**UNIVERSIDAD DE BUENOS AIRES**
**Facultad de Ciencias Exactas y Naturales**
**Departamento de Matemática**

**Tesis de Licenciatura**

# SELECCIÓN DE MODELOS GRÁFICOS NO-DIRIGIDOS EN EL CONTEXTO DE ALTA DIMENSIÓN

**Violeta Roizman**

**Directora:** Dra. Florencia Leonardi.
**Codirectora:** Dra. Mariela Sued.

Fecha de presentación: 29/05/2017

# Índice general

# Agradecimientos

Me gustaría agradecerle a algunas de las personas que me ayudaron a lo largo de la carrera.

A mi directora Florencia por animarse a dirigir la tesis a la distancia y sin conocerme. Gracias por leer cada uno mis larguísimos mails con preguntas filosóficas.

A Mariela, mi co-directora, que me ayudó desde más cerca con un montón de cosas y fue leyendo todo mientras estaba muy verde. También gracias por hacer que la estadística sea divertida.

A los jurados, Lucas y Daniela, por tomarse el tiempo de leer este trabajo.

A mi familia. Especialmente a ambos Marcelos por apoyarme incondicionalmente a lo largo de la carrera y a Clara y Félix por hacer mi vida más divertida.

A Nico por acompañarme siempre, escuchar mis preguntas sobre cómo se escribe una tesis y sobre cómo funciona el ranking ATP. Por el combo de cocinarme, lavar y hacerme té cuando estoy estudiando infinitas horas seguidas y no me despego de la compu. A Andrea y Sergio, que vinieron con Nico, y a los que recurrí para pedirles muchos consejos.

A la gente de matemática, física y computación con la que compartí cosas a lo largo de la carrera. Especialmente a Nahuel (con quien preparé infinitos finales), Tomás, Manu y Vir, Vero (que me ayudó a terminar el interminable trabajo final).

A mis amigas de antes de la facu por interesarse siempre.

¡GRACIAS!

# Introducción

El objetivo de esta tesis es estudiar los distintos métodos para la selección de modelos gráficos no-dirigidos en el contexto de alta dimensión. Introducimos a continuación qué significa cada uno de estos conceptos por separado y cómo se conectan entre sí.

Los **modelos gráficos** son modelos estadísticos multivariados que nos permiten visualizar fácilmente relaciones entre distintas variables aleatorias a partir de un grafo como el de la Figura 1, en donde cada nodo representa una variable y las aristas representan sus dependencias condicionales. Estos modelos nos proporcionan la libertad de dejar de lado las características paramétricas de las distribuciones a la hora de intentar responder ciertas preguntas con respecto a las relaciones entre variables: ¿Son **a** y **b** independientes conociendo el valor de **c**? ¿Es **a** una variable importante? ¿Hay grupos establecidos de variables? Además de utilizarse como una herramienta de visualización, son utilizados principalmente para abordar problemas clásicos del ámbito del aprendizaje estadístico, siendo uno de estos el problema de clasificación. El marco teórico de los modelos gráficos combina conceptos de probabilidades y de teoría de grafos, haciendo uso de las herramientas de ambos campos. Se utilizan, por ejemplo, la verosimilitud como método de estimación y algoritmos de grafos como los que buscan el árbol mínimo generador y el flujo máximo.

En general los modelos estadísticos surgen del análisis de muestras aleatorias finitas almacenadas en bases de datos. El **contexto de alta dimensión** refiere a que la cantidad de variables $p$ que tienen estos conjuntos de datos tiene un tamaño comparable o mayor que el tamaño de la muestra $n$. Muchos de los métodos clásicos asumen que $p < n$, lo que no permite aplicarlos en dicho contexto. Tomando como ejemplo el problema clásico de regresión lineal en el que tenemos una variable respuesta a predecir y variables explicativas, si la cantidad de co-variables es mayor que la cantidad de datos disponibles, el clásico método de cuadrados mínimos utilizado para resolverlo en datos de baja dimensión no está bien definido. En respuesta a esto, surge la idea de que al haber muchas

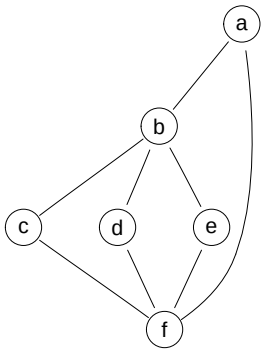



Figura 1: Ejemplo de grafo no dirigido

variables es probable que no todas sean útiles en la predicción de la variable respuesta y por ello, algunos métodos buscarán seleccionar un subconjunto de las variables originales. En particular, en el caso de los modelos gráficos en alta dimensión, se presenta el desafío de la computabilidad y de lograr encontrar modelos explicativos e interpretables esperando grafos relativamente ralos (con pocas aristas) para lograr este objetivo. Teniendo en cuenta que un grafo con $p$ nodos tiene $\frac{p\times(p-1)}{2}$ posibles aristas y por lo tanto existen $2^{(\frac{p\times(p-1)}{2})}$ posibles grafos con esa cantidad de nodos, los métodos exhaustivos pasan a ser intratables ya en el caso de valores medianos de $p$ (un valor de referencia podría ser $p = 10$).

Las dos principales clases de modelos gráficos, aunque existen muchas más, son las redes bayesianas (dirigidos) y las redes de Markov (**no-dirigidos**). Sus aplicaciones son muy variadas, se usan en distintos ámbitos tales como: genética (dirigidos y no-dirigidos, Wille et al. (2004)), procesamiento de lenguaje natural (Smith (2011)), reconocimiento de imágenes (no-dirigidos, Blake et al. (2011)) y reconocimiento del habla (modelos de markov ocultos, Gales y Young (2007)), entre otros. A veces la estructura gráfica del modelo es conocida o supuesta, como en el caso del procesado de imágenes, en el que una imagen se modela como una grilla no-dirigida en la que cada nodo es un píxel. Otras veces, se desconoce. Nos enfocaremos en este último caso y usaremos distintos métodos para estimar la estructura del grafo subyacente. A este proceso de estimación se lo llama **selección del modelo gráfico**.

En este trabajo hacemos una revisón biblográfica de algunos de los métodos existentes para la selección de modelos gráficos no-dirigidos en el contexto de alta dimensión, incluyendo simulaciones comparativas y aplicaciones a datos reales. La organización de esta tesis es la siguiente:

En el **Capítulo 1** realizamos una introducción a los modelos gráficos dirigidos y no-dirigidos poniendo énfasis en las propiedades de Markov que se codifican en los grafos.

En el **Capítulo 2** nos centramos en los modelos no-dirigidos en los que se representan normales multivariadas, estudiados en profundidad por sus propiedades particulares. Presentamos tres métodos para estimar los grafos: *Graphical Lasso* (basado en el método *lasso* para regresión lineal), *Nodewise Regression* y *Stability Selection*. Se presentan aplicaciones a datos genéticos y simulaciones para el estudio de la consistencia con los distintos métodos y del desempeño en la estimación de matrices de covarianza.

En el **Capítulo 3** nos centramos en el caso discreto, en particular el caso en el que las variables representadas son binarias. Presentamos tres métodos para la selección de modelos de este tipo: *Nodewise logistic Regression*, algoritmo *Chow-Liu* y un método no paramétrico demandante computacionalmente pero con condiciones débiles para la consistencia. Al final de este capítulo se presentan aplicaciones a datos reales y simulaciones para el estudio de la consistencia con los distintos métodos al estimar la estructura de grafos binarios.

Dejamos para el **Capítulo 4** algunos comentarios finales y el posible trabajo a futuro.

En el **Apéndice A** probamos los principales resultados sobre las distribuciones normales multivariadas.

Por último, en el **Apéndice B** presentamos una lista de los códigos escritos en el lenguaje **R** usados para los análisis de datos de este trabajo. Estos archivos se encuentran, junto con las bases de datos analizadas, en `https://github.com/violetr/tesis`.

# Capítulo 1

# Introducción a los modelos gráficos

En este capítulo empezamos repasando las definiciones básicas de teoría de grafos y de probabilidades necesarias para manejar el lenguaje de los modelos gráficos. Luego nos enfocamos en las propiedades de independencia condicional que pueden decodificarse de los modelos no-dirigidos y por último enunciamos los principales resultados para modelos dirigidos. La mayor parte del material de este capítulo fue extraída de los libros Lauritzen (1996) y Koller y N. Friedman (2009).

## 1.1. Definiciones básicas

### Grafos

Un **grafo** es un par $\mathcal{G} = (V, E)$ donde $V$ es un conjunto finito de **vértices** y el conjunto de **aristas** $E$ es un subconjunto de $V \times V$. Si $E$ es un conjunto de pares no-ordenados de vértices distintos diremos que el grafo es no-dirigido, en cambio si los pares son ordenados diremos que es dirigido y las aristas se representarán con flechas.

Dos vértices $i$ y $j$ son **adyacentes** si $(i, j) \in E$, lo notamos $i \sim j$. Una secuencia de vértices $(i_1, i_2, .., i_p)$ es un **camino** si $i_{k-1} \sim i_k$ para cada $2 \leq k \leq p$. Si $A$, $B$ y $C$ son tres conjuntos disjuntos de elementos de $V$, decimos que $C$ **separa a** $A$ de $B$ si todo camino entre un vértice de $A$ y otro de $B$ pasa por un vértice de $C$. En el ejemplo de la Figura 1.1 el conjunto de vértices $C = \{3, 2\}$ separa a $A = \{1\}$ de $B = \{4, 5\}$.

Llamamos **vecindario** de un vértice $v$ a todos los vértices adyacentes a $v$ en $\mathcal{G}$ y lo notamos $\mathcal{N}(v)$.

Un subconjunto $V' \subseteq V$ de vértices junto con un subconjunto $E' \subseteq (V' \times V' \cap E)$ es un **subgrafo** $\mathcal{G}' = (V', E')$ de $\mathcal{G}$. Dado un subconjunto $\widetilde{V} \subseteq V$, llamamos **subgrafo inducido** por $\widetilde{V}$ al subgrafo que tiene como aristas a todos los pares $(i, j)$ en $E$ tales que $i$ y $j$ pertenecen a $\widetilde{V}$, lo notamos $\mathcal{G}_{\widetilde{V}}$. Un grafo es **completo** si todo par de vértices es adyacente. Llamamos **clique** a un subgrafo $\mathcal{G}'$ completo de $\mathcal{G}$ que es maximal en el sentido de la inclusión, es decir, $\mathcal{G}'$ no está contenido en ningún subgrafo completo de $\mathcal{G}$ con mayor cantidad de vértices. En la Figura 1.1 tenemos tres cliques, una de tamaño 3 y las otras de 2 vértices.

### Modelos Gráficos

Un **modelo gráfico** es un modelo probabilístico multivariado que se representa a través de un grafo. En este grafo $\mathcal{G} = (V, E)$ cada vértice $i \in V$ representa a una variable aleatoria $X_i$ y las aristas codifican de alguna forma las independencias condicionales entre las variables. Estas variables aleatorias están definidas en espacios $\Omega_i$ con $\Omega = \times_{i \in V} \Omega_i$ un espacio de probabilidad

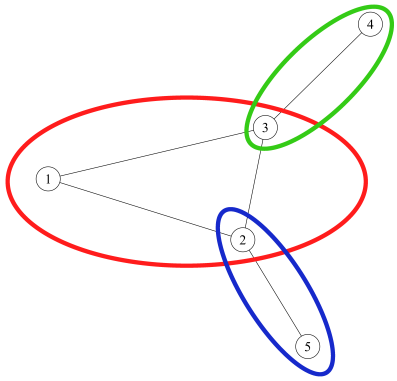



Figura 1.1: Ejemplo de grafo con sus cliques.

$(\Omega, \mathcal{F}, \mathbb{P})$ y tienen una distibución conjunta $\vec{X} = (X_1, ..., X_p) \sim \mathcal{D}$. Formalmente un modelo gráfico será un par $(\mathcal{G}, \mathcal{D})$.

Nuestro principal reto será la determinación de la estructura del grafo subyacente a la distribución de un vector aleatorio $\vec{X}$ a partir de la observación de una muestra aleatoria finita del vector. Particularmente nos centraremos en los modelos que tengan más parámetros a estimar que tamaño de muestra (alta dimensión).

Los tipos de modelos más estudiados son los dirigidos y los no-dirigidos. Algunas variantes de estas dos clases de modelos gráficos son los modelos bi-dirigidos, los modelos gráficos condicionales (CGM) y los modelos factoriales (FGM).

## Independencia condicional

El concepto de **independencia condicional** es central en este trabajo. Ilustramos la idea intuitiva de independencia condicional entre dos eventos en el siguiente ejemplo:

**Ejemplo 1.** *Cuando llueve vemos muchos pilotos de lluvia por la calle y hay un aumento en los accidentes de tránsito. A pesar de esto, sabemos que los pilotos de lluvia no causan accidentes de tráfico. Estos dos eventos están correlacionados sólo porque ambos están inducidos por la lluvia. Esto representa una independencia condicional: el número de pilotos usados es independiente de los accidentes de tránsito condicional a que esté lloviendo.*

A partir de esta idea intuitiva definimos formalmente el concepto:

**Definición 1.1.** *Un evento $A$ es condicionalmente independiente de otro evento $B$ dado $C$ (con respecto a $\mathbb{P}$) si $\mathbb{P}(A|B \cap C) = \mathbb{P}(A|C)$ o, equivalentemente, si $\mathbb{P}(A \cap B|C) = \mathbb{P}(A|C)\mathbb{P}(B|C)$. Lo notamos $A \perp B|C$.*

Generalizamos ahora la independencia condicional a variables aleatorias. Dadas $X$, $Y$ y $Z$ variables aleatorias, si las tres variables son discretas con función de probabilidad puntual $p$ definimos

$$X \perp Y|Z \iff p(x, y|z) = p(x|z)p(y|z). \tag{1.1}$$

En base a la definición dada tenemos una forma intuitiva de verificar si dos variables $X$ y $Y$ son condicionalmente independientes dada otra variable aleatoria $Z$. Suponiendo que dado el valor de $Z$ queremos adivinar el valor de $X$, ¿nos ayudaría conocer el valor de $Y$? En caso negativo tendremos que $X \perp Y|Z$.

Si las variables aleatorias son continuas y tienen una función de densidad definimos

$$X \perp Y|Z \iff f(x,y|z) = f(x|z)f(y|z) \text{ en casi todo punto.} \tag{1.2}$$

Equivalentemente vale que

$$X \perp Y|Z \iff \exists f, h \geq 0 \text{ tales que } f(x,y|z) = h(x,z)k(y,z) \text{ en casi todo punto.} \tag{1.3}$$

A esta útil caracterización la llamamos **criterio de factorización**.

Algunos de los resultados que usaremos sólo valdrán para distribuciones estrictamente positivas, es decir que tengan una densidad o una probabilidad puntual estrictamente positiva. El Ejemplo 2 de la Sub-sección 1.2.1 es un ejemplo de distribución discreta no-estrictamente positiva.

**Proposición 1.2.** *Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y $X, Y, Z$ variables aleatorias definidas en este espacio. Entonces se cumplen las siguientes propiedades:*

(*1*) *Simetría:* $X \perp Y|Z \Longrightarrow Y \perp X|Z$.

(*2*) *Descomposición:* $X \perp (Y,W)|Z \Longrightarrow X \perp Y|Z$.

(*3*) *Union débil:* $X \perp (Y,W)|Z \Longrightarrow X \perp Y|(Z,W)$.

(*4*) *Intersección: Si la distribución conjunta de $(X,Y,Z)$ es estrictamente positiva entonces $X \perp Y|(Z,W)$ y $X \perp W|(Z,Y) \Longrightarrow X \perp (Y,W)|Z$.*

## 1.2. Modelos no dirigidos

Los modelos no dirigidos, también llamados modelos de campos de Markov (MRF), representan relaciones de independencia condicional entre variables. Con ellos se modelan relaciones simétricas entre variables como se ve en la Figura 1.1. Si no existe una arista entre dos variables en el grafo diremos que estas son condicionalmente independientes dadas las demás variables del grafo.

### 1.2.1. Propiedades de Markov

Sea $V$ un conjunto de variables aleatorias con distribución $\mathcal{D}$, si construimos un grafo completo con una variable aleatoria por cada vértice y sacamos las aristas entre cada par de variables condicionalmente independiente dadas las demás variables entonces tenemos el grafo de Markov de a pares. Este grafo codifica el conjunto de relaciones de independencias condicionales de a pares.

Además de la propiedad de Markov de a pares existen las propiedades más generales que definimos a continuación.

**Definición 1.3.** *Dado un modelo gráfico $(\mathcal{G}, \mathcal{D})$ con un conjunto de variables aleatorias $\{X_i\}_i$ como vértices, decimos que $\mathcal{D}$ cumple:*

*(**P**) Propiedad de Markov de a pares con respecto a $\mathcal{G}$ si para todo par $(i,j)$ de vértices no adyacentes vale que*

$$X_i \perp X_j | X_{V \setminus \{i,j\}}. \tag{1.4}$$

*(**L**) Propiedad de Markov local con respecto a $\mathcal{G}$ si para todo vértice $i \in V$ vale que*

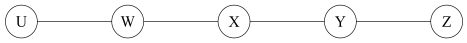
$$X_i \perp X_{V \setminus \mathcal{N}(i) \setminus \{i\}} | X_{\mathcal{N}(i)}. \tag{1.5}$$

*(**G**) Propiedad de Markov global con respecto a $\mathcal{G}$ si para $A$, $B$, $C$ conjuntos disjuntos de vértices de $V$ tales que $C$ separa a $A$ de $B$ en $\mathcal{G}$ entonces*

$$X_A \perp X_B | X_C. \tag{1.6}$$

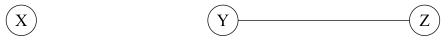
Los siguientes ejemplos, extraidos del libro Lauritzen (1996) ilustran las diferencias entre las distintas propiedades.

**Ejemplo 2.** *Sean $U, W, X, Y, Z$ variables binarias tales que $U$ y $Z$ son independientes y representan a una moneda equilibrada, $W = U$, $Y = Z$ y $X = WY$. La distribución conjunta satisface (**L**) pero no (**G**) para el siguiente grafo:*




*Se cumple la propiedad local pero no vale que $W \perp Y | X$.*

**Ejemplo 3.** *Sean las variables binarias $X, Y$ y $Z$ tales que $X = Y = Z$ y que representan una moneda equilibrada. Un grafo que satisface la propiedad (**P**) pero no (**L**) es el siguiente:*




*Se cumple que $X \perp Y | Z$ y que $X \perp Z | Y$ pero no vale que $X \perp Y, Z$.*

Veamos ahora algunos resultados que relacionan estas propiedades de Markov definidas.

**Proposición 1.4.** *Para cualquier distribución $\mathcal{D}$ definida en $\Omega$ valen las siguientes implicaciones: (**G**) $\Longrightarrow$ (**L**) $\Longrightarrow$ (**P**).*

*Demostración.* Veamos primero que si $\mathcal{D}$ cumple la propiedad (**L**) entonces cumple (**P**):

Sean $i, j \in V$ no adyacentes, entonces $j \notin \mathcal{N}(i)$ y como vale

$$X_i \perp X_{V \setminus \mathcal{N}(i) \setminus \{i\}} | X_{\mathcal{N}(i)}$$

entonces

$$X_i \perp (X_j, X_{V \setminus \mathcal{N}(i) \setminus \{i,j\}}) | X_{\mathcal{N}(i)}.$$

Finalmente por la propiedad de unión débil tenemos que

$$X_i \perp X_j | (X_{\mathcal{N}(i)}, X_{V \setminus \mathcal{N}(i) \setminus \{i,j\}}),$$

es decir $X_i \perp X_j | X_{V \setminus \{i,j\}}$.

Para probar (**G**) $\Longrightarrow$ (**L**) sólo es necesario observar que dado $i \in V$, entonces $C = \mathcal{N}(i)$ separa a $A = \{i\}$ de $B = V \setminus \mathcal{N}(i) \setminus \{i\}$ y reemplazar esos conjuntos en (**G**).

□

**Teorema 1.5.** *Sea un modelo gráfico no dirigido $(\mathcal{G}, \mathcal{D})$ con $\mathcal{G} = (V, E)$ y $\mathcal{D}$ una distribución estrictamente positiva, si $\mathcal{D}$ cumple (**P**) entonces cumple (**G**).*

*Demostración.* Queremos ver que para $A$, $B$ y $C$ disjuntos tales que $C$ separa a $A$ y $B$ vale que $X_A$ es independiente de $X_B$ condicional a $X_C$. Vamos a probar esto por inducción descendiente en el tamaño $k$ de $C$.

Caso $k = n - 2$: necesariamente debe ser $C = V \setminus \{i, j\}$ con $i$ y $j$ un par de vértices de $V$, es decir que estamos en el caso (**P**).

Supongamos ahora que vale para conjuntos $A$, $B$ y $C$ disjuntos con $|C| = k$ con $k \leq n - 2$ y veamos que vale para conjuntos $A$, $B$ y $C$ con $|C| = k - 1$. Estudiamos dos casos por separado:

En en primer caso consideramos $A$, $B$ y $C$ disjuntos con $|C| = k-1$ y tales que $A \cup B \cup C = V$. Entonces, como $|C| \leq n - 3$, necesariamente $A$ o $B$ tienen más de un vértice. Supongamos sin pérdida de generalidad que $|B| \geq 2$, entonces sea $b \in B$ consideremos $B' = B \setminus b$. $C$ sigue separando a $A$ y $B'$ en $\mathcal{G}$ y también separa a $A$ y $\{b\}$ en $\mathcal{G}$. Esto implica también que $C \cup \{b\}$ separa a $A$ y $B'$ y que por otro lado $C \cup B'$ separa a $A$ y $\{b\}$. Ambos conjuntos "separadores"tienen cardinal estrictamente mayor a $k - 1$, por lo tanto si aplicamos la hipótesis inductiva tenemos que

$$X_A \perp X_{B'} | X_{C \cup \{b\}}$$

y que

$$X_A \perp X_b | X_{C \cup B'}.$$

Al ser $\mathcal{D}$ positiva podemos aplicar la propiedad de intersección y obtenemos que $X_A \perp X_{B'}, X_b | X_C$ equivalente a $X_A \perp X_B | X_C$, es decir, $\mathcal{D}$ cumple (G).

En el segundo caso consideramos $A$, $B$ y $C$ disjuntos con $|C| = k - 1$ y tales que no son una partición de $V$. Ahora existe la posibilidad de que ambos $A$ y $B$ tengan sólo un vértice. Si no pasa esto, la demostración es igual que para el caso anterior. Veamos qué pasa en caso contrario. Consideramos entonces los conjuntos disjuntos $A = \{i\}$, $B = \{j\}$, $C$ y $D = (V \setminus \{i, j\} \setminus C)$. Como $C$ separa a $\{i\}$ de $\{j\}$, no puede haber un camino entre ellos que pase sólo por vértices de $D$. En particular existe un vértice $l \in D$ que no tiene caminos en $D$ hacia ambos $\{i\}$ y $\{j\}$. Supongamos sin pérdida de generalidad que en $D$ no hay un camino de $l$ a $i$, entonces $C \cup \{j\}$ separa a $l$ de $i$. Observemos que $C \cup \{j\}$ tiene cardinal mayor estricto que $k - 1$. Por otro lado como $C$ separa a $A$ y $B$ entonces $C \cup \{l\}$ (también con cardinal mayor estricto que $k - 1$) separa a $A$ y $B$ . Si aplicamos la hipótesis inductiva obtenemos

$$X_A \perp X_B | X_{C \cup \{l\}}$$

y

$$X_A \perp X_l | X_{C \cup B}.$$

Aplicando la propiedad de intersección obtenemos $X_A \perp X_B, X_l | X_C$. Finalmente usamos la propiedad de descomposición para ver que $X_A \perp X_B | X_C$.

□

Tenemos entonces que para modelos con distribución $\mathcal{D}$ positiva todas las propiedades de Markov definidas son equivalentes. Este teorema nos permite construir el grafo de manera simple de a pares y a partir del grafo ya construido deducir otras independencias condicionales usando la propiedad global de Markov.

### 1.2.2. Factorización

Definimos ahora el concepto de **factorización** que está directamente emparentado con las relaciones de independencia condicional en un modelo gráfico y que bajo ciertas condiciones resulta ser un enfoque equivalente al de separación.

**Definición 1.6.** *Decimos que $p$ (en el caso de distribuciones discretas la función de probabilidad puntual y en el de absolutamente continuas la función de densidad) se factoriza con respecto a $\mathcal{G}$ (o que cumple la propiedad (**F**) con respecto a $\mathcal{G}$) si*

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{A \,:\, \mathcal{G}_A \text{ subgrafo completo de } \mathcal{G}} \phi_A(\boldsymbol{x}_A),$$

*donde $Z$ es la constante de normalización también llamada función de partición y $\phi_A$ son funciones positivas que dependen solo de las variables $X_A$. A estas funciones se las llama potenciales o factores.*

Vale aclarar que esta representación no es única. Como cada subgrafo completo está incluido en una clique (puede ser él mismo), podemos escribir siempre a $p$ en función de factores dependientes de las variables de cliques maximales.

