EDA Bivariado Direccional Estadístico
Tesis
Que presenta

Paul Ramírez De la Cruz

26 jun 2009

Resumen

Los Algoritmos de Estimación de Distribución (EDAs por Estimation of Distribution Algorithms) son algoritmos de optimización evolutivos que representan un enfoque distinto, y de introducción más reciente, que los Algoritmos Genéticos y las Estrategias Evolutivas.

Existen distintas formas en las que un EDA puede aprender y muestrear de la distribución de los mejores individuos. Una de dichas formas consiste en aproximar la distribución multivariada a través solamente de distribuciones bivariadas. En esta Tesis se plantea un método de este tipo y se demuestra que es el mejor bosque de relaciones bivariadas para estimar una distribución multivariada con densidad acotada.

Una preocupación principal para mejorar el desempeño de los EDAs se refiere a evitar que converjan prematuramente por falta de diversidad. En este trabajo se propone un método direccional, que evita la convergencia prematura y mejora el desempeño del EDA, el cual está basado en un contraste estadístico multivariado de hipótesis y el uso del primer componente principal de los individuos de élite.

En los experimentos realizados en un conjunto estándar de funciones de prueba, se observa que el método propuesto iguala o supera el desempeño de métodos que utilizan aproximaciones más complejas a la distribución conjunta de los individuos de élite.

Índice general

Índice de figuras

Lista de Algoritmos

Capítulo 1

Introducción

En distintos ámbitos de la actividad humana se pueden encontrar situaciones en las que se utilizan recursos para alcanzar ciertos fines. En tales situaciones, dado que por lo general los recursos disponibles son limitados, existen ocasiones en las cuales resulta de interés distribuir dichos recursos "de la mejor manera posible".Un ejemplo típico de lo recién expuesto lo encontramos en la manufactura de productos donde se utilizan materias primas para la generación de algún bien, mediante algún tipo de transformación. En tal circunstancia, puede interesar encontrar la combinación de cantidades de materias primas que permitan producir el bien en cuestión con el menor desperdicio. También existen situaciones similares en la ingeniería, la medicina, la administración, la economía y otras áreas del conocimiento.

A las herramientas computacionales y matemáticas que permiten resolver problemas como los planteados en el párrafo anterior, se les denomina *Métodos de Optimización* (véase [34]). Estos pueden clasificarse en distintas categorías. Una clasificación básica que se puede considerar es aquella que separa a los métodos de optimización en determinísticos y probabilísticos.

Una de las principales características de los métodos de optimización determinísticos es que el modelo del problema a optimizar puede ser completamente especificado desde un inicio. Cuando no es este el caso, resultan de utilidad los métodos de optimización probabilísticos, los cuales utilizan conceptos derivados de la teoría de la probabilidad, como funciones de masa o de densidad de variables aleatorias o generación de números aleatorios, para solucionar el problema de obtención del óptimo (véase, por ejemplo, [43, p. 4 y ss.; p. 538]).

Algunos métodos de optimización probabilística introducen el uso de conceptos básicos de la teoría Darwiniana de la evolución de los seres vivos, por tanto trabajan con grupos o *poblaciones* de soluciones candidatas, a las cuales se les denomina *individuos*. Tales individuos son mejorados o *evolucionados* de manera iterativa, y cada iteración recibe el nombre de *generación*. A tales métodos se les denomina Algoritmos Evolutivos [4].

Entre los Algoritmos Evolutivos existen distintos enfoques, por ejemplo el Algoritmo Genético (AG o GA por *Genetic Algorithm*), uno de los primeros modelos de este

tipo, el cual fue introducido durante la segunda mitad del siglo XX (véase por ejemplo [20] o [30]). En el AG los individuos se representan como una codificación en cadenas binarias de las soluciones candidatas. Los AGs utilizan operadores de cruza (mezcla de subcadenas de individuos) y mutación (inversión de algunos bits del individuo) para producir la evolución que conduzca finalmente a la mejor solución obtenible mediante tal procedimiento (véase [30]). En la aplicación de dichos operadores se utilizan algunos conceptos básicos de probabilidad, como la generación de números aleatorios.

Un enfoque distinto al planteado por el AG lo constituyen las Estrategias Evolutivas (EE o ES por *Evolution Strategies*), surgidas casi al mismo tiempo que aquél, que representan directamente las soluciones como vectores decimales en la región de búsqueda. En comparación con el AG, las EE van un paso más allá en la incorporación de conceptos probabilísticos a los algoritmos de optimización, puesto que asumen que cada individuo es la media de una distribución normal multivariada y propician la mutación del individuo al permitirle desplazarse en cada dimensión involucrada una distancia igual a un múltiplo de la desviación estándar en dicha dirección [6]. Originalmente, las EE no consideraban el empleo de la cruza entre individuos, aunque versiones posteriores ya hacen uso de ella. La adición de variables que controlan el ángulo de los ejes del individuo (que puede pensarse como un hiperelipsoide de equidensidades) con respecto a los ejes coordenados, o bien la directa consideración de la existencia de correlación entre las variables de decisión proporciona una mayor flexibilidad a su desplazamiento por la región de búsqueda y una mejor convergencia hacia el óptimo [19].

Un tercer enfoque de muy reciente surgimiento entre los algoritmos evolutivos lo representan los Algoritmos de Estimación de Distribución (AED o EDA por *Estimation of Distribution Algorithm*), originalmente planteados en términos de variables binarias y posteriormente también con variables continuas (principalmente normales o gaussianas). Los AED trasponen el eje de atención para centrarlo no en los individuos, sino en las variables con las que dichos individuos están formados y asumen que tales variables siguen una distribución de probabilidad conjunta que puede estimarse a partir de tales soluciones candidatas. Una vez realizada la estimación, la generación siguiente se obtiene muestreando de la distribución. La iteración de este procedimiento conduce a la solución del problema planteado [25].

En este trabajo de Tesis se presenta un Algoritmo de Estimación de Distribución para variables continuas, basado en una distribución gaussiana multivariada, la cual aproxima utilizando solamente relaciones bivariadas estadísticamente significativas. Para ello se demuestra el resultado de Chow y Liu [14], originalmente presentado para variables binarias, generalizándolo para variables continuas con densidad acotada. Con base en dicho resultado se demuestra que el método presentado es el mejor bosque de relaciones bivariadas para aproximar una distribución multivariada. También se presenta una mejora del método recién descrito, agregándole un componente direccional basado en un contraste estadístico multivariado de hipótesis y el empleo del primer componente principal de los individuos de élite. Esta mejora representa una generalización estadística multivariada del método SDR-AVS-IDEA [12].

Se realizaron experimentos que muestran que el método presentado iguala o supera

los resultados obtenidos mediante otros algoritmos que utilizan relaciones más complejas entre variables. Estos experimentos se realizaron sobre un conjunto de funciones de prueba estándar en dimensiones 10 y 50.

El resto del documento está organizado de la siguiente forma: en el Capítulo 2 se habla sobre Algoritmos de Estimación de Distribución Discretos, en particular aquellos que utilizan relaciones bivariadas para aproximar una función de masa de probabilidad conjunta multivariada; en el Capítulo 3 se revisan las versiones continuas de los EDAs discretos presentados en el Cap. 2, junto con los principales resultados de teoría estadística referentes a las distribuciones normal univariada, normal bivariada y normal multivariada que son necesarios para la comprensión del método que se propone. En el Capítulo 4 se presenta el Algoritmo de Estimación de Distribución basado en la Aproximación de una Distribución Normal Multivariada mediante un Bosque de Dependencias Bivariadas, ADNMB, tema central de este trabajo de Tesis; y, finalmente, en el Capítulo 5 se habla sobre las modificaciones aplicadas al ADNMB para incorporarle un elemento direccional que guíe la búsqueda del óptimo, basado en resultados de estadística multivariada.

Capítulo 2

EDAs discretos

El enfoque presentado en este trabajo de tesis está basado en *Algoritmos de Estimación de Distribución*, AED, o bien EDA por *Estimation of Distribution Algorithms*, los cuales son algoritmos de optimización evolutiva que representan un paradigma distinto al planteado por otras metaheurísticas tales como los *Algoritmos Genéticos* (AG o GA por *Genetic Algorithms*) o las *Estrategias Evolutivas* (EE o ES, *Evolution Strategies*).

Los *Algoritmos Genéticos*, tal como fueron definidos originalmente por John Holland [20], se basan en la codificación-decodificación binaria de elementos en el espacio de búsqueda que se constituyen como soluciones candidatas cuya mejora progresiva se basa en operadores de cruza y mutación [30]. Las *Estrategias Evolutivas* no requieren de la parte de codificación, sino que directamente representan a los individuos como números reales. La evolución de las soluciones también está basada en métodos de cruza y mutación. A las soluciones se les agregan variables de control que regulan el alcance y orientación de los individuos que constituyen las nuevas generaciones [10].

Los *Algoritmos de Estimación de Distribución*, por su parte, no toman en cuenta las operaciones de cruza y mutación, sino que extraen información con respecto a la estructura probabilística de las mejores soluciones obtenidas en cada iteración y, a partir de dicho modelo de probabilidad estimado, obtienen una muestra de soluciones que forma la nueva generación [25]. Inicialmente se propusieron EDAs basados en variables binarias [32], pero en años recientes ha ido en aumento el interés por los EDAs basados en variables continuas [25]. A continuación se presenta una recopilación de los EDAs discretos en los que se inspira el método presentado en este trabajo de Tesis.

En la exposición de los distintos enfoques en EDAs para variables binarias que se presenta a continuación, se supondrá que:

- El espacio de búsqueda $\mathcal{B}$ es $n-dimensional$ sobre variables binarias $X_1, X_2, ..., X_n$
- Un individuo o solución sobre dicho espacio tiene la forma

$$x = (x_1, x_2, ..., x_n)$$

donde x_j es el valor observado de la variable X_j para $j = 1, ..., n$. Un individuo también puede representarse como

$$x = (x_1, x_2, ..., x_n; a)$$

si se incluye el valor de aptitud, a, del individuo, la cual se mide a través de la función $g : \mathcal{B} \to \mathbb{R}$, es decir, $g(x) = a$.

- Interesa minimizar el valor de la función de aptitud, lo cual no representa una limitante en cuanto al tipo de problemas que se puede resolver puesto que

$$\text{máx}\, g \equiv \text{mín}\, -g$$

El Algoritmo 2.1 muestra el esquema general de un EDA básico. Al inicio de cada iteración o generación de un EDA se cuenta con una población de m soluciones candidatas, por ejemplo

$$pob := Población = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix},$$

o

$$pob := Población = \left[\begin{array}{c|c} x_1 & a_1 \\ x_2 & a_2 \\ \vdots & \vdots \\ x_m & a_m \end{array}\right] = \left[\begin{array}{cccc|c} x_{11} & x_{12} & \dots & x_{1n} & a_1 \\ x_{21} & x_{22} & \dots & x_{2n} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} & a_m \end{array}\right]$$

o bien

$$pob := Población = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} & a_1 \\ x_{21} & x_{22} & \dots & x_{2n} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} & a_m \end{bmatrix}$$

en donde x_{ij} y a_i son la $j-ésima$ coordenada y la aptitud de la $i-ésima$ solución candidata, respectivamente, con $i = 1, 2, ..., m$, $j = 1, 2, ..., n$. Se supondrá, por simplicidad de la notación, que se ha ordenado los individuos de manera descendente por aptitud.

Se seleccionan entonces los $s \leq m$ individuos más aptos, que fungirán como "padres" de la siguiente generación, y se tiene:

$$Padres = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} & a_1 \\ x_{21} & x_{22} & \dots & x_{2n} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{s1} & x_{s2} & \dots & x_{sn} & a_s \end{bmatrix}$$

A partir de esta selección de padres se obtiene una nueva población de tamaño m de soluciones candidatas; pero a diferencia del Algoritmo Genético o las Estrategias Evolutivas en donde esto se lograría mediante la aplicación de operadores de cruza y mutación, en un Algoritmo de Estimación de Distribución se asume que los individuos que se ha seleccionado en esta generación son realizaciones de un vector aleatorio que sigue cierta distribución de probabilidad. Por tanto, interesa estimar la función de masa de probabilidad (*fmp*) conjunta de las X_j, $j = 1, ..., n$:

$$f_{X_1 X_2 ... X_n}(x_1, x_2, ..., x_n)$$

y después generar mediante simulación una nueva población tomando una muestra de la distribución estimada.

