cual representa una estimación sin sesgo de $\sum$ y una matriz objetivo (denominada *target matrix*) sesgada (se verá en el apartado 3.2.1). Para conocer el peso adecuado que se debe dar a cada matriz se necesita conocer la intensidad óptima *shrinkage* (λ) que es el valor que minimiza una función pérdida y que se define como el valor esperado de la suma de las desviaciones al cuadrado de los elementos de la matriz resultante y los valores reales correspondientes (apartado 3.3.3).

En el apartado 3.4 se muestra un ejemplo de la estimación *shrinkage* de la matriz de covarianzas y su inversa a partir del paquete de R `corpcor` utilizando la misma base de datos que se utiliza en el capítulo anterior: *Breast cáncer data*.

3.2 La estimación lineal *shrinkage* de la matriz de covarianzas poblacional

3.2.1 El método presentado por *Schäfer* y *Strimmer* (2005)

Existe una teoría que se suele aplicar a los problemas de estimación en el contexto de la alta dimensionalidad: Se supone $\Psi = (\psi_1,...,\psi_p)$ los parámetros de un modelo de interés sin restricciones y de

alta dimensión y $\Theta = (\theta_i)$ los parámetros de un submodelo de $\Psi = (\psi_1,...,\psi_p)$ con restricciones de menor dimensión. Por ejemplo, Ψ podría ser un vector de medias p-dimensional de un conjunto de variables con distribución normal y Θ un vector del correspondiente modelo restringido a que todas las medias fuesen iguales: $\theta_1 = ... = \theta_p$. Si se ajusta cada uno de los dos modelos a los datos observados asociados, se obtienen las estimaciones: $\hat{\Psi}$ y $\hat{\Theta}$, donde la estimación sin restricciones, $\hat{\Psi}$, tendrá una alta varianza debido al gran número de parámetros que se necesita ajustar mientras que la estimación de su homologo de menor dimensión, $\hat{\Theta}$, tendrá menor varianza aunque será un estimador sesgado del parámetro poblacional Ψ. Se suele solucionar el problema mediante el método linear *shrinkage*, que consiste en combinar ambos estimadores en una media poderada: $\hat{U} = (1-\lambda)\hat{\Psi} + \lambda\hat{\Theta}$, donde $\lambda \in [0,1]$ es la intensidad *shrinkage* y $\hat{U}$ el estimador *shrinkage*. Se obtiene así un estimador regularizado por el parámetro λ de Ψ que supera a los estimadores individuales $\hat{\Psi}$ y $\hat{\Theta}$ en precisión y eficiencia estadística [34].

Esta teoría se puede aplicar a la estimación de Σ: Se considera un conjunto de *p* variables aleatorias: $X_1,..., X_p$, con varianza: $s_{ii} > 0, i = 1,..., p$. Sea $\hat{V}$ la matriz de covarianzas muestrales del conjunto de datos donde a cada observación se le ha quitado la media de la variable correspondiente y sea $\hat{D}$ una matriz diagonal *n* x *n* cuyos elementos de la diagonal principal son las varianzas $s_{ii}, i = 1,..., p$. A esta matriz se la denomina *target matrix y* no tiene porqué ser siempre la misma (sobre esta cuestión se profundiza más en el apartado 3.3.2). Entonces, la estimación de Σ mediante el método linear *shrinkage* sería: $\hat{\Sigma} = (1-\lambda)\hat{V} + \lambda\hat{D}$.

Esta estimación garantiza que $\hat{\Sigma}$ es siempre una matriz definida positiva ya que $(1-\lambda)\hat{V}$ lo es y $\lambda\hat{D}$ también (la demostración se puede consultar en [35]).

Se puede observar que el caso de $\lambda = 0$ es el caso donde no se aplica regularización y por lo tanto no existiría ningún tipo de *shrinkage* (encogimiento) de las covarianzas y daría como resultado el estimador de máxima verosimilitud de siempre: La matriz de covarianzas muestrales para $\lambda = 1$ sería el caso contrario, el de una completa *shrinkage* de las covarianzas, es decir, se ignoran todas las covarianzas, todas tendrían valor cero. Los casos $\lambda < 0$ y $\lambda > 1$ no tienen sentido desde la perspectiva de la estimación *shrinkage.* El valor óptimo de la intensidad *shrinkage*, λ, es un valor que se encuentra entre los valores 0 y 1: $\lambda \in (0,1)$.La estimación del valor óptimo de la intensidad *shrinkage* (intensidad de encogimiento) es un asunto pendiente que se explicará en el apartado 3.3.

3.2.2 Ejemplo didáctico de la estimación linear *shrinkage*

En este apartado se quiere ilustrar a partir de un ejemplo, la estimación de Σ mediante el método linear *shrinkage* y mostrar cómo esta estimación da como resultado una matriz definida positiva. Para ello se utilizan los datos que se proporcionan en el ejemplo del artículo científico [35]. Estos datos se muestran en la Tabla 2.

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	10	12	9	-2	17	8	12
2	-9	-11	2	-5	-7	2	-2
3	16	5	8	5	18	8	9
4	6	-3	6	-13	1	4	2
5	1	4	-9	5	8	-16	-1
6	12	-1	2	22	11	6	10

Tabla 2 Datos del ejemplo del artículo científico [35]

La Tabla 2 describe un conjunto de datos con *p*=7 variables (las columnas) y *n*=6 observaciones (las filas), es decir, nos encontramos en el caso donde *p>n*.

El primer paso consiste en restar a cada una de las observaciones la media de la variable a la que corresponde. A continuación se calcula la matriz de covarianzas de esta matriz para así obtener la matriz $\hat{V}$ y se comprobará a partir de un análisis espectral, que esta matriz efectivamente no es una matriz definida positiva puesto que es la matriz de covarianzas muestral (calculada a partir de las observaciones a las que se les ha restado la media de la variable a la cual corresponden). Por último se calcula la estimación lineal *shrinkage* de Σ para una intensidad *shrinkage* de $\lambda = 0.3$, es decir, se calcula $\hat{\Sigma}$ tal que $\hat{\Sigma} = (1-0.3)\hat{V} + 0.3\hat{D}$ y se comprueba que esta si es una matriz definida positiva.