**Proposición 1.7.** *Para cualquier distribución $\mathcal{D}$ definida en $\Omega$ vale la siguiente implicación: (**F**) $\implies$ (**G**).*

*Demostración.* Sean $A$, $B$ y $C$ conjuntos disjuntos de vértices de $\mathcal{G} = (V, E)$ tales que $C$ separa a $A$ de $B$ consideremos $\mathcal{G}_{V \setminus C}$ el subgrafo inducido por $V \setminus C$. $A$ y $B$ están en distintas componentes conexas de $\mathcal{G}_{V \setminus C}$, consideremos entonces $V_A$ el conjunto de componentes conexas en $\mathcal{G}_{V \setminus C}$ que contienen a los vértices de $A$. $V_A$, $C$ y $V \setminus (C \cup V_A)$ son disjuntos y además valen las siguientes inclusiones: $A \subseteq V_A$ y $B \subseteq V \setminus (C \cup V_A)$. Un subgrafo completo inducido por $D \subseteq V$ puede estar incluido en los subgrafos inducidos por $V_A \cup C$ o por $V \setminus V_A$ ya que en otro caso existiría un camino entre $A$ y $B$ en $\mathcal{G}_{V \setminus C}$. Vamos a escribir la factorización de la función de probabilidad (densidad o probabilidad puntual) separando en estos dos conjuntos:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{D:D \text{ completo}} \phi_D(\mathbf{x}) = \frac{1}{Z} \prod_{D \subseteq V_A \cup C} \phi_D(\mathbf{x}) \times \prod_{\widetilde{D} \subseteq V \setminus V_A; \widetilde{D} \not\subseteq C} \phi_{\widetilde{D}}(\mathbf{x}).$$

Podemos reescribir el producto de la siguiente forma:

$$\mathbb{P}(X_1, ..., X_p) = \frac{1}{Z} f(X_{V_A}, X_C) g(X_{V \setminus (C \cup V_A)}, X_C).$$

De esta ecuación se deduce por el criterio de factorización que $V_A \perp V \setminus (C \cup V_A) | C$. A partir de esta independencia podemos concluir que $A \perp B | C$ aplicando dos veces la propiedad de descomposición. □

Probamos ahora que vale la vuelta para probabilidades continuas positivas aunque la demostración puede adaptarse a distribuciones discretas. Para eso probamos que bajo la condición de positividad (**P**) implica (**F**) y por lo tanto todas las propiedades de Markov definidas son equivalentes a (**F**). Usamos el siguiente lema.

**Lema 1.8.** *(Inversión de Möbius) Sean $\Phi$ y $\Psi$ funciones reales definidas en un conjunto de subconjuntos de $V$ finito. Entonces son equivalentes:*

(1) *Para todo* $A \subseteq V$*:* $\Psi(A) = \sum_{B:B\subseteq A} \Phi(B)$.

(2) *Para todo* $A \subseteq V$*:* $\Phi(A) = \sum_{B:B\subseteq A} (-1)^{|A\setminus B|}\Psi(B)$.

*Demostración.* Veamos que (2) implica (1), la otra implicación es análoga. Partimos de la definición de $\Psi(A)$:

$$\begin{aligned}\sum_{B:B\subseteq A} \Phi(B) &= \sum_{B:B\subseteq A}\sum_{C:C\subseteq B} (-1)^{|B\setminus C|}\Psi(C) \\ &= \sum_{C:C\subseteq A} \Psi(C)\left(\sum_{B:C\subseteq B\subseteq A} (-1)^{|B\setminus C|}\right) \\ &= \sum_{C:C\subseteq A} \Psi(C)\left(\sum_{H:H\subseteq A\setminus C} (-1)^{|H|}\right).\end{aligned}$$

Como cualquier conjunto no vacío tiene la misma cantidad de conjuntos pares e impares entonces la suma entre paréntesis de la última expresión sólo es distinta de cero en el caso en el que $A \setminus C$ es el conjunto vacío. Teniendo esto en cuenta solo queda el término en el que $C = A$ obteniendo lo que queríamos probar.

□

**Teorema 1.9.** ***(Hammersley-Clifford)*** *Sea un modelo gráfico* $(\mathcal{G}, \mathcal{D})$ *no-dirigido con* $\mathcal{G} = (V, E)$ *y* $\mathcal{D}$ *una distribución continua con densidad* $f$ *estrictamente positiva. Si* $\mathcal{D}$ *cumple* (***P***) *entonces cumple* (***F***).

*Demostración.* Queremos ver que la distribución conjunta puede escribirse como un producto de factores que dependen de las variables de los subgrafos completos de $\mathcal{G}$. Como $f$ es estrictamente positiva vamos a tomar logaritmo para trabajar con sumas en lugar de productos.

Sea $\breve{\mathbf{x}}$ un elemento fijado arbitrariamente del espacio de probabilidad y $A \subseteq V$ definimos

$$H_A = \log(f(\mathbf{x}_A, \breve{\mathbf{x}}_{V\setminus A})),$$

donde $(\mathbf{x}_A, \breve{\mathbf{x}}_{V\setminus A})$ es el elemento $\mathbf{y}_{(V)}$ que tiene como coordenadas $\mathbf{y}_{(v)} = \mathbf{x}_{(v)}$ si $v \in A$ e $\mathbf{y}_{(v)} = \breve{\mathbf{x}}_{(v)}$ si $v \notin A$. Al estar fijado $\breve{\mathbf{x}}$ tenemos que $H_A$ depende sólo de $\mathbf{x}_A$. También definimos para cada subconjunto $A \subseteq V$

$$\phi_A(\mathbf{x}) = \sum_{B:B\subseteq A} (-1)^{|A\setminus B|} H_B(\mathbf{x}).$$

Con esta definición también $\phi_A$ depende únicamente de $\mathbf{x}_A$. Aplicando el lema anterior obtenemos

$$\log(f(\mathbf{x})) = H_V(\mathbf{x}) = \sum_{A:A\subseteq V} \phi_A(\mathbf{x}).$$

Por lo tanto para ver que se cumple (**F**) falta ver que para todo subconjunto $\widetilde{A} \subseteq V$ cuyo subgrafo inducido no es completo $\phi_{\widetilde{A}}$ es 0.

Sea $\widetilde{A}$ un subconjunto de $V$ cuyo subgrafo inducido $\mathcal{G}_{\widetilde{A}}$ no es completo, sean $\alpha$ y $\beta$ dos vértices no adyacentes de $\mathcal{G}_{\widetilde{A}}$ y $C = \widetilde{A} \setminus \{\alpha, \beta\}$. Entonces, teniendo en cuenta las cuatro posibilidades (a

saber: que $B$ no contenga ni a $\alpha$ ni a $\beta$, que contenga a alguna de ellas o que contenga a ambas) reescribimos $\phi_{\widetilde{A}}$:

$$\phi_{\widetilde{A}}(\mathbf{x}) = \sum_{\widetilde{B}:\widetilde{B}\subseteq C} (-1)^{|C\setminus\widetilde{B}|}(H_{\widetilde{B}} - H_{\widetilde{B}\cup\{\alpha\}} - H_{\widetilde{B}\cup\{\beta\}} + H_{\widetilde{B}\cup\{\alpha,\beta\}}). \tag{1.7}$$

Veamos ahora que $\phi_{\widetilde{A}}$ es 0. Sea $D = V \setminus \{\alpha, \beta\}$ tenemos las siguientes igualdades:

$$\begin{aligned}
H_{\widetilde{B}\cup\{\alpha,\beta\}}(\mathbf{x}) - H_{\widetilde{B}\cup\{\alpha\}}(\mathbf{x}) &= \log\left(\frac{f(\mathbf{x}_{\widetilde{B}}, \mathbf{x}_\alpha, \mathbf{x}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}{f(\mathbf{x}_{\widetilde{B}}, \mathbf{x}_\alpha, \breve{\mathbf{x}}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}\right) \\
&= \log\left(\frac{f(\mathbf{x}_\alpha|\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})f(\mathbf{x}_{\widetilde{B}}, \mathbf{x}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}{f(\mathbf{x}_\alpha|\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})f(\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}\right) \\
&= \log\left(\frac{f(\breve{\mathbf{x}}_\alpha|\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})f(\mathbf{x}_{\widetilde{B}}, \mathbf{x}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}{f(\breve{\mathbf{x}}_\alpha|\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})f(\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}\right) \\
&= \log\left(\frac{f(\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_\alpha, \mathbf{x}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}{f(\mathbf{x}_{\widetilde{B}}, \breve{\mathbf{x}}_\alpha, \breve{\mathbf{x}}_\beta, \breve{\mathbf{x}}_{D\setminus\widetilde{B}})}\right) \\
&= H_{\widetilde{B}\cup\{\beta\}}(\mathbf{x}) - H_{\widetilde{B}}(\mathbf{x}).
\end{aligned}$$

En la primera igualdad simplemente usamos las definición de $H_{(.)}$, en la segunda usamos la definición de probabilidad condicional y usamos que la distribución de nuestro modelo cumple (**P**) y que $\alpha$ y $\beta$ no son adyacentes. De esta forma obtenemos en el numerador y en el denominador el mismo factor y lo que hacemos para obtener la siguiente expresión es cancelar para luego multiplicar y dividir por una expresión conveniente. Así, en la cuarta igualdad, volvemos a razonar de la misma forma que en la segunda para obtener la última expresión.

Por último, usando esta igualdad en (1.7) queda $\phi_{\widetilde{A}} \equiv 0$ para subgrafo inducidos por $\widetilde{A}$ no completo.

□

## 1.3. Modelos dirigidos

En esta sección sólo enunciamos los principales resultados de modelos dirigidos para completar la introducción a modelos gráficos.

Un modelo dirigido, también usualmente llamado modelo de red bayesiano, se representa con un grafo dirigido como los de las Figuras 1.2 y 1.3. Es un par $\mathcal{G} = (V, E)$ que consiste en un conjunto $V$ de vértices y $E$ un conjunto de pares ordenados de $V$. Si $(i, j) \in E$ entonces hay una flecha apuntando de $i$ hacia $j$.

Dos vértices $i$ y $j$ son adyacentes si están unidos con una flecha en algún sentido. Si hay una flecha de $i$ hacia $j$ entonces $i$ es **padre** de $j$ y $j$ es **hijo** de $i$. El conjunto de padres de $i$ se nota $pa(i)$. Un camino dirigido entre $i$ y $j$ es un conjunto de flechas, todas apuntando en el mismo sentido, uniendo ambas variables. $i$ es un **antecesor** de $j$ si existe un camino dirigido que va de $i$ hacia $j$. Decimos también que $j$ es **descendiente** de $i$. Un camino dirigido que empieza y termina en la misma variable es un ciclo dirigido. Un grafo dirigido es acíclico si no tiene ciclos dirigidos. Decimos que el grafo es dirigido y acíclico o que es un **DAG**. Por lo general, es este tipo de grafo dirigido el más estudiado.

Definamos la propiedad de Markov que se deduce de un modelo gráfico dirigido.

**Definición 1.10.** *Si $\mathcal{D}$ es la distribución de $\vec{X}$ con una función de densidad $f$, decimos que $\mathcal{D}$ cumple Markov con respecto a $\mathcal{G}$ si*

$$f(\boldsymbol{x}) = \prod_{i=1}^{k} f(\boldsymbol{x}_i | \boldsymbol{x}_{pa(i)}), \tag{1.8}$$

*donde $pa(i)$ son los padres de $X_i$. Al conjunto de distribuciones representados por $\mathcal{G}$ lo llamamos $\mathcal{M}(\mathcal{G})$.*

El siguiente resultado nos permite deducir independencias condicionales entre pares de variables conociendo a los padres de alguno de ellos. Es la generalización de las propiedades de las cadenas de Markov.

**Teorema 1.11.** *$\mathcal{D} \in \mathcal{M}(\mathcal{G})$ si y sólo si para toda variable $i$ en $\mathcal{G}$ vale que:*

$$X_i \perp X_j | X_{pa(i)}, \tag{1.9}$$

*donde $j$ es cualquier variable de $\mathcal{G}$ que no sea padre ni descendiente de $i$.*

Con el objetivo de encontrar una condición gráfica más general que la de este teorema para leer independencias condicionales de los modelos dirigidos se considera la **d-separación**. Para definir este concepto gráfico necesitamos introducir las definiciones que siguen.

Un **camino** entre $i$ y $j$ es una secuencia de distintos vértices adyacentes, por ejemplo: $i \to h \leftarrow l \leftarrow m \to j$. Decimos que un vértice $l$ de un camino entre $i$ y $j$ es un **colisionador** si el camino es de la forma $i \cdots \to l \leftarrow \cdots j$. Si no se tiene esta disposición de flechas decimos que es un no-colisionador.

En el grafo dirigido de la Figura 1.2 vemos que hay una **colisión** en $Y$, por lo tanto el vértice $Y$ es un colisionador. Hay que tener en cuenta que la definición de colisionador es siempre en relación a un camino en particular ya que un vértice puede ser colisionador en un camino pero no-colisionador en otro. Por este motivo siempre diremos que un vértice es colisionador con respecto a un cierto camino.

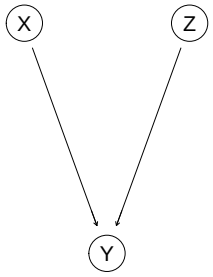



Figura 1.2: Ejemplo de una colisión en Y.

Decimos que un camino $\pi$ d-conecta incondicionalmente a $i$ con $j$ si $i$ y $j$ son los extremos de $\pi$ y no hay colisionadores en $\pi$. Si no existe un camino d-conectando a $i$ con $j$ entonces $i$ y $j$ están (incondicionalmente) d-separados.

**Lema 1.12.** *Dada cualquier distribución $\mathcal{D}$ que se factoriza con respecto a un grafo $\mathcal{G}$, si $i$ y $j$ están incondicionalmente d-separados entonces $X_i \perp X_j$.*

Definimos ahora formalmente el concepto general de separación dirigida:

**Definición 1.13.** *Sea $\mathcal{G}$ un grafo dirigido decimos que un camino $\pi$ d-conecta a $i$ con $j$ condicionalmente a un conjunto $C$ de vértices que no los contiene si ambos son los extremos del camino y además:*

(*1*) *Todo no-colisionador en $\pi$ no pertenece a $C$ y*

(*2*) *Todo colisionador en $\pi$ es un antecesor de $C$ o pertenece a $C$.*

*Si no existe ningún camino que d-conecte $i$ con $j$ condicionalmente al conjunto $C$ entonces $i$ y $j$ están d-separados dado $C$.*

*Sean $A$ y $B$ conjuntos no vacíos de vértices, estos están d-separados dado $C$ si para todo $i \in A$ y $j \in B$, $i$ y $j$ están d-separados dado $C$.*

El siguiente ejemplo ilustra este concepto:

**Ejemplo 4.** *En el **DAG** de la Figura 1.3 vemos que:*

- ***4** y **6** están d-conectados dado $C = \{\mathbf{7}\}$ por el camino $\pi = (\mathbf{4},\mathbf{5},\mathbf{6})$ ya que el único vértice interior de $\pi$ es **5** y es un colisionador en $\pi$ que es antecesor de $C$.*
- ***1** y **6** están d-conectados dado $C = \{\mathbf{7}\}$ ya que de los dos caminos posibles $\pi_1 = (\mathbf{1},\mathbf{3},\mathbf{4},\mathbf{5},\mathbf{6})$ y $\pi_2 = (\mathbf{1},\mathbf{4},\mathbf{5},\mathbf{6})$, $\pi_2$ los d-conecta dado $C$.*
- ***1** y **6** están d-separados dado $C = \{\mathbf{4},\mathbf{7}\}$ ya que ninguno de los dos caminos posibles los d-conecta dado $C$.*

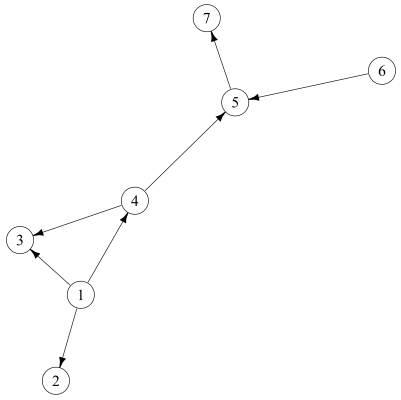
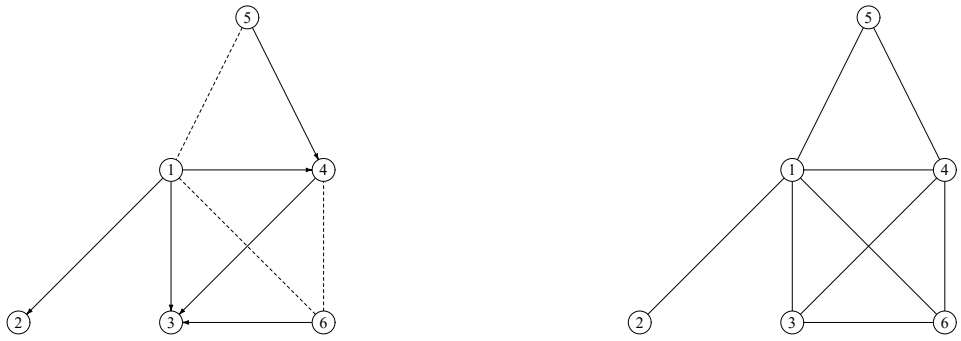



Figura 1.3: Ejemplo de grafo dirigido.

Una vez definido formalmente el concepto de d-separación enunciamos el siguiente teorema que relaciona directamente las independencias condicionales de un grafo dirigido con las d-separaciones en él.

**Teorema 1.14.** *Sean $A$, $B$ y $C$ conjuntos disjuntos de vértices. Si $A$ y $B$ están d-separados por $C$ entonces $X_A \perp X_B | X_C$.*

A partir de este resultado sabemos que lo que se deduzca del grafo con el concepto gráfico de d-separación será lo más general que se pueda decodificar de un grafo dirigido con respecto a las independencias condicionales.




(a) (b)

Figura 1.4: Proceso de moralización de un grafo. En la figura (a) vemos el grafo dirigido original con linea sólida y las conexiones que van a ser agregadas por ser entre padres de un hijo en común con linea punteada. En la figura (b) vemos el resultado final del proceso en el que agregamos las aristas necesarias y convertimos las flechas en aristas no dirigidas.

Existen algoritmos eficientes para determinar la d-separación entre dos vértices para los cuales no es necesario considerar enteros todos los caminos posibles entre ambos vértices.

Por último introducimos el concepto de **moralización** que relaciona los grafos dirigidos con los no-dirigidos. Es una transformación de un grafo dirigido en uno no-dirigido que consiste en dos pasos que se ven en la Figura 1.4: primero conectar entre sí a todos los vértices que tengan un hijo en común y después considerar a todas las flechas como aristas no dirigidas. Si $\mathcal{G}$ es un grafo dirigido notamos $\mathcal{G}^m$ al grafo que resulta de moralizar $\mathcal{G}$ y lo llamamos el grafo moral de $\mathcal{G}$.

Sea $A$ un conjunto de vértices notamos $\text{an}(A)$ al conjunto que resulta de unir los conjuntos de ancestros cada uno de los vértices que pertenecen a $A$. En base a esto definimos al **grafo ancestral** de $A$ como el subgrafo inducido por $\text{an}(A)$.

Finalmente el siguiente resultado relaciona la d-separación con la separación en el grafo moralizado.

**Teorema 1.15.** *Sean $A$, $B$ y $C$ conjuntos disjuntos de vértices de un grafo dirigido $\mathcal{G}$. Entonces $A$ y $B$ están d-separados dado $C$ en $\mathcal{G}$ si y sólo si están separados por $C$ en $(\mathcal{G}_{\text{an}(A\cup B\cup C)})^m$.*

A partir de este resultado tenemos un algoritmo para testear d-separación comprobando separación en grafos no-dirigidos.

## 1.4. Poder expresivo de los modelos gráficos dirigidos y no-dirigidos

Surge la pregunta de si alguna de las dos clases de modelo presentadas es más expresiva con respecto a las independencias condicionales que codifica que la otra. Es decir, si consideramos a $D$ y $U$ como los conjuntos de las relaciones de independencia condicional entre variables que pueden describirse con los DAGs y las redes de Markov respectivamente, nos gustaría saber si hay una relación de inclusión entre ellos. La respuesta es que ambos conjuntos tienen intersección pero no hay inclusión de ninguna de las dos partes. A continuación mostramos ejemplos.

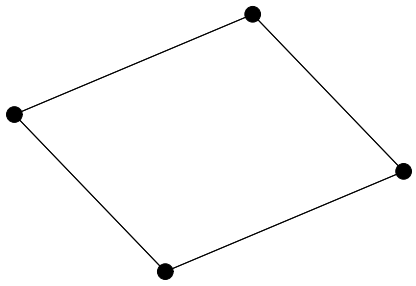


Figura 1.5: Ejemplo de grafo no dirigido con forma de diamante.

Cualquier grafo sin inmoralidades puede representarse tanto con DAGs como con redes de Markov.

Ningún DAG puede representar exactamente todas las (in)dependencias de un grafo como el de la Figura 1.5.

Ningun modelo de Markov puede expresar exactamente las relaciones de independencia condicional del grafo de la Figura 1.2.

# Capítulo 2

# Modelo Gaussiano

Uno de los casos de modelos gráficos más estudiados es el que representa un vector aleatorio normal multivariado. Esto se debe a la relación directa que hay entre las independencias condicionales de las distribuciones marginales y la inversa de la matriz de covarianza de esta distribución en particular. En este capítulo presentamos distintos métodos para la estimación del grafo subyacente para esta distribución en el contexto de altas dimensiones. Los métodos presentados son *Graphical Lasso* (J. Friedman, Trevor et al. (2008)), *Nodewise regression* (Meinshausen y Bühlmann (2006)) y *Stability Selection* (Meinshausen y Bühlmann (2010)). El primero es un método con una penalización análoga a la del método *lasso* para regresión lineal, el segundo se basa en realizar numerosas regresiones lineales con el método *lasso* y el tercero es un método que se basa en el resampleo para lograr estimaciones estables.

## 2.1. Propiedades

En el modelo gaussiano la distribución conjunta de las variables representadas en el grafo es la de una normal multivariada. Podemos identificar a una normal multivariada $\vec{X} = (X_1, .., X_p) \sim N(\boldsymbol{\mu}, \Sigma)$ por su media $\boldsymbol{\mu}$ y su matriz de covarianza $\Sigma$. A la inversa de $\Sigma$ la llamaremos matriz de precisión.

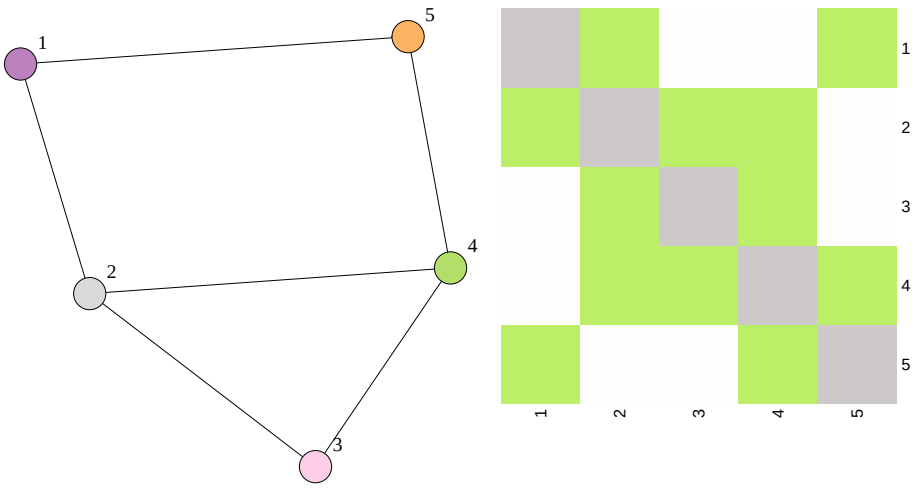
La propiedad de Markov de a pares es en este caso equivalente a la propiedad global de Markov por la Proposición 1.5 debido a que la distribución gaussiana es estrictamente positiva.

Esta distribución tiene la propiedad de que en la matriz de precisión $\Sigma^{-1}$ los ceros codifican las independencias condicionales entre las variables. Por lo tanto la idea será poner una arista entre la variable $i$ y la $j$ en el grafo si $\Sigma^{-1}_{ij} \neq 0$. La demostración de esta propiedad y de otros resultados teóricos sobre la normal que se usarán en este capítulo se encuentran desarrolladas en el **Apéndice A**. La Figura 2.1 ilustra esta relación estrecha entre los ceros de la matriz $\Sigma^{-1}$ y el grafo del modelo correspondiente.

El método *Graphical Lasso* es un método de estimación para modelos gráficos no dirigidos en el caso de altas dimensiones que está basado directamente en la penalización *lasso* utilizada en el caso de regresión con altas dimensiones que introducimos a continuación.

## 2.2. Regresión lineal con *lasso*

Tengamos en cuenta el siguiente problema: dadas las variables $x$ e $y$ y $n$ datos observados $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ nos gustaría poder explicar los valores de la variable $y$ con los valores de $x$. También nos gustaría dado un nuevo valor de $x$ poder predecir el valor de la variable $y$. Con esto en mente




(a) (b)

Figura 2.1: (a) Modelo gráfico gaussiano de 5 variables. (b) Patrón de ceros (representado por las casillas blancas) de la matriz de precisión correspondiente al vector aleatorio de los vértices de (a).

pensaremos que $y$ es una función de $x$ con una componente aleatoria aparte que no podremos controlar, en particular resulta simple pensar que la función es lineal. Buscaremos entonces la recta que mejor se ajuste a los puntos como se muestra en la Figura 2.2.

Este es el problema de regresión lineal simple, $X$ e $Y$ son dos variables aleatorias tales que $Y = \beta_0 + \beta_1 X + \varepsilon$. A $X$ la llamamos variable explicativa, a $Y$ variable respuesta y a $\varepsilon$ el error. Nuestro objetivo será estimar los coeficientes $\beta_i$ y una vez calculados podremos predecir fácilmente a $Y$ conociendo un nuevo valor $x^{(*)}$ de $X$ con la siguiente fórmula:

$$\widehat{y^{(*)}} = \widehat{\beta_0} + \widehat{\beta_1} x^{(*)}, \tag{2.1}$$

con $\widehat{\beta_0}$ y $\widehat{\beta_1}$ las estimaciones de los coeficientes correspondientes.

El problema de regresión lineal simple se generaliza al problema de regresión lineal múltiple donde hay muchas variables explicativas representadas en el vector aleatorio $\vec{X}$. Tradicionalmente, para resolver este problema se eligen los coeficientes $\boldsymbol{\beta}_i$ que minimizan el error cuadrático medio:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimizar}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left( y^{(i)} - \beta_0 - \sum_{j=1}^{p} \boldsymbol{x}_j^{(i)} \boldsymbol{\beta}_j \right)^2 \right\}, \tag{2.2}$$

con $\boldsymbol{\beta}$ el vector (indexado desde 1) tal que sus $i$-ésima componente es $\boldsymbol{\beta}_i$.

Cuando $p \leq n$ se pueden obtener los coeficientes a partir de las ecuaciones normales y son únicos. En cambio con $p > n$ la minimización de cuadrados mínimos no da un único óptimo.