Algoritmo 2.1 De Estimación de Distribución (EDA) Básico

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos de los que se encuentran en pob
4: Estime la distribución de probabilidad de los individuos más aptos, P, mediante $\widehat{P}$, calculada a partir de los individuos seleccionados
5: $hijos$ = Simule m individuos distribuidos según $\widehat{P}$
6: $pob = hijos$
7: **end while**

2.1. Algoritmo de Distribución Marginal Unvariada (UMDA)

El *Algoritmo de Distribución Marginal Univariada* (ADMU o UMDA, *Univariate Marginal Distribution Algorithm*) es uno de los más básicos en el ambiente de los algoritmos de estimación de distribución [32], [26], junto con el algoritmo PBIL (APBP, *Aprendizaje Progresivo Basado en Poblaciones* o *Population Based Incremental Learning*) [7]. A continuación se explica cuáles son las ideas básicas de UMDA.

La forma más básica de modelar $f_{X_1 ... X_n}$ es suponer que las X_j son todas independientes entre sí y que, por lo tanto, la *fmp* conjunta será igual al producto de las marginales:

$$f_{X_1 ... X_n}(x_1, ..., x_n) = \prod_{j=1}^{n} f_{Xj}(x_j)$$

La suposición de que las X_j son independientes dos a dos permite generar la nueva población "por columnas", en cualquier orden.

Para estimar la función de masa de probabilidad marginal de X_j, considérese la $j-$*ésima* columna de la matriz que representa a la población, que contiene los valores observados y seleccionados de dicha variable: $x_{1j}, x_{2j}, ..., x_{sj}$. Dado que X_j es una variable binaria, entonces puede suponerse que se distribuye Bernoulli con parámetro p_j, es decir

$$X_j \sim Ber\left(p_j\right), \quad j = 1, ..., n$$

así que su función de masa de probabilidad es (véase [31, p. 87]):

$$f_{X_j}\left(x_j\right) = p_j\left(1-p_j\right)^{1-x_j}; \qquad x_j = 0, 1; \qquad j = 1, ..., n$$

Puede verse que el estimador de máxima verosimilitud de p_j es ([31, p. 280])

$$\widehat{p}_j = \frac{1}{s}\sum_{i=1}^{s} x_{ij}; \qquad j = 1, ..., n \tag{2.1}$$

Dado lo anterior, la *fmp* conjunta de $X_1, X_2, ..., X_n$ es entonces

$$f_{X_1 ... X_n}\left(x_1, ..., x_n\right) = \prod_{j=1}^{n} p_j\left(1-p_j\right)^{1-x_j}; \quad x_j = 0, 1, \quad j = 1, ..., n$$

la cual puede estimarse mediante

$$\widehat{f}_{X_1 ... X_n}\left(x_1, ..., x_n\right) = \prod_{j=1}^{n} \widehat{p}_j\left(1-\widehat{p}_j\right)^{1-x_j}; \quad x_j = 0, 1, \quad j = 1, ..., n$$

Como ya se anotó, dado que se ha supuesto que las variables son independientes, es posible simular cada una de ellas por separado. En consecuencia, para producir la siguiente generación de soluciones candidatas lo que se requiere es estimar p_j mediante $\widehat{p}_j$ según se indica en la expresión (2.1) y entonces simular m números binarios que se distribuyan $Ber\left(\widehat{p}_j\right)$, por ejemplo

$$\begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{pmatrix}$$

Repitiendo el procedimiento anterior para cada $j = 1, 2, ..., n$ y luego yuxtaponiendo las columnas así obtenidas, habrá una nueva población:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mj} & \cdots & x_{mn} \end{bmatrix}.$$

En el Algoritmo 2.2 se presenta el UMDA para variables binarias.

La Figura 2.1 muestra la forma que tiene un grafo de UMDA para aproximar la distribución conjunta $p\left(X_1, X_2, X_3, X_4, X_5, X_6\right)$, que en la representación mostrada sería

$$\widehat{p}\left(X_1, X_2, X_3, X_4, X_5, X_6\right) = \widehat{p}\left(X_1\right)\widehat{p}\left(X_2\right)\widehat{p}\left(X_3\right)\widehat{p}\left(X_4\right)\widehat{p}\left(X_5\right)\widehat{p}\left(X_6\right).$$

Algoritmo 2.2 De Distribución Marginal Univariada, UMDA

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: **for** $j = 1, \ldots, n$ **do**
5: Estime la proporción de observaciones iguales a 1 en la variable X_j,

$$\hat{p}_j = \frac{1}{s} \sum_{i=1}^{s} x_{ij}$$

6: Simule m valores binarios distribuidos $Ber\,(\hat{p}_i)$
7: **end for**
8: $hijos$ = Los m individuos recién simulados
9: $pob_{previa} = padres \cup hijos$
10: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
11: **end while**

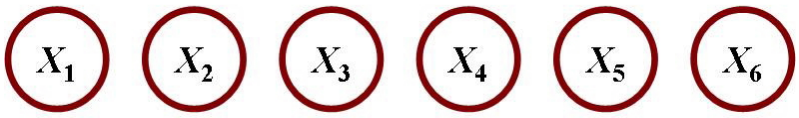


Figura 2.1: Grafo de una distribución conjunta aproximada mediante UMDA

2.2. Algoritmo MIMIC

Después de la suposición de independencia dos a dos de todas las variables, el siguiente nivel de complejidad en los métodos evolutivos que buscan aproximar una distribución conjunta se basa en la consideración de que las relaciones existentes se dan solamente entre pares de variables [24]. El Algoritmo de *Maximización de Información Mutua para Conglomeración de Información de Entrada* (MIMCIE o MIMIC, *Mutual Information Maximization for Input Clustering*) supone que las relaciones entre variables se dan por pares de manera encadenada o secuencial, según se describe en el artículo de 1997 de De Bonet, Isbell y Viola [11]. En el artículo, también se menciona que aunque MIMIC se presenta como un método que trabaja con cadenas binarias, puede reformularse fácilmente para utilizarse con cualquier alfabeto finito.

Este algoritmo forma un grafo con mínima entropía. La entropía es una medida de incertidumbre utilizada en Teoría de la Información. Las siguientes definciones se tomaron de [40]:

Definición 1 (Entropía discreta de Shannon) *Sea X una variable aleatoria discreta con k posibles resultados $\{x_1, x_2, ..., x_k\}$ y sea p_X su función de masa de probabilidad. La información que proporciona el conocimiento de que X ha tomado el valor x_i es $-\log p_{X_i}(x_i)$. La entropía de Shannon (o medida de incertidumbre) de X se denota por $H(X)$ y se define como el valor esperado de la información:*

$$H(X) = -\sum_{i=1}^{k} p_X(x_i) \log p_X(x_i).$$

Si se toma el logaritmo de base 2, entonces el valor de la entropía se da en *bits*. Si se utiliza el logaritmo natural, entonces se dice que la entropía se mide en *nats* [17, p. 18]. Durante todo el desarrollo de este trabajo se utilizó el logaritmo natural.

Definición 2 (Entropía conjunta discreta) *Sean X y Y variables aleatorias discretas que asumen valores $\{x_1, ..., x_k\}$ y $\{y_1, ..., y_l\}$, respectivamente. Sea p_{XY} la fmp conjunta de X y Y. La medida conjunta de incertidumbre, o entropía conjunta, de X y Y es*

$$H(X,Y) = -\sum_{i=1}^{k} \sum_{j=1}^{l} p_{XY}(x_i, y_j) \log p_{XY}(x_i, y_j). \tag{2.2}$$

Sean p_X la fmp de X, y p_Y la fmp de Y. Dado que estas fmp marginales de X y Y se pueden expresar respectivamente como

$$p_X(x_i) = \sum_{j=1}^{l} p_{XY}(x_i, y_j); \quad i = 1, ..., k$$

y

$$p_Y(y_j) = \sum_{i=1}^{k} p_{XY}(x_i, y_j); \quad j = 1, ..., l$$

entonces se hace la siguiente

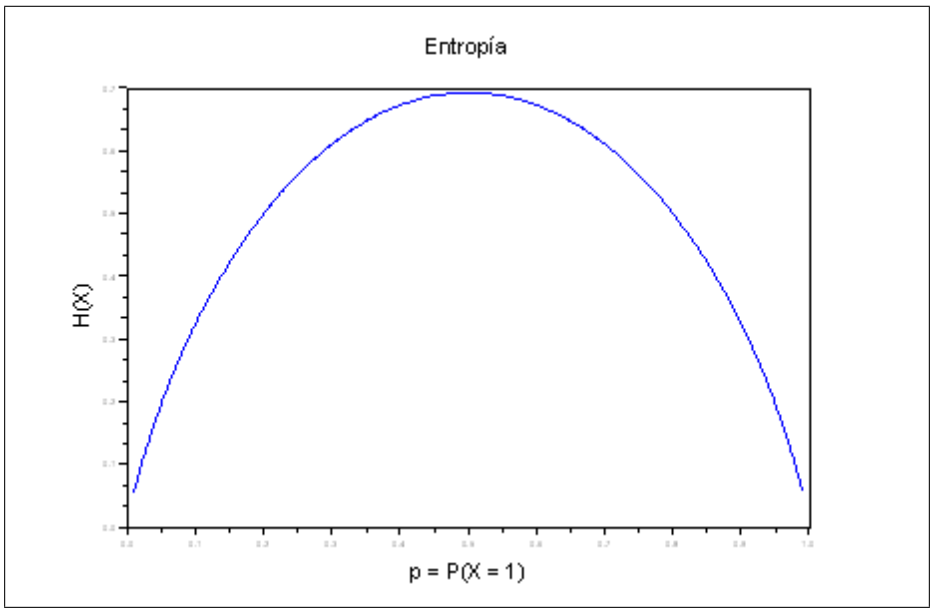


Figura 2.2: Gráfica de $H(X)$ con $X \sim Ber(p)$ y $p \in (0,1)$

Afirmación 3 *Las entropías marginales de X y Y se pueden expresar como*

$$H(X) = -\sum_{i=1}^{k}\sum_{j=1}^{l} p_{XY}(x_i, y_j) \log p_X(x_i).$$

y

$$H(Y) = -\sum_{j=1}^{l}\sum_{i=1}^{k} p_{XY}(x_i, y_j) \log p_Y(y_j).$$

Prueba. *Se sigue directamente de las definiciones de fmp marginal y de entropía conjunta.* ■

Definición 4 (Entropía condicional discreta) *Sean X y Y variables aleatorias discretas que asumen valores $\{x_1, ..., x_k\}$ y $\{y_1, ..., y_l\}$, respectivamente. La entropía condicional de Y dado $X = x_i$ es*

$$H(Y \mid X = x_i) = -\sum_{j=1}^{l} p_{Y|X}(y_j|x_i) \log p_{Y|X}(y_j|x_i).$$

Definición 5 (Divergencia de Kullback-Leibler) *Es una medida de disimilitud entre dos distribuciones. Si X es una variable aleatoria discreta definida en $\mathcal{D}_X$, p_X es la función de masa de probabilidad de X y $\widehat{p}_X$ es una distribución que aproxima a p_X, entonces la Divergencia de Kullback-Leibler entre p_X y $\widehat{p}_X$, $D(p_X \,||\, \widehat{p}_X)$ se define como*

$$D(p_X \,||\, \widehat{p}_X) = \sum_{x \in \mathcal{D}_X} p_X(x) \log \frac{p_X(x)}{\widehat{p}_X(x)}.$$

Observación 6 *Note que haciendo un poco de álgebra con la expresión para la Divergencia de Kullback-Leibler resulta*

$$\begin{aligned} D\left(p_X \,||\, \widehat{p}_X\right) &= \sum_{x \in \mathcal{D}_X} p_X(x) \left[\log p_X(x) - \log \widehat{p}_X(x)\right] \\ &= E_{p_X}\left(\log p_X\right) - E_{p_X}\left(\log \widehat{p}_X\right) = -H(X) + \widehat{H}(X) \end{aligned}$$

o bien

$$D\left(p_X \,||\, \widehat{p}_X\right) = \widehat{H}(X) - H(X). \tag{2.3}$$

Observación 7 $D\left(p_X \,||\, \widehat{p}_X\right) \geq 0$, *y la igualdad se cumple si* $\widehat{p}_X \equiv p_X$ *(véase la demostración, por ejemplo, en [40]).*