Los resultados obtenidos son:

```
#matriz con los datos

s<-matrix(c(10,12,9,-2,17,8,12,-9,-11,2,-5,-7,2,-2,16,5,8,5,18,8,9,6,-3,6,-13,1,4,2,1,4,-9,5,8,-16,-1,12,-
1,2,22,11,6,10),ncol=7,nrow=6,byrow=TRUE)

#medias de cada variable

m1<-mean (s[,1])
m2<-mean (s[,2])
…
```

NOTA: Se calculan las medias del resto de variables

```
#a cada observación se le quita la media de la variable a la cual corresponde

v1<-s[,1]
v11<- numeric (6)
for (i in 1:6)
   v11[i]<-c(v1[i]-m1)

v2<-s[,2]
v12<- numeric (6)
for (i in 1:6)
  v12[i]<-c(v2[i]-m2)
…
```

NOTA: El procedimiento se repite para el resto de variables

```
#la nueva matriz
```

```
s2<-c(v11,v12,v13,v14,v15,v16,v17)
s2<-matrix(s2,ncol=7,nrow=6,,byrow=FALSE)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    4   11    6   -4    9    6    7
[2,]  -15  -12   -1   -7  -15    0   -7
[3,]   10    4    5    3   10    6    4
[4,]    0   -4    3  -15   -7    2   -3
[5,]   -5    3  -12    3    0  -18   -6
[6,]    6   -2   -1   20    3    4    5
```

Se calcula la matriz de covarianzas de esta matriz y de esta forma obtenemos $\hat{V}$:

```
matriz.covar<-cov(s2)
```

```
      [,1]  [,2]  [,3]   [,4]  [,5]  [,6]  [,7]
[1,] 80.4  47.4  28.6  44.8  75.8  39.6  46.6
[2,] 47.4  62.0  10.4  16.2  68.2   4.0  32.2
[3,] 28.6  10.4  43.2 -20.6  19.0  56.8  25.4
[4,] 44.8  16.2 -20.6 141.6 52.8  -2.0  32.0
[5,] 75.8  68.2  19.0  52.8  92.8  22.4  48.8
[6,] 39.6   4.0  56.8  -2.0  22.4  83.2  37.6
[7,] 46.6  32.2  25.4  32.0  48.8  37.6  36.8
```

Se comprueba que efectivamente $\hat{V}$ no es una matriz definida positiva a partir de un análisis espectral en el que se observa que los dos últimos autovalores son cero:

```
list<-eigen(matriz.covar, only.values = TRUE)
list
```

```
$values
[1]  2.948831e+02  1.487367e+02  8.168183e+01  1.270684e+01  1.991508e+00
[6] -1.772621e-15 -1.338192e-14
```

Y por último se estima $\hat{\Sigma}$ tal que $\hat{\Sigma} = (1-0.3)\hat{V} + 0.3\hat{D}$ y se comprueba que en este caso $\hat{\Sigma}$ si es una matriz definida positiva ya que todos sus autovalores son positivos.

```
matriz.shrinkage<- matriz.covar
for (i in 1:7)
  for (j in 1:7)
    if (i!=j) matriz.shrinkage[i,j]<-matriz.covar[i,j]*0.7
```

```
      [,1]   [,2]   [,3]    [,4]   [,5]   [,6]   [,7]
[1,] 80.40  33.18  20.02  31.36  53.06  27.72  32.62
[2,] 33.18  62.00   7.28  11.34  47.74   2.80  22.54
[3,] 20.02   7.28  43.20 -14.42  13.30  39.76  17.78
[4,] 31.36  11.34 -14.42 141.60  36.96 - 1.40  22.40
[5,] 53.06  47.74  13.30  36.96  92.80  15.68  34.16
[6,] 27.72   2.80  39.76  -1.40  15.68  83.20  26.32
[7,] 32.62  22.54  17.78  22.40  34.16  26.32  36.80
```

```
#se comprueba que la matriz es definida positiva

list<-eigen(matriz.shrinkage, only.values = TRUE)
list
```

```
$values
[1] 232.89143 135.64678  84.19945  32.18013  24.93202  16.93463  13.21556
```

Cabe observar, para conocer la esencia de la estimación lineal *shrinkage,* como las covarianzas al aplicar esta estimación han sido todas *shrinkage,* es decir, han disminuido si se comparan con las covarianzas de $\hat{V}$. También se observa que no se reducen las varianzas de las variables, se reducen solamente las

covarianzas, las varianzas de $\hat{\Sigma}$ son las mismas que en $\hat{V}$. Esto es una característica del tipo de *target matrix* que se ha seleccionado para este ejemplo.

3.3 El método de *Ledoit-Wolf.* Estimación óptima de la intensidad *shrinkage*

3.3.1 El método de *Ledoit-Wolf*

Una cuestión pendiente y que es fundamental en la estimación lineal *shrinkage* es seleccionar un valor óptimo de la intensidad *shrinkage,* λ . Un método que se puede usar para hallar el valor óptimo de λ es el método de la validación cruzada que se explicó en el capítulo anterior. Este método es computacionalmente intenso [36]. Otros métodos que se han utilizado se plantean dentro del contexto empírico de Bayes ([37]; [38]). Solo en [39] se puede encontrar una revisión de estimadores empíricos de Bayes *shrinkage.* Sin embargo para estimar la intensidad *shrinkage,* λ en el caso del presente trabajo estas estimaciones no se pueden aplicar ya que tienen el inconveniente de que están restringidos a datos donde *p* < *n* y/o son computacionalmente costosos como el método de la validación cruzada.

Estos inconvenientes se evitan con el método propuesto por *Ledoit-Wolf* [31], donde el parámetro de regularización λ se puede obtener de forma analítica y no requiere procedimientos computacionales costosos. Para aplicarlo no se necesita conocer la distribución de la cual provienen los datos, solamente comprobar que existen los dos primeros momentos de los datos desde los que se han estimado $U = \hat{\Psi}$ y $T = \hat{\Theta}$. La propuesta consiste entonces en hallar el valor de λ que minimiza el error al cuadrado medio, $R(\lambda) = E\left(\sum_{i=1}^{p} \left(\hat{u}_i - \psi_i \right)^2 \right) = E\left(\sum_{i=1}^{p} \left(\left(\lambda t_i + (1-\lambda) u_i \right) - \psi_i \right)^2 \right)$.

El resultado es:

$$\lambda = \frac{\sum_{i=1}^{p} v(u_i) - c(t_i, u_i) - s(u_i) E(t_i - u_i)}{\sum_{i=1}^{p} E\left[(t_i - u_i)^2 \right]}$$

Donde v es la varianza, c la covarianza y s el sesgo (La demostración se puede ver en [34]*)*. De la fórmula se deduce que λ existe siempre y además es único.

Los autores de [31] plantean que para hallar el resultado de λ los parámetros de la ecuación deben ser estimados de forma consistente, sin embargo, se considera que este es un requerimiento débil ya que la consistencia es una propiedad asintótica y un requerimiento básico de cualquier estimador, además, tal y como ya hemos comentado, este trabajo se centra en tamaños muestrales pequeños por lo que las propiedades asintóticas no nos garantizan nada en nuestro caso. ¿Cómo se podrían entonces estimar los parámetros de esta ecuación? En el trabajo presentado en [34] se sugiere reemplazar todas la varianzas y covarianzas de la ecuación por sus homólogo muestrales sin sesgo. Es decir:

$$\hat{\lambda} = \frac{\sum_{i=1}^{p} \hat{v}(u_i) - \hat{c}(t_i, u_i) - \hat{s}(u_i)(t_i - u_i)}{\sum_{i=1}^{p} (t_i - u_i)^2}$$

Podría suceder que el valor de $\hat{\lambda}$ sea mayor que uno o un valor negativo. Para solucionar este problema se trunca el valor de $\hat{\lambda}$ al valor: $\hat{\lambda}^* = \max(0, \min(1, \hat{\lambda}))$.

Se observa que la ecuación es válida independientemente del tamaño de la muestra, es decir, se puede utilizar en el caso de *p*>>*n*, que es la situación de interés en esta investigación.