Al estar en el contexto de altas dimensiones estamos en peligro de sobreajuste, esto significa que dado mi conjunto de datos ajuste los coeficientes considerando todas las variables que tengo alcanzando un error muy pequeño pero que este ajuste prediga con mucho error a la variable

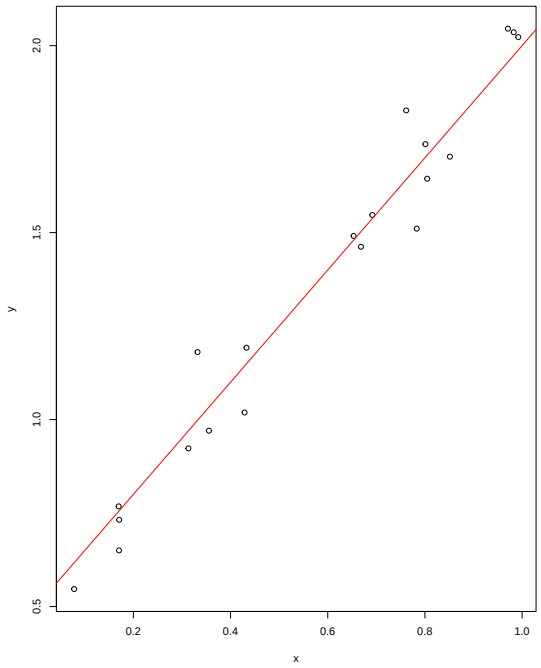



Figura 2.2: Ejemplo un conjunto de puntos en el plano ajustado a una recta con el método de cuadrados mínimos.

respuesta en un conjunto nuevo de datos $\{\boldsymbol{x}^{(*,i)}\}_{i=1}^{m}$. Vamos a querer identificar un conjunto más pequeño de variables explicativas que tengan una mayor capacidad predictiva a la hora de intentar predecir la variable respuesta.

Basándose en esto, en Tibshirani (1996) se introduce el método *lasso* (**l**east **a**bsolute **s**hrinkage and **s**election **o**perator) que consiste en combinar la pérdida de mínimos cuadrados con una restricción para la norma $l_1$ del vector de coeficientes $\boldsymbol{\beta}$:

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimizar}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left( y^{(i)} - \beta_0 - \sum_{j=1}^{p} \boldsymbol{x}_j^{(i)} \boldsymbol{\beta}_j \right)^2 \right\} \text{ sujeto a } \|\boldsymbol{\beta}\|_1 \leq t. \tag{2.3}$$

Esta restricción tiene el efecto de reducir el valor absoluto de los coeficientes, hasta incluso volver cero algunos, dejando de lado al término independiente que no penalizaremos. Este método introduce un sesgo en la estimación $\widehat{\boldsymbol{\beta}}$ de los coeficientes, que el tradicional no tenía, pero a cambio de eso gana en precisión y en interpretabilidad del resultado obtenido (ver Sub-sección 2.2.3). Esta característica nos da además un criterio de selección de variables explicativas.

Otra forma conveniente y equivalente de escribir el método *lasso* es sumándole al error cuadrático medio una penalización de la forma

$$\underset{\boldsymbol{\beta},\beta_0}{\text{minimizar}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} \left( y^{(i)} - \beta_0 - \sum_{j=1}^{p} \boldsymbol{x}_j^{(i)} \boldsymbol{\beta}_j \right)^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{2.4}$$

para algún $\lambda \geq 0$. Estas dos formas de plantear el problema tienen una correspondencia uno a uno: para cada $\lambda$ existe un $t$ tal que los dos problemas tienen la misma solución y viceversa.

Este es un problema convexo y por lo tanto relativamente fácil de optimizar con distintos métodos. Los algoritmos más usados para resolverlo son: el llamado algoritmo de "descenso coordinado"(propuesto por Hastie y Tibshirani) y el método LARS que calcula la optimización para un camino de valores de $\lambda$ y estima por interpolación lineal a los $\lambda$s intermedios.

### 2.2.1. ¿ Por qué norma 1 ?

Existen otros métodos de regularización similares a *lasso* como los propuestos por Frank y Friedman (1993) usando una penalización de la norma $l_q$ con cualquier $q$; el método *ridge* que usa una penalización de la norma $l_2$ al cuadrado o *elastic net* (J. Friedman et al. (2010)) que es una combinación convexa entre *lasso* y *ridge*. El método *ridge* en su planteo es análogo a *lasso* usando la norma $l_2$ al cuadrado en lugar de la norma $l_1$. Esta diferencia en apariencia menor lleva a grandes diferencias en su funcionamiento: *ridge* estima todas las componentes de $\widehat{\boldsymbol{\beta}}$ distintas de cero mientras que *lasso* estima a lo sumo $n$ componentes de $\widehat{\boldsymbol{\beta}}$ distintas de cero. Una idea geométrica de por qué *lasso* es mejor que *ridge* para anular coeficientes consiste en pensar en el planteo de minimización (2.2) en el cual queremos minimizar una función

$$f(\boldsymbol{\beta}, \beta_0) = \frac{1}{2n} \sum_{i=1}^{n} \left( y^{(i)} - \beta_0 - \sum_{j=1}^{p} \boldsymbol{x}_j^{(i)} \boldsymbol{\beta}_j \right)^2$$

con la variable sujeta a la restricción de que su norma 1 sea menor que $t$ (o en el caso *ridge* norma 2 al cuadrado menor que $t$). Para que sea más intuitivo tomamos el caso $p = 2$ y sin término independiente $\beta_0$ que se ve en la Figura 2.3 tomada del libro Bühlmann y Van de Geer (2011). Analicemos primero el caso del método *lasso* : dado un $t \geq 0$ la región en donde minimizo a $f$ queda restringida a un rombo. Sea $\widehat{\boldsymbol{\beta}}$ un mínimo de $f(\boldsymbol{\beta})$ sin restricciones en las variables, que sería lo que obtendríamos al hacer cuadrados mínimos tradicional, veamos que bajo ciertas condiciones las curvas de nivel de $f$ son elipses centradas en $\widehat{\boldsymbol{\beta}}$ que se van expandiendo a medida que $f$ toma valores más altos. La ecuación implícita general de una cónica cualquiera es

$$ax^2 + 2hxy + by^2 + 2gx + 2fy + c = 0,$$

si se cumple que $h^2 < ab$ y que el invariante cúbico $\Delta$ es estrictamente mayor que 0 (el determinante de cierta matriz de $3 \times 3$ formada por los seis constantes $a, b, h, g, f, c$) entonces se trata de una elipse. Desarrollamos la curva de $f(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ para el valor $k$:

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \left( y^{(i)} - (\boldsymbol{x}_1^{(i)} \boldsymbol{\beta}_1 + \boldsymbol{x}_2^{(i)} \boldsymbol{\beta}_2) \right)^2 - 2nk \\ &= (\sum_{i=1}^{n} \boldsymbol{x}_1^{(i)}) \boldsymbol{\beta}_1^2 + 2(\sum_{i=1}^{n} \boldsymbol{x}_1^{(i)} \boldsymbol{x}_2^{(i)}) \boldsymbol{\beta}_1 \boldsymbol{\beta}_2 + (\sum_{i=1}^{n} \boldsymbol{x}_2^{(i)}) \boldsymbol{\beta}_2^2 + \sum_{j=1}^{2} (-2 \sum_{i=1}^{n} y^{(i)} \boldsymbol{x}_j^{(i)}) \boldsymbol{\beta}_j + \sum_{i=1}^{n} (y^{(i)})^2 - 2nk, \end{aligned}$$

y vemos que efectivamente bajo ciertas condiciones de los datos sus curvas de nivel son elipses tomando $a = \sum_{i=1}^{n} \boldsymbol{x}_1^{(i)}$, $b = \sum_{i=1}^{n} \boldsymbol{x}_2^{(i)}$ y $h = \sum_{i=1}^{n} \boldsymbol{x}_1^{(i)} \boldsymbol{x}_2^{(i)}$. Usando la desigualdad de Cauchy-Schwarz sabemos que $h^2 \leq ab$. Si $\Delta < 0$ y además no existe $q$ real tal que $\boldsymbol{x}_1^{(i)} = q \boldsymbol{x}_2^{(i)}$ para todo $i = 1, ..., n$ entonces las curvas son elipses. Están centradas en el mínimo $\widehat{\boldsymbol{\beta}}$ pues es el único mínimo (vale en el contexto de bajas dimensiones y como $p = 2$ se aplica) y se expanden a medida que el valor $k$ aumenta.

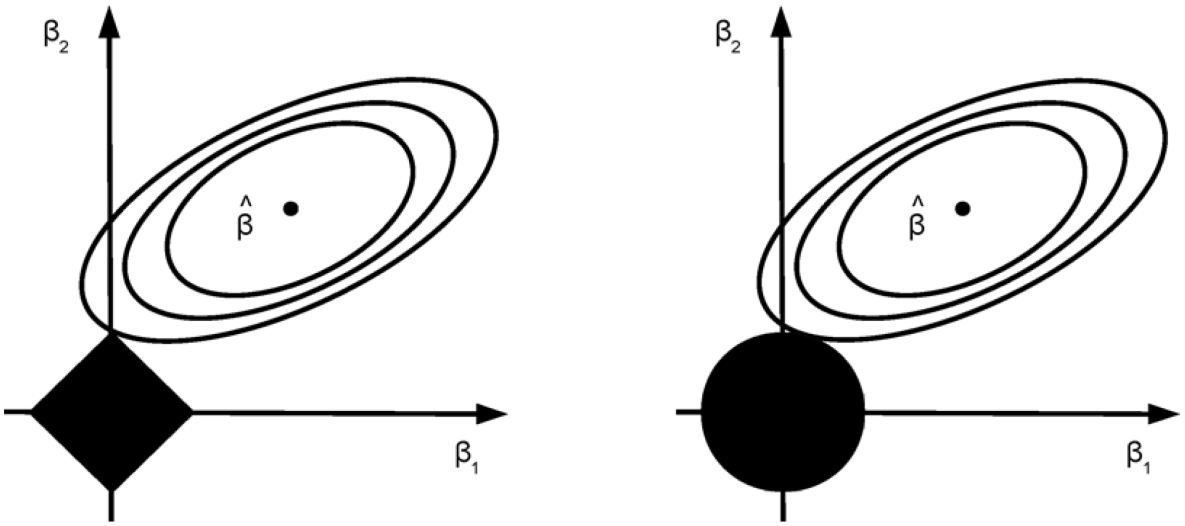



Figura 2.3: Idea geométrica de regresión con *lasso* y con *ridge* para el caso $p = 2$. Esta imagen fue extraída del libro Bühlmann y Van de Geer (2011).

Salvo que uno de los ejes de las elipses quede justo paralelo a uno de los lados del rombo, las elipses van a encontrarse con el rombo en alguno de los vértices y estos corresponden a que alguno de los coeficientes de $\boldsymbol{\beta}$ sea nulo. En cambio, en el caso de *ridge* la región factible del problema de optimización es un círculo con el cual en contadas excepciones las elipses curvas de nivel van a encontrarse cuando alguna de los coeficientes de $\boldsymbol{\beta}$ se anule. Esta idea intuitiva se traslada al caso con más dimensiones a pesar de que no pueda justificarse estos mismos argumentos.

### 2.2.2. Elección del parámetro $\lambda$

Cuando estamos en un contexto de aprendizaje en el cual tenemos varias opciones de modelos vamos a querer elegir al mejor modelo. Según cuál sea nuestro objetivo hay muchos criterios que podemos tener en cuanto a qué significa "ser mejor". El mejor puede ser por ejemplo el que sea más preciso, el más exacto o el que tenga mayor interpretabilidad. En general, vamos a querer un balance entre estas cosas. La forma más común de medir la bondad de un ajuste en general de cierto modelo con una variable respuesta numérica continua es con el error cuadrático medio (MSE) que corresponde a la definición

$$\text{MSE} = \frac{1}{m}\sum_{k=1}^{m}(y^{(k)} - \widehat{y^{(k)}})^2, \tag{2.5}$$

donde $\{y^{(k)}\}_{k=1}^m$ son los datos de la variable respuesta e $\{\widehat{y^{(k)}}\}_{k=1}^m$ son las correspondientes predicciones. En el caso de regresión lineal múltiple las predicciones corresponden a la fórmula $\widehat{y^{(k)}} = \widehat{\beta}_0 + \sum_{j=1}^{m} \widehat{\boldsymbol{\beta}_j}\boldsymbol{x}_j^{(k)}$ donde los coeficientes $\widehat{\boldsymbol{\beta}_i}$ son calculados en base a un conjunto de datos que llamaremos conjunto de **entrenamiento**.

Esta cantidad, como medida de calidad, tiene sentido calcularla para un nuevo conjunto de datos que no sea el conjunto de entrenamiento usado para estimar los coeficientes, en otro caso estaremos sobreajustando el modelo. Algunas veces tenemos datos suficientes para separar una porción del

conjunto de datos original para usar de conjunto de **validación** y por ejemplo dejaremos de lado al 30 % de los datos con este fin. Otras, tenemos un tamaño de muestra muy limitado y no podemos permitirnos esto. Para estos casos podemos usar el método **validación cruzada** con $k$ grupos ($k$-fold CV).

El método de validación cruzada se basa en la división del conjunto de datos en $k$ grupos, con $k$ algún número arbitrario entre 1 y el tamaño de la muestra $n$. Dado un modelo que estemos considerando para explicar la naturaleza de nuestros datos, realizaremos $k$ iteraciones en las que en cada una de ellas el conjunto $i$-ésimo de los $k$ actúa como conjunto de validación y los demás $k-1$ grupos actúan en conjunto como datos de entrenamiento. En cada iteración ajustamos el modelo con el conjunto transitorio de entrenamiento y calculamos su calidad (en el caso de regresión lineal por ejemplo con MSE) en el de validación. Al finalizar las $k$ iteraciones tendremos $k$ medidas de error por lo que promediaremos estos errores para tener una idea de como se comporta ese modelo en particular en el conjunto de datos. Repetiremos este procedimiento para cada modelo que tengamos en consideración, así, al finalizar, compararemos los promedios de errores y eligiremos el modelo cuyo valor sea el menor. Este método introduce la decisión del parámetro $k$ que en general se lo toma como 5 o 10. Además, este método tiene como ventajas que no desperdicia datos de entrenamiento y que estabiliza el valor del error cuadrático medio (en comparación con el método del único conjunto de validación) pero como desventaja que aumenta la cantidad de cómputos necesarios para realizar la misma tarea (elegir el modelo ganador).

El método de validación cruzada puede aplicarse no sólo para la elección entre distintos métodos, sino también cuando tenemos un sólo modelo pero un parámetro a elegir como en el caso de $\lambda$ en el método *lasso*. En este caso, como tengo infinitas posibilidades de elección para $\lambda$ tendremos que seleccionar un conjunto finito de opciones en un rango conveniente a elección. En general elegiremos un $\lambda_{\max}$ (uno que fuerce al vector $\widehat{\boldsymbol{\beta}}$ a ser el vector nulo) y en base a este elegiremos un camino de valores de $\lambda$ descendientes.

Otros métodos comúnmente usados para la elección de este parámetro son los métodos AIC y BIC. Ambos maximizan la función de verosimilitud penalizando la cantidad de parámetros estimados para evitar sobreajuste.

### 2.2.3. Comparación de cuadrados mínimos y *lasso*

Dadas $\boldsymbol{x}_*$ e $y_*$ realizaciones de $\vec{X}$ e $Y$ respectivamente. Sea $f$ la función que describe la relación entre $\vec{X}$ e $Y$ de la forma $Y = f(\vec{X}) + \varepsilon$ y sea $\widehat{f}$ la función que dado un valor de $\boldsymbol{x}$ devuelve su predicción correspondiente al modelo ajustado. Usando que para $X$ una variable aleatoria con segundo momento se cumple $\mathrm{V}[X]=\mathrm{E}[X^2]-\mathrm{E}[X]^2$ deducimos

$$\mathrm{E}[(y_* - \widehat{f}(\boldsymbol{x}_*))^2] = sesgo^2(\widehat{f}(\boldsymbol{x}_*)) + \mathrm{V}(\widehat{f}(\boldsymbol{x}_*)) + \mathrm{V}(\varepsilon), \tag{2.6}$$

donde el sesgo de un estimador se define como la esperanza de la diferencia entre el parámetro estimado y el estimador.

Esto nos indica que entre el sesgo y la varianza hay un balance. Salvo que la esperanza del error sea 0 (es decir que tengamos un estimador perfecto) cuando tengamos muy bajo sesgo seguramente tendremos alta varianza y viceversa.

Veamos ahora empíricamente, con una simulación, el balance entre el sesgo y la varianza para distintos valores de $\lambda$. Para eso generamos 100 veces un conjunto de datos en el cual las variables explicativas $\{X^{(i)}\}_{i=1}^{1000}$ son normales estándar independientes y la variable respuesta responde a la ecuación $Y = 2X_1 + X_{400} + 0.5X_3 + \varepsilon$, con $\varepsilon$ también con distribución normal estándar independiente. Elegimos además un conjunto de valores de $\lambda$, los valores pequeños representarán más flexibilidad mientras que los valores más grandes representarán menor flexibilidad debido a que valores más

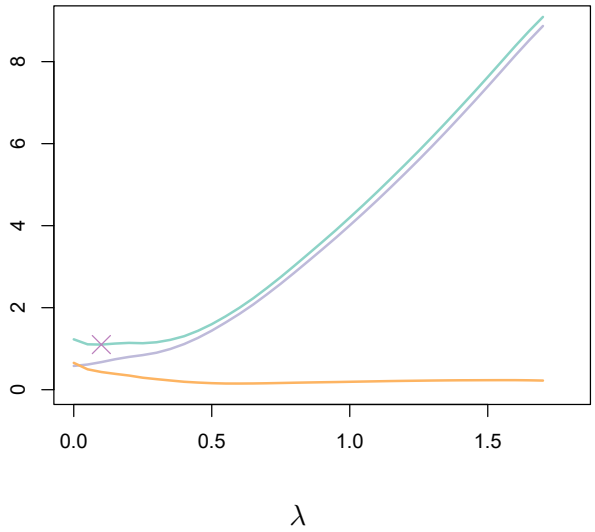



Figura 2.4: Gráfico del error cuadrático medio (en turquesa), sesgo al cuadrado (en lila) y varianza (en naranja) en función del valor de $\lambda$ usado como penalización en el método *lasso*. El valor de $\lambda$ con menor error cuadrático medio se encuentra señalado con una cruz.

altos de $\lambda$ se traducen en mayor cantidad de coeficientes estimados como 0. Para cada uno de los conjuntos de datos generados y para cada $\lambda$ ajustamos el modelo con el método *lasso* y luego estimamos la variable respuesta para un valor $\boldsymbol{x}_*$ escogido al azar y estimamos las medidas de la ecuación (2.6). Con las estimaciones para cada $\lambda$ realizamos la Figura 2.4 en la que se puede ver la forma característica de U del error cuadrático medio, el sesgo creciente con respecto a $\lambda$ y la varianza decreciente con respecto a $\lambda$. Observamos entonces al valor particular de $\lambda = 0$ que corresponde a cuadrados mínimos tradicional en contraposición con los demás valores de $\lambda$. En el caso de cuadrados mínimos vemos el menor sesgo pero también la mayor varianza que es lo que habíamos adelantado en la Sección 2.2: el método *lasso* introduce un sesgo en la estimación de los coeficientes a cambio de una disminución en la varianza con respecto a cuadrados mínimos.

### 2.2.4. Consistencia en la selección de variables

Más allá del problema de la estimación de los coeficientes de regresión para la predicción, en el área de modelos gráficos nos va a interesar particularmente el problema de selección de variables. Este problema consiste en recuperar el conjunto $S_*$ de variables cuyo coeficiente real $\beta_*$ de la regresión es distinto de cero:

$$S_* = \{j \in \{1, ..., p\} | (\boldsymbol{\beta}_j)_* \neq 0\}, \tag{2.7}$$

y dejar de lado las demás. En el contexto de alta dimensión el problema de selección de variables es de gran relevancia ya que existen $2^p$ posibles subconjuntos de variables, que con $p$ grande son muchas variables, demasiadas. Además de esto tenemos la complicación de que en los estudios asintóticos a medida que $n$ aumenta, $p$ también puede hacerlo.

Una forma de hacer la selección de variables es sumar a la función de mínimos cuadrados

una penalización proporcional a la cantidad de coeficientes no nulos ($\sum_{i=1}^{p} I(\boldsymbol{\beta}_i \neq 0)$). Este método penaliza la cantidad alta de variables seleccionadas pero tiene la desventaja de no resultar una función convexa en $\boldsymbol{\beta}$ y por lo tanto resulta una función difícil de optimizar cuando $p$ es grande. Según la bibliografía usar *lasso* para la selección de variables en altas dimensiones es útil pero en general se sobreestima a $S_*$, es decir que no se cumple en general la consistencia definida de la siguiente forma. Sea $\widehat{S}$ la estimación $S_*$ decimos que el método es consistente si

$$P(\widehat{S} = S_*) \longrightarrow 1(n \to \infty). \tag{2.8}$$

De todos modos se definen las (restrictivas) condiciones equivalentes *neighborhood stability* (Meinshausen y Bühlmann (2006)) y *irrepresentable condition* que se piden para mostrar consistencia.

La condición de irrepresentabilidad fue introducida en Zou (1996) y en Zhao y Yu (2006). Sea $S_* = \{1, ..., s_*\}$ el conjunto de variables explicativas relevantes (las reordenamos), $\widehat{S}$ su estimador y $\widehat{\Sigma}$ la matriz empírica de covarianza. Dividimos en bloques a $\widehat{\Sigma}$ de la siguiente forma:

$$\begin{pmatrix} \widehat{\Sigma}_{1,1} & \widehat{\Sigma}_{1,2} \\ \widehat{\Sigma}_{2,1} & \widehat{\Sigma}_{2,2} \end{pmatrix},$$

donde $\widehat{\Sigma}_{1,1}$ tiene dimensión $s_* \times s_*$, $\widehat{\Sigma}_{1,2} = \widehat{\Sigma}_{2,1}^T$ y $\widehat{\Sigma}_{2,2}$ tiene dimensión $(p - s_*) \times (p - s_*)$. La condición de irrepresentabilidad de define como:

$$\|\widehat{\Sigma}_{1,2}\widehat{\Sigma}_{1,2}^{-1}\text{sign}((\boldsymbol{\beta}_1)_*, ..., (\boldsymbol{\beta}_{(s_*)})_*)\|_\infty \leq \theta \text{ para algún } 0 < \theta < 1, \tag{2.9}$$

donde $\text{sign}((\boldsymbol{\beta}_1)_*, ..., (\boldsymbol{\beta}_{(s_*)})_*) = (\text{sign}((\boldsymbol{\beta}_1)_*), ..., \text{sign}((\boldsymbol{\beta}_{(s_*)})_*))$. Es una condición suficiente y "esencialmente" necesaria para probar 2.8. "Esencialmente"se refiere a que para la condición necesaria se usa en la ecuación (2.9) la condición $\theta \leq 1$ mientras que para la condición suficiente se usa la desigualdad estricta $\theta < 1$.

Por otro lado las variaciones que a continuación presentamos logran un mejor resultado a la hora de la selección de variables.

### 2.2.5. Variaciones del método *lasso*

Motivados por diversas razones como por ejemplo reducir el sesgo, mejorar la complejidad algorítmica y mejorar la selección de variables existen muchas variaciones del método *lasso*. Dos ejemplos que vamos a usar de estas variaciones son *lasso* adaptable (*adaptive lasso*) y *lasso* truncado (*thresholded lasso*).

#### Adaptive *lasso*

El método *adaptive* propuesto en Zou (1996) consiste en realizar la regresión en dos pasos de los cuales el primero es *lasso* tradicional. Ya contando con los coeficientes $\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}$ estimados del primer paso, procedemos a aplicar el método *lasso* nuevamente pero agregando penalidades $1/|\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}|$ a los coeficientes $\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}$ distintos de 0 y una penalidad grande (por ejemplo mayor al máximo de las penalidades ya definidas) a los que ya eran 0. En particular, el método agrega una penalidad muy grande a los coeficientes inicialmente estimados que eran casi cero y logra tener menos sesgo que *lasso*. El planteo del segundo paso del método es

$$\underset{\boldsymbol{\beta}, \beta_0}{\text{minimizar}} \left\{ \frac{1}{2n}\|\mathbf{y} - \beta_0 - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} \frac{|\boldsymbol{\beta}_j|}{|\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}|} \right\}, \tag{2.10}$$

donde $\mathbf{X}$ es la matriz de datos que contiene a las $p$ variables como columnas, $\mathbf{y}$ es el vector de valores respuesta y $\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}$ son las componentes de $\boldsymbol{\beta}$ estimados en el primer paso del método.

Resumiendo, este método tiene la siguiente propiedad:

$$\widehat{\boldsymbol{\beta}_{j_{\text{init}}}} = 0 \Longrightarrow \widehat{\boldsymbol{\beta}_{j_{\text{adapt}}}} = 0 \text{ para todo } j = 1, ..., p. \tag{2.11}$$

El método reduce el número de falsos positivos (variables seleccionadas que no son relevantes). Esta propiedad es buena ya que sabemos que el método *lasso* tiene la propiedad de que el conjunto seleccionado de variables incluye a las variables relevantes originales con alta probabilidad.