Por la definición de probabilidad condicional y como se menciona en el artículo de De Bonet *et al.* [11], la distribución de probabilidad conjunta $p_{X_1 X_2 \ldots X_n}$ de n variables aleatorias $X_1, X_2, ..., X_n$ se puede expresar en términos de probabilidades condicionales como:

$$\begin{aligned} p_{X_1 X_2 \ldots X_n}(X_1, X_2, ..., X_n) &= p(X_1 | X_2, X_3, ..., X_n)\, p(X_2 | X_3, X_4, ..., X_n) \\ &\quad ... p(X_{n-1} | X_n)\, p(X_n). \end{aligned}$$

Considerando la restricción de utilizar solamente probabilidades marginales y condicionales basadas en pares de variables que se den de manera secuencial, se busca obtener una aproximación a la distribución conjunta a través de una permutación de los números naturales $1, 2, ..., n$ dada por $\eta = \{i_1, i_2, ..., i_n\}$. Con base en η se consideran solamente distribuciones conjuntas del tipo

$$\begin{aligned} p_{X_{i1} X_{i2} \ldots X_{in}}(X_{i_1}, X_{i_2}, ..., X_{i_n}) &= p(X_{i_1} | X_{i_2})\, p(X_{i_2} | X_{i_3}) \\ &\quad ... p\left(X_{i_{n-1}} | X_{i_n}\right) p(X_{i_n}). \end{aligned} \tag{2.4}$$

Si se aproxima $p_{X_1 X_2 \ldots X_n}$ por una factorización como la dada por la expresión (2.4), entonces aplicando (2.3) puede verse que

$$\begin{aligned} D\left(p_X \,||\, \widehat{p}_X\right) &= \widehat{H}(X) - H(X) = -H(X) + \widehat{H}(X) \\ &\leq -H(X) + H(X_{i_1} | X_{i_2}) + H(X_{i_2} | X_{i_3}) + \\ &\quad \cdots + H\left(X_{i_{n-1}} | X_{i_2}\right) + H(X_{i_n}). \end{aligned}$$

Como $H(X)$ no depende de la permutación η, la función que se minimiza mediante el algoritmo MIMIC es

$$J(\eta, X_1, ..., X_n) = H(X_{i_1} | X_{i_2}) + H(X_{i_2} | X_{i_3}) + \cdots + H\left(X_{i_{n-1}} | X_{i_2}\right) + H(X_{i_n}).$$

Para la selección de la permutación adecuada, se utiliza un algoritmo *voraz* o *glotón* (*greedy*, en Inglés) que encuentra:

Algoritmo 2.3 Mutual Information Maximization for Input Clustering

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: Sean $k = 0$ e $I = \emptyset$
3: **while** no se cumpla el criterio de finalización **do**
4: $padres$ = Seleccione los $s \leq m$ individuos más aptos
5: Coloque como raíz del grafo a la variable con la mínima entropía empírica. Es decir, sea

$$X_{i_1} = \underset{j=1,\ldots,n}{\arg\min} \widehat{H}\left(X_j\right)$$

6: Haga $I = \{i_1\}$
7: Sea $i = i_1$
8: **for** $l = 2, \ldots, n$ **do**
9: $j_l = \underset{j=1,\ldots,n;\, j \neq i}{\arg\min} \widehat{H}\left(X_j \mid X_i\right)$
10: $I = I \cup \{j_l\}$
11: $i = j_l$
12: **end for**
13: $hijos$ = Genere aleatoriamente una población de tamaño m a partir de la distribución conjunta estimada:

$$\widehat{p}\left(X_1, X_2, \ldots, X_n\right) = \widehat{p}\left(X_{i_1}\right) \widehat{p}\left(X_{i_2} \mid X_{i_1}\right) \widehat{p}\left(X_{i_3} \mid X_{i_2}\right) \ldots \widehat{p}\left(X_{i_n} \mid X_{i_{n-1}}\right)$$

14: $pob_{previa} = padres \cup hijos$
15: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
16: **end while**

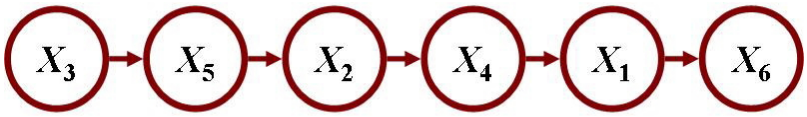


Figura 2.3: Forma típica de un grafo obtenido mediante MIMIC para aproximar una distribución conjunta

1. La variable X_j con la mínima entropía.

2. Entre las X_j que aún no han sido procesadas, aquella que tiene la mínima entropía dada la más recientemente seleccionada.

La Figura 2.3 muestra la forma que tiene un grafo de MIMIC para aproximar la distribución conjunta $p(X_1, X_2, X_3, X_4, X_5, X_6)$, que en la representación mostrada sería

$$\widehat{p}(X_1, X_2, X_3, X_4, X_5, X_6) = \widehat{p}(X_3)\,\widehat{p}(X_5|X_3)\,\widehat{p}(X_2|X_5)\,\widehat{p}(X_4|X_2)\,\widehat{p}(X_1|X_4)\,\widehat{p}(X_6|X_1)\,.$$

Observación 8 *El algoritmo MIMIC considera una manera óptima de obtener la variable raíz, aunque limita mucho la forma de construir el grafo de relaciones, ya que asume que todas las dependencias entre variables se dan en forma de "cadena". Esta suposición se mejora, como ser verá, en el siguiente algoritmo, denominado BMDA.*

2.3. Algoritmo de Distribución Marginal Bivariada (BMDA)

El *Algoritmo de Distribución Marginal Bivariada* (ADMA o BMDA por *Bivariate Marginal Distribution Algorithm*) es un EDA discreto que considera relaciones bivariadas. La estimación de la distribución conjunta de las variables se realiza mediante un grafo acíclico dirigido con forma de "bosque", una estructura en donde hay uno o más árboles que no están conectados entre sí y en la que se incluyen solamente relaciones que son estadísticamente significativas.

La estimación de la distribución conjunta de las mejores soluciones se hace con base en una hipermatriz P, la cual, para variables binarias, es de tamaño $n \times n \times 4$. La proyección de P sobre sus dos primeras dimensiones, $P_{n\times n}$, es una matriz que considera todas las combinaciones posibles de pares de variables:

$$P_{n\times n} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix}$$

donde P_{ij} es la distribución conjunta de X_i y X_j para $i, j \in \{1, 2, ..., n\}$; es decir que considerando la dimensión adicional de P se tiene que

$$\begin{aligned} P_{ij} &= (P(X_i = 0, X_j = 0), P(X_i = 0, X_j = 1) \\ &\quad P(X_i = 1, X_j = 0), P(X_i = 1, X_j = 1)) \\ i, j &= 1, 2, ..., n. \end{aligned}$$

lo cual se denotará por

$$P_{ij} := (P_{ij00}, P_{ij01}, P_{ij10}, P_{ij11}); \quad i, j = 1, 2, ..., n$$

Nétese que la distribución marginal de las X_j, $j = 1, 2, .., n$, está dada por

$$p := diag(P_{n\times n})$$

si bien se trata de información redundante, porque para cada $i = 1, 2, ..., n$

$$P_{ii} = (P(X_i = 0), 0, 0, P(X_i = 1))$$

También obsérvese que $P_{n\times n}$ no es exactamente simétrica, porque, por ejemplo

$$\begin{aligned} P_{12} &= (P(X_1 = 0, X_2 = 0), P(X_1 = 0, X_2 = 1), \\ &\quad P(X_1 = 1, X_2 = 0), P(X_1 = 1, X_2 = 1)) \end{aligned}$$

mientras que

$$\begin{aligned} P_{21} &= (P(X_2 = 0, X_1 = 0), P(X_2 = 0, X_1 = 1), \\ &\quad P(X_2 = 1, X_1 = 0), P(X_2 = 1, X_1 = 1)); \end{aligned}$$

pero la información que proporciona su parte triangular inferior es equivalente a la que da su parte triangular superior.

En la construcción del grafo de dependencias solamente se tiene en cuenta a las relaciones entre pares de variables que son estadísticamente significativas. Por tanto en la construcción del grafo de dependencias solamente se tiene en cuenta las celdas de $P_{n\times n}$ para las cuales X_i y X_j no son independientes. A fin de establecer dicha situación, se hace uso de un contraste de hipótesis de independencia basado en el estadístico χ^2 (véase [35]).

Interesa poner a prueba la hipótesis de que las clasificaciones X_i y X_j son independientes, es decir que para todo $x_i, x_j \in \{0, 1\}$ [13, p. 22]

$$P(X_i = x_i \,|\, X_j = x_j) = P(X_i = x_i)$$

y

$$P\left(X_j = x_j \,|\, X_i = x_i\right) = P\left(X_j = x_j\right),$$

relaciones que se cumplen si

$$\begin{aligned} P\left(\{X_i = x_i\} \cap \{X_j = x_j\}\right) &= P\left(X_i = x_i\right) P\left(X_j = x_j\right) \\ \forall x_i, x_j &\in \{0,1\} \end{aligned}$$

Recuérdese que el vector $p \equiv diag\left(P\right)$ contiene las probabilidades marginales para las $X_j,\ j = 1, ..., n$, y que la parte triangular superior (o inferior) de la hipermatriz P contiene las probabilidades conjuntas, por tanto se tiene que la hipótesis nula se puede plantear en términos de

$$\begin{gathered} H_0 : P_{ij00} = \left(1 - p_i\right)\left(1 - p_j\right),\ P_{ij01} = \left(1 - p_i\right) p_j, \\ P_{ij10} = p_i\left(1 - p_j\right),\ P_{ij11} = p_i p_j \\ H_a : \text{Las variables no son independientes} \end{gathered}$$

Para realizar dicha prueba se hará uso del estadístico χ^2_{Calc}, que se define a continuación. Note que una vez observados los valores de X_i y X_j, es posible formar la siguiente tabla de contingencia:

	X_j		
X_i	0	1	Total
0	o_{00}	o_{01}	$o_{0\cdot} = o_{00} + o_{01}$
1	o_{10}	o_{11}	$o_{1\cdot} = o_{10} + o_{11}$
Total	$o_{\cdot 0}$	$o_{\cdot 1}$	$o_{\cdot\cdot} = o_{0\cdot} + o_{1\cdot} = o_{\cdot 0} + o_{\cdot 1}$

donde, para cada $i, j = 1, 2, ..., n$ el número o_{ij} es la cantidad observada de soluciones candidatas en las que $X_i = x_i$ y $X_j = x_j$, con $x_i, x_j \in \{0, 1\}$.

El planteamiento de Pelikan y Muhlenbein [35] recurre a la aproximación de la distribucion binomial mediante la distribución normal, por lo cual hace uso del estadístico

$$\begin{aligned} \chi^2_{Calc} &= \sum_{i,j} \frac{\left[o_{ij} - e_{ij}\right]^2}{e_{ij}} \\ \chi^2_{Calc} &= \sum_{i,j} \frac{\left[o_{\cdot\cdot} p_{ij} - o_{\cdot\cdot} p_i p_j\right]^2}{o_{\cdot\cdot} p_i p_j} \end{aligned}$$

el cual se distribuye aproximadamente χ^2 con (no. de renglones $- 1$) (no. de columnas $- 1$) $= (2 - 1)(2 - 1) = 1$ grado de libertad. Por tanto, se rechaza H_0 al nivel α si $\chi^2_{Calc} > \chi^2_{1,\alpha} \equiv \chi^2_{Tabla}$. Si se emplea el valor de $\alpha = 0.05$, entonces se tiene que $\chi^2_{Tabla} \equiv \chi^2_{1,\alpha} = \chi^2_{1,0.05} = 3.84$.

Se resumen las ideas generales de BMDA en el Algoritmo 2.4.

Para la construcción del grafo de dependencias se requiere también de la siguiente

Notación 9 *Para el algoritmo de construcción del grafo de dependencias en BMDA:*

- *Vértice := Variable*

Algoritmo 2.4 De Distribución Marginal Bivariada, BMDA

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime P =Hipermatriz de distribuciones marginales y bivariadas
5: Obtenga el grafo de dependencias $G = (V, E, R)$ a partir de la hipermatriz P
6: $hijos$ = Genere aleatoriamente una población de tamaño m con base en G y P
7: $pob_{previa} = padres \cup hijos$
8: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
9: **end while**

- *Arista := Par de variables*
- *D := Conjunto de aristas tales que las variables participantes son dependientes entre sí*
- *V := Conjunto de vértices*
- *A := Conjunto de vértices sin procesar hasta el momento*
- *E := Conjunto de aristas en el grafo final*
- *R := Conjunto de vértices raíz del grafo de dependencias*

La construcción del grafo de dependencias se explica en el Algoritmo 2.5.