3.3.2 Tipos de matriz objetivo (*target matrix*)

En el apartado 3.2.1, la matriz objetivo (*target matrix*), $\hat{D}$, era una matriz cuyos elementos de la diagonal principal eran las varianzas, $s_{ii}, i = 1,..., p$ y el resto de los elementos de la matriz ceros. Sin embargo, existen más tipos de *target matrix* que en determinadas circunstancias podrían ser más adecuadas. En el trabajo presentado en [34] se muestran los seis tipos más comunes:

- *Diagonal, unit variance*, que se suele utilizar en la regresión *Ridge y* la regularización *Tikhonov* ([40]).
- *Diagonal, common variance* para estimar el valor de λ mediante la *cross-validation* ([37]).
- *Common (co)variance,* que es la que menos se utiliza,
- *Diagonal, unequal variance,* que es la *target matrix* $\hat{D}$ que se ha utilizado en el apartado 3.2.2.
- *Perfect positive correlation* y la *constant correlation* que se introdujeron con el fin de modelar los rendimientos de las acciones ([31] *y* [32]).

Cabe destacar que el presente trabajo se centra en la *target matrix*: D*iagonal, unequal variance,* que al igual que las matrices: *Diagonal, unit variance* y d*iagonal, common variance* trata de encoger las covarianzas muestrales a cero y además, en este caso, deja las entradas de la diagonal principal de la matriz $\hat{V} = S$ intactas, es decir, esta *target matrix* no reducen las varianzas de las variables, reduce solamente las covarianzas. En el ejemplo del apartado 3.2.2, que se utiliza esta *diagonal, unequal variance,* se puede observar esta característica.

La descripción de cada una de estas *target matrix* así como la fórmula específica para la estimación del parámetro λ se encuentra en [34].

3.3.3 Estimación óptima de la intensidad *shrinkage* cuando la *target matrix* es del tipo: *Diagonal, unequal variance*

En este apartado se muestra la estimación de la intensidad *shrinkage* óptima a partir de la *target matrix: Diagonal, unequal variance* que es la matriz elegida por [34] para la estimación de la matriz de covarianzas poblacional y de la matriz de correlaciones en su análisis de perfiles de expresión de un experimento de E. coli y por [35] en su ilustración didáctica de la estimación lineal *shrinkage* de la matriz de covarianzas.

La fórmula específica para calcular la intensidad *shrinkage* óptima a partir de la *target matrix: Diagonal, unequal variance* es:

$$\lambda = \frac{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P} v(s_{ij})}{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\left[v(s_{ij}) + \sigma_{ij}^2\right]},$$

Donde v es la varianza, s_{ij} es la covarianza muestral entre las variables *i* y *j* y σ_{ij} la covarianza poblacional entre las variables *i* y *j.* La demostración se puede consultar en [35].

Se observa que tanto el numerador como el denominador son positivos siendo siempre mayor el denominador. Luego, en este caso particular, $0 < \lambda < 1$, por lo que no será necesario aplicar en ningún momento el truncamiento citado en el apartado 3.3.1.

Esta estimación del valor de λ, requiere calcular los valores de $v(s_{ij})$ y σ_{ij}^2, para ello, se utiliza el mismo método que se describe en [41] cuyo procedimiento es el siguiente:

Primero se calcula es el valor de la variable aleatoria: $w_{ijn} = (x_{in} - \bar{x}_i)(x_{jn} - \bar{x}_j),\ n = 1,\ldots,N$, cuya media es: $\bar{w}_{ij} = \frac{1}{N}\sum_{n=1}^{N}(x_{in} - \bar{x}_i)(x_{jn} - \bar{x}_j)$. Los elementos de la matriz de covarianzas se pueden escribir entonces como: $s_{ij} = \frac{1}{N-1}\sum_{n=1}^{N}(x_{in} - \bar{x}_i)(x_{jn} - \bar{x}_j) = \frac{N}{N-1}\bar{w}_{ij}$, por lo que, la estimación de las varianzas de s_{ij} y de $\bar{w}_{ij}$ ($\hat{v}(s_{ij}), \hat{v}(\bar{w}_{ij})$) se relacionan de la siguiente forma: $\hat{v}(s_{ij}) = \frac{N^2}{(N-1)^2}\hat{v}(\bar{w}_{ij})$, y, debido a que la distribución de la media muestral de una variable aleatoria basada en *n* observaciones tiene una varianza muestral que es solamente $\frac{1}{N}$ de la varianza muestral de la variable, la $\hat{v}(s_{ij})$ se puede determinar como: $\hat{v}(s_{ij}) = \frac{N}{(N-1)^2}\hat{v}(w_{ijn}) = \frac{N}{(N-1)^3}\sum_{n=1}^{N}(w_{ijn} - \bar{w}_{ij})^2$.

Para la estimación de los σ_{ij} se utiliza el método propuesto por [35]. Debido a que $E(s_{ij}) = \sigma_{ij}$, la covarianza muestral s_{ij} proporciona un estimador sin sesgo de σ_{ij}, por lo tanto, la estimación de λ puede obtenerse de la forma:

$$\hat{\lambda} = \frac{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\hat{v}(s_{ij})}{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\left[\hat{v}(s_{ij}) + s_{ij}^2\right]}$$

3.3.4 Ejemplo didáctico de la estimación óptima de la intensidad *shrinkage* cuando la *target matrix* es del tipo: *Diagonal, unequal variance.*

Para ilustrar como se calcula la intensidad *shrinkage* óptima *c*uando la *target matrix* es del tipo*: Diagonal, unequal variance* se retoma el ejemplo del apartado 3.2.2 y se explica cada paso hasta llegar a la solución, es decir, al valor óptimo: $\hat{\lambda}$.

Se parte de la matriz de covarianzas muestrales calculada en este apartado, $\hat{V}$, y se multiplica a cada elemento por el factor $\frac{N-1}{N}$, es decir, $\frac{5}{6}$. De esta forma, se obtiene la matriz de los $\overline{w}_{ij}$ (como $s_{ij} = \frac{N}{N-1}\overline{w}_{ij} \Rightarrow \overline{w}_{ij} = \frac{N-1}{N}s_{ij}$):

```
meanwij<-(5/6)*matriz.covar
```

```
      [,1]     [,2]      [,3]      [,4]     [,5]     [,6]     [,7]
[1,] 67.00000 39.500000 23.833333 37.333333 63.16667 33.000000 38.83333
[2,] 39.50000 51.666667  8.666667 13.500000 56.83333  3.333333 26.83333
[3,] 23.83333  8.666667 36.000000 -17.166667 15.83333 47.333333 21.16667
[4,] 37.33333 13.500000 -17.166667 118.000000 44.00000 -1.666667 26.66667
[5,] 63.16667 56.833333 15.833333 44.000000 77.33333 18.666667 40.66667
[6,] 33.00000  3.333333 47.333333 -1.666667 18.66667 69.333333 31.33333
[7,] 38.83333 26.833333 21.166667 26.666667 40.66667 31.333333 30.66667
```

El siguiente paso corresponde a calcular los casos $(w_{ijn} - \overline{w}_{ij})^2$ para cada variable:

```
#variable 1

matrizvar1<-matrix(ncol=7,nrow=6)
 for (j in 1:7)
   for (i in 1:6)
     if (j!=1) matrizvar1[i,j]<-(((s2[i,1]*s2[i,j])-meanwij[1,j]))^2
```

```
     [,1]     [,2]      [,3]         [,4]         [,5]     [,6]     [,7]
[1,]   NA    20.25 2.777778e-02 2844.44444    738.0278     81  117.361111
[2,]   NA 19740.25 7.802778e+01 4578.77778 26190.0278 1089 4378.027778
[3,]   NA     0.25 6.846944e+02   53.77778 1356.6944  729    1.361111
[4,]   NA  1560.25 5.680278e+02 1393.77778 3990.0278 1089 1508.027778
[5,]   NA  2970.25 1.308028e+03 2738.77778 3990.0278 3249   78.027778
[6,]   NA  2652.25 8.900278e+02 6833.77778 2040.0278   81   78.027778
```

…

NOTA: Este procedimiento se repite para el resto de las variables. Si se quieren conocer los resultados obtenidos para cada una de las variables se puede consultar el apendice 1.