Podemos convertir el planteo de (2.10) en un problema *lasso* para aplicar el algoritmo *lasso* tradicional de la siguiente forma:

$$\tilde{X}_j = |\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}| X_j, \ \widetilde{\boldsymbol{\beta}}_j = \frac{\boldsymbol{\beta}_j}{|\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}|}.$$

La función a minimizar resulta ser

$$\frac{1}{2n}\|\mathbf{y} - \widetilde{\boldsymbol{\beta}}_0 - \widetilde{\mathbf{X}}\widetilde{\boldsymbol{\beta}}\|_2^2 + \lambda\|\widetilde{\boldsymbol{\beta}}\|^{(1)}. \tag{2.12}$$

Luego reconstruimos los coeficientes:

$$\widehat{\boldsymbol{\beta}_{j_{\text{adapt}}}} = |\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}|\widehat{\widetilde{\boldsymbol{\beta}}_j} \ \text{ para todo } \ j = 1, ..., p.$$

**Thresholded *lasso***

El método truncado consiste en fijar un umbral crítico $\tau$ para determinar el valor de los coeficientes. Si un coeficiente calculado con *lasso* tiene valor absoluto menor que $\tau$ entonces este método lo determina como cero con la siguiente fórmula:

$$\widehat{\boldsymbol{\beta}_{j_{\text{thres}}}} = \widehat{\boldsymbol{\beta}_{j_{\text{init}}}} I(|\widehat{\boldsymbol{\beta}_{j_{\text{init}}}}| > \tau). \tag{2.13}$$

Las variables seleccionadas serán $\widehat{S}_{\text{thres}} = \{j; \widehat{\boldsymbol{\beta}_{j_{\text{thres}}}} \neq 0\}$ pero haremos un reajuste en los coeficientes estimando con el método de cuadrados mínimos tradicional tomando en cuenta sólo las variables cuyo coeficiente supera al umbral $\tau$ (notar que como *lasso* selecciona a lo sumo $n$ variables entonces podemos usar las ecuaciones normales para resolver):

$$\widehat{\boldsymbol{\beta}}_{\text{thres-refit}} = (\mathbf{X}_{\widehat{S}_{\text{thres}}}^T \mathbf{X}_{\widehat{S}_{\text{thres}}})^{-1} \mathbf{X}_{\widehat{S}_{\text{thres}}}^T \mathbf{y},$$

donde $\mathbf{X}$ es la matriz de datos que contiene a las $p$ variables como columnas y para un subconjunto $S \subseteq \{1, ..., p\}$, $\mathbf{X}_S$ es la matriz formada por las columnas en $S$ de $\mathbf{X}$.

Para la selección del parámetro $\tau$ procedemos secuencialmente, primero con validación cruzada seleccionamos el valor de $\lambda$ y luego con este parámetro fijo seleccionamos con CV el valor de $\tau$.

A pesar de su simplicidad, este método obtiene iguales o mejores resultados que adaptive *lasso* para la selección de variables predictoras.

## 2.3. *Graphical Lasso*

En el caso de modelos gráficos gaussianos el objetivo será, como dijimos antes, estimar $\Sigma^{-1}$ para luego colocar la arista $(i, j)$ del grafo por cada $\widehat{\Sigma_{ij}^{-1}} \neq 0$.

Dada la función de densidad de la normal multivariada

$$f(\boldsymbol{x};\boldsymbol{\mu},\Sigma)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \tag{2.14}$$

y sea $\{\boldsymbol{x}^{(i)}\}_{i=1}^n$ un conjunto de realizaciones $\boldsymbol{x}^{(i)}=(\boldsymbol{x}_1^{(i)},..,\boldsymbol{x}_p^{(i)})$ del vector aleatorio $\vec{X}$, planteamos la función de verosimilitud. Tomando logaritmo y sacando constantes obtenemos

$$l(\boldsymbol{\mu},\Sigma;\boldsymbol{x}^{(1)},..,\boldsymbol{x}^{(n)})=-n\log(\det(\Sigma^{1/2}))-\frac{1}{2}\sum_{i=1}^n(\boldsymbol{x}^{(i)}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}^{(i)}-\boldsymbol{\mu}).$$

Para condensar un poco la expresión llamamos $\Theta=\Sigma^{-1}$ a la matriz de precisión y usamos propiedades de la función traza. Usamos que $\mathrm{tr}(ABC)=\mathrm{tr}(CAB)=\mathrm{tr}(BCA)$ y que para una constante $c$ vale $\mathrm{tr}(c)=c$. Estimando además a $\boldsymbol{\mu}$ por su estimador de máxima verosimilitud $\widehat{\boldsymbol{\mu}_{MV}}=\frac{1}{n}\sum\limits_{i=1}^n\boldsymbol{x}^{(i)}$ (ver cuenta en el **Apéndice A**), la función de verosimilitud queda

$$l(\Theta;\boldsymbol{x}^{(1)},..,\boldsymbol{x}^{(n)})=\frac{n}{2}\log(\det(\Theta))-\frac{n}{2}\mathrm{tr}\left(\frac{1}{n}\sum_{i=1}^n(\boldsymbol{x}^{(i)}-\widehat{\boldsymbol{\mu}}_{MV})(\boldsymbol{x}^{(i)}-\widehat{\boldsymbol{\mu}}_{MV})^T\Theta)\right).$$

Sacando las constantes y agregando el término de regularización *lasso* con una penalización de la norma $l_1$ queda la expresión

$$l_\lambda(\Theta;\boldsymbol{x}^{(1)},..,\boldsymbol{x}^{(n)})=\log(\det(\Theta))-\mathrm{tr}(\mathbf{S}\Theta)-\lambda\|\Theta\|_1, \tag{2.15}$$

con $\mathbf{S}=n^{-1}\sum_{i=1}^n(\boldsymbol{x}^{(i)}-\widehat{\boldsymbol{\mu}}_{MV})(\boldsymbol{x}^{(i)}-\widehat{\boldsymbol{\mu}}_{MV})^T$ la matriz empírica de covarianza y la norma $\|\Theta\|_1=\sum\limits_{j<k}|\Theta_{jk}|$ sin incluir a los valores de la diagonal.

El método *Graphical Lasso* (*GLasso*) consiste en maximizar la función (2.15) sobre el espacio de matrices simétricas definidas positivas.

El valor de $\lambda$ representa qué tanto regularizamos el modelo. En el caso de $\lambda$ pequeño estamos cerca del caso sin penalización en el cual el grafo estimado tendrá muchas aristas. Por el contrario, en el caso de $\lambda$ muy grande, estamos ante un modelo bastante forzado en el que quizás muchas aristas importantes son desechadas. Podemos elegir el valor de este parámetro, al igual que en el caso de regresión, mediante un proceso de *k-fold cross-validation* para el cual se divide al conjunto de datos en k conjuntos de similar tamaño y se hacen k iteraciones. Consideramos un conjunto de valores de $\lambda$ a elegir en algún rango que sea conveniente. En la iteración k-ésima el subconjunto k pasa a cumplir el papel de conjunto de validación y el resto de los datos es el conjunto de entrenamiento. Para cada iteración se calcula, para cada $\lambda$ considerado, la matriz simétrica positiva $\Theta$ que maximiza la ecuación (2.15) en los datos de entrenamiento y se calcula el valor de la log-verosimilitud sin penalizar para el conjunto k de validación:

$$\log(\det(\widehat{\Theta_{\text{train}}}))-\mathrm{tr}(\mathbf{S}_{\text{val}}\widehat{\Theta_{\text{train}}}), \tag{2.16}$$

con $\widehat{\Theta_{\text{train}}}$ la matriz que maximiza (2.15) para los datos de entrenamiento y $\mathbf{S}_{\text{val}}$ la matriz empírica de covarianza calculada en los datos de validación. Al finalizar el proceso tendremos k valores de log-verosimilitud para cada posible valor $\lambda$. Para cada $\lambda$ tomamos el promedio de esas k medidas de ajuste y nos quedamos con el parámetro que tenga el mayor de estos promedios.

Una vez ya elegido el valor de $\lambda$, vamos a estimar el conjunto $E$ de aristas del grafo de la siguiente forma sin hacer ningún test de significación:

$$\widehat{E}=\{(j,k)\in V\times V;\widehat{\Sigma_{jk}^{-1}}\neq 0\}. \tag{2.17}$$

Este estimador es razonablemente calculable en altas dimensiones y consistente si se piden ciertas condiciones (restrictivas) (Ver Sección 2.6). Asumiendo algunas condiciones como una dispersión suficientemente grande en la matriz de precisión (cantidad de ceros en $\Theta_*$) y una limitación en sus autovalores tenemos que se cumple:

$$E_* \subseteq \widehat{E}. \tag{2.18}$$

Es decir que en general esta estimación resulta por encima del conjunto real de aristas. Para resolver esto se propone el método *adaptive GLasso* inspirado directamente en el *adaptive lasso*. Se usa como $\Sigma^{-1}$ inicial a uno obtenido con *GLasso* que llamaremos $\widehat{\Sigma^{-1}}_{\text{init}}$ y se efectúa la siguiente minimización:

$$\widehat{\Sigma^{-1}} = \underset{\Sigma^{-1}\succ 0}{\operatorname{argmin}} \left( -\log(\det(\Sigma^{-1})) + \operatorname{tr}(\mathbf{S}\Sigma^{-1}) + \lambda \sum_{j<k} w_{jk}\Sigma^{-1}_{jk} \right), \tag{2.19}$$

con $w_{jk} = 1/|\widehat{\Sigma^{-1}_{\text{init};jk}}|$. El objetivo de esta segunda etapa es la de volver cero algunos de los valores de la matriz $\widehat{\Sigma^{-1}_{\text{init}}}$ que en el primer paso quedaron con valor absoluto chico pero sin ser cero.

**Medidas de ajuste de la estimación de $\widehat{\Sigma^{-1}}$:**

Una vez ya estimada $(\Sigma^{-1})_*$ y estando en el caso de una simulación (en donde conocemos el valor real de $\Sigma^{-1}$) podemos medir qué tan buena es la estimación con alguna de las medidas siguientes:

- divergencia Kullback-Leibler: $\rho_{KL}(\widehat{\Sigma^{-1}}, (\Sigma^{-1})_*) = \operatorname{tr}(\Sigma_*\widehat{\Sigma^{-1}}) - \log|\Sigma_*\widehat{\Sigma^{-1}}| - p$,
- Norma Frobenius: $\|\widehat{\Sigma^{-1}} - (\Sigma^{-1})_*\|_F = \sum_{j,k}(\widehat{(\Sigma^{-1})}_{jk} - \big((\Sigma^{-1})_*\big)_{jk})^2$,
- norma-$l_q$.

## 2.4. Método de estimación *Nodewise Regression*

Ante el problema de estimar el modelo gráfico correspondiente a una normal multivariada tenemos otro método basado en realizar sucesivas regresiones lineales. Este método llamado *Nodewise Regression* es presentado en Meinshausen y Bühlmann (2006) y se basa en el siguiente resultado demostrado en el **Apéndice A**: cada coeficiente $\boldsymbol{\beta}_k^{(j)}$ correspondiente a la regresión lineal de $X_j$ con las variables explicativas $\{X_k\}_{k\in V, k\neq j}$ tiene una correspondencia a menos de una constante estrictamente positiva con el valor $\Theta_{jk}$ de la matriz de precisión de $\vec{X}$.

Sea $j$ tal que $1 \leq j \leq p$ entonces

$$X_j = -\sum_{k\in V\setminus\{j\}} \frac{\Theta_{jk}}{\Theta_{jj}} X_k + \varepsilon_j, \tag{2.20}$$

donde $\mathrm{E}[\varepsilon_j] = 0$ y $\varepsilon_j$ es independiente de $\{X^{(r)}; r \in V \setminus \{j\}\}$.

Luego el vecindario de un nodo del modelo gráfico en el caso normal multivariado corresponde a las siguientes definiciones:

$$\mathcal{N}(j) = \{k \in V \setminus \{j\} : \Theta_{jk} \neq 0\} = \{k \in V \setminus \{j\} : \boldsymbol{\beta}_k^{(j)} \neq 0 \text{ y } \boldsymbol{\beta}_j^{(k)} \neq 0\}.$$

En el método anterior estimábamos la matriz de precisión $\Theta$ y luego estimamos el vecindario con $\widehat{\Theta}$ de la forma

$$\widehat{\mathcal{N}(j)} = \{k \in V \setminus \{j\} : \widehat{\Theta_{jk}} \neq 0\},$$

en este método hacemos una regresión lineal con el método *lasso* de cada una de las $p$ variables del modelo en función de las restantes $p-1$ la cual resulta razonable en un contexto de altas dimensiones y con una cantidad limitada de coeficientes $\boldsymbol{\beta}_j$ distintos de 0. Estimamos el vecindario de cada nodo de la siguiente forma

$$\widehat{\mathcal{N}(j)} = \{k \in V \setminus \{j\} : \widehat{\boldsymbol{\beta}_k^{(j)}} \neq 0\}.$$

El detalle en esta estimación es que en un grafo sucede que si $k \in \mathcal{N}(j)$ entonces $j \in \mathcal{N}(k)$ mientras que los conjuntos estimados con este método $\widehat{\mathcal{N}(j)}$ y $\widehat{\mathcal{N}(k)}$ no cumplen necesariamente esta propiedad. Para solucionar esto surgen la regla "**AND**" y la "**OR**". Dada la regresión en la que tenemos a la variable $X_j$ como variable respuesta y a las demás variables del modelo como variables explicativas con $\boldsymbol{\beta}_k^{(j)}$ el coeficiente correspondiente a la variable explicativa $X_k$ la regla "**OR**" consiste en poner una arista si alguno de los dos coeficientes $\widehat{\boldsymbol{\beta}_k^{(j)}}$ o $\widehat{\boldsymbol{\beta}_j^{(k)}}$ es distinto de 0. Por otro lado la regla "**AND**", más conservadora, consiste en poner una arista si ambos coeficientes son distintos de 0. Notemos que estas reglas no son necesarias en el caso de la estimación del grafo con *Graphical Lasso* ya que se realiza una optimización sobre el espacio de las matrices simétricas. Resaltamos que una gran ventaja de este método es la capacidad de paralelización de su cómputo. Presentamos a continuación el algoritmo que lo resume:

**Algoritmo 1:** Regresión basada en el vecindario para distribuciones gaussianas

1 **para cada** $i \in \{1, ..., p\}$ **hacer**
2 Estimar los coeficientes $\boldsymbol{\beta}$ de la regresión lineal de $X_i$ en función de $X_{(V\setminus\{i\})}$ con el método *lasso* ;
3 Considerar la estimación $\widehat{N(i)} = soporte(\boldsymbol{\beta})$ del vecindario de $i$ ;
4 **fin**
5 Combinar las estimaciones $\{\widehat{N(s)}, s \in V\}$ con la regla "**AND**" o "**OR**" para obtener el grafo estimado $\widehat{\mathcal{G}}$;

Las condiciones que plantean Meinshausen y Bühlmann para la consistencia de este método son las siguientes:

(1) la cantidad de variables $p$ puede crecer como potencia de $n$,

(2) la matriz de covarianza debe cumplir ciertas condiciones de regularidad,

(3) condiciones de dispersión (cantidad de ceros) de la matriz de precisión,

(4) los valores distintos de cero de la matriz de precisión deben tener un valor absoluto acotado por abajo y

(5) **condición de estabilidad de los vecindarios**: debe existir un valor pequeño de $\lambda$ distinto de cero tal que para un nodo $i$ la estimación $\widehat{\mathcal{N}(i)}$ (sin penalización $l_1$) coincida con la estimación $\widehat{\mathcal{N}(i)(\lambda)}$ (con penalización $l_1$).

Por último notamos que como lo que importa en cada regresión es la selección de variables, podemos remplazar al método *lasso* por alguna de sus variaciones.

## 2.5. Estimación de la matriz de covarianza a partir del grafo

Suponiendo que conocemos el grafo no-dirigido $\mathcal{G}$ subyacente en una distribución normal multivariada, podemos estimar la matriz $\Sigma$ de covarianza haciendo una estimación de máxima verosimilitud con restricciones. Vamos a maximizar el logaritmo de la función de verosimilitud sobre todas las matrices simétricas y definidas positivas que tengan un cero en todos los lugares $(\Sigma^{-1})_{ij}$ con $i, j$ vértices tales que no estén unidos en $\mathcal{G}$. Nos queda este problema

$$\widehat{\Sigma^{-1}} = \underset{\Sigma^{-1}\succ 0; C(\Sigma^{-1}\leftrightarrow\mathcal{G})}{\operatorname{argmax}} \log(\det(\Sigma^{-1})) - \operatorname{tr}(\mathbf{S}\Sigma^{-1}), \tag{2.21}$$

donde notamos a esta restricción de ajuste al grafo $C(\Sigma^{-1}, \mathcal{G})$. Luego, invirtiendo $\widehat{\Sigma^{-1}}$ obtendremos el estimador para $\Sigma$.

En caso de no tener la estructura del grafo $\mathcal{G}$ podemos usar un método mixto en el que primero estimamos el grafo con algún método de los vistos en las secciones anteriores (*GLasso* o *Nodewise regression*) y con esta estimación del grafo $\widehat{\mathcal{G}}$ realizamos la maximización de (2.21). A la combinación de estos métodos los llamamos métodos híbridos. La ventaja de usar *NodewiseRegression-MV*, en comparación con *GLasso-MV*, es que el método *Nodewise Regression* es consistente para la estimación de un grafo pidiendo restricciones más relajadas que *GLasso*. En Zhou et al. (2011) se introduce un método híbrido llamado ***Gelato*** (**G**raph **e**stimation with **la**sso and **t**hresh**o**lding)) que es del tipo *NodewiseRegression-MV* pero realiza regresiones con *lasso* truncado. En general este método mejora la performance de *GLasso-MV* para estimaciones de la matriz de covarianza en el caso en que tenemos matrices ralas y con los valores distintos de cero con alto valor absoluto.

Existen otros métodos propuestos para estimar la matriz de covarianza directamente sin recurrir a estimar el grafo primero algunos ejemplos son el método *CLIME* y el que usa una penalización del tipo *SCAD*.

## 2.6. Condiciones para la consistencia en la estimación del grafo

Por un lado, en relación a las condiciones bajo las cuales el método *Nodewise Regression* es consistente en la estimación del grafo subyacente en el caso de modelos gaussianos no dirigidos, en Meinshausen y Bühlmann (2006) se especifican varias restricciones (ver Sección 2.4) entre las que se encuentra la condición de estabilidad de los vecindarios. En Zhao y Yu (2006) se presenta la condición de irrepresentabilidad (ver Sub-sección 2.2.4) para la consistencia en la selección de variables del método *lasso*, además en este trabajo se realiza un análisis sobre la influencia de esta condición en la consistencia y otro sobre la fuerza de la condición. Según el libro Bühlmann y Van de Geer (2011) estas condiciones introducidas son equivalentes y por lo tanto, al estudiar la consistencia del método basado en los vecindarios, podremos considerar la segunda condición que resulta más intuitiva y fácil de calcular.

Pasando al método *GLasso*, en Meinshausen (2008) se presenta un ejemplo sencillo de un grafo para el cual su estimación con el método *GLasso* no es consistente. En Ravikumar, Raskutti et al. (2008) se presentan condiciones para la consistencia del método en la estimación del grafo subyacente en el caso de normales multivariadas. Las dos condiciones son:

(1) **condición de incoherencia**:

Sea $\Gamma_*$ el Hessiano con respecto a $\Theta$ de la función a maximizar en la ecuación (2.15) evaluada en la verdadera matriz de precisión $\Theta_*$ del modelo. Puede demostrarse que $\Gamma_* = \Theta_* \otimes \Theta_*$, con $\otimes$ el producto Kronecker de matrices. Sean $S = \{(i,j)|(\Theta_{ij})_* \neq 0\}$ y $S^c = \{(i,j)|(\Theta_*)_{ij} = 0\}$ entonces la condición es

$$\|\Gamma_{*,S^cS}(\Gamma_{*,SS})^{-1}\|_\infty \leq 1. \tag{2.22}$$

(2) deben existir contantes $K_1$ y $K_2$ que controlen las normas de la covarianza y del Hessiano: $\|\Sigma_*\|_\infty < K_1$ y $\|\Gamma_*\|_\infty < K_2$.

Además, en el trabajo de Ravikumar, se estudia bajo qué condiciones el ejemplo base de Meinhausen resulta consistente en la estimación con los métodos *Nodewise Regression* y *Glasso*.

## 2.7. *Stability Selection*

Ya sabemos que la estimación de una estructura discreta como variables seleccionadas en regresión o la estimación de grafos en modelos gráficos resulta una tarea complicada, sobretodo en modelos de altas dimensiones. En las secciones anteriores se indican algunas condiciones para lograr la consistencia de los distintos métodos propuestos. El método *Stability Selection* presentado en Meinshausen y Bühlmann (2010) consiste en utilizar el submuestreo o el método *bootstrap* para obtener estimaciones más estables. Además, este método fija una tolerancia máxima de falsos positivos esperados.

La diferencia entre el método *subsampling* y el *bootstrap* consiste en que en general el primero tiene un tamaño de sub-muestra menor que $n$ (en nuestro caso $\lfloor n/2 \rfloor$) y las muestras se obtienen sin reposición. El método *bootstrap* toma muestras de tamaño $n$ pero con reposición.

### 2.7.1. En regresión lineal

El problema consiste en estimar $S_*$ el conjunto de variables explicativas cuyo coeficiente (real) $(\boldsymbol{\beta}_j)_*$ es distinto de cero. Para esto se toman $I_1, ..., I_B$ sub-muestras de los datos sin reposición y de tamaño $\lfloor n/2 \rfloor$. Para cada sub-muestra $I_l$ se estima con *lasso* la variable respuesta y se considera el conjunto:

$$\widehat{S}_l(\lambda) = \{j; (\widehat{\boldsymbol{\beta}_j})_l \neq 0\}.$$

Luego, simplemente calculamos $\widehat{\Pi}_j(\lambda)$ como la frecuencia relativa de $j$ en los conjuntos $\widehat{S}_l(\lambda)$. Sea $\pi_{thres}$ el límite de corte elegido, el método seleccionará todas las variables $j$ que cumplan con la condición de que la frecuencia relativa $\widehat{\Pi}_j(\lambda)$ sea mayor o igual que este límite. Luego, tendremos

$$\widehat{S}^{\text{stable}} = \{j; \widehat{\Pi}_j > \pi_{\text{thres}}\}. \tag{2.23}$$

Seleccionamos entonces las variables con alta probabilidad de selección con *lasso*, con la dificultad agregada de seleccionar un valor adecuado de corte para $\pi_{\text{thres}}$.

Existe además un teorema para controlar, bajo ciertas condiciones, el valor esperado de la cantidad de falsos positivos en el proceso de selección.

### 2.7.2. En modelos gráficos

El método *Stability Selection* para seleccionar modelos gráficos es análogo al método para regresión. Para cada submuestra $I_l$ se estima con *GLasso* la matriz de precisión $\Sigma^{-1}$. Conservamos los $q$ valores que tengan mayor valor absoluto del triangulo superior de $\widehat{\Sigma^{-1}}_l$ (por la simetría de la matriz de precisión). Sintetizamos el método en el siguiente algoritmo.

**Algoritmo 2:** Método *Stability Selection* para la selección de modelos gráficos

1 Seleccionar el conjunto de parámetros de regularización $\Lambda$, el valor límite $\pi_{\text{thres}}$ y el número de sub-muestras $B$ ;
2 **para cada** $\lambda \in \Lambda$ **hacer**
3  **para cada** $l \in \{1, ..., B\}$ **hacer**
4  Generar una sub-muestra $I_l$ tomada sin reposición del conjunto original de datos y de tamaño $\lfloor n/2 \rfloor$ ;
5  Estimar el conjunto de aristas $\widehat{S}_l(\lambda)$ con el método *GLasso* con parámetro de penalización $\lambda$ ;
6  **fin**
7  **para cada** $i \in \{1, ..., p\}$ **hacer**
8  Calcular $\widehat{\Pi}_i(\lambda) = \frac{1}{B} \sum_{l=1}^{B} I(i \in \widehat{S}_l(\lambda))$;
9  **fin**
10 **fin**
11 Finalmente estimar $\widehat{S}^{\text{stable}} = \left\{ j; \max_{\lambda \in \Lambda} \widehat{\Pi}_j(\lambda) > \pi_{\text{thres}} \right\}$;

## 2.8. Simulaciones y análisis de datos

### 2.8.1. Problema de regresión: *lasso*, *ridge*, *adaptive lasso* y *thresholded lasso*

Realizamos una simulación de regresión en altas dimensiones ($n \ll p$) en la cual tenemos $p = 1000$ covariables normales estándar independientes y $n = 50$ realizaciones de cada una. Creamos una variable respuesta que es una función lineal de las variables 1, 3 y 400 con un ruido normal con $\sigma = 0.5$ ($S_* = \{1, 3, 400\}$, $(\boldsymbol{\beta}_1)_* = 2$, $(\boldsymbol{\beta}_3)_* = 0.7$ y $(\boldsymbol{\beta}_{400})_* = 1$). El objetivo es comparar el desempeño de los métodos *lasso*, *ridge*, *adaptive lasso* y *thresholded lasso* en el problema de predicción y de selección de variables. El análisis fue realizado con la biblioteca `glmnet` (J. Friedman et al. (2010)) en **R** en la que está implementado el método *elastic net* (combinación convexa entre *lasso* y *ridge*). Para los métodos *lasso*, *adaptive lasso* y *ridge* elegimos el valor de penalización $\lambda$ con el método de validación cruzada usando el error cuadrático medio (MSE) como función de pérdida. Para el método *thresholded lasso* primero elegimos el valor de $\lambda$ con validación cruzada y luego para ese valor fijo elegimos el valor de $\tau$ con validación cruzada también. Realizamos este proceso $nrep = 100$ veces y en cada una de estas corridas calculamos para la estimación de un conjunto de test el error cuadrático medio, la cantidad de variables seleccionadas y si estas variables incluyen a las verdaderas 1, 3 y 400. Calculamos los promedios de estas medidas y los volcamos en el Cuadro 2.1. Para la primera corrida también graficamos los coeficientes estimados con cada método en las Figuras 2.5, 2.6, 2.7 y 2.8.

Observando el Cuadro 2.1 vemos que el menor error cuadrático medio se obtiene con el método *lasso* adaptable seguido por *lasso* truncado y que el método con peores resultados es el *ridge*. Vemos

además que el método *ridge* es el peor en cuanto a interpretabilidad del modelo ajustado, mientras que el método adaptable es el que mejor selecciona variables en cuanto a falsos positivos. En cuanto a falsos negativos es el que más falla con el 5 % de los casos aunque la diferencia no es grande.

| | **MSE** | **Variables seleccionadas** | **Incluye a $S_*$** |
|---|---|---|---|
| ***lasso*** | 0.55 | 11.78 | 0.99 |
| ***ridge*** | 5.68 | 1000.00 | 1.00 |
| ***adaptive lasso*** | 0.37 | 3.01 | 0.95 |
| ***thresholded lasso*** | 0.42 | 10.29 | 0.99 |

Cuadro 2.1: Comparación de los métodos *lasso*, *ridge*, *adaptive lasso* y *thresholded lasso* para regresión lineal en altas dimensiones

En las imágenes vemos que los cuatro métodos estiman con $\boldsymbol{\beta}_j$ distinto de 0 a los coeficientes de las variables utilizadas para generar la variable respuesta (1, 3 y 400). Comparando la Figura 2.5.(a) con los gráficos de la Figura 2.6 vemos el efecto selector del método *lasso* en contraposición al método *ridge*. Por un lado *lasso* selecciona variables y por el otro *ridge* no anula ninguno de los coeficientes estimados, quedando todos distintos que cero pero muy pequeños. Al comparar los métodos de las Figuras 2.5 y 2.7, observamos que tanto el método *lasso* adaptable como el *lasso* truncado anulan algunos coeficientes que el método original no había anulado, manteniendo seleccionadas las variables que realmente son las relevantes.