Algoritmo 2.5 Construcción del grafo de dependencias en BMDA

1: Sean $V = \{1, 2, \ldots, n\},\ A = V,\ E = \emptyset,\ R = \emptyset$
2: Sea $D = \{(i, j) \mid i, j \in V,\ i \neq j,\ \chi^2_{Calc} > \chi^2_{Tabla}\}$
3: **while** $A \neq \emptyset$ **do**
4: Sea v un vértice elegido aleatoriamente de A
5: Sea $R = R \cup \{v\}$
6: **if** existe al menos un vértice (v, v') tal que $v \in A$ y $v' \in V \setminus A$ **then**
7: Sea $v = \arg\max\{\chi^2_{vv'} \mid (v, v') \in D,\ v \in A \text{ y } v' \in V \setminus A\}$
8: $E = E \cup \{v\}$
9: $A = A \setminus \{v\}$
10: **end if**
11: **end while**

La Figura 2.4 muestra la forma que tiene un grafo de BMDA para aproximar la distribución conjunta $p(X_1, X_2, X_3, X_4, X_5, X_6)$, que en la representación mostrada sería

$$\widehat{p}(X_1, X_2, X_3, X_4, X_5, X_6) = \widehat{p}(X_1)\,\widehat{p}(X_2|X_1)\,\widehat{p}(X_6|X_1)\,\widehat{p}(X_5|X_1)\,\widehat{p}(X_3)\,\widehat{p}(X_4|X_3).$$

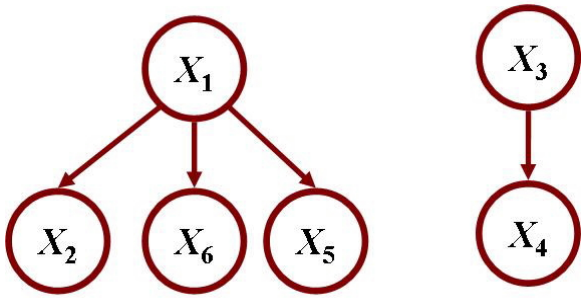


Figura 2.4: Forma típica de un grafo obtenido mediante BMDA para aproximar una distribución conjunta

La simulación de la siguiente generación de individuos a partir de P y G en BMDA implica el recorrido en *entreorden* de un árbol $n-ario$, es decir, una generalización del recorrido en *entreorden* de un árbol binario. El caso de BMDA se presenta en el Algoritmo 2.6.

Algoritmo 2.6 Recorrido del grafo de dependencias en BMDA

Input: padre $\equiv X_{j(i)}$, nodo $\equiv X_i$

1: **if** $X_i \in R$ **then** // El nodo es una raíz
2: Simule m valores distribuidos $Ber(\widehat{p}_{X_i})$
3: **else** // El nodo X_i está condicionado por $X_{j(i)}$
4: Calcule $\widehat{p}_0 = P(X_i = 1|X_{j(i)} = 0)$ y $\widehat{p}_1 = P(X_i = 1|X_{j(i)} = 1)$
5: Simule m valores distribuidos $Ber(\widehat{p}_0)$ o $Ber(\widehat{p}_1)$, según corresponda dado el valor observado de $X_{j(i)}$
6: **end if**
7: **while** X_i tenga hijos **do**
8: Ejecute el Algoritmo 2.6 con nodo, hijo
9: **end while**

Observación 10 *BMDA "falla" en la selección de la raíz inicial de cada árbol, puesto que no utiliza algún criterio de optimización, sino que realiza una selección aleatoria. El algoritmo propuesto en esta Tesis se insipira en BMDA, pero mejora dicha selección inicial al realizarla en una forma similar al método MIMIC (véase la sección 2.2)*

2.4. Árboles discretos de Chow y Liu

En su artículo de 1968, C. Chow y C. Liu propusieron un método para aproximar distribuciones conjuntas discretas mediante un árbol con dependencias bivariadas [14] cuya magnitud se evalúa a través de la *información mutua* entre las variables

participantes en tales dependencias. Los autores no dieron un nombre particular a su algoritmo, y simplemente lo denominan de *Aproximación de Distribuciones Discretas de Probabilidad mediante la construcción de un Árbol de Dependencias*. En esta Tesis, se le denominará a este método ADDPAD. A continuación se presentan las ideas principales de dicho método.

Dada una distribución conjunta de n variables discretas,

$$p\left(X_1, X_2, ..., X_n\right),$$

el algoritmo de Chow y Liu proporciona la mejor aproximación a p, según se demuestra en la Sección 2.5, utilizando una distribución marginal y $n-1$ distribuciones condicionales bivariadas, de la forma

$$\widehat{p}\left(X_1, X_2, ..., X_n\right) = p\left(X_{i_1}\right) p\left(X_{i_2}|X_{j(i_2)}\right) ... p\left(X_{i_n}|X_{j(i_n)}\right)$$

en donde $i_1, i_2, ..., i_n$ es una permutación de los índices $1, ..., n$; y $j\left(i_l\right) \in \{1, ..., n\} \setminus \{i_l\}$ es la variable que condiciona a i_l, para $l = 2, ..., n$. La factorización anterior se puede representar mediante un grafo dirigido acíclico en donde cada variable tiene su distribución condicionada a lo más por otra variable.

En el método descrito por Chow y Liu la generación de poblaciones se hace con base en una hipermatriz P, idéntica a la introducida en la discusión de BMDA (véase la sección 2.3).

A partir de las distribuciones marginales y bivariadas, y haciendo uso de las informaciones mutuas entre todos los pares de variables, el ADDPAD obtiene un árbol de dependencias, $G = (E, R)$, dado por una raíz, R, y un conjunto de aristas $E = \{(v, v') \mid v, v' = 1, 2, ..., n;\ v \neq v'\}$ que indica la relación de dependencias de las variables X_v y $X_{v'}$. El Algoritmo 2.7 resume el método descrito por Chow y Liu.

Algoritmo 2.7 De Chow y Liu para Aproximación de una Distribución Discreta de Probabilidad mediante la construcción de un Árbol de Dependencias

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime P = Hipermatriz de distribuciones marginales y bivariadas
5: Obtenga el grafo de dependencias $G = (E, R)$ a partir de la hipermatriz P
6: $hijos$ = Genere aleatoriamente una población de tamaño m con base en G y P
7: $pob_{previa} = padres \cup hijos$
8: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
9: **end while**

El algoritmo de aprendizaje del grafo de dependencias, utiliza el concepto de información mutua discreta.

Definición 11 (Información mutua discreta) *Sean X_i y X_j dos variables aleatorias discretas con recorridos D_i y D_j, respectivamente. La información mutua entre X_i y X_j, denotada $I(X_i, X_j)$, está dada por*

$$I(X_i, X_j) = \sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i, X_j = x_j)}{P(X_i = x_i) P(X_j = x_j)} \right) \quad (2.5)$$

Nótese que en la definición anterior X_i y X_j no están limitadas a ser binarias, sino que pueden tomar cualquier número finito de valores.

Afirmación 12 *Sean X_i y X_j dos variables aleatorias discretas con recorridos D_i y D_j, respectivamente; entonces*

$$I(X_i, X_j) \geq 0.$$

Prueba. *En esta prueba se sigue la idea presentada en [17, p. 21].*
Por definición tenemos que

$$I(X_i, X_j) = \sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i, X_j = x_j)}{P(X_i = x_i) P(X_j = x_j)} \right)$$

y por las propiedades de los logaritmos, $\log \frac{a}{b} = -\log \frac{b}{a}$, así que

$$\begin{aligned} I(X_i, X_j) &= -\sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i) P(X_j = x_j)}{P(X_i = x_i, X_j = x_j)} \right) \\ &\Leftrightarrow \quad -I(X_i, X_j) = \sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i) P(X_j = x_j)}{P(X_i = x_i, X_j = x_j)} \right) \end{aligned}$$

Ahora recuérdese la relación básica

$$\log x \leq x - 1$$

con lo que

$$\begin{aligned} & \sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \log \left(\frac{P(X_i = x_i) P(X_j = x_j)}{P(X_i = x_i, X_j = x_j)} \right) \\ \leq & \sum_{x_i \in D_i} \sum_{x_j \in D_j} P(X_i = x_i, X_j = x_j) \left(\frac{P(X_i = x_i) P(X_j = x_j)}{P(X_i = x_i, X_j = x_j)} - 1 \right) \\ = & \sum_{x_i \in D_i} \sum_{x_j \in D_j} \left[P(X_i = x_i, X_j = x_j) \frac{P(X_i = x_i) P(X_j = x_j)}{P(X_i = x_i, X_j = x_j)} - P(X_i = x_i, X_j = x_j) \right] \\ = & \sum_{x_i \in D_i} \sum_{x_j \in D_j} \left[P(X_i = x_i) P(X_j = x_j) - P(X_i = x_i, X_j = x_j) \right] \\ = & \ 1 - 1 = 0 \end{aligned}$$

De lo anterior resulta que $-I(X_i, X_j) \leq 0$, es decir,

$$I(X_i, X_j) \geq 0$$

■

Observación 13 *De las expresiones 2.2 y 2.5 obsérvese que se podría interpretar la información mutua entre X_i y X_j como la divergencia de Kullback-Leibler entre la distribución conjunta verdadera de las variables y la distribución conjunta que tendrían si fueran independientes.*

Para establecer la forma en que el método de Chow y Liu trabaja, se requiere de la siguiente

Definición 14 *Sea t un árbol de dependencia para n variables. Se dice que t es de peso máximo si para cualquier árbol t' para n variables se cumple que*

$$\sum_{i=1}^{n} I\left(x_i, x_{j(i)}\right) \geq \sum_{i=1}^{n} I\left(x_i, x_{j'(i)}\right).$$

Como ya vimos que $I(x_i, x_j) \geq 0$, se puede obtener un árbol de peso máximo obteniendo la combinación correcta de ramas con información máxima. En el método de Chow y Liu, se construye un árbol de peso máximo mediante un algoritmo glotón que elige en cada iteración de entre todas las ramas posibles aquella con máxima información.

La construcción del grafo de dependencias se explica mediante el Algoritmo 2.8 en donde se hace uso de la siguiente notación

Notación 15 *Para el algoritmo de construcción del grafo de dependencias en ADDPAD:*

- *Vértice := Variable*
- *Arista := Par de variables*
- *D := Conjunto de todas las aristas posibles*
- *V := Conjunto de vértices*
- *A := Conjunto de vértices sin procesar hasta el momento*
- *$I(v, v')$ = Valor de la información en la arista (v, v')*
- *E := Conjunto de aristas en el grafo final*
- *R := Vértice raíz del grafo de dependencias*

La Figura 2.5 muestra la forma que tiene un grafo del ADDPAD de Chow y Liu para aproximar la distribución conjunta $p(X_1, X_2, X_3, X_4, X_5, X_6)$, que en la representación mostrada sería

$$\widehat{p}(X_1, X_2, X_3, X_4, X_5, X_6) = \widehat{p}(X_4)\,\widehat{p}(X_5|X_4)\,\widehat{p}(X_2|X_5)\,\widehat{p}(X_1|X_5)\,\widehat{p}(X_6|X_1)\,\widehat{p}(X_3|X_4).$$

La simulación de la siguiente generación de individuos a partir de P y G en ADDPAD se hace de manera similar al caso de BMDA (véase el Algoritmo 2.6), con la diferencia de que en ADDPAD existe una única raíz. Se presenta la simulación de la siguiente generación en el método de Chow y Liu en el Algoritmo 2.9.