Una vez obtenidos todos los $\left(w_{ijn}-\overline{w}_{ij}\right)^2$, se puede calcular:

$$\sum_{j=1}^{P}\sum_{n=1}^{N}\left(w_{ijn}-\overline{w}_{ij}\right)^2=\sum_{j=1}^{7}\sum_{n=1}^{6}\left(w_{ijn}-\overline{w}_{ij}\right)^2,\forall i=1,...,P=1,...,7:$$

```
#suma de los casos (wij-meanwij)^2 para la variable 1

sumvars1j=0
for (j in 1:7)
  for (i in 1:6)
    if (j!=1) sumvars1j<-sumvars1j+matrizvar1[i,j]

[1] 99699.33
```

NOTA: Este procedimiento se repite para el resto de las variables. Si se quieren conocer los resultados obtenidos para cada una de las variables se puede consultar el apendice 1.

Con todo esto ya se tienen los datos para poder hallar el denominador de la fórmula de $\hat{\lambda}$:

$$\sum_{i=1}^{P}\sum_{j=1}^{P}\hat{v}\left(s_{ij}\right)=\frac{N}{\left(N-1\right)^3}\sum_{i=1}^{P}\sum_{j=1}^{P}\sum_{n=1}^{N}\left(w_{ijn}-\overline{w}_{ij}\right)^2=\frac{6}{125}\sum_{i=1}^{7}\sum_{j=1}^{7}\sum_{n=1}^{6}\left(w_{ijn}-\overline{w}_{ij}\right)^2$$

```
sumvarsij<-(6/125)*(sumvars7j+sumvars6j+sumvars5j+sumvars4j+sumvars3j+sumvars2j+sumvars1j)

[1] 25786.91
```

Quedaría calcular el denominador de la fórmula, para ello, se necesita conocer los valores: s_{ij}^2, es decir, el cuadrado de los valores de la matriz de covarianzas muestrales, $\hat{V}$, calculada previamente.

```
#cálculo de la matriz de covarianzas al cuadrado

matriz.covar2<-matrix(ncol=7,nrow=7)

for (j in 1:7)
  for (i in 1:7)
    if (j!=i) matriz.covar2[i,j]<-matriz.covar[i,j]^2
```

Se calcula la suma de todos estos valores para poder obtener el valor total del denominador:

```
#sumamos los elementos de la matriz


summatriz.covar2=0

for (j in 1:7)
  for (i in 1:7)
    if (j!=i) summatriz.covar2<-summatriz.covar2+matriz.covar2[i,j]

[1] 66802.72
```

Ya se tienen todos los valores necesarios para poder calcular la estimación óptima *shrinkage* que buscábamos. El valor de $\hat{\lambda}$ es:

$$\hat{\lambda} = \frac{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\hat{v}(s_{ij})}{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\left[\hat{v}(s_{ij})+s_{ij}^2\right]} = \frac{25786{,}91}{25786{,}91+66802{,}72} = 0{,}278508$$

3.4 Ejemplo final con el paquete de R *corpcor*

La estimación lineal *shrinkage* de $\sum$ y $\sum^{-1}$ que se explica en este capítulo y basada en el *paper* de [34] se puede realizar con el paquete de R `corpcor`. Para la estimación lineal *shrinkage* de $\sum$ se debe utilizar la función `cov.shrink`:

```
Cov.shrink(x, lambda, lambda.var, w, verbose=TRUE)
```

`x`: Se tiene que introducir una matriz de datos.

`lambda`: Se tiene que indicar el valor de la intensidad shrinkage de correlación (valores entre 0 y 1), sino se especifica ninguno, se calcula un valor mediante la fórmula que se indica en [34].

`lambda.var`: Se tiene que indicar el valor de la intensidad shrinkage de la varianza (valores entre 0 y 1), sino se especifica ninguno, se calcula un valor mediante la fórmula que se indica en [42].

`w`: Es opcional, se utiliza si se quiere especificar pesos concretos a los datos.

`verbose`: Se muestra en la salida algunos mensaje de estado mientras se realiza el cálculo. Por defecto es TRUE.

Por ejemplo, la estimación lineal *shrinkage* de $\sum$ del ejemplo del apartado 3.2.2 para un valor de $\lambda = 0{,}3$ sería:

```
a<-cov.shrink(s, lambda=0.3, lambda.var=0)
```

```
      [,1]  [,2]   [,3]   [,4]   [,5]  [,6]  [,7]
[1,] 80.40 33.18  20.02  31.36  53.06 27.72 32.62
[2,] 33.18 62.00   7.28  11.34  47.74  2.80 22.54
[3,] 20.02  7.28  43.20 -14.42  13.30 39.76 17.78
[4,] 31.36 11.34 -14.42 141.60  36.96 -1.40 22.40
[5,] 53.06 47.74  13.30  36.96  92.80 15.68 34.16
[6,] 27.72  2.80  39.76  -1.40  15.68 83.20 26.32
[7,] 32.62 22.54  17.78  22.40  34.16 26.32 36.80
```

Es decir, se obtiene el mismo resultado que se muestra en el apartado 3.2.2.

Para la estimación lineal *shrinkage* de $\sum^{-1}$ se utiliza la función `invcov.shrink`:

```
Invcov.shrink(x, lambda, lambda.var, w, verbose=TRUE)
```

`x`: Se tiene que introducir una matriz de datos.

`lambda:` Se tiene que indicar el valor de la intensidad shrinkage de correlación (valores entre 0 y 1), sino se especifica ninguno, se calcula un valor mediante la fórmula que se indica en [34].

`lambda.var:` Se tiene que indicar el valor de la intensidad shrinkage de la varianza (valores entre 0 y 1), sino se especifica ninguno, se calcula un valor mediante la fórmula que se indica en [42].

`w:` Es opcional, se utiliza si se quiere especificar pesos concretos a los datos.

`verbose:` Se muestra en la salida algunos mensaje de estado mientras se realiza el cálculo. Por defecto es TRUE.