Este ejemplo no implica que siempre el método *lasso* sea más adecuado que *ridge* para seleccionar variables, se pueden crear ejemplos en los que sea mejor usar *ridge* que *lasso*. Por ejemplo, sabiendo que el método *lasso* selecciona como máximo $n$ (el tamaño de la muestra) de las $p$ variables, podemos generar un ejemplo con $m > n$ variables relevantes y en este caso lo más probable es que sea más conveniente usar el método *ridge* para no perder información. Al querer hacer una regresión lineal en datos reales no sabemos a priori si el número de variables relevantes es mayor o menor que $n$, por lo tanto una técnica como validación cruzada nos será útil para determinar cual de los modelos utilizar para un conjunto de datos en particular.

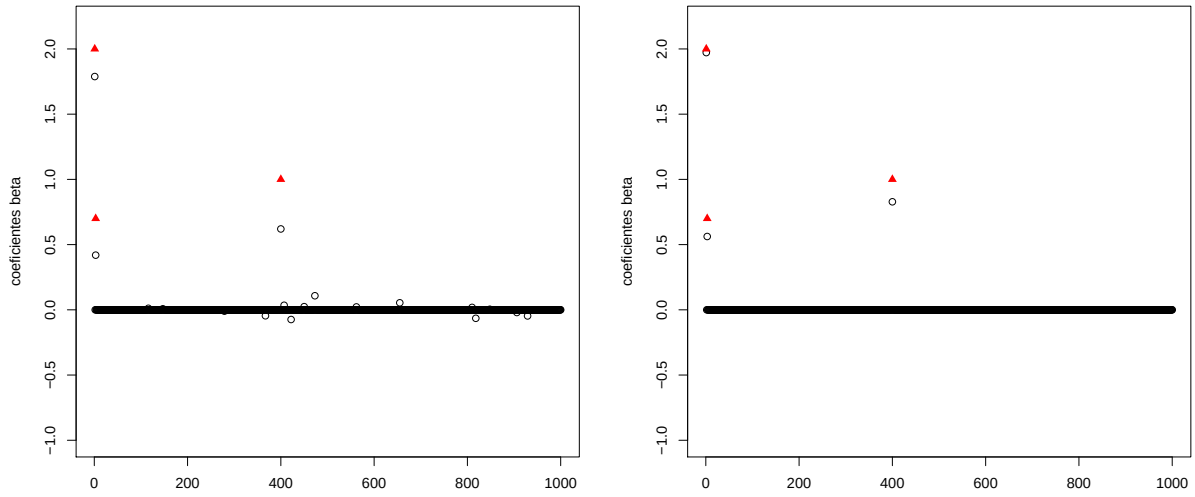
Por último en la Figura 2.8 podemos apreciar la diferencia entre los caminos de coeficientes dibujados en función del logaritmo de $\lambda$ para los métodos *lasso* y *ridge*. Mientras que para cualquier valor de $\lambda$ el método *ridge* estima todos los coeficientes como distintos de cero como puede verse en la escala superior del gráfico, vemos que a medida que $\lambda$ aumenta cada vez más coeficientes son estimados como cero en el método *lasso*.

### 2.8.2. Consistencia de *GLasso* en la estimación de $\Sigma$

Realizamos una simulación para estudiar la consistencia del método *GLasso* en la estimación de la matriz de presición para tres valores distintos de $p$ fijos. Tenemos en cuenta que en el problema de estimación *GLasso* estima aproximadamente $\frac{p \times p}{2}$ parámetros por la simetría de la matriz de precisión y por lo tanto tomaremos valores de $n$ comparables con este valor para quedarnos enmarcados en el contexto de altas dimensiones.

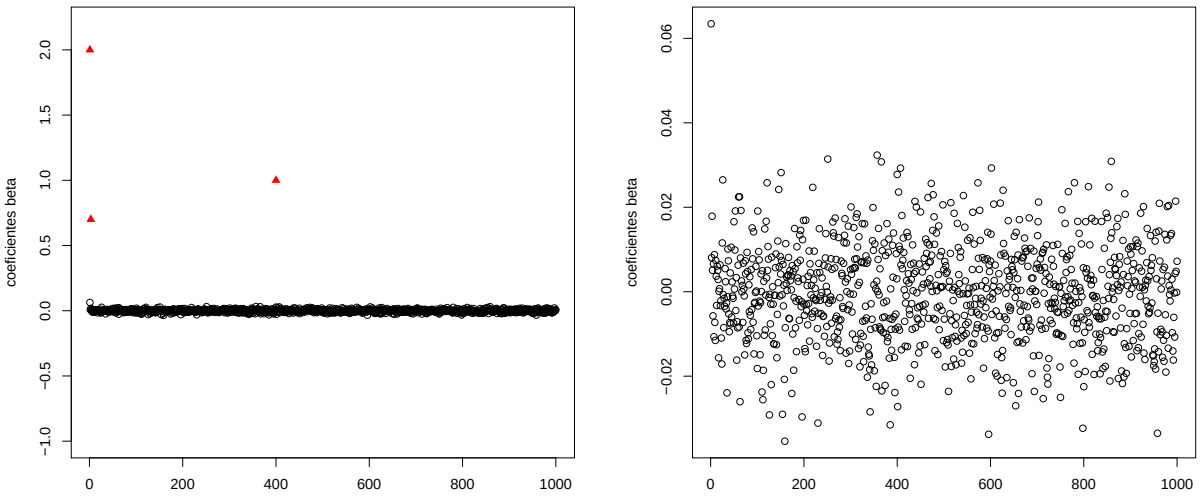
Para cada valor $p \in \{64, 120, 350\}$ generamos $n$ realizaciones de una normal multivariada con la matriz de precisión de la forma

$$(\Theta_*)_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0.2 & \text{si } |i-j|{=}1 \\ 0 & \text{si no} \end{cases} \tag{2.24}$$




(a) (b)

Figura 2.5: (a) Coeficientes $\boldsymbol{\beta}$ estimados por método *lasso*. Nro de variables seleccionadas= 19. (b) Coeficientes $\boldsymbol{\beta}$ estimados por método adaptive *lasso*. Nro de variables seleccionadas= 3. En ambos gráficos se representan en negro estimaciones y en rojo coeficientes reales distintos de cero.




(a) (b)

Figura 2.6: (a) Coeficientes $\boldsymbol{\beta}_j$ estimados por método *ridge*. Se representan en negro estimaciones y en rojo coeficientes reales distintos de cero. Nro de variables seleccionadas= 1000. (b) Ampliación de (a).

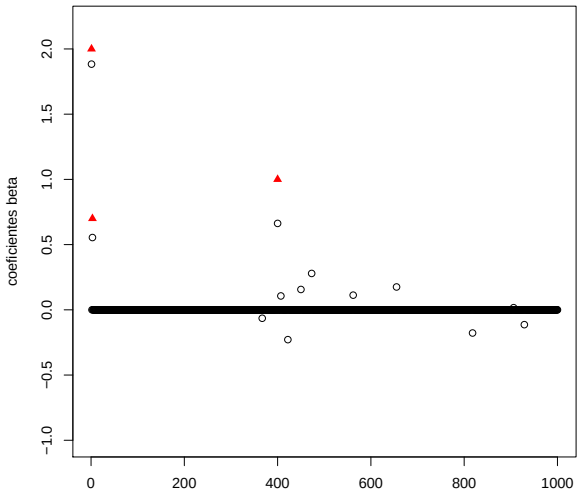
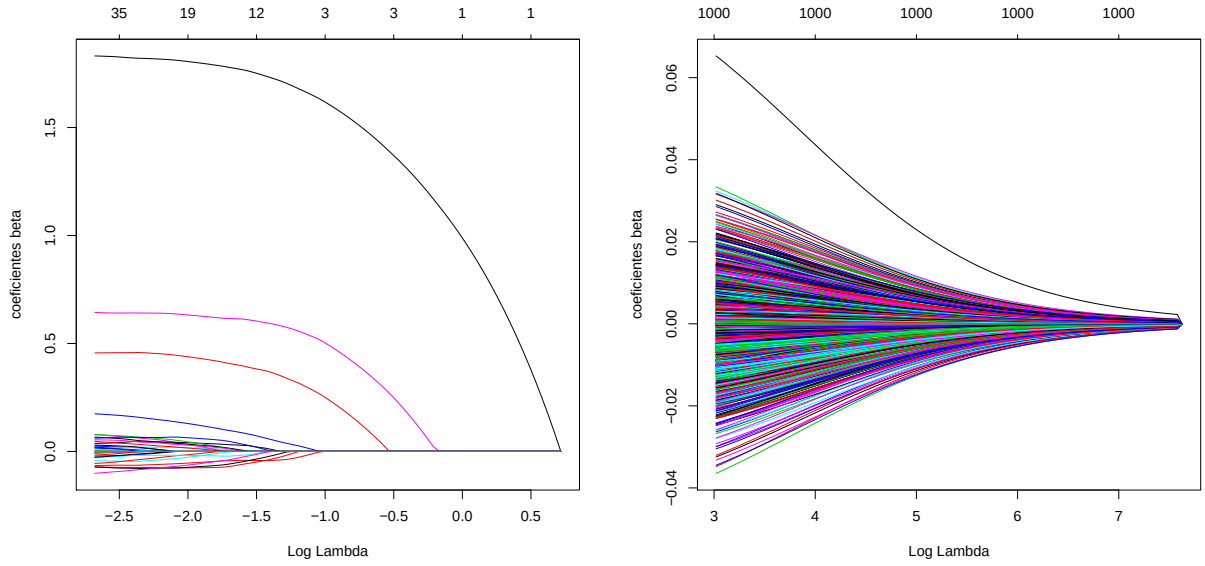



Figura 2.7: Coeficientes $\boldsymbol{\beta}_j$ estimados por método *thresholded lasso*. Se representan en negro estimaciones y en rojo coeficientes reales distintos de cero. Nro de variables seleccionadas= 13.




(a) (b)

Figura 2.8: (a) Caminos de los coeficientes $\boldsymbol{\beta}_j$ estimados por método *lasso* en función de los distintos valores de $\lambda$. (b) Caminos de los coeficientes $\boldsymbol{\beta}_j$ estimados por método *ridge* en función de los distintos valores de $\lambda$. En (a) y (b) se indican en el eje horizontal inferior los distintos valores del logaritmo de $\lambda$ mientras que en el eje superior se indica cantidad de variables explicativas con coeficiente beta distinto de cero para cada valor de $\lambda$ representado.

, tomamos $\lambda = 2\sqrt{\log(p)/n}$ como se sugiere en el Teorema 1 del trabajo Ravikumar, Wainwright et al. (2011) para estimar la matriz de precisión $\Theta_*$ y medimos el error con la norma matricial 2. Repetimos el procedimiento $nrep = 50$ veces y tomamos el error promedio. Los resultados obtenidos son los de la Figura 2.9 en la que observamos que el valor del error disminuye tendiendo a 0 a medida que el tamaño $n$ de la muestra aumenta para todos los valores de $p$ escogidos. Debido a que cuanto mayor es $p$ la cantidad de parámetros a estimar aumenta cuadráticamente vemos una clara diferencia entre los errores para $p \in \{64, 120\}$ y para $p = 350$.

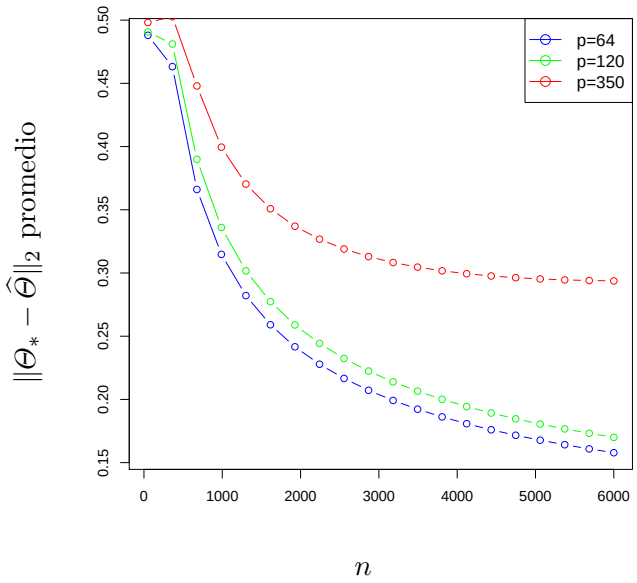
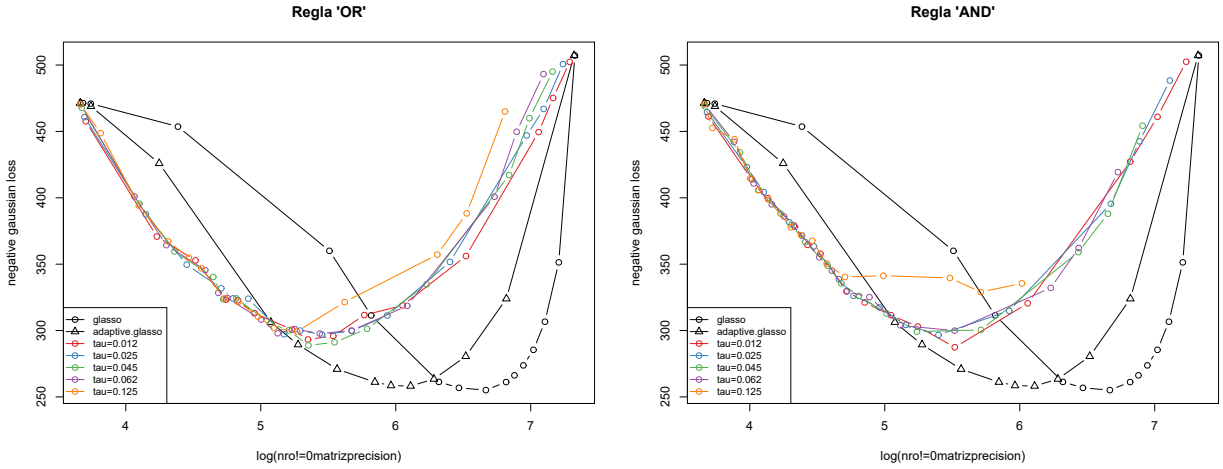



Figura 2.9: Error promedio medido con la norma matricial 2 de la estimación de la matriz de precisión $\Theta_*$ en función del tamaño de la muestra $n$ para tres valores distintos de $p$.

### 2.8.3. Estimación de la matriz de precisión de `arabidopsis thaliana`

Analizamos la base de datos con $p = 39$ variables y tamaño de muestra $n = 118$ extraída del vínculo `gene expression - arabidopsis thaliana` (2004) que contiene la expresión de 39 genes en la síntesis de isoprenoides en el caso de la especie *arabidopsis thaliana*. Con el objetivo de conocer sobre las relaciones entre los genes estudiamos el modelo gráfico no-dirigido estimando el grafo por el método *GLasso*, por el método *Adaptive GLasso* y por otro lado la estimación de la matriz de covarianza por máxima verosimilitud a partir del grafo estimado con *Nodewise Regression* con *thresholded lasso* (***Gelato***). Para las estimaciones usamos los paquetes de **R** `glmnet` (J. Friedman et al. (2010)) y `glasso` (J. Friedman et al. (2014)).

La Figura 2.10 es una réplica de la Figura 13.3 del libro Bühlmann y Van de Geer (2011) con el agregado de un método adicional. La Figura representa los gráficos de la función de pérdida negativa del logaritmo de la verosimilitud en función de la medida de regularización (representada por el número de valores de la matriz de precisión estimada distintos de cero) para los distintos métodos utilizados. Mirando la figura vemos que el método *GLasso* es el que toma el menor valor de la función de pérdida igualado prácticamente por el método *Adaptive GLasso*. Al hacer un análisis en función de la dispersión de la matriz $\Theta$ vemos que en el caso de matrices menos ralas, con el método *GLasso* se alcanza el menor valor con una diferencia significativa con respecto al método




(a) (b)

Figura 2.10: Graficos de la función de pérdida en función del logaritmo de la cantidad de valores de la matriz estimada $\widehat{\Theta}$ del conjunto de datos `riboflavin` para los métodos *GLasso*, *Adaptive GLasso* y el método híbrido ***Gelato*** con distintos valores de $\tau$. El gráfico (a) corresponde al uso de la regla "**OR**" para la estimación del grafo $\mathcal{G}$ con *Nodewise Regression* y el (b) al uso de la regla "**AND**" .

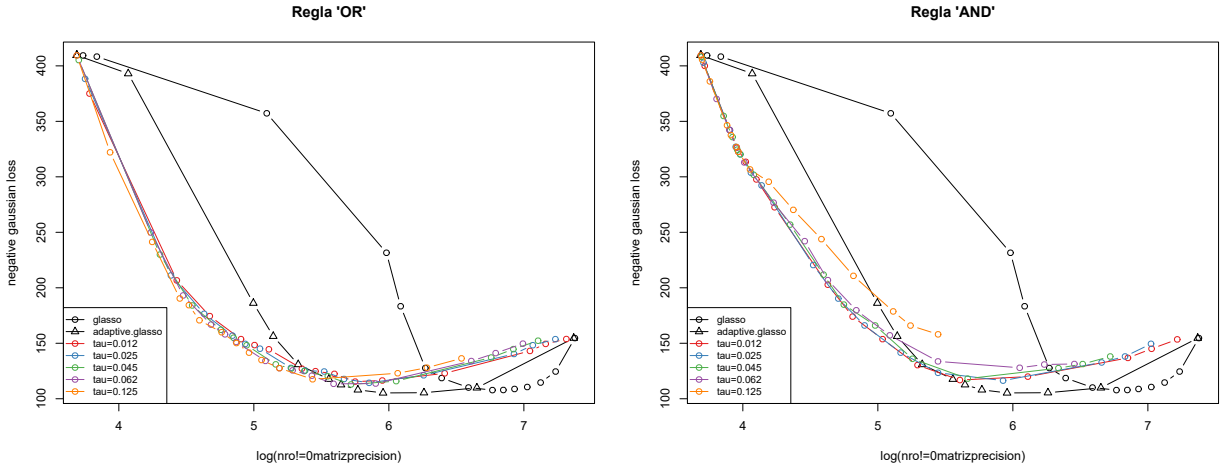
híbrido para los distintos valores del parámetro $\tau$. Por otro lado, para estimaciones más ralas el método ***Gelato*** parece ser un mejor estimador. El método *Adaptive GLasso* es el que mejor se comporta para las estimaciones con grados de libertad intermedios. La diferencia entre el gráfico de la izquierda y el de la derecha consiste en la regla usada para la estimación del grafo con el método *Nodewise Regression*: el de la izquierda usa la regla "**AND**" y el de la derecha usa la regla "**OR**". En este aspecto ambos gráficos resultan ser muy similares.

Como no sabemos si la distribución de esta muestra es la de una normal multivariada ni si se aproxima, decidimos hacer una simulación con una muestra con esta distribución. Para generar la matriz de precisión $\Theta_*$ generamos al azar una matriz de $p \times p$ simétrica definida positiva, a la que luego agregamos ceros para tener una dispersión tal que el logaritmo del numero de lugares distinto de cero fuera aproximadamente 6 (en nuestro caso 6.2). Con esta matriz generamos una muestra normal multivariada de tamaño $n = 180$ y media $\mathbf{0}$.

Los resultados se muestran en la Figura 2.11. De nuevo, *GLasso* y *Adaptive GLasso* son los métodos que minimizan la función de pérdida. Pero por un lado el mínimo alcanzado por *Adaptive GLasso* es con 6.1 cantidad de lugares distintos de cero en la matriz estimada que se acerca mucho más al valor real que los 6.7 de *GLasso*. El método *Gelato* se comporta de manera similar al caso analizado anteriormente, sin grandes variaciones entre las dos distintas reglas.

### 2.8.4. Selección del modelo para `riboflavin`: *Glasso* y *Stability Selection*

Dado el conjunto de datos `riboflavin` ubicado en el paquete `hdi` (Dezeure et al. (2015)) de **R** con $p = 160$ y $n = 71$ que contiene las expresiones de genes en la síntesis de la proteína riboflavin estimamos el grafo no dirigido subyacente con los métodos *GLasso* y *Stability Selection* replicando los resultados del trabajo Meinshausen y Bühlmann (2010) correspondientes a la Figura 13.2 de Bühlmann y Van de Geer (2011).




(a) (b)

Figura 2.11: Graficos de la función de pérdida en función del logaritmo de la cantidad de valores de la matriz estimada $\widehat{\Theta}$ de la muestra normal generada sintéticamente para los métodos *GLasso*, *Adaptive GLasso* y el método híbrido ***Gelato*** con distintos valores de $\tau$. El gráfico (a) corresponde al uso de la regla "**OR**" para la estimación del grafo $\mathcal{G}$ con *Nodewise Regression* y el (b) al uso de la regla "**AND**" .

Además de estimar el grafo para este conjunto de datos generamos a partir de este mismo dataset un nuevo conjunto de datos permutando al azar todas las filas. Así generamos un conjunto de datos con todas las variables independientes y por lo tanto con grafo subyacente vacío.

En el primer caso vemos en la Figura 2.12 que mientras que las estimaciones con *GLasso* varían bastante en cantidad de grafos entre el menor y el mayor de los valores de $\lambda$, las estimaciones de *Stability Selection* son mucho más similares en cuanto a número de aristas y aristas seleccionadas más allá del parámetro $\lambda$ del método base.

En el caso del dataset permutado vemos en la Figura 2.13 que mientras *GLasso* sobreestima muchísimo la cantidad de aristas del grafo con variaciones importantes entre los distintos valores de $\lambda$, las estimaciones de *Stability Selection* develan que todos las aristas estimadas por *GLasso* son muy inestables quedando en todas las estimaciones grafos vacíos.

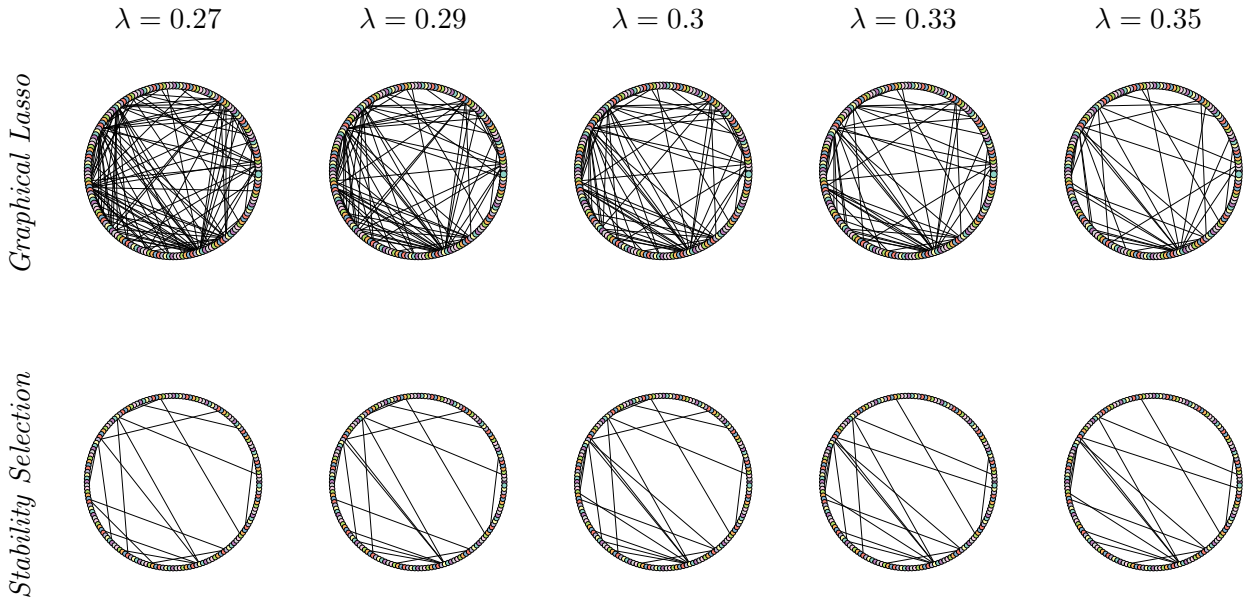



Figura 2.12: Comparación de la estimación del grafo para el conjunto de datos `riboflavin` ***GLasso*** vs ***Stability Selection*** con distintos valores del parámetro $\lambda$

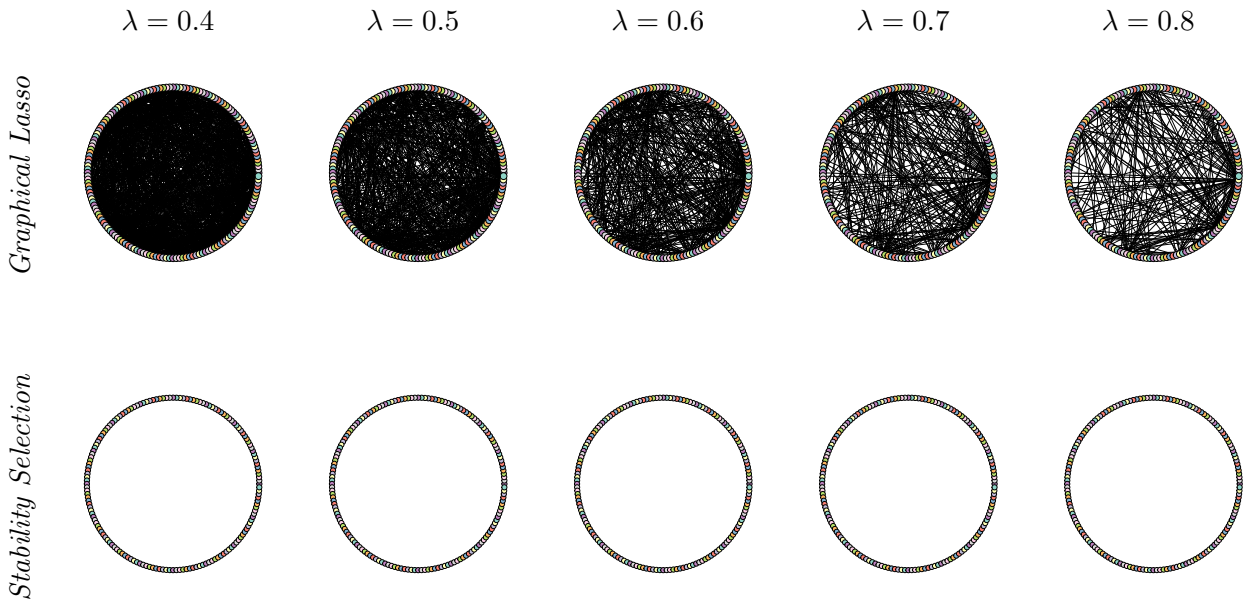



Figura 2.13: Comparación de la estimación del grafo para el conjunto de datos `riboflavin` permutado para que las variables queden independientes con los métodos ***GLasso*** vs ***Stability Selection*** con distintos valores del parámetro $\lambda$

# Capítulo 3

# Modelos discretos

Aparte de los modelos gaussianos, otra clase de modelos gráficos muy estudiada es la constituida por sistemas donde todas las variables son discretas, en particular los más populares son los que tienen todas las variables binarias. En estadística clásica los modelos discretos son estudiados con el nombre de modelos de Poisson log-lineales (Lauritzen (1996), Wasserman (2004)) mientras que en la literatura de aprendizaje automático los modelos gráficos no-dirigidos binarios que permiten nodos ocultos se llaman "*Boltzmann machines*".