Algoritmo 2.8 Construcción del grafo de dependencias según Chow y Liu

1: Sean $V = \{1, 2, ..., n\}$, $A = V$, $E = \emptyset$, $B = A \times V$
2: **for** $i = 1, \ldots, n-1$ **do**
3: **if** $B \neq \emptyset$ **then**
4: Sea (v, v') aquella arista que satisfaga

$$(v, v') = \arg\max_{(u,w) \in B} I(u, w)$$

5: Sea $A = V \setminus \{v, v'\}$
6: Sea $B = A \times (V \setminus A)$
7: Sea $E = E \cup \{(v, v')\}$
8: **end if**
9: **end for**
10: Sea R el único vértice que no forma parte de alguna arista en E

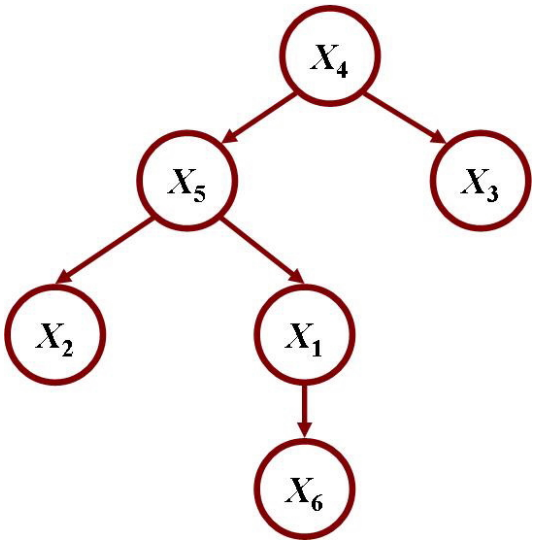


Figura 2.5: Forma típica de un grafo obtenido mediante el algoritmo de Chow y Liu para aproximar una distribución conjunta

Algoritmo 2.9 Recorrido del grafo de dependencias en el método de Chow y Liu

Input: padre $\equiv X_{j(i)}$, nodo $\equiv X_i$

1: **if** $X_i = R$ **then** // El nodo es la raíz
2: Simule m valores distribuidos $Ber(\widehat{p}_{X_i})$
3: **else** // El nodo X_i está condicionado por $X_{j(i)}$
4: Calcule $\widehat{p}_0 = P(X_i = 1|X_{j(i)} = 0)$ y $\widehat{p}_1 = P(X_i = 1|X_{j(i)} = 1)$
5: Simule m valores distribuidos $Ber(\widehat{p}_0)$ o $Ber(\widehat{p}_1)$, según corresponda dado el valor observado de $X_{j(i)}$
6: **end if**
7: **while** X_i tenga hijos **do**
8: Ejecute el Algoritmo 2.9 con nodo, hijo
9: **end while**

Observación 16 *Observe que existe una situación susceptible de mejora en ADDPAD: la forma de elegir el nodo raíz. Se afirma esto porque el método se basa en la maximización de la suma de las informaciones mutuas en las ramas; pero resulta que aquel que tal vez se pueda considerar como el nodo más importante, la raíz, se designa como la única variable que nunca fue seleccionada por maximizar $I(X_i, X_j)$ para algún par de nodos X_i y X_j. La selección de la raíz es mejor si se realiza del modo en que se hace en el algoritmo MIMIC, que se describió en la sección (2.2), tomando de entre las variables disponibles aquella con la mínima entropía.*

2.5. El mejor árbol de dependencia, caso discreto

El resultado más importante que se presenta en el artículo de Chow y Liu [14] es aquel que se refiere a que el árbol de dependencias definido por dicho algoritmo es el mejor árbol de dependencias bivariadas para aproximar una función de masa de probabilidad multivariada discreta. El documento mencionado esboza una demostración al respecto.A continuación se presenta una demostración del resultado en cuestión para variables discretas.

Afirmación 17 *La función de masa de probabilidad definida por el árbol t, $P_t(\mathbf{x})$, es una aproximación óptima a $P(\mathbf{x})$ si y sólo si t es un árbol de dependencia de peso máximo.*

Prueba. *Se puede considerar que la distribución P_t aproxima a P de manera óptima en el sentido de proporcionar el mínimo valor para la divergencia de Kullback-Liebler, si es la solución de*

$$\begin{aligned} P_t &= \arg\min_t D(P\|P_t) \\ &= \arg\min_t \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P_t(\mathbf{x})} \end{aligned}$$

Se demostrará que

$$\min_t \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P_t(\mathbf{x})}$$

equivale a

$$\max_t \sum_{i=1}^{n} I\left(x_i, x_{j(i)}\right)$$

Para encontrar el árbol que produce P_t tal que minimice[1] *$D(P\|P_t)$ obsérvese que*

$$\begin{aligned} D(P\|P_t) &= \sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{P_t(\mathbf{x})} \\ &= \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log \frac{P(x_1, ..., x_n)}{P_t(x_1, ..., x_n)} \end{aligned}$$

Luego

$$\begin{aligned} D(P\|P_t) &= \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \left[\log P(x_1, ..., x_n) - \log P_t(x_1, ..., x_n)\right] \\ &= \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log P(x_1, ..., x_n) \\ &\quad - \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log P_t(x_1, ..., x_n) \end{aligned}$$

$$\begin{aligned} D(P\|P_t) &= -H(X) \\ &\quad - \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log P_t(x_1, ..., x_n) \end{aligned}$$

$$D(P\|P_t) = -H(X) - \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log \prod_{i=1}^{n} P_{X_i}(x_i) P_{X_i|X_{j(i)}}\left(x_i \mid x_{j(i)}\right) \quad (2.6)$$

$$= -H(X) \quad (2.7)$$

$$- \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \log P(x_{i_1}) \quad (2.8)$$

$$- \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log P\left(x_i \mid x_{j(i)}\right) \quad (2.9)$$

Nótese que:

1. *$H(X)$ no depende de t.*
2. *Puede demostrarse que (2.8) tampoco depende de t*

[1] En el artículo de Chow y Liu se escribe $I(P, P_t)$ en vez de $D(P\|P_a)$, pero se trata sólo de notación

Por tanto es posible omitir ambas expresiones en el proceso de minimización de $D(P\|P_t)$. *La expresión (2.9), por otro lado, no se comporta como se asume en el artículo de Chow y Liu ([14]), sin embargo se logra probar la afirmación de interés.*

Se procede inicialmente a probar que (2.8) no depende de t. *Se tiene entonces*

$$-\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\log P(x_{i_1})$$

donde i_1 *puede ser* $1,2,...,n$. *Supóngase que* $i_1=1$, *entonces*

$$\begin{aligned} & -\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\log P(x_{i_1}) \\ = & -\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\log P(x_1) \\ = & -\sum_{x_1}\log P(x_1)\sum_{x_2}\cdots\sum_{x_n} P(x_1,...,x_n) \end{aligned}$$

pero por la definición de distribución marginal, se tiene que $\sum_{x_2}\cdots\sum_{x_n} P(x_1,...,x_n) = P(x_1)$, *luego*

$$\begin{aligned} & -\sum_{x_1}\log P(x_1)\sum_{x_2}\cdots\sum_{x_n} P(x_1,...,x_n) \\ = & -\sum_{x_1}\log P(x_1)\left[P(x_1)\right] = -\sum_{x_1} P(x_1)\log P(x_1) \\ = & H(x_1) = H(x_{i_1}). \end{aligned}$$

Si $i_1=2,3,...,n$ *el resultado es análogo, así que (2.8) no depende de* t *y se le puede omitir del proceso de minimización.*

Aquí, sin embargo, cabe hacer la siguiente consideración:

El valor verdadero de $H(x_{i_1})$ *no depende de* t, *pero si se aproxima* H *mediante* $\widehat{H}$, *obtenida de la distribución que proporciona el árbol* t, *entonces sí resulta de interés en la minimización. En particular, se quiere que sea mínima, para que así* $\widehat{H}(x_{i_1})$ *proporcione la mínima contribución a* D. *Recuérdese que* x_{i_1} *es la variable que se coloca en la raíz del árbol, y que en el algoritmo ADNMB (el que se está desarrollando) el criterio para elegir dicha variable es que su entropía sea mínima.*

Hasta aquí, todo bien; pero se procede ahora a pasar a la parte complicada, en donde, tal parece, no se cumple completamente lo que afirman Chow y Liu. Se trabajará ahora con la expresión (2.9), que es:

$$\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{\substack{i=1\\ j(i)\neq 0}}\log P\left(x_i \mid x_{j(i)}\right)$$

En el artículo de Chow y Liu mencionado, se afirma que

$$\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{i=1}^{n}\log P\left(x_i \mid x_{j(i)}\right) \tag{2.10}$$

$$= \sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{i=1}^{n}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)} \tag{2.11}$$

$$= \sum_{i=1}^{n} I\left(x_i,x_{j(i)}\right) \tag{2.12}$$

Sin embargo, se demostrará que lo que se cumple es:

1. $\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{\substack{i=1\\ j(i)\neq 0}}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)P\left(x_{j(i)}\right)} = \sum_{\substack{i=1\\ j(i)\neq 0}}^{n} I\left(x_i,x_{j(i)}\right)$, *es decir (2.11) = (2.12)*

2. $\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{\substack{i=1\\ j(i)\neq 0}}\log P\left(x_i \mid x_{j(i)}\right) \leq \sum_{\substack{i=1\\ j(i)\neq 0}} I\left(x_i,x_{j(i)}\right)$, *o bien (2.10)* $\leq$ *(2.11). No obstante esto, se consigue el objetivo inicial de minimizar D maximizando I, como se explicará más adelante.*

A continuación se considera ambas expresiones por separado. Primero se probará que (2.11) y (2.12) son equivalentes. Comenzando con

$$\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{\substack{i=1\\ j(i)\neq 0}}^{n}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}$$

debe notarse que

$$\sum_{\substack{i=1\\ j(i)\neq 0}}^{n}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}$$

es una suma en donde aparecen todas las variables, excepto la que funge como raíz del árbol. Se supondrá que dicha raíz es x_1; entonces

$$\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\sum_{i=2}^{n}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}$$

$$= \sum_{i=2}^{n}\left[\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}\right]$$

$$= \sum_{i=2}^{n}\left[\sum_{x_i}\sum_{x_{j(i)}}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}\sum_{x_1}\cdots\sum_{x_n} P(x_1,...,x_n)\right]$$

$$= \sum_{i=2}^{n}\left\{\sum_{x_i}\sum_{x_{j(i)}}\log\frac{P\left(x_i,x_{j(i)}\right)}{P(x_i)\,P\left(x_{j(i)}\right)}\left[P\left(x_i,x_{j(i)}\right)\right]\right\}$$

donde la última igualdad se cumple porque la sumatoria múltiple se realiza sobre todas las x desde 1 hasta n, excepto sobre x_i y $x_{j(i)}$ y debido a la definición de distribución conjunta. De lo anterior se sigue que

$$\begin{aligned} & \sum_{x_i} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right) P(x_i)} \\ = & \sum_{i=2}^{n} \left\{ \sum_{x_i} \sum_{x_{j(i)}} P\left(x_i, x_{j(i)}\right) \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right) P(x_i)} \right\} \\ = & \sum_{i=2}^{n} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Si la raíz fuera otra variable distinta de x_1, el razonamiento sería análogo al presentado, con lo cual se ha probado que

$$\begin{aligned} & \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P(x_i) P\left(x_{j(i)}\right)} \\ = & \sum_{\substack{i=1 \\ j(i)\neq 0}} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Se probará a continuación que la expresión (2.10) ***no es igual*** *a (2.11), sino que (2.10) $\leq$ (2.11).*

Nuevamente se simplifica $x_{i_1} = x_1$. Obsérvese que la igualdad no se cumple porque por la definición de probabilidad condicional:

$$\begin{aligned} & \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{i=2}^{n} \log P\left(x_i \mid x_{j(i)}\right) \\ = & \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{i=2}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right)} \\ \neq & \sum_{x_1} \cdots \sum_{x_n} P(x_1, ..., x_n) \sum_{i=2}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P(x_i) P\left(x_{j(i)}\right)}. \end{aligned}$$

No obstante lo anterior, se puede probar que (2.10)$\leq$(2.11)

Para ello, nótese que

$$0 \leq P(x_i) \leq 1$$

por tanto

$$\frac{1}{P(x_i)} \geq 1$$

y entonces

$$\frac{P\left(x_i, x_{j(i)}\right)}{P(x_i) P\left(x_{j(i)}\right)} \geq \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right)}.$$

Además, como la función log *es creciente,*

$$\log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_i\right) P\left(x_{j(i)}\right)} \geq \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right)},$$

luego

$$\sum_{i=2}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_i\right) P\left(x_{j(i)}\right)} \geq \sum_{i=2}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_{j(i)}\right)},$$

de modo que

$$\begin{aligned} & \sum_{x_1} \cdots \sum_{x_n} P\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log P\left(x_i \mid x_{j(i)}\right) \\ \leq & \sum_{x_1} \cdots \sum_{x_n} P\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_i\right) P\left(x_{j(i)}\right)} \qquad (2.13) \\ \equiv & \sum_{i=2}^{n} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Luego

$$\begin{aligned} & -\sum_{x_1} \cdots \sum_{x_n} P\left(x_1, ..., x_n\right) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log P\left(x_i \mid x_{j(i)}\right) \\ \geq & -\sum_{x_1} \cdots \sum_{x_n} P\left(x_1, ..., x_n\right) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log \frac{P\left(x_i, x_{j(i)}\right)}{P\left(x_i\right) P\left(x_{j(i)}\right)} \qquad (2.14) \\ \equiv & -\sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right). \qquad (2.15) \end{aligned}$$

Recuérdese de la expresión (2.6) que la divergencia de Kullback-Leibler entre la distribución P y la aproximación proporcionada por el árbol t tiene la forma

$$\begin{aligned} D\left(P \| P_t\right) = & A - B \\ & -\sum_{x_1} \cdots \sum_{x_n} P\left(x_1, ..., x_n\right) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log P\left(x_i \mid x_{j(i)}\right) \end{aligned}$$

donde A y B no dependen de t. Luego

$$D\left(P \| P_t\right) \geq A - B - \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right)$$

y entonces

$$\begin{aligned} \min_t D\left(P \| P_t\right) & \geq \min \left[-\sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right) \right] \\ \min_t D\left(P \| P_t\right) & \geq \max \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Esto implica que si se obtiene el árbol de dependencia t de peso máximo, la distribución de probabilidad que define dicho árbol, P_t, es la que minimiza el valor de la divergencia de Kullback-Leibler con respecto a P, con lo cual queda concluida la demostración. ■

Capítulo 3

EDAs continuos

El siguiente paso en el estudio de los Algoritmos de Estimación de Distribución consiste en extender la idea planteada para variables binarias a variables continuas. La suposición inicial para lograr dicho cambio es que las variables $X_1, X_2, ..., X_n$ no siguen una distribucion Bernoulli, como en el caso de los EDAs discretos, sino determinada distribución de probabilidad continua, usualmente la distribución normal multivariada.