Si se aplica al ejemplo del apartado 3.2.2:

```
b<-invcov.shrink(s, lambda=0.3, lambda.var=0)
```

```
         [,1]           [,2]           [,3]           [,4]           [,5]
[1,]  0.024800325 -0.0037075910 -0.0040153265 -2.474972e-03 -0.0075290466
[2,] -0.003707591  0.0300964340  0.0005646894  3.254395e-03 -0.0118711691
[3,] -0.004015327  0.0005646894  0.0493430036  7.710166e-03 -0.0008336697
[4,] -0.002474972  0.0032543951  0.0077101664  9.955221e-03 -0.0027467091
[5,] -0.007529047 -0.0118711691 -0.0008336697 -2.746709e-03  0.0252275533
[6,] -0.002464811  0.0055101465 -0.0183646743 -7.617078e-05  0.0010935238
[7,] -0.007514119 -0.0103227867 -0.0114113612 -6.980212e-03 -0.0081802223
          [,6]           [,7]
[1,] -2.464811e-03 -0.007514119
[2,]  5.510146e-03 -0.010322787
[3,] -1.836467e-02 -0.011411361
[4,] -7.617078e-05 -0.006980212
[5,]  1.093524e-03 -0.008180222
[6,]  2.468498e-02 -0.010941030
[7,] -1.094103e-02  0.065338077
```

La base de datos sacada del artículo científico [35] y que se ha estado utilizando durante todo el capítulo para mostrar en que consiste y como se calcula la estimación lineal *shrinkage* de $\sum$ y la intensidad óptima *shrinkage* no es una base de datos con una dimensión muy alta por lo que, para completar el capítulo se considera oportuno mostrar cómo funcionaría esta estimación lineal *shrinkage* con una base de datos real de alta dimensión. Se retoma la base de datos utilizada en el capítulo anterior (las 100 primeras variables de *the breast cancer data*) en la que se tiene la situación objeto de estudio en la presente investigación: *p* >> *n*.

Para realizar la estimación lineal *shrinkage* se necesita conocer a priori la intensidad *shrinkage* óptima. El paquete de R *corpcor*, estima la intensidad *shrinkage* óptima utilizando fórmulas basadas en correlaciones parciales y varianzas parciales diferentes a la fórmula explicada en el presente capítulo (basada en el artículo de [35]). Por este motivo, no se utiliza el paquete de R *corpcor* para calcular el valor de la intensidad *shrinkage* óptima. El valor de la intensidad *shrinkage* óptima estimada a partir de la fórmula del artículo [35], $\hat{\lambda} = \dfrac{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\hat{v}(s_{ij})}{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\left[\hat{v}(s_{ij})+s_{ij}^2\right]}$, en la que se considera como *target matrix* una matriz d*iagonal, unequal variance,* de las 100 primeras variables de la base de datos *the breast cancer data (*la programación se puede consultar en el apéndice 1*)* es:

El resultado del valor del numerador es: `[1] 7.24946`

El resultado del valor del denominador es: `[1] 169.0535`

El valor óptimo de $\hat{\lambda}$ es:

$$\hat{\lambda} = \frac{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\hat{v}(s_{ij})}{\sum_{i=1}^{P-1}\sum_{j=i+1}^{P}\left[\hat{v}(s_{ij})+s_{ij}^2\right]} = \frac{7{,}25}{7{,}25+169{,}05} = 0{,}04$$

A continuación se realiza la estimación *shrinkage* de $\sum$ y $\sum^{-1}$ mediante el paquete *corpcor* para este valor calculado de la estimación óptima *shrinkage* ($\lambda = 0{,}04$):

```
a<-cov.shrink(mibbdd, lambda=0.04, lambda.var=0)
b<-invcov.shrink(mibbdd, lambda=0.04, lambda.var=0)
```

Por ejemplo los valores de $\hat{\sum}$ de las últimas 4 variables de la base de datos son:

```
             <NA>          <NA>          ACTB          ACTB
<NA>  0.152905983 -0.0227109433 -0.0261966171  0.0824847550
<NA> -0.022710943  0.1481774250 -0.0241240020  0.0769082577
ACTB -0.026196617 -0.0241240020  0.5007752046  0.0522016253
ACTB  0.082484755  0.0769082577  0.0522016253  0.7377443416
```

Al igual que en el capítulo anterior se utiliza la función *rank.condition* para comprobar que efectivamente se ha realizado una estimación de $\sum$ que está mejor *well-conditioned* que la clásica matriz de covarianzas muestral cuando *p* > *n*.

```
a<-cov.shrink(mibbdd, lambda=0.04, lambda.var=0)
rank.condition(a)
```

El resultado es:

```
$condition
[1] 4046.214
```

El valor del *ratio* sigue siendo muy alto: 4046,21 pero no tan alto como en caso de la matriz de covarianzas muestral que resultó ser infinito tal y como se indica en el capítulo anterior. Por lo que, podemos concluir que, cuando *p* > n, la estimación lineal *shrinkage* es también una mejor estimación de $\sum$ en comparación a la clásica matriz de covarianzas muestral.

Conclusiones

Todo el estudio realizado en este trabajo parte del hecho, extendido en el campo de la estadística, de la importancia de la matriz de covarianzas en el análisis multivariante. En el pasado, cuando se ha buscado un estimador de la matriz de covarianza poblacional, $\sum$, siendo la población de partida normal $N(\mu, \sum)$, el método más utilizado ha sido el de máxima verosimilitud dando como resultado el estimador de máxima verosimilitud que es la matriz de covarianzas muestral: $S = \frac{1}{n-1}\sum_{\alpha=1}^{n}(x_\alpha - \bar{x})(x_\alpha - \bar{x})'$. Se sabe que este estimador es un estimador adecuado de $\sum$ y funciona bien (es siempre una matriz definida positiva y es una matriz *well-conditioned*) cuando *n>p.*

Hoy en día en muchos de los conjuntos de datos, la dimensión, *p*, es comparable o incluso mayor al tamaño muestral, *n*, por lo que es necesario, como se ha explicado en el presente trabajo de investigación, encontrar un estimador de $\sum$ que funcione bien también en los casos donde $p \geq n$.

Para conseguir este objetivo en primer lugar se ha planteado que sucede con el estimador clásico S cuando $p \geq n$, es decir, se plantea si en estos casos se podría seguir utilizando la matriz de covarianzas muestral como estimador de $\sum$. Tal y como se muestra en el trabajo, S no es un buen estimador en estos casos puesto que muchos de sus autovalores llegan a tomar el valor de cero con las consecuencias que ello conlleva, como, por ejemplo, que no se podría calcular su inversa.

A partir de la investigación realizada se puede afirmar que el método para conseguir un estimador adecuado de $\sum$ cuando $p \geq n$ pasa por la utilización del algoritmo *Graphical Lasso* (se explicó en el capítulo 2) o por la estimación lineal *shrinkage* (se explicó en el capítulo 3). A partir de ambos métodos se consigue el objetivo del trabajo.

En el caso de la solución mediante el algoritmo *Graphical Lasso* se asume que las variables aleatorias tienen una distribución Gaussiana multivariante de media μ y matriz de covarianzas $\sum$. La teoría en la cual se basa proviene de la propuesta de [21] que consiste en encontrar el máximo de la función log-verosimilitud parcialmente maximizada por el parámetro μ con una penalización *Lasso:* $\mathrm{L}(\Theta) = \log|\Theta| - tr(S\Theta) - \lambda\|\Theta\|_1$, con respecto a $\Theta = \Sigma^{-1}$. Este problema se denomina problema G*raphical Lasso.* La resolución de este problema proporciona un estimador tanto de $\sum$ y de Σ^{-1} en el contexto de la alta dimensionalidad con las características deseadas (son siempre matrices definidas positivas y son matrices *well-conditioned).*

Vemos que este método se basa en la regularización, en concreto en la denominada regularización *Lasso* cuyo objetivo no es más que incrementar el número de ceros de Σ^{-1}, es decir, hallar una estimación dispersa de Σ^{-1} y conseguir de esta forma reducir la alta dimensionalidad.