En el caso de modelos discretos, la computabilidad pasa a ser una limitación importante a la hora de estimar un modelo gráfico ya que la evaluación de la función de verosimilitud es muy demandante computacionalmente. En este capítulo presentamos distintos métodos para la estimación del grafo subyacente para estas distribuciones en el contexto de altas dimensiones. Los métodos presentados son *Nodewise logistic regression* (Wainwright et al. (2006)) para el caso particular de variables binarias y para variables discretas en general presentamos brevemente el método *Chow-Liu* (Chow y C. Liu (2006)) para árboles y un método basado en una pseudo-verosimilitud (Frondana (2016)) no eficientemente computable en altas dimensiones pero no-paramétrico y con condiciones para la consistencia menos restrictivas que las de los anteriores métodos.

Existen en la literatura otros métodos con enfoques distintos a los que desarrollaremos, mencionamos algunos de ellos a continuación. En Jalali et al. (2011) se presenta un método greedy eficientemente computable que mejora las cotas de error de *Nodewise logistic Regression* y relaja las condiciones. En Loh y Wainwright (2013) se realiza un estudio sobre la relación entre el soporte de la inversa de una generalización de la matriz de covarianza y la estructura de grafos discretos, inspirándose en la propiedad de los modelos gaussianos. Este trabajo encuentra una clase de grafos (que incluye a los grafos cadena) para los cuales la estimación del grafo con el método *Glasso* es consistente. En Wasserman et al. (2014) se aborda la estimación no paramétrica de este tipo de grafos con test múltiples, aunque la demostración de la consistencia del método se restringe a bajas dimensiones.

## 3.1. Propiedades del modelo Ising

Dentro de los modelos binarios el más utilizado es el Ising. Es un modelo que proviene de la física estadística que modela un sistema de $p$ partículas fijas con un spin positivo o negativo (1 o 0 en nuestro caso) que interactúan bajo el efecto de un campo magnético. Este modelo fue usado no sólo en el ámbito de la física, sino también por ejemplo en reconocimiento de imágenes (Blake et al. (2011)), en psicología (Van Borkulo et al. (2014)) y en el modelado de redes sociales (Banerjee et al. (2008)).

En el modelo Ising la función de probabilidad puntual es

$$p(\mathbf{x}, \Theta) = \exp\left[\sum_{j\in V} \Theta_j \mathbf{x}_j + \sum_{(j,k)\in E} \Theta_{jk}\mathbf{x}_j\mathbf{x}_k - \Phi(\Theta)\right] \text{ para } \mathbf{x} \in \chi, \tag{3.1}$$

con $\chi = \{0,1\}^p$ y $\Phi(\Theta)$, generalmente llamada función de partición, el factor de normalización que hace que la suma de todas las probabilidades resulte 1:

$$\Phi(\Theta) = \log \sum_{\mathbf{x}\in\chi} \exp\left(\sum_{j\in V} \Theta_j \mathbf{x}_j + \sum_{(j,k)\in E} \Theta_{jk}\mathbf{x}_j\mathbf{x}_k\right). \tag{3.2}$$

Dados $j$ y $k$ en $V$, el parámetro $\Theta_{jk}$ es la fuerza del vínculo entre $X_j$ y $X_k$ y $\Theta_j$ es el potencial para el nodo $j$. Si $\Theta_{jk}$ es positivo resulta más probable que ambas variables $X_j$ y $X_k$ valgan 1 ya que en este caso se aporta un término $\Theta_{jk}$ al exponente de la probabilidad, mientras que si es negativo esta disposición resulta menos probable que las otras. Por otro lado, si el parámetro $\Theta_j$ es positivo hay una afinidad de $X_j$ por ser 1 y si es negativo vale lo contrario.

Este modelo, que sólo tiene potenciales de a pares, puede extenderse agregando interacciones de mayor orden en la factorización, por ejemplo agregando términos en los que interactúen 3 variables $j, k, l$ con nuevos parámetros $\Theta_{jkl}$.

De (3.1) deducimos por el criterio de factorización (ver (1.3)) que para el modelo Ising vale que

$$X_j \perp X_k | X_{V\setminus\{j,k\}} \iff \Theta_{kj} \neq 0 \text{ y } \Theta_{jk} \neq 0. \tag{3.3}$$

Teniendo esto en cuenta, parece una opción razonable estimar a estos parámetros $\Theta_{jk}$ para todo $(j,k) \in V \times V$ como los que maximizan la verosimilitud y a partir de esto reconstruir el grafo con los parámetros que queden distintos de cero. El problema de este método es que es intratable en altas dimensiones ya que la función de partición es una suma de $|\chi| = 2^p$ términos. Por lo tanto, el desafío de este problema es encontrar métodos que preferentemente no usen esta función en la estimación, para conseguir algoritmos eficientes.

Por último notamos, también a partir de (3.1), que como para cualquier elemento $\mathbf{x} \in \chi$ la probabilidad es estrictamente positiva entonces por Hammersley-Clifford (ver Teorema 1.9) podemos estimar el grafo usando la propiedad de Markov de a pares y luego sacar conclusiones con la propiedad global.

## 3.2. Método basado en los vecindarios para modelos binarios

Al igual que en el caso gaussiano, el método basado en los vecindarios *Nodewise logistic regression* se basa en estimar paralelamente el vecindario para cada una de las variables a partir de regresiones. En este caso, las regresiones son logísticas por tratarse la variable respuesta de una variable binaria. Presentamos a continuación el método de regresión logística como método de clasificación en general y la regresión logística con penalización para el caso de alta dimensión.

### 3.2.1. Clasificación con regresión logística con penalización

El modelo de regresión lineal, presentado en la Sección 2.2, asume que la variable respuesta es cuantitativa. Pero existen muchas situaciones en las cuales la variable respuesta no es cuantitativa sino cualitativa, es decir, toma valores entre una cantidad finita de categorías. Por ejemplo, el color de ojos de una persona es una variable cualitativa: puede clasificarse en las categorías verde, azul o

marrón. El problema de predecir variables categóricas se llama **clasificación**. Un ejemplo sencillo de clasificación es el problema de clasificar a un mail en spam o no spam (variable respuesta) según sus características como palabras incluidas, destinatarios, emisor, archivos incluidos, etc. (variables explicativas). Existen numerosos métodos conocidos para clasificar entre los que se encuentran análisis de discriminación lineal (LDA), análisis de discriminación cuadrático (QDA), regresión logística, k-vecinos más cercanos, árboles de decisión , etc. En muchos casos, inspirados en la regla de Bayes, estos métodos usados para clasificar predicen la probabilidad de pertenecer a cada una de las categorías posibles y clasifican en base a cuál es la mayor probabilidad.

Uno de estos métodos de clasificación que resulta de una generalización de la regresión lineal es el método de **regresión logística**. En general este método se utiliza para variables respuesta dicotómicas (con dos categorías posibles). Continuamos con la notación anteriormente usada para regresión notando $\vec{X}$ al vector aleatorio de co-variables e $Y \in \{0,1\}$ a la variable respuesta. Como sabemos que el modelo de regresión lineal modela a la variable respuesta en el conjunto de los reales $\mathbb{R}$, a la hora de predecir una probabilidad $0 < p < 1$, haremos una transformación de los números reales al conjunto $(0,1)$. A esta transformación la llamaremos función logística y aplicada a la función lineal en función de $\vec{X}$ la interpretaremos como la probabilidad de que la variable respuesta tome el valor 1:

$$p(\vec{X}) = \frac{e^{\beta_0+\boldsymbol{\beta}_1 X_1+...+\boldsymbol{\beta}_p X_p}}{1+e^{\beta_0+\boldsymbol{\beta}_1 X_1+...+\boldsymbol{\beta}_p X_p}} = P(Y=1|\vec{X}), \tag{3.4}$$

de donde fácilmente podemos comprobar que la imagen se encuentra en el intervalo deseado. Una vez estimados los coeficientes $(\beta_0,\boldsymbol{\beta})$ podremos estimar la probabilidad de que $Y$ valga 1 evaluando la ecuación (3.4) en las estimaciones $(\widehat{\beta_0},\widehat{\boldsymbol{\beta}})$. Con esta probabilidad estimada, podremos predecir la categoría de una nueva observación $\mathbf{x}^{(*)}$ como 1 si $p(\mathbf{x}^{(*)}) > 0.5$ y como 0 en caso contrario. Con un poco de manipulación en (3.4), tenemos

$$\frac{p(\vec{X})}{1-p(\vec{X})} = e^{\beta_0+\boldsymbol{\beta}_1 X_1+...+\boldsymbol{\beta}_p X_p},$$

y por último tomando logaritmo a ambos lados obtenemos

$$\log\left(\frac{p(\vec{X})}{1-p(\vec{X})}\right) = \beta_0 + \boldsymbol{\beta}_1 X_1 + ... + \boldsymbol{\beta}_p X_p, \tag{3.5}$$

donde al primer miembro de la ecuación lo llamaremos $logit(p(\vec{X}))$.

Para estimar a $p(\vec{X})$ debemos calcular el vector de coeficientes $\boldsymbol{\beta}$. Sean $\{(\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^n$ las observaciones del conjunto de entrenamiento queremos que se cumpla que para los $y^{(i)} = 1$ la probabilidad estimada $\widehat{p(\mathbf{x}^{(i)})}$ sea cercana a 1 y para los $y^{(i)} = 0$ $\widehat{p(\mathbf{x}^{(i)})}$ sea cercana a 0. Con este objetivo usaremos la función de verosimilitud de una distribución binomial con $p(\mathbf{x})$ la probabilidad de éxito y la maximizaremos sobre $\boldsymbol{\beta}$:

$$L(\beta_0,\boldsymbol{\beta}) = \prod_{y^{(i)}=1} p(\mathbf{x}^{(i)}) \prod_{y^{(i)}=0} (1-p(\mathbf{x}^{(i)})) = \prod_{i=1}^{n} (p(\mathbf{x}^{(i)}))^{y^{(i)}} (1-p(\mathbf{x}^{(i)}))^{1-y^{(i)}}. \tag{3.6}$$

Tomando logaritmo nos queda finalmente la función a maximizar

$$l(\beta_0,\boldsymbol{\beta}) = \sum_{i=1}^{n} y^{(i)} \log(p(\mathbf{x}^{(i)})) + (1-y^{(i)}) \log(1-p(\mathbf{x}^{(i)})). \tag{3.7}$$

El método que describimos hasta ahora es el de regresión logística clásico. En el contexto de alta dimensión usaremos una penalización de la norma 1 del vector de coeficientes $\boldsymbol{\beta}$ (dejando de lado el coeficiente correspondiente a la ordenada al origen) en la función de verosimilitud (3.7):

$$l_\lambda(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} y^{(i)} \log(p(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - p(\mathbf{x}^{(i)})) - \lambda ||\boldsymbol{\beta}||_r, \tag{3.8}$$

con $\lambda > 0$ y $w$ un valor positivo. Suele tomarse $r = 1$ (penalización $l_1$) o $r = 2$ (penalización $l_2$).

**Ejemplo de clasificación**

Analizamos los datos `spls::prostate` en **R** [a] que corresponden a la expresión de 6033 genes obtenidas de 102 muestras de la próstata de pacientes operados. El objetivo es clasificar a estas muestras como tumores o no tumores siendo que en el conjunto de datos hay 52 muestras de prostatas con tumores y 50 muestras de próstatas normales. Para realizar la clasificación, los datos fueron divididos al azar en 2 subconjuntos: entrenamiento (75 %) y validación (25 %).

Los resultados obtenidos son:

| | **Porcentaje de acierto** | **Variables seleccionadas** |
|---|---|---|
| **penalización** $l_1$ | 84 % | 23 |
| **penalización** $l_2$ | 92 % | todas |

Cuadro 3.1: Resultados obtenidos en la clasificación con penalización $l_1$ y $l_2$

Analizando el porcentaje de acierto se ve que con la penalización $l_2$ obtenemos una mejor clasificación que con la penalización $l_1$. Por otro lado vemos que el modelo ajustado con la penalización $l_1$ es mucho más interpretable ya que selecciona sólo 23 de las 6033 variables del conjunto de datos.

[a] Este dataset es una modificación del usado originalmente en Prostate cancer data set (Singh et al. 2002)

### 3.2.2. Método *Nodewise logistic regression*

En Wainwright et al. (2006) se plantea el método basado en los vecindarios en el que, análogamente al caso gaussiano (ver Sección 2.4), se estima la estructura del grafo con regresiones logísticas. Para la selección del modelo nos basamos en la propiedad (3.3) del modelo Ising que nos permite estimar la estructura del modelo gráfico a partir de encontrar los $(j, k) \in V \times V$ cuyo coeficiente $\Theta_{jk}$ es distinto que cero. Si para $j \in V$ calculamos $logit(P[X_j = 1 | X_{(V \setminus j)}])$ (recordemos que $logit(p) = log(p/(1-p))$ para $0 < p < 1$) a partir de (3.1) obtenemos

$$logit(P[X_j = 1 | X_{V \setminus j}]) = \Theta_j + \sum_{k \neq j} \Theta_{jk} X_k. \tag{3.9}$$

Es decir que si estimamos los coeficientes $\boldsymbol{\beta}^{(j)} = (\beta_0^{(j)}, \boldsymbol{\beta}_1^{(j)}, \boldsymbol{\beta}_2^{(j)}, ..., \boldsymbol{\beta}_{j-1}^{(j)}, \boldsymbol{\beta}_{j+1}^{(j)}, ..., \boldsymbol{\beta}_p^{(j)})$ de una regresión logística con $X_j$ como variable respuesta y las variables restantes como variables explicativas estaremos estimando los parámetros $\Theta_{ij}$ de nuestro modelo y por lo tanto la estructura del grafo de nuestro modelo:

$$\boldsymbol{\beta}_j^{(k)} = \Theta_{kj} \text{ y entonces } \boldsymbol{\beta}_j^{(k)} \neq 0 \iff \Theta_{kj} \neq 0. \tag{3.10}$$

Al estar en el contexto de altas dimensiones agregamos una penalización $l_1$ a las regresiones logísticas y estimaremos el vecindario de cada nodo, al igual que en el caso gaussiano, de la siguiente forma:

$$\widehat{\mathcal{N}_\lambda(j)} = \{k \in V \setminus \{j\} : \widehat{\boldsymbol{\beta}_j^{(k)}(\lambda)} \neq 0\}.$$

Nuevamente el detalle en esta estimación es que en un grafo sucede que si $k \in \mathcal{N}(j)$ entonces $j \in \mathcal{N}(k)$ mientras que los conjuntos estimados con este método $\widehat{\mathcal{N}_\lambda(j)}$ y $\widehat{\mathcal{N}_\lambda(k)}$ no cumplen necesariamente esta propiedad. Para solucionar esto podemos usar al igual que en el caso gaussiano la regla "**OR**":

$$(j,k) \in \widehat{E}_{\text{“}\mathbf{OR}\text{”}} \iff k \in \widehat{\mathcal{N}_\lambda(j)} \text{ o } j \in \widehat{\mathcal{N}_\lambda(k)}, \tag{3.11}$$

o la versión más conservadora "**AND**":

$$(j,k) \in \widehat{E}_{\text{“}\mathbf{AND}\text{”}} \iff k \in \widehat{\mathcal{N}_\lambda(j)} \text{ y } j \in \widehat{\mathcal{N}_\lambda(k)}. \tag{3.12}$$

El algoritmo es equivalente al Algoritmo 1 cambiando en el segundo paso la función a maximizar por (3.8).

Existen distintos criterios para seleccionar el parámetro de penalización $\lambda$. Cabe aclarar que, por lo mencionado anteriormente sobre la intratabilidad de la evaluación de la función de partición, no usaremos la función de verosimilitud total deducible de (3.1) para medir la bondad del ajuste. En su lugar, lo que haremos será usar algún criterio de elección de $\lambda$ para cada una de las regresiones logísticas por separado. Algunos de estos criterios, de los cuales hablamos en el caso de regresiones lineales en alta dimensión en el capítulo 2, son validación cruzada, stability selection, AIC, BIC y EBIC. Los últimos tres son criterios de información usualmente usados para seleccionar modelos y se basan en maximizar la verosimilitud con una penalización a los grados de libertad del modelo para evitar el sobreajuste. Su filosofía es lograr un balance entre bondad de ajuste y simplicidad.

En particular, el criterio EBIC (Extended Bayesian Informaton Criterion) sugerido por Van Borkulo et al. (2014) y Barber y Drton (2015) para aplicar a este método de sucesivas regresiones tiene, para cada regresión logística de una variable $X_j$, con $j \in V$, en función de las demás variables $X_{V\setminus\{j\}}$, la fórmula

$$\text{BIC}_{j,\gamma}(\lambda) = -2l(\widehat{\boldsymbol{\beta}^{(j)}(\lambda)}) + |J| \log(n) + 2\gamma|J| \log(p-1), \tag{3.13}$$

donde $l(\widehat{\boldsymbol{\beta}^{(j)}(\lambda)})$ es la verosimilitud logarítmica (3.7) de la regresión logística con penalización $\lambda$ evaluada en la estimación de $\boldsymbol{\beta}^{(j)}(\lambda)$, $|J|$ es el número de vecinos seleccionados en la regresión logística, $p-1$ es el numero de variables explicativas de la regresión y $\gamma$ es un hiperparámetro positivo que regula la penalización del último término y cuyo valor fijaremos arbitrariamente. Si $\gamma$ vale cero estamos en el caso del criterio BIC tradicional. Definida esta formula, el método consistirá en elegir para cada $j \in V$ el $\lambda \in \Lambda$ que minimice $\text{BIC}_{j,\gamma}(\lambda)$. De esta forma quedarán determinados los vecindarios para cada variable.

## 3.3. Métodos para variables discretas en general

### 3.3.1. Algoritmo *Chow-Liu*

Con el fin de contar con un método con el cual comparar el método basado en los vecindarios introducimos el algoritmo de *Chow-Liu* (Chow y C. Liu (2006)). Para la descripción de este método necesitamos las siguientes definiciones de la teoría de grafos.

Un grafo es **conexo** si dado cualquier par de vértices del grafo existe un camino que los une. Un **árbol** es un grafo conexo y sin ciclos. Un **árbol generador** de un grafo $\mathcal{G} = (V, E)$ es un subgrafo $\mathcal{G}' = (V, E')$ $(E' \subset E)$ que es un árbol. Notemos que si $\mathcal{G}$ no es conexo entonces no tiene árbol generador. Por último, si tenemos un grafo $\mathcal{G} = (V, E, w)$ con pesos en las aristas (es decir que tenemos una función $w : E \mapsto \mathbb{R}$) entonces el **árbol generador máximo** de $\mathcal{G}$ es el árbol generador de $\mathcal{G}$ que maximiza la suma de los pesos de sus aristas.

El método Chow-Liu es muy eficiente pero su uso se reserva para el caso en el que el verdadero grafo subyacente es un árbol. Puede probarse que los modelos que se representan con árboles se factorizan de la siguiente forma:

$$p(\mathbf{x}) = \prod_{i \in V} p_i(\mathbf{x}_i) \prod_{(i,j) \in E} \frac{p_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{p_i(\mathbf{x}_i) p_j(\mathbf{x}_j)}, \tag{3.14}$$

donde, para cada $i, j$ en $V$, $p_{ij}$ es la distribución conjunta de $(X_i, X_j)$ y $p_i$, $p_j$ son las distribuciones marginales de $X_i$ y $X_j$ respectivamente. En Tan et al. (2011) se muestra que maximizar la verosimilitud de la muestra sobre el conjunto de distribuciones que se factorizan como en (3.14) es equivalente a minimizar la divergencia de Kullback-Leiber [1] $D(\widehat{P}|Q)$ sobre el mismo conjunto de distribuciones, donde $\widehat{P}$ es la distribución empírica de la muestra y $Q$ pertenece al conjunto de distribuciones que satisfacen (3.14). Además, se prueba que la estructura del grafo de la distribución que minimiza esta divergencia se obtiene buscando el árbol generador máximo en un grafo completo en el cual cada arista tiene como peso el número de información mutua $\widehat{I_{ij}}$ entre las marginales empíricas de ambos extremos de la arista. El número de información mutua entre dos distribuciones se define de la siguiente forma:

$$I_{ij} = I(X_i, X_j) = \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \chi^2} p_{ij}(\mathbf{x}_i, \mathbf{x}_j) \log \frac{p_{ij}(\mathbf{x}_i, \mathbf{x}_j)}{p_i(\mathbf{x}_i) p_j(\mathbf{x}_j)}.$$

En cuanto a la eficiencia del método, sabemos que el problema de buscar el árbol generador máximo cuando todas las aristas son positivas puede ser resuelto con el algoritmo *Kruskal* en tiempo $\mathcal{O}(p \times \log(p))$. Esquematizamos a continuación el algoritmo descripto.

**Algoritmo 3:** Algoritmo *Chow-Liu* para encontrar el árbol subyacente

**para cada** $i \in V$ **hacer**
  Calcular la marginal empírica $\widehat{p_i}(\mathbf{x}_i)$ ;
  **para cada** $j \in V$ **hacer**
    Calcular $\widehat{p_{ij}}(\mathbf{x}_i, \mathbf{x}_j)$ ;
    Calcular el número de información mutua $\widehat{I_{ij}}$ ;
  **fin**
**fin**
Buscar con el algoritmo *Kruskal* el árbol generador máximo del grafo completo con los vértices $V$ y para cada arista $(i, j)$ el peso $w_{ij} = \widehat{I_{ij}}$ ;

Para finalizar la descripción de este método mencionamos que puede generalizarse a distribuciones cuyo grafo subyacente sea un **bosque** (conjunto de árboles).

[1] Recordamos la definición de la divergencia de Kullback-Leiber: $D(P|Q) = \sum_{\mathbf{x} \in \chi^p} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})}$

### 3.3.2. Algoritmo basado en la pseudo-verosimilitud

Presentamos el método general no-paramétrico para variables discretas con un diccionario $A$ finito propuesto en Frondana (2016). Inspirado en la intratabilidad de la evaluación de la función de partición de los métodos de verosimilitud para variables discretas en altas dimensiones, se propone un método de pseudo-verosimilitud que consiste en maximizar el producto de las probabilidades condicionales (que no implican la evaluación de una función de partición). Este método utiliza un criterio similar a AIC y BIC que penaliza los grados de libertad de la estimación.

Definimos para un vértice $j$ y un conjunto $W \subset V \setminus \{j\}$ la pseudo-verosimilitud condicional:

$$l_{j|W} = \prod_{i=1}^{n} p(x_j^{(i)} | x_W^{(i)}). \tag{3.15}$$

Estimamos esta cantidad a partir de estimar no-paramétricamente las probabilidades condicionales necesarias para el cálculo y notamos $\widehat{l_{j|W}}$ a la estimación.

Luego, estimaremos el vecindario de cada variable en forma exhaustiva maximizando la pseudo-verosimilitud penalizada sobre los $2^{|V|-1}$ conjuntos posibles de vecindarios del nodo $j$:

$$\widehat{\mathcal{N}(j)} = \underset{W \subset V \setminus \{j\}}{\operatorname{argmax}} \{\log(\widehat{l_{j|W}}) - c|A|^{|W|} \log_{|A|} n\}, \tag{3.16}$$

donde $c$ es un parámetro a elección que regula el peso de la penalización a la complejidad del modelo.

Usamos una regla "**AND**" u "**OR**" para resolver la falta de simetría en la estimación de los vecindarios al igual que en los otros métodos basados en los vecindarios. Se prueba que esta estimación de los vecindarios es consistente.

El código en **C** que implementa este método se encuentra disponible en `http://github.com/yoshiomori/neighborhoods.git`.

## 3.4. Simulaciones y análisis de datos reales

### 3.4.1. Estimación del grafo: red de palabras

El objetivo de este análisis es generar un grafo que sea una especie de red de campos léxicos de palabras y que cumpla las propiedades de un modelo gráfico no dirigido. Para eso usamos un subconjunto del conjunto de datos `news` (2008) que es una matriz binaria en la cual encontramos la medición de las apariciones de $p = 100$ palabras en un conjunto de $n = 1000$ noticias. Para cada noticia $l$, la columna $j$ vale 1 si la palabra $j$ aparece en la noticia $l$ y 0 en caso contrario. Como siempre en la estimación de grafos, recordemos que el contexto de alta dimensión viene de la necesidad de la estimación de $\frac{p\times(p-1)}{2} + p$ parámetros que en este caso son 5050.