En la exposición de los distintos enfoques en EDAs para variables continuas que se presenta a continuación, se supondrá que:

- El espacio de búsqueda $\mathcal{B}$ es $n-dimensional$ sobre variables continuas $X_1, X_2, ..., X_n$
- Un individuo o solución sobre dicho espacio tiene la forma

  $$x = (x_1, x_2, ..., x_n)$$

  o bien

  $$x = (x_1, x_2, ..., x_n, a)$$

  si se incluye el valor de aptitud, a, del individuo
- En cada individuo x el valor x_j es el valor observado de la variable X_j para $j = 1, ..., n$
- La aptitud de los individuos se mide a través de la función $g : \mathcal{B} \to \mathbb{R}$
- Interesa minimizar el valor de la función de aptitud

Recordemos que el Algoritmo 2.1 en la página 11 muestra el esquema general de un EDA básico en el cual, al inicio de cada generación se cuenta con una población de m soluciones candidatas, por ejemplo

$$Población = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} & a_1 \\ x_{21} & x_{22} & \dots & x_{2n} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} & a_m \end{bmatrix},$$

de donde se seleccionan los $s \leq m$ individuos más aptos, que fungirán como "padres" de la siguiente generación, y se tiene:

$$Padres = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} & a_1 \\ x_{21} & x_{22} & \dots & x_{2n} & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{s1} & x_{s2} & \dots & x_{sn} & a_s \end{bmatrix}$$

A partir de esta selección de padres se obtiene una nueva población de tamaño m de soluciones candidatas mediante la suposición de que los individuos que se ha seleccionado en esta generación son realizaciones de una distribución de probabilidad continua. Por tanto, interesa estimar la función de densidad de probabilidad (*fdp*) conjunta de las X_j, $j = 1, ..., n$:

$$f_{X_1 X_2 \dots X_n}(x_1, x_2, ..., x_n)$$

y después generar mediante simulación una nueva población tomando una muestra de la distribución estimada.

Como en el caso de los EDAs discretos, también es posible clasificar a los Algoritmos de Estimación de Distribución en Dominio Continuo en tres categorías, atendiendo al tipo de dependencias que consideran entre las variables:

- Sin dependencias
- Con dependencias bivariadas
- Con dependencias múltiples

A continuación se hablará un poco sobre dos EDAs continuos: UMDA Continuo Gaussiano y EDA Gaussiano Multivariado con factorización de Cholesky.

3.1. $UMDA_c^g$

El *Algoritmo Continuo Gaussiano de Distribución Marginal Univariada* (AcgDMU, o $UMDA_c^g$ *por Univariate Marginal Distribution Algorithm, continuous gaussian*) es el EDA continuo más simple en cuanto a las dependencias entre variables que considera, ya que, al igual que en el caso discreto, supone que para cada $j_1, j_2 \in \{1, 2, ..., n\}$ las variables X_{j_1} y X_{j_2} son independientes si $j_1 \neq j_2$; por lo tanto, la *fdp* conjunta será igual al producto de las marginales:

$$f_{X_1 \dots X_n}(x_1, ..., x_n) = \prod_{j=1}^{n} f_{Xj}(x_j)$$

Además, en el $UMDA_c^g$ se asume que cada variable sigue una distribución normal, $X_j \sim N\left(\mu_j, \sigma_j^2\right), \quad j = 1, 2, ..., n$. Es decir, se trata de la misma idea expresada en

el caso discreto en UMDA, pero ahora considerando variables gaussianas en vez de Bernoulli.

Para estimar la función de desidad de probabilidad marginal de X_j, se considera la columna correspondiente que contiene los valores observados y seleccionados de dicha variable: $x_{1j}, x_{2j}, ..., x_{sj}$. Dado que se supone que X_j es una variable Normal con media μ y varianza σ^2, entonces su función de densidad de probabilidad es (véase [31, p. 107]):

$$f_{X_j}(x_j) = \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2\right\}; \qquad -\infty < x_j < \infty; \qquad j = 1, ..., n$$

Puede verse los estimadores de máxima verosimilitud de μ_j y σ_j^2 son ([31, p. 281])

$$\widehat{\mu}_j = \frac{1}{s}\sum_{i=1}^{s} x_{ij} \equiv \overline{x}_j; \qquad j = 1, ..., n \tag{3.1}$$

y

$$\widehat{\sigma}_j^2 = \frac{1}{s}\sum_{i=1}^{s} (x_{ij} - \overline{x}_j)^2; \qquad j = 1, ..., n \tag{3.2}$$

Dado lo anterior, la *fdp* conjunta de $X_1, X_2, ..., X_n$ es entonces

$$\begin{aligned} f_{X_1...X_n}(x_1, ..., x_n) &= \prod_{j=1}^{n} \frac{1}{\sigma_j\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2\right\}; \\ -\infty &< x_j < \infty; \quad j = 1, ..., n \end{aligned}$$

la cual puede estimarse mediante

$$\begin{aligned} \widehat{f}_{X_1...X_n}(x_1, ..., x_n) &= \prod_{j=1}^{n} \frac{1}{\widehat{\sigma}_j\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x-\widehat{\mu}_j}{\widehat{\sigma}_j}\right)^2\right\}; \\ -\infty &< x_j < \infty; \quad j = 1, ..., n \end{aligned}$$

Dado que se ha supuesto que las las variables son independientes, es posible simular cada una de ellas por separado. En consecuencia, para producir la siguiente generación de soluciones candidatas lo que se requiere es estimar μ_j y σ_j^2 según se indica en las ecuaciones (3.1) y (3.2) y entonces simular m números que se distribuyan $N\left(\mu_j, \sigma_j^2\right)$, por ejemplo

$$\begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{pmatrix}$$

Repitiendo el procedimiento anterior para cada $j = 1, 2, ..., n$ y luego yuxtaponiendo las columnas así obtenidas, habrá una nueva población:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1j} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2j} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mj} & \cdots & x_{mn} \end{bmatrix}.$$

La estructura del algoritmo $UMDA_c^g$ es idéntica a la de UMDA que se mostró en el Algoritmo 2.2 de la página 13.

3.2. Algunos conceptos sobre distribuciones normales univariadas, bivariadas y multivariadas

Para extender el algoritmo de Chow y Liu al caso de variables gaussianas, se requiere introducir algunos conceptos, lo cual se hace a continuación.

Definición 18 (Distribución Normal Multivariada) *Sea el vector aleatorio.*

$$X = (X_1, X_2, ..., X_n)^T$$

Se dice que X sigue la distribución normal multivariada con vector de medias

$$\mu = (\mu_1, \mu_2, ..., \mu_n)^T$$

y matriz de covarianzas

$$\Sigma = \begin{bmatrix} \sigma_{X_1}^2 & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_n} \\ \sigma_{X_1X_2} & \sigma_{X_2}^2 & \cdots & \sigma_{X_2X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_1X_n} & \sigma_{X_2X_n} & \cdots & \sigma_{X_n}^2 \end{bmatrix},$$

lo cual se denota por $X \sim N_n(\mu, \Sigma)$, si su función de densidad de probabilidad conjunta está dada por

$$\begin{aligned} f_X(x) &\equiv f_{X_1,...,X_n}(x_1,..., x_n) \\ &= \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\}; \\ -\infty &< x_i < \infty;\ i = 1, ..., n. \end{aligned} \tag{3.3}$$

A continuación se presentan, sin demostración, dos resultados que son importantes para el desarrollo del método propuesto en esta Tesis. Estos resultados se citan en [36, p. 43, expresiones 363 y 364]

Teorema 19 *Sean $Z = (Z_1, ..., Z_n)^T \sim N_n(0, I)$, Σ definida positiva y A una matriz $n \times n$ tal que $AA^T = \Sigma$. Defínase*

$$X = AZ + \mu.$$

Entonces $X \sim N_n(\mu, \Sigma)$.

Prueba. *La demostración puede verse en [18, p. 101]* ■

Nótese que el Teorema recién enunciado es la generalización multivariada de la relación entre una variable aleatoria normal y una variable aleatoria normal estándar.

Teorema 20 *Sea* $X \sim N_n(\mu, \Sigma)$ *particionada en la forma*

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix},$$

donde X_1 *tiene dimensiones* $n_1 \times 1$ *y* X_2 *tiene dimensiones* $n_2 \times 1$, *con* $n_2 = n - n_1$. *Supóngase además que el vector de medias y la matriz de covarianzas de* X *se particionan en la forma*

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

siendo μ_1 *de dimensiones* $n_1 \times 1$ *y* μ_2 *de dimensiones* $n_2 \times 1$, *y*

$$\begin{aligned} \Sigma &= \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \\ &= \begin{bmatrix} cov(X_1) & cov(X_1, X_2) \\ cov(X_2, X_1) & cov(X_2) \end{bmatrix}. \end{aligned}$$

con Σ_{11} *de dimensiones* $n_1 \times n_1$; Σ_{12} *de* $n_1 \times n_2$; Σ_{21}, *de* $n_2 \times n_1$ *y* Σ_{22}, *de* $n_2 \times n_2$. *Entonces se cumple que* $X_1 \sim N_{n_1}(\mu_1, \Sigma_{11})$ *y* $X_2 \sim N_{n_2}(\mu_2, \Sigma_{22})$.

Prueba. *La demostración puede verse en [18, pp. 102 - 103]* ■

En la sección anterior se discutió la forma en que se puede aproximar la distribución dada por la expresión (3.3) simplemente a través del producto de las distribuciones marginales. En el caso en que no todas las variables sean independientes dos a dos, la aproximación por el producto de distribuciones marginales no resulta adecuada. La forma más simple de incorporar al modelo las relaciones existentes entre n variables es suponer que aquellas se presentan solamente entre pares de variables. Entonces se está interesado en aproximar una distribución normal multivariada mediante un producto de distribuciones normales bivariadas, más específicamente, para el caso en que existe correlación entre las variables. Se introduce la distribución normal bivariada en la siguiente

Definición 21 (Distribución normal bivariada) *En el caso de que* $n = 2$, *la expresión 3.3 se transforma en*

$$\begin{aligned} f_{X,Y}(x, y) &= \frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)} \exp\left\{ -\frac{1}{2(1-\rho^2)} \left[\left(\frac{x-\mu_X}{\sigma_X} \right)^2 \right. \right. \\ &\quad \left. \left. -2\rho \frac{x-\mu_X}{\sigma_X} \frac{y-\mu_Y}{\sigma_Y} + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right] \right\} \\ -\infty &< x < \infty, \; -\infty < y < \infty \end{aligned} \tag{3.4}$$

donde ρ es el coeficiente de correlación entre X y Y, definido por

$$\rho = \frac{Cov\,(X,Y)}{\sigma_X \sigma_Y} \equiv \frac{\sigma_{XY}}{\sigma_X \sigma_Y}.$$

El modo de obtener los estimadores de máxima verosimilitud de los cinco parámetros de la distribución normal bivariada se presenta en el siguiente

Teorema 22 *Sean $(X_1, Y_1), ..., (X_m, Y_m)$ variables normales bivariadas independientes e idénticamente distribuidas, donde todos los cinco parámetros (μ_X, μ_Y, σ_X^2, σ_Y^2 y ρ) son desconocidos. Los estimadores de máxima verosimilitud de dichos parámetros son, respectivamente:*

$$\widehat{\mu}_X \equiv \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i, \qquad \widehat{\mu}_Y \equiv \overline{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$$

$$\widehat{\sigma}_X^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \widehat{\mu}_X)^2, \qquad \widehat{\sigma}_Y^2 = \frac{1}{m}\sum_{i=1}^{m} (y_i - \widehat{\mu}_Y)^2$$

$$\widehat{\rho} = \frac{1}{m}\sum_{i=1}^{m} (x_i - \widehat{\mu}_X)(y_i - \widehat{\mu}_Y) / \widehat{\sigma}_X \widehat{\sigma}_Y$$

Prueba. *Véase [13, p. 358]* ■

A partir de la definición anterior podemos obtener la distribución normal bivariada condicional, que resulta fundamental en el desarrollo del método que se presenta en este trabajo de Tesis.