El presente trabajo de investigación propone para la resolución del problema *graphical Lasso* el algoritmo de [22] que se denomina algoritmo *Graphical Lasso* y que efectivamente, como se muestra en el presente trabajo, resuelve el problema y por lo tanto proporciona estimadores adecuados de $\sum$ y de Σ^{-1}.

Mediante R, se ha investigado como realizar el algoritmo *Graphical Lasso* y aplicarlo a conjuntos de datos. Se ha averiguado los paquetes de R huge y `glasso` resuelven este algoritmo incluso para bases de datos de alta dimensión además de proporcionar el correspondiente MGG. Es decir, el algoritmo G*raphical Lasso* proporciona una estimación de $\sum$ y Σ^{-1} cuando $p \geq n$ que cumplen con las características deseadas. Además se cuenta con los paquetes de R huge y `glasso` para su completa resolución.

Otro de los métodos que se ha investigado es el de la estimación lineal *shrinkage*. La principal ventaja de este método es que su cálculo es muy sencillo y por lo tanto no es computacionalmente costoso.

La teoría en la cual se basa proviene de la propuesta desarrollada en [34] que consiste en hallar el estimador de $\sum$ a partir de una media ponderada entre una matriz de covarianzas muestrales, V, y una matriz objetivo denominada *target matrix:* $\hat{\Sigma} = (1-\lambda)\hat{V} + \lambda\hat{D}$. Esta estimación garantiza que $\hat{\Sigma}$ es siempre una matriz definida positiva y *well-conditioned*.

Al igual que en el caso del problema *Graphical Lasso,* en este caso, el parámetro λ, se denomina intensidad *shrinkage* y el objetivo es encoger (*shrinkage)* las covarianzas muestrales de $\hat{V}$ hacia cero.

Igualmente, se ha investigado en R como realizar una estimación lineal *shrinkage* de $\sum$ de un conjunto de datos. El paquete de R `corpcor` calcula esta estimación incluso para bases de datos de alta dimensión. Sin embargo, en este caso, para hallar el valor óptimo del parámetro λ, no se utiliza este paquete puesto que *corpcor* estima la intensidad *shrinkage* óptima utilizando una fórmula que calcula este valor cuando la estimación *shrinkage* que se pretende conseguir es la de la matriz de correlaciones, la cual no es el objetivo de la presente investigación. Por ese motivo, para hallar la estimación *shrinkage* óptima se ha utilizado la fórmula presentada en [35] y se ha creado su programación en R.

Apéndice I

1. Análisis espectral

```
#Matriz de covarianzas poblacional, matriz de Toeplitz 0.2, 10 variables

sigma0.2<-matrix(ncol=10,nrow=10)
for (i in 1:10)
  for (j in 1:10)
    sigma0.2[i,j]<-matrix(0.2**abs(i-j))

#autovalor superior e inferior de la matriz de covarianzas poblacional sigma=0.2

list<-eigen(sigma0.2, only.values = TRUE)
auto_s0.2=list$values[1]
auto_i0.2=list$values[10]

####n=50, sigma=0.2####
#simulación de 100 muestras de matrices de 10 variables cada una y n=50 con distribución normal de media 0 y matriz de covarianzas sigma 0.2

k<-100; n<-50; r<-10
a<-array(NA, c(n,r,k))

for (t in 1:100)
  a[,,t]<- rmvnorm(n=50, mean=c(0,0,0,0,0,0,0,0,0,0), sigma=sigma0.2)

#cálculo de las matrices de covarianzas de las 100 matrices simuladas

k<-100; n<-10; r<-10
b<-array(NA, c(n,r,k))

for (t in 1:100)
  b[,,t]<- cov(a[,,t])

#nos quedamos con el menor y mayor autovalor de cada matriz

auto_i<-c(NA,dim=100)
auto_s<-c(NA,dim=100)

for (t in 1:100)
{list<-eigen(b[,,t], only.values = TRUE)
 auto_s[t]=list$values[1]
 auto_i[t]=list$values[10]}

auto_n50_inf_0.2=auto_i
auto_n50_sup_0.2=auto_s

####n=10, sigma=0.2########
#simulación de 100 muestras de matrices de 10 variables cada una y n=10 con distribución normal de media 0 y matriz de covarianzas sigma 0.2

k<-100; n<-10; r<-10
a<-array(NA, c(n,r,k))

for (t in 1:100)
  a[,,t]<- rmvnorm(n=10, mean=c(0,0,0,0,0,0,0,0,0,0), sigma=sigma0.2)

#cálculo de las matrices de covarianzas de las 100 matrices simuladas

k<-100; n<-10; r<-10
b<-array(NA, c(n,r,k))

for (t in 1:100)
  b[,,t]<- cov(a[,,t])

#nos quedamos con el menor y mayor autovalor de cada matriz
```

```
auto_i<-c(NA,dim=100)
auto_s<-c(NA,dim=100)

for (t in 1:100)
{list<-eigen(b[,,t], only.values = TRUE)
 auto_s[t]=list$values[1]
 auto_i[t]=list$values[10]}

auto_n10_inf_0.2=auto_i
auto_n10_sup_0.2=auto_s

####n=5, sigma=0.2####
#simulación de 100 muestras de matrices de 10 variables cada una y n=5 con distribución normal de media 0 y matriz de covarianzas sigma 0.2

k<-100; n<-5; r<-10
a<-array(NA, c(n,r,k))

for (t in 1:100)
  a[,,t]<- rmvnorm(n=5, mean=c(0,0,0,0,0,0,0,0,0,0), sigma=sigma0.2)

#cálculo de las matrices de covarianzas de las 100 matrices simuladas

k<-100; n<-10; r<-10
b<-array(NA, c(n,r,k))

for (t in 1:100)
  b[,,t]<- cov(a[,,t])

#nos quedamos con el menor y mayor autovalor de cada matriz

auto_i<-c(NA,dim=100)
auto_s<-c(NA,dim=100)

for (t in 1:100)
{list<-eigen(b[,,t], only.values = TRUE)
 auto_s[t]=list$values[1]
 auto_i[t]=list$values[10]}

auto_n5_inf_0.2=auto_i
auto_n5_sup_0.2=auto_s

#CASO sigma 0.2#
#cambios de la variabilidad del autovalor inferior

boxplot(auto_p_inf_0.2,auto_n50_inf_0.2,auto_n10_inf_0.2,auto_n5_inf_0.2,ylim = c(0,1),col
="blue",rango=0,boxwex=0.4,outline = FALSE,names=c("población",'n>p','n=p','n<p'))

#cambios de la variabilidad del autovalor superior

boxplot(auto_p_sup_0.2,auto_n50_sup_0.2,auto_n10_sup_0.2,auto_n5_sup_0.2,ylim = c(0,10.5),col
="blue",rango=0,boxwex=0.4,outline = FALSE,names=c("población",'n>p','n=p','n<p'))
...
```