Para estimar la estructura del grafo aplicamos el método *Nodewise logistic Regression* usando distintos criterios para seleccionar el valor de penalización $\lambda$. Los criterios utilizados fueron: validación cruzada, stability selection y el criterio EBIC .

Una cosa a tener en cuenta es que podemos encontrar aristas poco intuitivas en algunos de los grafos estimados, esto puede deberse en parte a que la independencia condicional se interpreta en el sentido de la afinidad de dos variables a ser iguales pero también en el sentido de la afinidad a ser distintas. Esto se refleja en el signo de los parámetros $\Theta_{ij}$.

Tomamos por ejemplo la Figura 3.1, y damos ejemplos de cómo interpretar el grafo usando la propiedad global de Markov. Por un lado tenemos componentes conexas aisladas de la gran masa de nodos, por ejemplo la relacionada con juegos (*games, team, players*, etc.) y las duplas *israel-jews*

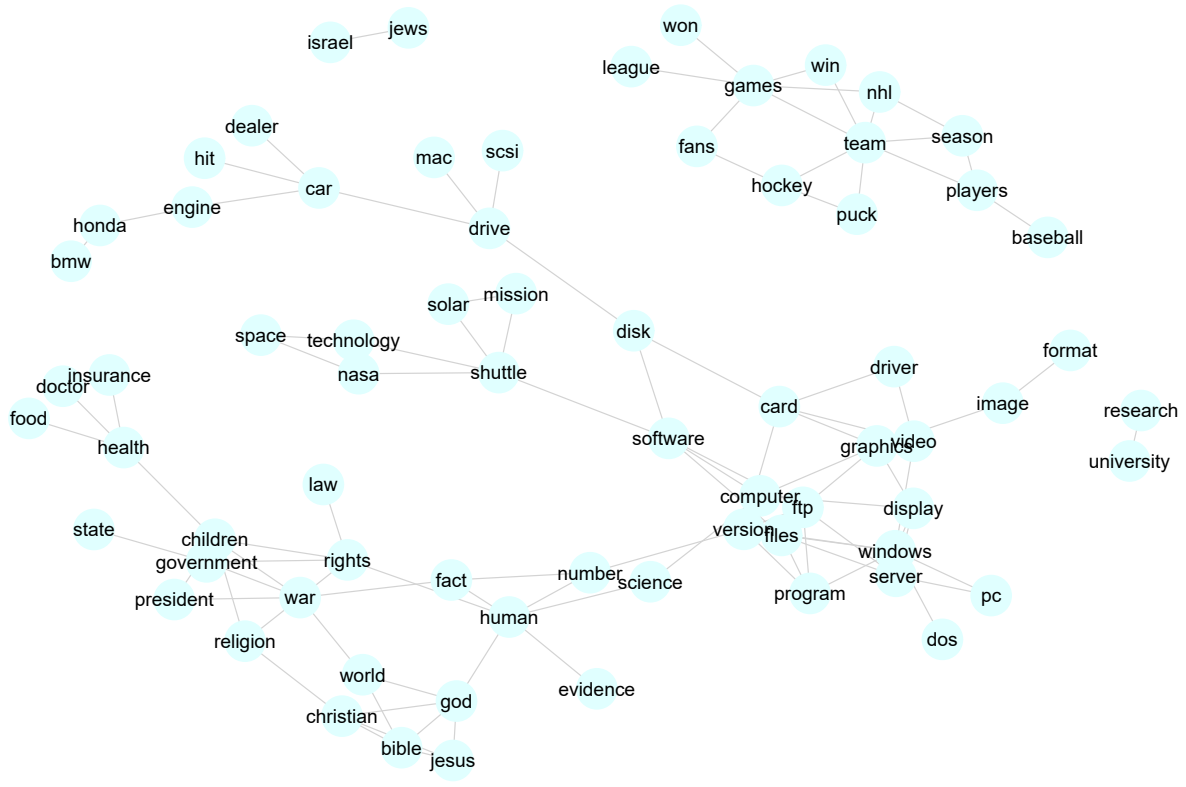



Figura 3.1: Estimación a partir del método *Nodewise logistic regression* de la estructura del grafo para el modelo de aparición de palabras en un cuerpo de noticias. La regularización fue elegida con el criterio EBIC.

y *research-university*. Por otro lado tenemos también reconocibles grupos (sin llegar a ser cliques) que se unen con otros a través de algunos pocos nodos: usando la propiedad global de Markov deducimos que la aparición del grupo de palabras relacionados con el espacio (*space, shuttle, nasa*, etc.) es condicionalmente independiente de la aparición del de tecnología (*files, program*, etc.) y del de autos (*car, dealer*, etc.) dada la aparición de la palabra *software*. También vemos que la aparición de la palabra *food* es condicionalmente independiente de la palabra *insurance* dada la aparición de *health*.

### 3.4.2. Estimación del grafo: cámara de diputados

El objetivo de este análisis es encontrar un grafo de diputados según sus votos pasados al estilo del trabajo Banerjee et al. (2008). Para esto utilizamos los conjuntos de datos `diputados`, `asuntos-diputados`, `bloques-diputados` y `votaciones-diputados` disponibles en la página `decada votada` (2016) para generar una matriz de datos con las votaciones de los diputados en asuntos seleccionados entre 2013 y 2015. Seleccionamos asuntos entre estos dos años para intentar que el cuerpo de diputados no cambiara mucho por el recambio legislativo habitual, además eliminamos a los diputados con muchas ausencias imputadas. El conjunto de datos tiene tamaño de muestra $n = 290$ (asuntos tratados) y $p = 130$ variables (diputados seleccionando sólo los partidos más representativos: FPV, PRO, UCR, Nuevo Encuentro, ARI). Los votos negativos, abstenciones y ausencias fueron computados como una misma clase. Observando la Figura 4.1 vemos una clara


Figura 3.2: Estimación a partir del método *Nodewise logistic regression* de la estructura del grafo para el modelo de aparición de palabras en un cuerpo de noticias. La regularización fue elegida con el criterio de validación cruzada.

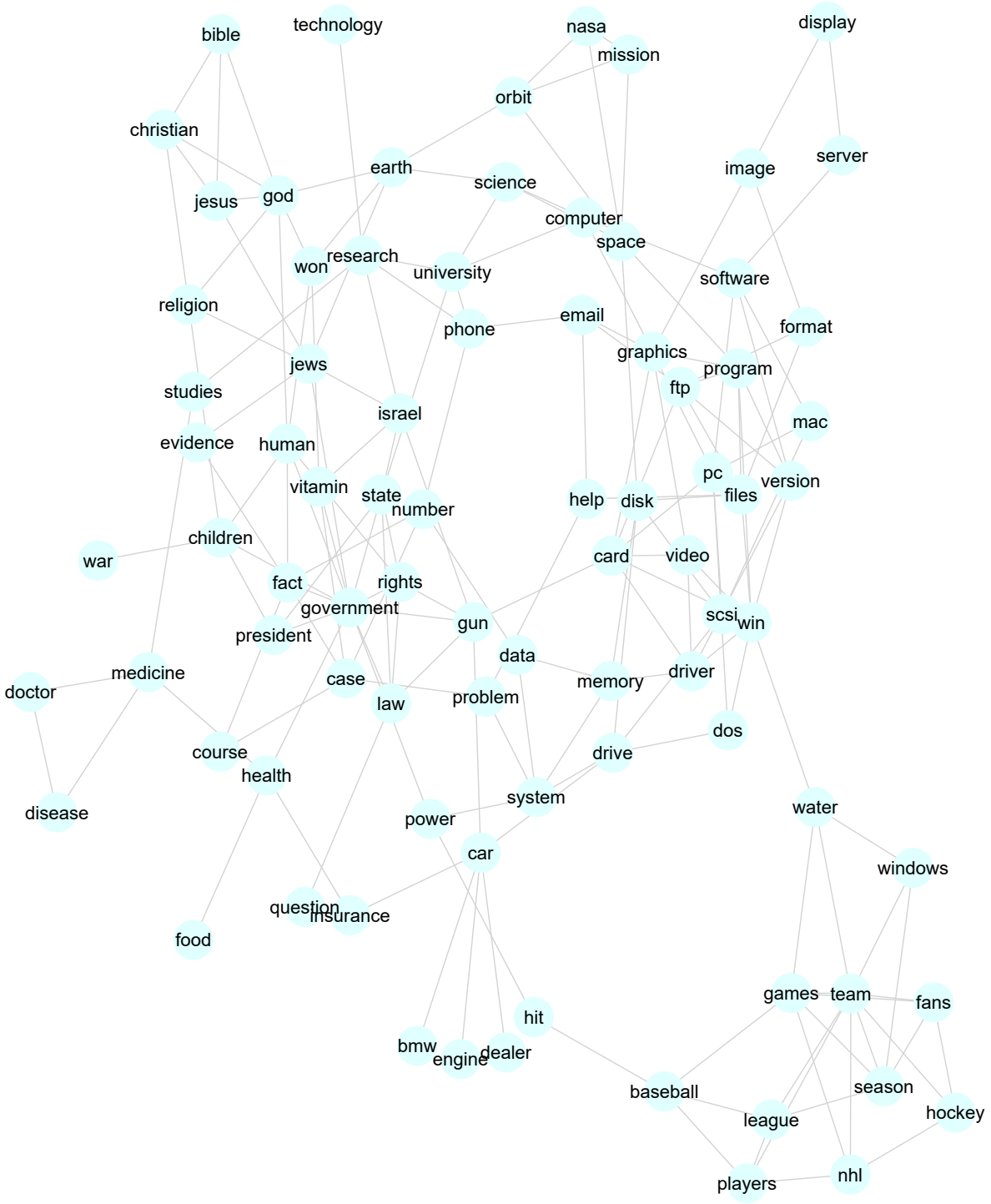



Figura 3.3: Estimación a partir del método *Nodewise logistic regression + Stability Selection* de la estructura del grafo para el modelo de aparición de palabras en un cuerpo de noticias.

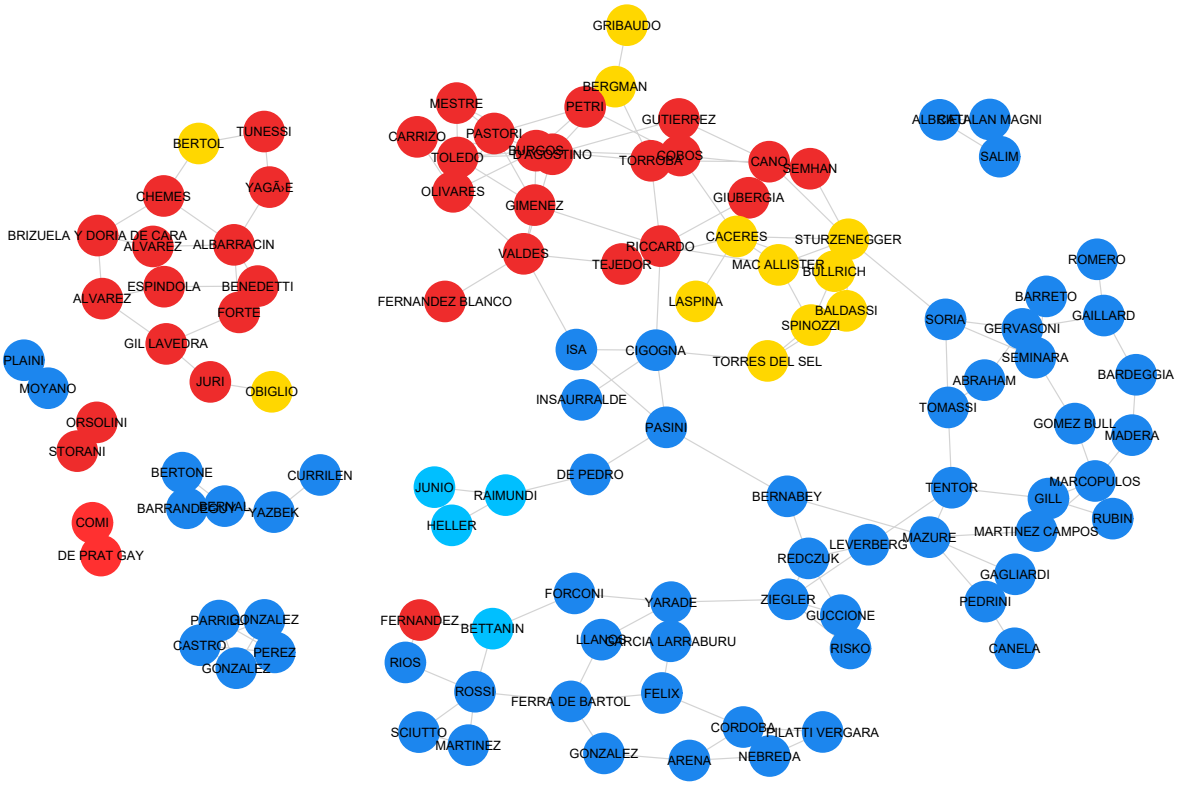



Figura 3.4: Estimación a partir del método *Nodewise logistic regression* con criterio EBIC de la estructura del grafo para el modelo de los votos de los diputados argentinos entre 2013 y 2015. El color de cada nodo representa el bloque de cada diputado: Rojo= UCR/ARI, Amarillo=PRO, Azul=FPV y Celeste=Nuevo Encuentro.

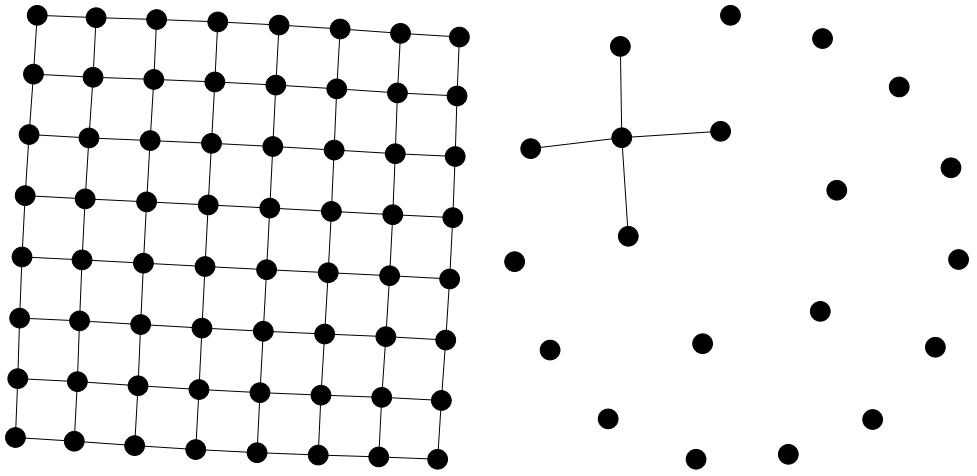
separación entre los distintos partidos (con UCR cercano a PRO y ARI y Nuevo Encuentro cercano a FPV) con pocas relaciones inter-partidos.

### 3.4.3. Simulación de estimación de grafos

El objetivo es hacer una simulación para comparar el desempeño de los distintos métodos presentados. Queremos comparar los métodos *Nodewise logistic regression* con los criterios de selección de regularización CV y EBIC, *Chow-Liu* y el basado en pseudo-verosimilitud. Con este fin, realizamos un análisis similar al de Ravikumar, Wainwright et al. (2011) utilizando grafos de $p = 64$ variables. Las clases de grafos que utilizamos fueron los grafos tipo grilla con 4 vecinos y los conjuntos de estrellas que se aprecian en la Figura 3.5.

El primer tipo de grafos es muy utilizado para modelar imágenes, cada nodo representa un píxel. Como no es un árbol, el método *Chow-Liu* no recupera nunca la estructura original. El segundo tipo de grafo es un bosque (conjunto de árboles) por lo que el método *Chow-Liu* podrá estimar la estructura del grafo.

Para ambos tipos de grafos generamos la matriz de parámetros de un modelo Ising para luego, con el paquete `IsingSample` de **R**, generar $nrep = 20$ muestras de tamaño variable. Estimamos la estructura del grafo para tamaños de muestra $n = \{500, 1000, 5000, 10000\}$ para calcular el porcentaje de acierto.



(a) (b)

Figura 3.5: Grafos utilizados para la simulación. En (a) vemos una grilla del estilo 4 vecinos con 64 variables y en (b) un grafo del tipo conjunto de estrellas con una estrella de grado 4.

Para estimar con el método *Chow-Liu* utilizamos la función `minForest` del paquete `gRapHD` de **R**. Para estimar con el método *Nodewise Logistic Regression* con el criterio EBIC utilizamos la función `IsingFit` del paquete `IsingFit` de **R** y para estimar con el criterio de validación cruzada y con el criterio *stability selection* utilizamos la función disponible en `https://github.com/violetr/tesis`.

Para medir la bondad de las estimaciones realizadas utilizamos el porcentaje de acierto del grafo completo para cada uno de los tamaños de muestra y las medidas Precision y Recall. Dada una estimación de un conjunto de variables binarias, definimos TP como la cantidad de verdaderos que se estiman como verdaderos, FP como la cantidad de falsos que fueron estimados como verdaderos y FN como la cantidad de positivos clasificados como falsos. En el caso de la estimación del grafo, cada variable binaria que se estima representa la existencia de cada una de las posibles aristas del grafo. Precision es un valor entre 0 y 1 que se define con la fórmula

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

Representa la proporción de los positivos correctamente identificados. Por otro lado, Recall se define con la fórmula

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Representa la proporción de positivos que fue detectada. Estas medidas fueron volcadas en los Cuadros 3.2,3.3, 3.4 y 3.5.

De estos cuadros concluimos que el método *Nodewise Logistic Regression* con el método de validación cruzada se comporta muy mal con ambos tipos de grafos, sobreestimando siempre en cantidad al conjunto de aristas. Esto se debe a la sobreestimación de conjunto de variables en el proceso de predición. Es por esto que el criterio stability selection mejora notablemente los resultados. Observando el comportamiento del método EBIC, vemos que, para muestras de tamaño pequeño, el criterio tiende a sub-estimar el conjunto de aristas. Esto se ve reflejado en los valores

| método — $n$ | 500 | 1000 | 5000 | 10000 |
|---|---|---|---|---|
| Chow-Liu | 0.04309829 | 0.14273217 | 0.5075068 | 0.6063413 |
| NLR - EBIC | NaN | NaN | 1 | 1 |
| NLR - CV | 0.02723124 | 0.05919155 | 0.1396104 | 0.1485191 |
| NLR - SS | 0.04761905 | NaN | 0.7092328 | 0.7007781 |

Cuadro 3.2: Valores promedio de Precision en la estimación de la estructura de un grafo del tipo estrella con 64 vértices y grado 7. Los valores se muestran según el método de selección de modelo y según el tamaño de la muestra.

| método — $n$ | 500 | 1000 | 5000 | 10000 |
|---|---|---|---|---|
| Chow - Liu | 0.17142857 | 0.3785714 | 0.9714286 | 1 |
| NLR - EBIC | 0 | 0 | 0.7428571 | 1 |
| NLR - CV | 0.17857143 | 0.3642857 | 0.9714286 | 1 |
| NLR - SS | 0.02142857 | 0.1571429 | 0.9642857 | 1 |

Cuadro 3.3: Valores promedio de Recall en la estimación de la estructura de un grafo del tipo estrella con 64 vértices y grado 7. Los valores se muestran según el método de selección de modelo y según el tamaño de la muestra.

| método — $n$ | 500 | 1000 | 5000 | 10000 |
|---|---|---|---|---|
| NLR - EBIC | NaN | NaN | 0.99 | 0.98 |
| NLR - CV | 0.28 | 0.31 | 0.38 | 0.39 |
| NLR - SS | 0.45 | 0.60 | 0.94 | 0.96 |

Cuadro 3.4: Valores promedio de Precision en la estimación de la estructura de un grafo del tipo grilla con 64 vértices. Los valores se muestran según el método de selección de modelo y según el tamaño de la muestra.

| método — $n$ | 500 | 1000 | 5000 | 10000 |
|---|---|---|---|---|
| NLR - EBIC | 0.03 | 0.03 | 0.34 | 0.69 |
| NLR - CV | 0.25 | 0.29 | 0.82 | 0.93 |
| NLR - SS | 0.05 | 0.08 | 0.49 | 0.75 |

Cuadro 3.5: Valores promedio de Recall en la estimación de la estructura de un grafo del tipo grilla con 64 vértices. Los valores se muestran según el método de selección de modelo y según el tamaño de la muestra.

NaN de Precision que provienen del hecho de que el denominador de la fracción que la define es 0 pues el método estima el grafo vacío. Por el contrario, para muestras de mayor tamaño el criterio EBIC se comporta muy bien, estimando al grafo correctamente en la mayoría de los casos. Por otro lado, al estudiar los porcentajes de acierto obtenidos, observamos que estos no se condicen con los correspondientes en el trabajo Ravikumar, Wainwright et al. (2011). Queda como trabajo futuro, detectar las causas de las diferencias en los resultados.

Por el tiempo que tarda en correr el algoritmo basado en pseudo-verosimilitud, debido al carácter exponencial de su complejidad, decidimos finalmente no incluirlo en la comparación. De todos modos, realizamos estimaciones de grafos con un número menor de variables ($n = 10, n = 20$) que resultaron diferir en muy poco del grafo real.

# Capítulo 4

# Comentarios finales y trabajo a futuro

Para concluir, a lo largo de este trabajo realizamos un recorrido por algunos de los diferentes métodos para la selección de modelos gráficos no-dirigidos, abarcando asimismo los problemas de regresión lineal y logística en altas dimensiones.

Gracias a las simulaciones, verificamos empíricamente la consistencia en la selección de los grafos no-dirigidos bajo ciertas condiciones y pudimos comparar el desempeño de los distintos métodos entre sí. Vimos que la elección del parámetro $\lambda$ de penalización es clave en la selección del modelo y que el método *stability selection* sirve para disminuir esta dependencia tan fuerte del parámetro. Por otra parte, aplicamos los métodos presentados a distintos conjuntos de datos reales, pudiendo utilizar la estructura seleccionada para la visualización de las relaciones de interdependencia entre las variables y su interpretación.

Los tipos de modelos estudiados a lo largo del trabajo fueron el modelo normal multivariado y los modelos con variables exclusivamente discretas. Como trabajo a futuro, queda pendiente indagar sobre las diferentes clases de modelos que quedaron afuera. Una de ellas es aquella en la que las variables tienen la condición de ser continuas pero, sin embargo, no aparentan tener una distribución gaussiana (H. Liu et al. (2009)). Otro es el caso que corresponde a los modelos en los cuales hay variables discretas y continuas simultáneamente, comúnmente llamados modelos mixtos (Fan et al. (2017)). Existen también métodos que contemplan conjuntos de datos con datos faltantes. Por último, un caso interesante es el de los modelos que tienen variables ocultas (no observadas). Un ejemplo concreto de aplicación de esta clase de modelos es el problema de segmentación de una imagen en primer plano y fondo en donde las variables observadas son los píxeles y por cada pixel hay una variable binaria oculta que clasifica con su valor al píxel correspondiente en "primer plano" o "fondo".

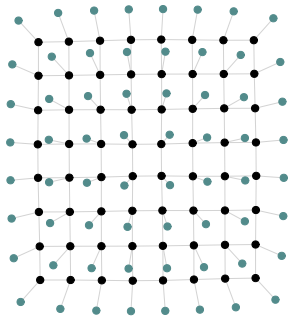


Figura 4.1: Estructura del modelo gráfico con variables ocultas utilizado para segmentar una imagen en fondo y figura principal.

# Apéndice A

# Normal Multivariada

Un vector aleatorio $\vec{X}$=($X_1$,..,$X_p$) tiene una distribución normal multivariada con media $\boldsymbol{\mu}$ y matriz de covarianza $\Sigma$, con $\Sigma$ simétrica y positiva, si su función de densidad es :

$$f(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{A.1}$$

lo notamos $\vec{X} \sim N(\boldsymbol{\mu},\Sigma)$.

Al igual que la distribución gaussiana en una variable, la normal multivariada de 2 dimensiones tiene un gráfico en forma de campana tridimensional. Esta campana está centrada en la media $\boldsymbol{\mu}$ y su forma está dada por su matriz de covarianza como se ve en el ejemplo de la Figura A.1.

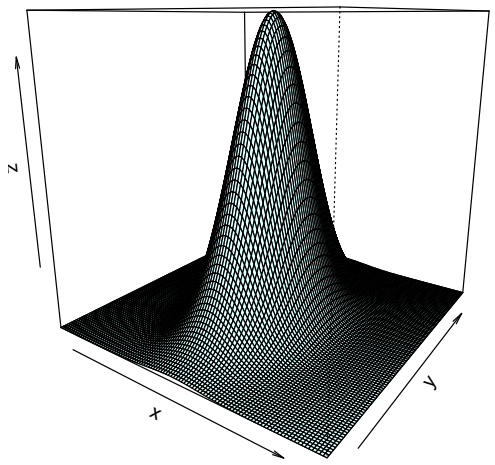



Figura A.1: Gráfico de la densidad de una distribución normal multivariada en 2 dimensiones con $\boldsymbol{\mu} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ y $\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$.

A continuación resumimos los principales resultados de esta distribución:

- la suma de gaussianas independientes es gaussiana,
- una transformación afín de una gaussiana resulta una gaussiana,

- sus distribuciones marginales son normales,
- sus distribución condicional resulta también normal y
- las independencias condicionales se encuentran codificadas en los ceros de la matriz de precisión.