Teorema 23 (Distribución normal bivariada condicional) *Si el vector aleatorio (X, Y) tiene una distribución normal bivariada, entonces la distribución de X dado $Y = y$ es normal con media $\mu_{X|Y=y} = \mu_X + \rho\frac{\sigma_X}{\sigma_Y}(y - \mu_Y)$ y varianza $\sigma^2_{X|Y=y} = \sigma_X^2(1 - \rho^2)$.*

Prueba. *Por definición (véase [31, p. 146])*

$$f_{X|Y=y}(x \mid y) = \frac{f_{XY}(x,y)}{f_Y(y)},$$

luego, a partir de (3.4), se tiene

$$\begin{aligned} f_{X|Y=y}(x \mid y) &= \left\{ \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \exp\left\{ -\frac{1}{2(1-\rho^2)} z \right\} \right. \\ &\quad \left. \Big/ \left\{ \frac{1}{\sqrt{2\pi}\sigma_Y} \exp\left\{ -\frac{1}{2}\left(\frac{y-\mu_Y}{\sigma_Y} \right)^2 \right\} \right\}, \end{aligned}$$

con

$$z = \left(\frac{x-\mu_X}{\sigma_X} \right)^2 - 2\rho \frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y} + \left(\frac{y-\mu_Y}{\sigma_Y} \right)^2.$$

La constante de normalización se simplifica a

$$\begin{aligned} & \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho}} \Big/ \frac{1}{\sqrt{2\pi}\sigma_Y} \\ = \quad & \frac{1}{\sqrt{2\pi}\sigma_X\sqrt{1-\rho^2}}. \end{aligned}$$

Por otro lado, simplificado el argumento de las exponenciales, y omitiendo para efectos de simplicidad el factor $-\frac{1}{2(1-\rho^2)}$*, resulta*

$$\begin{aligned} = \quad & \left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y} + \left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 - \left(1-\rho^2\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 \\ = \quad & \left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y} + \rho^2\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2 \end{aligned}$$

$$\begin{aligned} = \quad & \left(\frac{x-\mu_X}{\sigma_X} - \rho\frac{y-\mu_Y}{\sigma_Y}\right)^2 \\ = \quad & \frac{1}{\sigma_X^2}\left[x-\mu_X-\rho\frac{\sigma_X}{\sigma_Y}\left(y-\mu_Y\right)\right]^2. \end{aligned}$$

Por tanto

$$f_{X|Y=y}\left(x \,|\, y\right) = \frac{1}{\sqrt{2\pi}\sigma_X\sqrt{1-\rho}} \exp\left\{-\frac{1}{2\left(1-\rho^2\right)\sigma_X^2}\left[x-\mu_X-\rho\frac{\sigma_X}{\sigma_Y}\left(y-\mu_Y\right)\right]^2\right\}.$$

Es decir que $X \,|\, Y = y \sim N\left(\mu_X + \rho\frac{\sigma_X}{\sigma_Y}\left(y-\mu_Y\right), \left(1-\rho^2\right)\sigma_X^2\right)$ ■

Recuérdese que el método de Chow y Liu emplea conceptos de teoría de información para la construcción del árbol de dependencias. A continuación se presentan dichos conceptos, referidos a variables continuas en general y gaussianas en particular.

Definición 24 (Entropía Diferencial) *Sea* X *una variable aleatoria continua definida en* $\mathbb{R}$ *con fdp* f_X*. La entropía o Entropía Diferencial de* X *está dada por*

$$\begin{aligned} H\left(X\right) &= E\left(-\log f_X\left(x\right)\right) \\ H\left(X\right) &= -\int_{-\infty}^{\infty} f_X\left(x\right)\log f_X\left(x\right)dx \end{aligned}$$

si la integral existe [40].

Definición 25 (Entropía Diferencial Conjunta) *Sean* X *y* Y *variables aleatorias continuas con fdp conjunta* f_{XY}*. La Entropía Diferencial Conjunta de* X *y* Y *está dada por*

$$H\left(X,Y\right) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}\left(x,y\right)\log f_{XY}\left(x,y\right)dydx$$

si la integral existe [39, p. 35].

Definición 26 (Información Mutua Continua) *Sean X y Y variables aleatorias continuas con funciones de densidad de probabilidad f_X y f_Y, respectivamente, y fdp conjunta f_{XY}. La Información Conjunta de X y Y está dada por*

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log\frac{f_{XY}(x,y)}{f_X(x)\,f_Y(y)}dydx$$

si la integral existe [17, p. 37].

Afirmación 27 *Sean X_i y X_j dos variables aleatorias continuas con recorridos D_i y D_j, respectivamente; entonces*

$$I(X_i,X_j) \geq 0.$$

Prueba. *En esta prueba se sigue la idea presentada en [17, p. 21].*

Sean f_{X_i} y f_{X_j} las fdp marginales de X_i y X_j; respectivamente, y sea $f_{X_iX_j}$ la fdp conjunta de X_i y X_J. Por definición tenemos que

$$I(X_i,X_j) = \int_{D_i}\int_{D_j} f_{X_iX_j}(x_i,x_j)\log\left(\frac{f_{X_iX_j}(x_i,x_j)}{f_{X_i}(x_i)\,f_{X_i}(x_i)}\right)dx_jdx_i$$

y por las propiedades de los logaritmos, $\log\frac{a}{b} = -\log\frac{b}{a}$, así que

$$\begin{aligned} I(X_i,X_j) &= -\int_{D_i}\int_{D_j} f_{X_iX_j}(x_i,x_j)\log\left(\frac{f_{X_i}(x_i)\,f_{X_i}(x_i)}{f_{X_iX_j}(x_i,x_j)}\right)dx_jdx_i \\ &\Leftrightarrow \quad -I(X_i,X_j) = \int_{D_i}\int_{D_j} f_{X_iX_j}(x_i,x_j)\log\left(\frac{f_{X_i}(x_i)\,f_{X_i}(x_i)}{f_{X_iX_j}(x_i,x_j)}\right)dx_jdx_i \end{aligned}$$

Ahora recuérdese la relación básica

$$\log x \leq x-1$$

con lo que

$$\begin{aligned} & \int_{D_i}\int_{D_j} f_{X_iX_j}(x_i,x_j)\log\left(\frac{f_{X_i}(x_i)\,f_{X_i}(x_i)}{f_{X_iX_j}(x_i,x_j)}\right)dx_jdx_i \\ \leq\ & \int_{D_i}\int_{D_j} f_{X_iX_j}(x_i,x_j)\left(\frac{f_{X_i}(x_i)\,f_{X_i}(x_i)}{f_{X_iX_j}(x_i,x_j)}-1\right)dx_jdx_i \\ =\ & \int_{D_i}\int_{D_j}\left[f_{X_i}(x_i)\,f_{X_i}(x_i) - f_{X_iX_j}(x_i,x_j)\right]dx_jdx_i \\ =\ & 1(1)-1=0 \end{aligned}$$

De lo anterior resulta que $-I(X_i,X_j) \leq 0$, es decir,

$$I(X_i,X_j) \geq 0$$

■

Teorema 28 (Entropía de una variable aleatoria normal) *Sea* $X \sim N(\mu, \sigma^2)$, *entonces la entropía de* X *es*

$$H(X) = \frac{1}{2}\log\left(2\pi e\sigma^2\right).$$

Prueba. *Se sabe que la entropía* H *de una variable aleatoria* X *con función de densidad de probabilidad* f_X *se define como el valor esperado de la función* $-\log(f_X)$, *esto es:*

$$H(X) = E(-\log f_X(x)).$$

Sea $X \sim N(\mu, \sigma^2)$, *entonces su fdp es*

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}; \quad -\infty < x < \infty,$$

así que el valor esperado de $-\log(f_X)$ *es*

$$\begin{aligned} H(X) &= E(-\log f_X(x)) - E(\log f_X(x)) = -\int_{-\infty}^{\infty} f_X(x)\log f_X(x)\,dx \\ &= -\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}\log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}\right)dx \\ &= -\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}\left[-\log\left(\sqrt{2\pi}\sigma\right) - \frac{(x-\mu)^2}{2\sigma^2}\right]dx \end{aligned}$$

Distribuyendo la fdp de X *en el corchete, queda*

$$\begin{aligned} H(X) &= \log\left(\sqrt{2\pi}\sigma\right)\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}dx \\ &\quad + \frac{1}{2\sigma^2}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}(x-\mu)^2\exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}dx, \end{aligned}$$

Ahora, note que el primer integrando es, precisamente, la fdp de una variable aleatoria, por tanto su integral es igual a 1, *mientras que la segunda integral es por definición la varianza de* X, *así que se obtiene*

$$\begin{aligned} H(X) &= \log\left(\sqrt{2\pi}\sigma\right) + \frac{1}{2\sigma^2}Var(X) = \log\left(\sqrt{2\pi}\sigma\right) + \frac{1}{2\sigma^2}\sigma^2 \\ &= \log\left(\sqrt{2\pi}\sigma\right) + \frac{1}{2}. \end{aligned}$$

De lo anterior, se llega a

$$\begin{aligned} H(X) &= \log\left(\sqrt{2\pi}\sigma\right) + \frac{1}{2} = \log\left(\sqrt{2\pi}\sigma\right) + \frac{1}{2}\log e \\ &= \frac{1}{2}\left[2\log\left(\sqrt{2\pi}\sigma\right) + \log e\right] = \frac{1}{2}\left[\log\left(2\pi\sigma^2\right) + \log e\right] \end{aligned}$$

de donde

$$H(X) = \frac{1}{2}\log\left(2\pi\sigma^2 e\right).$$

■

Teorema 29 (Entropía conjunta normal bivariada) *Si el vector aleatorio (X,Y) sigue la distribución normal bivariada $N_2(\mu,\Sigma)$, entonces la entropía conjunta de X y Y está dada por*

$$H(X,Y) = \frac{1}{2}\log\left(\left[2\pi e\right]^2|\Sigma|\right).$$

Prueba. *Sea (X,Y) un vector aleatorio que se distribuye $N_2(\mu,\Sigma)$, con $\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}$ y $\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix}$.*
Entonces

$$\begin{aligned} H(X,Y) &= E\left(-\log f_{XY}(x,y)\right) \\ &= -\int_x\int_y f_{XY}(x,y)\log f_{XY}(x,y)\,dydx \end{aligned}$$

$$\begin{aligned} = & -\int_x\int_y \frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2\right.\right. \\ & \left.\left.-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\} \\ & \times\log\left(\frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2\right.\right.\right. \\ & \left.\left.\left.-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}\right)dydx \end{aligned}$$

$$\begin{aligned} H(X,Y) = & -\int_x\int_y \frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2\right.\right. \\ & \left.\left.-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\} \\ & \times\left[\log\left(\frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\right)-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2\right.\right. \\ & \left.\left.\left.-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right]dydx \end{aligned}$$

$$H(X,Y) = -\log\left(\frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\right)\int_x\int_y \frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2 -2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}dydx \quad (3.5)$$

$$+\int_x\int_y \frac{1}{2\pi\sigma_X\sigma_Y(1-\rho^2)}\left\{\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}$$
$$\times\exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]\right\}dydx \quad (3.6)$$