NOTA: El procedimiento se repite para el caso de $\rho = 0.5$

2. Resolución del ejemplo de *Whittaker* (1990)

```
#se crea la matriz de covarianzas muestrales del ejemplo de Whittaker (1990)

s<-matrix(c(10,1,5,4,1,10,2,6,5,2,10,3,4,6,3,10),ncol=4,nrow=4,byrow=FALSE)

#se crea la matriz que necesita el argumento 'zero' en la cual se indican las posiciones de la inversa de la matriz de
covarianzas que son ceros
#en el ejemplo actual, por filas, las posiciones 1,3 y 2,4. Al ser una matriz simétrica no es necesario indicar
#las demás posiciones

s2<-matrix(c(1,3,2,4),ncol=2,byrow=TRUE)

#se aplica la función glasso para la resolución del problema con rho=0
#puesto que en este caso no existe regularización

a<-glasso(s,0,zero=s2)
```

3. Resultados obtenidos de los casos $\left(w_{ijn} - \overline{w}_{ij}\right)^2$ en la estimación óptima *shrinkage*

```
#variable 2
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	20.25	NA	3287.11111	3306.25	1778.0278	3927.11111	2516.6944
[2,]	19740.25	NA	11.11111	4970.25	15170.0278	11.11111	3268.0278
[3,]	0.25	NA	128.44444	2.25	283.3611	427.11111	117.3611
[4,]	1560.25	NA	427.11111	2162.25	831.3611	128.44444	220.0278
[5,]	2970.25	NA	1995.11111	20.25	3230.0278	3287.11111	2010.0278
[6,]	2652.25	NA	44.44444	2862.25	3948.0278	128.44444	1356.6944

```
#variable 3
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	2.777778e-02	3287.11111	NA	46.694444	1456.6944444	128.4444	434.027778
[2,]	7.802778e+01	11.11111	NA	584.027778	0.6944444	2240.4444	200.694444
[3,]	6.846944e+02	128.44444	NA	1034.694444	1167.3611111	300.4444	1.361111
[4,]	5.680278e+02	427.11111	NA	774.694444	1356.6944444	1708.4444	910.027778
[5,]	1.308028e+03	1995.11111	NA	354.694444	250.6944444	28448.4444	2584.027778
[6,]	8.900278e+02	44.44444	NA	8.027778	354.6944444	2635.1111	684.694444

```
#variable 4
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	2844.44444	3306.25	46.694444	NA	6400	498.777778	2988.4444
[2,]	4578.77778	4970.25	584.027778	NA	3721	2.777778	498.7778
[3,]	53.77778	2.25	1034.694444	NA	196	386.777778	215.1111
[4,]	1393.77778	2162.25	774.694444	NA	3721	802.777778	336.1111
[5,]	2738.77778	20.25	354.694444	NA	1936	2738.777778	1995.1111
[6,]	6833.77778	2862.25	8.027778	NA	256	6669.444444	5377.7778

```
#variable 5
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	738.0278	1778.0278	1456.6944444	6400	NA	1248.44444	498.7777778
[2,]	26190.0278	15170.0278	0.6944444	3721	NA	348.44444	4138.7777778
[3,]	1356.6944	283.3611	1167.3611111	196	NA	1708.44444	0.4444444
[4,]	3990.0278	831.3611	1356.6944444	3721	NA	1067.11111	386.7777778
[5,]	3990.0278	3230.0278	250.6944444	1936	NA	348.44444	1653.7777778
[6,]	2040.0278	3948.0278	354.6944444	256	NA	44.44444	658.7777778

```
#variable 6
```

```
     [,1]      [,2]          [,3]         [,4]         [,5]        [,6]       [,7]
[1,]   81 3927.11111    128.4444   498.777778   1248.44444    NA      113.77778
[2,] 1089   11.11111   2240.4444     2.777778    348.44444    NA      981.77778
[3,]  729  427.11111    300.4444   386.777778   1708.44444    NA       53.77778
[4,] 1089  128.44444   1708.4444   802.777778   1067.11111    NA     1393.77778
[5,] 3249 3287.11111   8448.4444  2738.777778    348.44444    NA     5877.77778
[6,]   81  128.44444    2635.1111 6669.444444     44.44444    NA      128.44444
```

#variable 7

```
        [,1]         [,2]        [,3]         [,4]         [,5]         [,6]      [,7]
[1,]  117.361111  2516.6944   434.027778   2988.4444   498.7777778 113.77778   NA
[2,] 4378.027778 3268.0278   200.694444    498.7778 4138.7777778  981.77778   NA
[3,]    1.361111   117.3611     1.361111    215.1111    0.4444444   53.77778   NA
[4,] 1508.027778   220.0278   910.027778    336.1111   386.7777778 1393.77778   NA
[5,]   78.027778 2010.0278  2584.027778   1995.1111 1653.7777778 5877.77778   NA
[6,]   78.027778 1356.6944   684.694444   5377.7778  658.7777778  128.44444   NA
```

4. Resultados obtenidos de los casos $\sum_{j=1}^{P}\sum_{n=1}^{N}\left(w_{ijn}-\overline{w}_{ij}\right)^2$ en la estimación óptima *shrinkage*

#variable 2

```
[1] 88799.33
```

#variable 3

```
[1] 57088
```

#variable 4

```
[1] 73310.33
```

#variable 5

```
[1] 96465.17
```

#variable 6

```
[1] 74102.67
```

#variable 7

```
[1] 47762.5
```

5. Intensidad *shrinkage* óptima estimada a partir de la fórmula del artículo de *Clarence C. Y. Kwan*

#base de datos

```
mibbdd<-west.mat.clean[,1:100]
```

#a cada variable se le resta la media de la variable correspondiente

```
for (i in 1:100)
  mibbdd[,i]<-c(mibbdd[,i]-mean(mibbdd[,i]))
```

#Se calcula la matriz de covarianzas

```
bbdd<-cov(mibbdd)
```

```
#se obtiene la matriz de los wij estimados

meanwij<-(99/100)*bbdd

#########################################################################
                #Se calcula el numerador
#########################################################################

#Se calculan los casos wiln-wijest^2  para cada variable

k<-100; n<-100; r<-49
a<-array(NA, c(r,n,k))
for (t in 1:100)
   for (j in 1:100)
   for (i in 1:49)
     if (j!=t) a[,,t][i,j]<-(((mibbdd[i,t]*mibbdd[i,j])-meanwij[t,j]))^2

#Se calcula la suma de los casos (wij-meanwij)^2

k<-100
b<-array(0, c(k))
for (t in 1:100)
  for (j in 1:100)
  for (i in 1:49)
    if (j!=t) b[t]<-b[t]+a[,,t][i,j]

#se calcula el valor del numerador

sumvarsij<-0
for (i in 1:100)
sumvarsij<-sumvarsij+b[i]
sumvarsij<-(100/970299)*sumvarsij
sumvarsij

#########################################################################
                #Se calcula el denominador
#########################################################################

#cálculo de la matriz de covarianzas al cuadrado

matriz.covar2<-matrix(ncol=100,nrow=100)
for (j in 1:100)
for (i in 1:100)
   if (j!=i) matriz.covar2[i,j]<-bbdd[i,j]^2

#sumamos los elementos de la matriz

summatriz.covar2=0
for (j in 1:100)
for (i in 1:100)
   if (j!=i) summatriz.covar2<-summatriz.covar2+matriz.covar2[i,j]
```

Bibliografía

[1] Dobbin, K.K., and Simon, R.M., (2007). Sample Size Planning for Developing Classifiers Using High-Dimensional DNA Microarray Data. *Biostatistics*, 8(1), 101-117.