**Lema A.1.** *Sean $\vec{X} \sim N(\boldsymbol{\mu}, \Sigma)$ con $\Sigma \in \mathbb{R}^{p\times p}$ simétrica y definida positiva, $f$ la función de densidad de $\vec{X}$, $C \in \mathbb{R}^{p\times p}$ inversible y $\boldsymbol{b} \in \mathbb{R}^p$. Entonces el vector aleatorio $\vec{Y} = C\vec{X} + \boldsymbol{b} \sim N(C\boldsymbol{\mu} + \boldsymbol{b}, C\Sigma C^T)$ con $C\Sigma C^T \succ 0$.*

*Demostración.* Como $C$ es inversible entonces $(C\Sigma C^T)^{-1} = (C^T)^{-1}\Sigma^{-1}C^{-1}$ existe y sea $\mathbf{y} = (C\mathbf{x}+\mathbf{b})$ podemos escribir $\mathbf{x} = C^{-1}(\mathbf{y}-\mathbf{b})$. Luego, podemos escribir a $g(\mathbf{y})$, la función de densidad de $C\mathbf{x}+\mathbf{b}$, de la siguiente forma por el teorema de cambio de variables:

$$\begin{aligned} g(\mathbf{y}) &= f(C^{-1}(\mathbf{y}-\mathbf{b}))|\det(C^{-1})| \\ &= \frac{1}{(2\pi)^{p/2}(\det(\Sigma))^{1/2}} \exp\left[-\frac{1}{2}((C^{-1}(\mathbf{y}-\mathbf{b})-\boldsymbol{\mu})^T\Sigma^{-1}(C^{-1}(\mathbf{y}-\mathbf{b})-\boldsymbol{\mu}))\right]|\det(C^{-1})|. \end{aligned}$$

Sabemos que $det(C^{-1}) = 1/\det(C) > 0$, luego $(\det(\Sigma))^{-1/2}|\det(C^{-1})| = (\det(C^T\Sigma C))^{-1/2}$. Usando esto y reordenando el exponente de la exponencial queda

$$g(\mathbf{y}) = \frac{1}{(2\pi)^{p/2}(\det(\Sigma_{\vec{Y}}))^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}_{\vec{Y}})^T\Sigma_{\vec{Y}}^{-1}(\mathbf{y}-\boldsymbol{\mu}_{\vec{Y}})\right), \tag{A.2}$$

con $\Sigma_{\vec{Y}} = C^T\Sigma C$ y $\boldsymbol{\mu}_{\vec{Y}} = C\boldsymbol{\mu} + \mathbf{b}$. Reconocemos la distribución normal multivariada con esta función de densidad.

□

Consideramos la normal estándar multivariada $\vec{Z}_p$ definida como $N(\mathbf{0}, \mathrm{Id}_p)$. Su densidad es

$$f(\mathbf{z}) = \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}\sum_{j=1}^{p}(\mathbf{z}_j)^2\right), \tag{A.3}$$

de donde concluimos que las componentes del vector aleatorio $Z_p$ son normales estándar (univariadas) independientes.

Usando que dada $\Sigma$ una matriz simétrica y definida positiva existe $C$ inversible tal que $CC^T = \Sigma$ en tenemos que

$$\vec{X} \sim N(\boldsymbol{\mu}, \Sigma) \iff \exists\, C \text{ inversible tal que } CC^T = \Sigma \text{ y } X \sim C\vec{Z}_p + \boldsymbol{\mu}. \tag{A.4}$$

Para probar algunas propiedades usaremos la función característica del vector aleatorio $\vec{X}$ definida de la siguiente forma: $\psi_{\vec{X}}(\mathbf{t}) = E[e^{i\mathbf{t}.\vec{X}}]$.

Calculamos la función característica de la normal estándar multivariada:

$$\psi_{\vec{Z}_p} = E\left[\varepsilon^{i\mathbf{t}.\vec{Z}_p}\right] = E\left[e^{i\sum_{j=1}^{p}\mathbf{t}_j Z_j}\right] \tag{A.5}$$

$$= E\left[e^{i\sum_{j=1}^{p}\mathbf{t}_j Z_j}\right] = E\left[\prod_{j=1}^{p} e^{i\mathbf{t}_j X_j}\right] \tag{A.6}$$

$$= \prod_{j=1}^{p} E\left[e^{i\mathbf{t}_j X_j}\right] = \prod_{j=1}^{p}\psi_Z(\mathbf{t}_j) \tag{A.7}$$

$$= e^{-\frac{1}{2}\mathbf{t}^T\mathbf{t}}. \tag{A.8}$$

donde usamos la independencia de las componentes y en la última igualdad la fórmula de la función característica de una normal estándar univariada.

A partir de este cálculo y de la caracterización (A.4) es fácil ver que la función característica de una distribución $\vec{X} \sim N(\boldsymbol{\mu}, \Sigma)$ es $\psi_{\vec{X}} = e^{it^T\boldsymbol{\mu}-\frac{1}{2}\mathbf{t}^T\Sigma\mathbf{t}}$.

Para probar los siguientes resultados usaremos que podemos reconocer unívocamente a una distribución por su función característica y el siguiente hecho fácil de probar. Sea $\vec{X} = \begin{pmatrix} X_A \\ X_B \end{pmatrix}$ con función característica $\psi_{\vec{X}}(\mathbf{t})$ entonces la función característica de $X_A$ es $\psi_{X_A}(\mathbf{t}_A) = \psi_{\vec{X}}(\mathbf{t}_A, \mathbf{o}_B)$.

**Lema A.2.** *Sean $\vec{X} \sim N(\boldsymbol{\mu}_{\vec{X}}, \Sigma_{\vec{X}})$ e $\vec{Y} \sim N(\boldsymbol{\mu}_{\vec{Y}}, \Sigma_{\vec{Y}})$ dos vectores aleatorios independientes con medias y matrices de covarianza de igual tamaño respectivamente. Entonces $\vec{X} + \vec{Y} \sim N(\boldsymbol{\mu}_{\vec{X}} + \boldsymbol{\mu}_{\vec{Y}}, \Sigma_{\vec{X}} + \Sigma_{\vec{Y}})$.*

*Demostración.* Se prueba calculando la función característica de la suma y usando que la esperanza del producto es el producto de las esperanzas por ser vectores independientes:

$$\psi_{\vec{X}+\vec{Y}}(t) = E\left[e^{i\mathbf{t}.(\vec{X}+\vec{Y})}\right] = E\left[e^{i\mathbf{t}.\vec{X}+i\mathbf{t}.\vec{Y}}\right] = E\left[e^{i\mathbf{t}.\vec{X}}\right] E\left[e^{i\mathbf{t}.\vec{Y}}\right].$$

□

**Lema A.3.** *Sean $A$ y $B$ dos subconjuntos disjuntos de $\{1, ..., p\}$ y un vector aleatorio $\vec{X} = \begin{pmatrix} X_A \\ X_B \end{pmatrix} \sim N(\boldsymbol{\mu}, \Sigma)$. Sea $\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{pmatrix}$ y la matriz de covarianza $\Sigma$ separada por bloques de la siguiente forma:*

$$\Sigma = \begin{pmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{pmatrix}.$$

*Entonces las distribuciónes marginales son normales y vale que $X_A \sim N(\boldsymbol{\mu}_A, \Sigma_{AA})$ y $X_B \sim N(\boldsymbol{\mu}_B, \Sigma_{BB})$.*

*Demostración.* Calculamos la función característica de $X_A$:

$$\psi_{X_A}(\mathbf{t}_A) = \psi_{\vec{X}}(\mathbf{t}_A, \mathbf{o}_B) = e^{i\mathbf{t}_A^T\boldsymbol{\mu}_A - \frac{1}{2}\mathbf{t}_A{}^T\Sigma_{AA}\mathbf{t}_A}.$$

de donde concluimos directamente $X_A \sim N(\boldsymbol{\mu}_A, \Sigma_{AA})$. El resultado para $X_B$ se demuestra análogamente.

□

**Lema A.4.** *Sean $A$ y $B$ dos subconjuntos disjuntos de $\{1,...,p\}$ y un vector aleatorio $\vec{X} = \begin{pmatrix} X_A \\ X_B \end{pmatrix} \sim N(\boldsymbol{0}, \Sigma)$. Sea $\Theta = \Sigma^{-1}$ la matriz de precisión de $\vec{X}$ separada por bloques de la siguiente forma:*

$$\Theta = \begin{pmatrix} \Theta_{AA} & \Theta_{AB} \\ \Theta_{BA} & \Theta_{BB} \end{pmatrix}.$$

*Entonces la distribución condicional de $X_A$ dada $X_B$ es una gaussiana con distribución $N(-\Theta_{AA}^{-1}\Theta_{AB}X_B, \Theta_{AA}^{-1})$. Equivalentemente vale que $X_A = -\Theta_{AA}^{-1}\Theta_{AB}X_B + \varepsilon_A$ donde $\varepsilon_A \sim N(0, \Theta_{AA}^{-1})$ es independiente de $X_B$.*

*Demostración.* Sean $g(\mathbf{x}_A, \mathbf{x}_B)$ y $g(\mathbf{x}_B)$ las densidades de las distribuciones conjunta y marginal de $X_B$ respectivamente, calculamos la densidad condicional

$$g(\mathbf{x}_A|\mathbf{x}_B) = \frac{g(\mathbf{x}_A, \mathbf{x}_B)}{g(\mathbf{x}_B)} = \frac{|\Sigma_{BB}|^{1/2}}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}\mathbf{x}_A^T\Theta_{AA}\mathbf{x}_A - \mathbf{x}_A^T\Theta_{AB}\mathbf{x}_B - \frac{1}{2}\mathbf{x}_B^T(\Theta_{BB} - \Sigma_{BB}^{-1})\mathbf{x}_B\right],$$

con $\Sigma_{BB}$ la matriz de covarianza de $X_B$ por el lema anterior y $k$ el cardinal del conjunto $A$. Usando que $\Sigma_{BB}^{-1} = \Theta_{BB} - \Theta_{BA}\Theta_{AA}^{-1}\Theta_{AB}$ y $|\Sigma| = |\Sigma_{AA}||\Sigma_{BB} - \Sigma_{AB}^T\Sigma_{AA}^{-1}\Sigma_{AB}|$ por propiedades de la multiplicación de matrices por bloques tenemos

$$g(\mathbf{x}_A|\mathbf{x}_B) = \frac{1}{(2\pi)^{k/2}|\Sigma_{AA}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_A + \Theta_{AA}^{-1}\Theta_{AB}\mathbf{x}_B)^T\Theta_{AA}(\mathbf{x}_A + \Theta_{AA}^{-1}\Theta_{AB}\mathbf{x}_B)\right].$$

Y reconocemos que esta es la densidad de una gaussiana $N(-\Theta_{AA}^{-1}\Theta_{AB}\mathbf{x}_B, \Theta_{AA}^{-1})$.

□

**Corolario A.5.** *Para cualquier $a \in \{1,...,p\}$ vale que*
$X_a = -\sum_{b:b\neq a} \frac{\Theta_{ab}}{\Theta_{aa}} X_b + \epsilon_a$ *donde $\epsilon_a \sim N(0, \Theta_{aa}^{-1})$ es independiente de $\{X_b : b \neq a\}$.*

*Demostración.* Aplicamos el lema anterior con $A = \{a\}$ y $B = A^c$. □

Veamos ahora que la matriz de precisión codifica las independencias condicionales de las normales multivariadas.

**Proposición A.6.** *Sea $\vec{X} \sim N(0, \Sigma)$ con $\Sigma \in \mathbb{R}^{p\times p}$ simétrica y definida positiva entonces para cada par $i, j \in V = \{1,...,p\}$ se cumple*

$$X_i \perp X_j | X_{V\setminus\{i,j\}} \iff \Theta_{ij} = 0 \quad y \quad \Theta_{ji} = 0, \tag{A.9}$$

*con $\Theta = \Sigma^{-1}$.*

*Demostración.* Por el criterio de factorización tenemos que dados $i, j \in V = \{1,...,p\}$ entonces $X_i \perp X_j | X_{V\setminus\{i,j\}}$ si y sólo si $f(\mathbf{x})$ admite una factorización de la forma

$$f(\mathbf{x}) = g(\mathbf{x}_{V\setminus\{i,j\}}, \mathbf{x}_i)h(\mathbf{x}_{V\setminus\{i,j\}}, \mathbf{x}_j).$$

Si escribimos a la forma cuadrática $\mathbf{x}^T\Theta\mathbf{x}$ de la forma

$$\mathbf{x}^T\Theta\mathbf{x} = \sum_{k=1}^{p}\sum_{l=1}^{p}\Theta_{kl}\mathbf{x}_k\mathbf{x}_l, \tag{A.10}$$

tenemos que si y sólo si $\Theta_{ij} = 0$ y $\Theta_{ji} = 0$ podemos escribir la densidad de $\mathbf{x}$ de la forma

$$f(\mathbf{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}\sum_{k=1,k\neq j}^{p}\sum_{l=1,l\neq j}^{p}\Theta_{kl}\mathbf{x}_k\mathbf{x}_l\right)\exp\left(-\frac{1}{2}\sum_{k=1,k\neq i}^{p}\sum_{l=1,l\neq i}^{p}\Theta_{kl}\mathbf{x}_k\mathbf{x}_l\right).$$

Luego, probamos lo que queríamos ver. □

**Proposición A.7.** *Dado un vector aleatorio $\vec{X} = (X_1, ..., X_p) \sim N(\boldsymbol{\mu}, \Sigma)$, si tenemos $\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$ $(p < n)$ realizaciones del vector, entonces el estimador de máxima verosimilitud de $(\boldsymbol{\mu}, \Sigma)$ es $(\hat{\boldsymbol{\mu}}_{MV}, \boldsymbol{S})$ donde $\hat{\boldsymbol{\mu}}_{MV} = \sum_{i=1}^{n}\boldsymbol{x}^{(i)}$ y $\boldsymbol{S} = n^{-1}\sum_{i=1}^{n}(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}}_{MV})(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{\mu}}_{MV})^T$ .*

*Demostración.* A partir de la expresión (A.1) planteamos la función de verosimilitud:

$$l(\boldsymbol{\mu},\Sigma;\mathbf{x}_1,..,\mathbf{x}^{(n)}) = -n\log(\det(\Sigma)^{1/2}) - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu}).$$

Para condensar un poco la expresión usamos propiedades de la función traza. Usando que $\mathrm{tr}(ABC) = \mathrm{tr}(CAB) = \mathrm{tr}(BCA)$ y que para una constante podemos usar $\mathrm{tr}(c) = c$ obtenemos

$$l(\boldsymbol{\mu},\Sigma;\mathbf{x}_1,..,\mathbf{x}^{(n)}) = \frac{n}{2}\log(\det(\Sigma^{-1})) - \frac{n}{2}\mathrm{tr}\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T\Sigma^{-1}\right).$$

Para encontrar los estimadores de máxima verosimilitud buscamos los valores de $\mu$ y de $\Sigma$ que maximizan la verosimilitud. Para esto, derivamos e igualamos a 0 para buscar extremos locales.

Usamos las siguientes identidades para derivar con respecto a matrices: $\frac{\partial \mathrm{tr}[BA]}{\partial A} = B^T$, $\frac{\partial \log[|A|]}{\partial A} = A^{-T}$ con A y B matrices y por último $\frac{\partial x^T a}{\partial x} = a$ con $x$ y $a$ vectores.

Usando las propiedades anteriores tenemos: $\frac{\partial x^T Ax}{\partial A} = \frac{\partial \mathrm{tr}[x^T xA]}{\partial A} = [xx^T]^T = xx^T$.

Finalmente nos quedan las ecuaciones

$$\frac{\partial l(\boldsymbol{\mu},\Sigma)}{\partial \boldsymbol{\mu}} = -\frac{1}{2}n\Sigma^{-1}\boldsymbol{\mu} + \frac{1}{2}\Sigma^{-1}\sum_{i=1}^{n}\mathbf{x}^{(i)} = 0$$

y

$$\frac{\partial l(\boldsymbol{\mu},\Sigma)}{\partial \Sigma^{-1}} = \frac{n}{2}\Sigma^T - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^T = 0,$$

de las que despejamos:

$$\widehat{\boldsymbol{\mu}}_{MV} = \frac{1}{n}\sum_{i=1}^{n}\vec{X}^{(i)} \tag{A.11}$$

y

$$\widehat{\Sigma}_{MV} = \mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\vec{X}^{(i)} - \hat{\boldsymbol{\mu}}_{MV})(\vec{X}^{(i)} - \hat{\boldsymbol{\mu}}_{MV})^T. \tag{A.12}$$

□

# Apéndice B

# Guía de códigos y datos

Todos los códigos propios utilizados en las simulaciones y en los análisis de datos junto con las bases de datos analizadas se encuentran en `https://github.com/violetr/tesis`. A continuación describimos los archivos.

## Código de los métodos utilizados

A continuación damos los nombres de las funciones usadas. Si no se especifica el paquete de **R** entonces el código es propio salvo para el caso del método pseudo-likelihood (Frondana (2016)) que se encuentra implementado en el lenguaje **C** en la dirección especificada. Los archivos se encuentran en la carpeta `metodos`.

### Regresión lineal

- *lasso, ridge*: `glmnet::glmnet/cv.glmnet`
- *stability selection*: `hdi::stability`

En el archivo `regresion_lineal.R` se encuentran las siguientes funciones:

- adaptive *lasso* cv: `cv.adaptive.lasso`
- thresholded *lasso* cv: `cv.tau.threslasso`

### Regresión logística

- regresión logística con penalización: `glmnet::glmnet/cv.glmnet`

### Estimación del grafo normal

- *GLasso*: `glasso::glasso`

En el archivo `seleccion_modelo_normal.R` se encuentran las siguientes funciones:

- *Adaptive GLasso*: `adaptive.glasso`
- *(Adaptive)GLasso* CV: `cv.glasso`
- *Nodewise regression*: `nodewisereg`

- *Stability Selection*: `stability.selection.grafico`
- *Gelato (nodewise+threshold)*: `cv.Gelato`

### Estimación del grafo discreto

- *Chow-Liu*: `gRapHD::minForest`
- pseudo-likelihood: `http://github.com/yoshiomori/neighborhoods.git`.

En el archivo `seleccion_modelo_binario.R` se encuentran las siguientes funciones:

- *Nodewise Logistic Regression*: `nodewiselogreg` (con criterio EBIC, CV, stability selection)

## Simulaciones y análisis de datos

Los archivos se encuentran en la carpeta `analisis`.

### Modelo normal

- `3_2_1_3_tradeoff.R`: gráfico error, sesgo y varianza vs. $\lambda$ para el método *lasso*.
- `3_7_1_reg.R`: simulación regresión lineal con *lasso*, *adaptive*, *thresholded* y *ridge*.
- `3_7_2_consistencia.R`: gráfico de la consistencia en la estimación de la matriz de precisión para normal simulada.
- `3_7_3_grafperdida.R`: gráfico de la pérdida con *GLasso*, *Adaptive GLasso* y *Gelato* para `gene expression - arabidopsis thaliana` (2004) y normal simulada.
- `3_7_4_ribo.R`: selección de modelo para `hdi::riboflavin` con *Stability Selection*.

### Modelos discretos

- `4_2_1_prostate.R`: clasificación de los datos `spls::prostate`.
- `4_4_1_palabras.R`: selección de modelo para datos `news` (2008).
- `4_4_2_dipus.R`: selección de modelo para datos de diputados en `decada votada` (2016).
- `4_4_3_bin.R`: simulación de selección de modelo para datos binarios.

## Bases de datos

Las bases de datos se encuentran en la carpeta `datos`.

# Bibliografía


Banerjee, O., El Ghaoui, L. y d'Aspremont, A. (2008). "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data". En: *J. Mach. Learn. Res.* 9, págs. 485-516. ISSN: 1532-4435.

Barber, R. F. y Drton, M. (2015). "High-dimensional Ising model selection with Bayesian information criteria". En: *Electron. J. Statist.* 9.1, págs. 567-607. DOI: 10.1214/15-EJS1012. URL: http://dx.doi.org/10.1214/15-EJS1012.

Blake, A., Kohli, P. y Rother, C. (2011). *Markov Random Fields for Vision and Image Processing*. The MIT Press. ISBN: 0262015773, 9780262015776.

Bühlmann, P. y Van de Geer, S. (2011). *Statistics for High-Dimensional Data*.

Chow, C. y Liu, C. (2006). "Approximating Discrete Probability Distributions with Dependence Trees". En: *IEEE Trans. Inf. Theor.* 14.3, págs. 462-467. ISSN: 0018-9448.

decada votada (2016). Conjunto de archivos con datos de las votaciones de las cámaras de diputados y senadores de Argentina entre los años 2003 y 2016. URL: http://decadavotada.com.ar/doc.html.

Dezeure, R., Bühlmann, P., Meier, L. y Meinshausen, N. (2015). "High-Dimensional Inference: Confidence Intervals, p-values and R-Software hdi". En: *Statistical Science* 30.4, págs. 533-558.

Fan, J., Liu, H., Ning, Y. y Zou, H. (2017). "High dimensional semiparametric latent graphical model for mixed data". En: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.2, págs. 405-421. ISSN: 1467-9868.

Friedman, J., Hastie, T. y Tibshirani, R. (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". En: *Journal of Statistical Software* 33.1, págs. 1-22. URL: http://www.jstatsoft.org/v33/i01/.

Friedman, J., Hastie, T. y Tibshirani, R. (2014). *glasso: Graphical lasso- estimation of Gaussian graphical models*. URL: https://CRAN.R-project.org/package=glasso.

Friedman, J., Trevor, H. y Tibshirani, R. (2008). "Sparse inverse covariance estimation with the graphical lasso". En: *Biostatistics* 9.3, pág. 432. eprint: /oup/backfile/Content_public/Journal/biostatistics/9/3/10.1093/biostatistics/kxm045/2/kxm045.pdf. URL: +%20http://dx.doi.org/10.1093/biostatistics/kxm045.

Frondana, I. M. (2016). "Model selection for discrete Markov random fields on graphs". Tesis doct. Instituto de Maatemática y Estadística de la Universidad de São Paulo.

Gales, Mark y Young, Steve (2007). "The Application of Hidden Markov Models in Speech Recognition". En: *Found. Trends Signal Process.* 1.3, págs. 195-304. ISSN: 1932-8346.

`gene expression - arabidopsis thaliana` (2004). Data file 1 de la versión online de Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. URL: `http://genomebiology.biomedcentral.com/articles/10.1186/gb-2004-5-11-r92`.

Giraud, C. (2014). *Introduction to High-Dimensional Statistics*.

Jalali, A., Johnson, C. C. y Ravikumar, P. (2011). "On Learning Discrete Graphical Models using Greedy Methods". En: *Advances in Neural Information Processing Systems 24*. Ed. por Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F. C. N. y Weinberger, K. Q., págs. 1935-1943. URL: `http://books.nips.cc/papers/files/nips24/NIPS2011%5C_1098.pdf`.

Koller, D. y Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Lauritzen, S. L. (1996). *Graphical models*.

Liu, H., Lafferty, J. D. y Wasserman, L. A. (2009). "The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs". En: *Journal of Machine Learning Research* 10, págs. 2295-2328. URL: `http://jmlr.org/papers/volume10/liu09a/liu09a.pdf`.

Loh, P. y Wainwright, M. J. (2013). "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses". En: *Ann. Statist.* 41.6, págs. 3022-3049. DOI: `10.1214/13-AOS1162`. URL: `http://dx.doi.org/10.1214/13-AOS1162`.

Meinshausen, N. (2008). "A note on the Lasso for Gaussian graphical model selection". En: *Statistics & Probability Letters* 78.7, págs. 880-884.

Meinshausen, N. y Bühlmann, P. (2006). "High-dimensional graphs and variable selection with the Lasso". En: *Ann. Statist.* 34.3, págs. 1436-1462. DOI: `10.1214/009053606000000281`. URL: `http://dx.doi.org/10.1214/009053606000000281`.

Meinshausen, N. y Bühlmann, P. (2010). "Stability selection". En: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.4, págs. 417-473. ISSN: 1467-9868. URL: `http://dx.doi.org/10.1111/j.1467-9868.2010.00740.x`.

`news` (2008). Conjunto de datos con las ocurrencias de 100 palabras en un conjunto de 16242 noticias. URL: `http://www.cs.toronto.edu/~roweis/data.html`.

Ravikumar, P., Raskutti, G., Wainwright, M. J. y Yu, B. (2008). "Model Selection in Gaussian Graphical Models: High-Dimensional Consistency of $l_1$-Regularized MLE". En:

Ravikumar, P., Wainwright, M. J., Raskutti, G. y Yu, B. (2011). "High-dimensional covariance estimation by minimizing $l_1$-penalized log-determinant divergence". En: *Electron. J. Statist.* 5, págs. 935-980. DOI: `10.1214/11-EJS631`. URL: `http://dx.doi.org/10.1214/11-EJS631`.

Smith, Noah A. (2011). *Linguistic Structure Prediction (Synthesis Lectures on Human Language Technologies)*. 1.ª ed. Morgan & Claypool Publishers.

Tan, V. Y. F., Anandkumar, A., Tong, L. y Willsky, A. S. (2011). "A Large-Deviation Analysis of the Maximum-Likelihood Learning of Markov Tree Structures". En: *IEEE Transactions on Information Theory* 57.3, págs. 1714-1735. ISSN: 0018-9448.

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". En: *Journal osf the Royal Statistics Society* 58.

Van Borkulo, C. D., Borsboom, D., Epskamp, S., Blanken, T. F., Boschloo, L., Schoevers, R. A. y Waldorp, L. J. (2014). "A new method for constructing networks from binary data". En: *Scientific Reports* 4, pág. 5918. ISSN: 2045-2322. DOI: 10.1038/srep05918.

Wainwright, M. J., Ravikumar, Pradeep y Lafferty, John D. (2006). "High-dimensional Graphical Model Selection Using $l_1$-regularized Logistic Regression". En: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Canada: MIT Press, págs. 1465-1472.

Wasserman, L. A. (2004). *All of Statistics*.

Wasserman, L. A., Kolar, M. y Rinaldo, A. (2014). "Berry-Esseen bounds for estimating undirected graphs". En: *Electron. J. Statist.* 8.1, págs. 1188-1224. DOI: 10.1214/14-EJS928. URL: http://dx.doi.org/10.1214/14-EJS928.

Wille, A. et al. (2004). "Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana." En: *Genome biology* 5.11, R92+. ISSN: 1474-760X. DOI: 10.1186/gb-2004-5-11-r92. URL: http://dx.doi.org/10.1186/gb-2004-5-11-r92.

Zhao, P. y Yu, B. (2006). "On Model Selection Consistency of Lasso". En: *J. Mach. Learn. Res.* 7, págs. 2541-2563. ISSN: 1532-4435.

Zhou, S., Rütimann, P., Xu, M. y Bühlmann, P. (2011). "High-dimensional Covariance Estimation Based On Gaussian Graphical Models". En: *J. Mach. Learn. Res.* 12, págs. 2975-3026. ISSN: 1532-4435.

Zou, H. (1996). "The Adaptive Lasso and Its Oracle Properties". En: *Journal of the Royal Statistics Society* 58.