[2] Yao, J., Chang, C., Salmi, M.L., Hung, Y.S., Loraine, A., and Roux, S.J., (2008). Genome-scale Cluster Analysis of Replicated Microarrays Using Shrinkage Correlation Coefficient. *BMC Bioinformatics, 9:288.*

[3] Alan Julian Izenman (2008). Modern multivariate statistical techniques.

[4] Beerenwinkel, N., Antal, T., Dingli, D., Traulsen, A., Kinzler, K.W., Velculescu, V.E., Vogelstein, B., and Nowak, M.A., (2007). Genetic Progression and the Waiting Time to Cancer. PLoS Computational Biology, 3(11), 2239-2246.

[5] Tuo Zhao, Han Liu, Kathryn Roeder, John Lafferty, Larry Wasserman, (2013). High-dimensional Undirected Graph Estimation.

[6] Alan Genz, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, Friedrich Leisch, Fabian Scheipl, Bjoern Bornkamp, Torsten Hothorn, (2014). Mvtnorm: Multivariate Normal and t Distributions.

[7] Juliane Schafer, Rainer Opgen-Rhein, Verena Zuber, Miika Ahdesm\"aki, A. Pedro Duarte Silva, and Korbinian Strimmer, (2013). Corpcor: Efficient Estimation of Covariance and (Partial) Correlation.

[8] Jerome Friedman, Trevor Hastie and Rob Tibshirani, (2011). Glasso: Graphical lasso- estimation of Gaussian graphical models.

[9] Mike West, Carrie Blanchette, Holly Dressman, Erich Huang, Seiichi Ishida, Rainer Spang, Harry Zuzan, John A. Olson, Jr., Jeffrey R. Marks and Joseph R. Nevins, (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. vol. 98 no. 20, 11462-11467.

[10] Richard A. Johnson, Dean W. Wichern (third edition). Applied multivariate statistical analysis.

[11] Anderson, T.W. (2003). An Introduction to Multivariate Statistical Analysis.

[12] Stein, C. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In Proc. Third Berkeley Symp. Math. Statist. Probab. I 197-206. Univ. California Press, Berkeley. MR0084922.

[13] Johnstone, I. M. (2001). On the distribution of the largest eigenvalue in principal components analysis. Ann. Statist. 29 295-327. MR 1863961.

[14] R. Tibshirani, (1996) Regression Shrinkage and Selection via the Lasso. The Royal Statistical Society. Series B (Methodological), vol. 58, Nº 1.

[15] Donoho, D. and Johnstone, I. (1995). Adapting to unknown smoothness via wavelet shrinkage. J. Amer. Statist. Assoc. 90 1200-1224. MR1379464.

[16] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion), Annals of Statistics 32 (2): 407–499.

[17] Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), "Pathwise coordinate optimization", Annals of Applied Statistics, Vol.1, No. 2, 302-332.

[18] Boyd, S. & Vandenberghe, L. (2004). Convex Optimization, Cambridge University Press.

[19] Whittaker, J. (1990). Graphical Models in Applied Multivariate Statistics, Wiley, Chichester.

[20] Meinshausen, N. and Buhlmann, P. (2006). Highdimensional graphs and variable selection with the lasso. Ann. Statist. 34 1436-1462. MR2278363.

[21] Yuan, M., and Lin, Y. (2007). Model Selection and Estimation in the Gaussian Graphical Model", Biometrika, 94, 19-35. [893,897].

[22] Friedman, J., Hastie, T., Tibshirani, R. (2008b), Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9: 432-441.

[23] Banerjee, O., Ghaoui, L.E. & D'aspremont, A. (2007), 'Model selection through sparse maximum likelihood estimation', To appear, J. Machine Learning Research 101.

[24] Wu, T. & Lange, K (2007), Coordinate descent procedures for lasso penalized regression

[25] Referencia: Ryan Tibshirani (march 26 2013). Model selection and validation: Cross-validation.

[26] H. Liu, K. Roeder, and L. Wasserman (2010). Stability approach to regularization selection for high dimensional graphical models. Advances in Neural Information Processing Systems.

[27] S. Lysen (2009). Permuted Inclusion Criterion: A variable Selection Technique. PhD thesis, University of Pennsylvania.

[28] R. Foygel and M. Drton (2010). Extended Bayesian information criteria for Gaussian graphical models. Advances in Neural Information Processing Systems.

[29] Daniela M. Witten, Jerome H. Friedman, and Noah Simon (2011). New insights and faster computations for the graphical Lasso. Jounal of Computational and Graphical Statistics, volume 20, number 4, pages 892-900.

[30] Mazumder, R., and Hastie, T. (2011), "Exact covariance thresholding into connected components for large-scale graphica Lasso". http://arxiv.org/abs/1108.3829. [893].

[31] Ledoit, O. and M. Wolf (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. J. Empir. Finance 10, 603-621.

[32] Ledoit, O. and M. Wolf (2004a). Honey, I shrunk the sample covariance matrix. J. Portfolio Management 30, 110-119.

[33] Ledoit, O. and M. Wolf (2004b). A well-conditioned estimator for large-dimensional covariance matrices. J. Multiv. Anal. 88, 365-411.

[34] Schafer, J., and Strimmer, K., (2005). A Shrinkage Approach to Large-Scale Covariance Matrix estimation and Implications for Functional Genomics. Statistical Applications in Genetics and Molecular Biology, 4(1), Article 32.

[35] Kwan, Clarence C. Y. (2011) "An Introduction to Shrinkage Estimation of the Covariance Matrix: A Pedagogic Illustration," Spreadsheets in Education (eJSiE): Vol. 4: Iss. 3, Article 6.

[36] Friedman, J.H. (1989). Regularized discriminant analysis. J. Amer. Statist. Assoc. 84, 165-175.

[37] Morris, C. N. (1983). Parametric empirical Bayes inference: theory and applications. J. Amer. Statist. Assoc. 78, 47-55.

[38] Greenland, S. (2000). Principles of multilevel modelling. Intl. J. Epidemiol. 29, 158-167.

[39] Daniels, M. J. and R. E. Kass (2001). Shrinkage estimators for covariance matrices. Biometrics 57, 1173-1184.

[40] Hastie, T., R. Tibshirani, and J. Friedman (2001). The elements of Statistical Learning. New York: Springer.

[41] Kwan, C.C. Y., (2009). Estimation Error in the Correlation of Two Random Variables: A Spreadsheet-Based Exposition. Spreadsheets in Education, 3(2), Article 2.

[42] Opgen-Rhein R, Strimmer K (2007) Accurate ranking of differentially expressed genes by a distributionfree shrinkage approach. Stat Appl Genet Mol Biol 6:9.