Nótese que el integrando de la expresión (3.5) es la fdp de una variable aleatoria normal bivariada, por tanto queda

$$(3.5) = \log\left(2\pi\sigma_X\sigma_Y\left(1-\rho^2\right)\right).$$

Por otro lado, para simplificar la expresión (3.6), nótese que el argumento de la función exponencial en la fdp normal multivariada general (3.3) es $-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$, *que en el caso de* $n=2$ *resulta ser* $a := \frac{-1}{2(1-\rho^2)}\left[\left(\frac{x-\mu_X}{\sigma_X}\right)^2-2\rho\frac{x-\mu_X}{\sigma_X}\frac{y-\mu_Y}{\sigma_Y}+\left(\frac{y-\mu_Y}{\sigma_Y}\right)^2\right]$.
Así que tenemos

$$\begin{aligned}(3.6) &= -\int_x\int_y \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}a\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}dydx\\ &= -E(a) = -E\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)\\ &= \frac{1}{2}E\left((x-\mu)^T\Sigma^{-1}(x-\mu)\right) = \frac{1}{2}Tr\left(\Sigma^{-1}\Sigma\right) \quad (3.7)\\ &= \frac{1}{2}2\\ &= 1\end{aligned}$$

en donde la igualdad (3.7) se cumple porque si $X\sim N(\mu,\Sigma)$, *entonces* $E\left((x-m)^T A(x-m)\right) = (\mu-m)^T A(\mu-m) + Tr(A\Sigma)$ *(véase [36, p.37 identidad 302]).*

Así que

$$\begin{aligned}H(X,Y) &= \log\left(2\pi\sigma_X\sigma_Y\left(1-\rho^2\right)\right)+1\\ &= \frac{1}{2}\log\left(\left[2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}\right]^2\right)+\frac{1}{2}2\log e\\ &= \frac{1}{2}\log\left(\left[2\pi\sigma_X\sigma_Y\right]^2\left[1-\rho^2\right]\right)+\frac{1}{2}\log e^2\\ &= \frac{1}{2}\log\left(\left[2\pi e\right]^2\sigma_X^2\sigma_Y^2\left[1-\rho^2\right]\right)\end{aligned}$$

Ahora, recuérdese que

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix}$$

por tanto

$$\begin{aligned} |\Sigma| &= \sigma_X^2\sigma_Y^2 - \sigma_{XY}^2 = \sigma_X^2\sigma_Y^2 - \rho^2\sigma_X^2\sigma_Y^2 \\ &= \sigma_X^2\sigma_Y^2\left(1-\rho^2\right). \end{aligned}$$

Es decir que

$$H(X,Y) = \frac{1}{2}\log\left(\left[2\pi e\right]^2|\Sigma|\right)$$

■

Teorema 30 (Entropía condicional normal bivariada) *Si el vector aleatorio* (X,Y) *sigue la distribución normal bivariada* $N_2(\mu,\Sigma)$, *entonces la entropía condicional de* X *dado* $Y=y$ *está dada por*

$$H(X \mid Y=y) = \frac{1}{2}\log\left(2\pi e\sigma_X^2\left(1-\rho^2\right)\right).$$

Prueba. *En efecto, por el Teorema 23, si el vector aleatorio* (X,Y) *se distribuye normal bivariado con vector de medias*

$$\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}$$

y matriz de covarianzas

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix},$$

entonces condicional en $Y=y$, *la variable* X *se distribuye* $N\left(\mu_X + \rho\frac{\sigma_X}{\sigma_Y}y - \mu_Y, \left(1-\rho^2\right)\sigma_X^2\right)$.

Además, por el Teorema 28 se sabe que la entropía de una variable aleatoria $X \sim N(\mu,\sigma^2)$ *es* $H(X) = \frac{1}{2}\log\left(2\pi\sigma^2 e\right)$, *entonces*

$$H(X \mid Y=y) = \frac{1}{2}\log\left(2\pi e\sigma_X^2\left(1-\rho^2\right)\right)$$

■

Teorema 31 (Información mutua) *Sea* $I(X,Y)$ *la información mutua de* X *y* Y. *Entonces*

$$I(X,Y) = H(X) + H(Y) - H(X,Y).$$

Prueba. *Se tiene que*

$$I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log\frac{f_{XY}(x,y)}{f_X(x)\,f_Y(y)}dydx,$$

luego

$$\begin{aligned} I(X,Y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\left[\log f_{XY}(x,y) - \log f_X(x) - \log f_Y(y)\right] dydx \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log f_{XY}(x,y)\, dydx - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log f_X(x)\, dydx \\ &\quad - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{XY}(x,y)\log f_Y(y)\, dydx \end{aligned}$$

de donde

$$\begin{aligned} I(X,Y) &= -H(X,Y) - \int_{-\infty}^{\infty}\log f_X(x)\left[\int_{-\infty}^{\infty} f_{XY}(x,y)\, dy\right] dx \\ &\quad - \int_{-\infty}^{\infty}\log f_Y(y)\left[\int_{-\infty}^{\infty} f_{XY}(x,y)\, dx\right] dy. \end{aligned}$$

Luego queda

$$\begin{aligned} I(X,Y) &= -H(X,Y) - \int_{-\infty}^{\infty}\log f_X(x)\left[f_X(x)\right] dx - \int_{-\infty}^{\infty}\log f_Y(y)\left[f_Y(y)\right] dy \\ &= -H(X,Y) + H(X) + H(Y) \end{aligned}$$

Es decir,

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

■

Teorema 32 (Información mutua normal) *Si el vector aleatorio (X,Y) sigue la distribución normal bivariada $N_2(\mu,\Sigma)$, entonces la información mutua de X y Y está dada por*

$$I(X,Y) = -\frac{1}{2}\log\left(1-\rho^2\right).$$

Prueba. *Del Teorema 28 tenemos que*

$$H(X) = \frac{1}{2}\log\left(2\pi\sigma_X^2 e\right)$$

y

$$H(Y) = \frac{1}{2}\log\left(2\pi\sigma_Y^2 e\right)$$

Además, por el Teorema 29,

$$\begin{aligned} H(X,Y) &= \frac{1}{2}\log\left(\left[2\pi e\right]^2|\Sigma|\right) \\ &= \frac{1}{2}\log\left(\left[2\pi e\right]^2\sigma_X^2\sigma_Y^2\left[1-\rho^2\right]\right) \end{aligned}$$

Así que por el resultado anteior,

$$\begin{aligned} I(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= \frac{1}{2}\log\left(2\pi\sigma_X^2 e\right) + \frac{1}{2}\log\left(2\pi\sigma_Y^2 e\right) - \frac{1}{2}\log\left(\left[2\pi e\right]^2 \sigma_X^2\sigma_Y^2\left[1-\rho^2\right]\right) \\ &= \frac{1}{2}\left[\log\left(2\pi\sigma_X^2 e\right) + \log\left(2\pi\sigma_Y^2 e\right) - \log\left(\left[2\pi e\right]^2 \sigma_X^2\sigma_Y^2\left[1-\rho^2\right]\right)\right] \\ &= \frac{1}{2}\log\left(\frac{\left[2\pi\sigma_X\sigma_Y e\right]^2}{\left[2\pi e\right]^2 \sigma_X^2\sigma_Y^2\left[1-\rho^2\right]}\right) \end{aligned}$$

de donde

$$I(X,Y) = -\frac{1}{2}\log\left(1-\rho^2\right)$$

■

3.3. EDA gaussiano multivariado con factorización de Cholesky

El EDA gaussiano multivariado con factorización de Cholesky, mostrado en el Algoritmo 3.1, está basado en lo expresado en el Teorema 19 con respecto a que si $Z = (Z_1, ..., Z_n)^T \sim N_n(0, I)$, entonces $X = AZ + \mu \sim N_n(\mu, \Sigma)$, con $\Sigma = AA^T$.

Nótese que este EDA hace uso de la distribución normal multivariada completa para producir cada generación.

Algoritmo 3.1 EDA gaussiano multivariado con Factorización de Cholesky

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime μ y Σ
5: Obtenga la factorización de Cholesky de Σ:

$$\Sigma = AA^T \tag{3.8}$$

6: Genere Z = Vector de observaciones $N_n(0, I)$
7: $hijos$ = Genere m individuos de acuerdo con la expresión

$$hijos = AZ + \mu \tag{3.9}$$

8: $pob_{previa} = padres \cup hijos$
9: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
10: **end while**

3.4. Algoritmo de Chow y Liu para variables gaussianas

Recuérdese de la Sección 2.4 que el método de Chow y Liu establece una forma para aproximar distribuciones conjuntas discretas mediante un árbol con dependencias bivariadas [14] cuya magnitud se evalúa a través de la *información mutua* entre variables. En esta sección se extenderá dicho método para aplicarlo en la aproximación de distribuciones normales multivariadas.

Dada una función de densidad de probabilidad conjunta en n variables, $f_{X_1X_2...X_n}(x_1, x_2, ..., x_n)$, acotada, el algoritmo de Chow y Liu proporciona la mejor aproximación a f, según se demuestra en la Sección 3.5, utilizando una densidad marginal y $n-1$ densidades condicionales bivariadas, de la forma

$$\widehat{f}(x_1, x_2, ..., x_n) = \widehat{f}(x_{i_1})\,\widehat{f}\left(x_{i_2}|x_{j(i_2)}\right)...\widehat{f}\left(x_{i_n}|x_{j(i_n)}\right)$$

en donde $i_1, i_2, ..., i_n$ es una permutación de los índices $1, ..., n$; y $j(i_l) \in \{1, ..., n\}\backslash\{i_l\}$ es la variable que condiciona a i_l, para $l = 2, ..., n$.

En el método descrito por Chow y Liu para la aproximación de distribuciones conjuntas discretas, la generación de poblaciones se hace con base en una hipermatriz P y un árbol de dependencias, G. La hipermatriz P contiene las distribuciones marginales de todas las variables y las distribuciones conjuntas para todos los pares posibles de variables. Por su parte, $G = (E, R)$ está dado por una raíz, R, y un conjunto de aristas $E = \{(v, v') \mid v, v' = 1, 2, ..., n;\ v \neq v'\}$ que indica la relación de dependencias de las variables X_v y $X_{v'}$ (véase la sección 2.4).

Para el caso gaussinano se requiere del vector de medias

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}$$

y la matriz de varianzas y covarianzas

$$\Sigma = \begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_n} \\ \sigma_{X_1X_2} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_1X_n} & \sigma_{X_2X_n} & \cdots & \sigma^2_{X_n} \end{bmatrix}$$

además del árbol de dependencias $G = (E, R)$ formado igual que en el caso discreto por la raíz, R, y el conjunto de aristas $E = \{(v, v') \mid v, v' = 1, 2, ..., n;\ v \neq v'\}$ que indica la relación de dependencias de las variables X_v y $X_{v'}$

El Algoritmo 3.2 resume el método descrito por Chow y Liu, extendido para aproximar distribuciones normales multivariadas.

En dicho algoritmo se requiere el cálculo de los estimadores de máxima verosimilitud para el vector de medias y la matriz de covarianzas bivariados, $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$. Por el Teorema 20 de la p. 38, estos estimadores son los mismos dados en el Teorema 22 de la p. 39.

Algoritmo 3.2 De Chow y Liu para Aproximación de una Distribución Normal Multivariada mediante la Construcción de un Árbol de Dependencias

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime μ y Σ mediante $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
5: Obtenga el grafo de dependencias $G = (E, R)$ a partir de $\widehat{\Sigma}_{MV}$
6: $hijos$ = Genere aleatoriamente una población de tamaño m con base en G, $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
7: $pob_{previa} = padres \cup hijos$
8: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
9: **end while**

Para la obtención del grafo de dependencias, se hace uso del concepto de información mutua continua. Como se vio anteriormente, se puede obtener un árbol de peso máximo obteniendo la combinación correcta de ramas con información máxima, lo cual en el método de Chow y Liu, se realiza mediante la construcción de un árbol de peso máximo a través de un algoritmo glotón que elige en cada iteración de entre todas las ramas posibles aquella con máxima información. La construcción del grafo de dependencias se explica mediante el Algoritmo 3.3 en donde se hace uso de la siguiente

Notación 33 *Para el algoritmo de construcción del grafo de dependencias en el Algoritmo de Chow y Liu para variables gaussianas:*

- *Vértice :=* $Variable$
- *Arista :=* $Par\ de\ variables$
- $D :=$ *Conjunto de todas las aristas posibles*
- $V :=$ *Conjunto de vértices*
- $A :=$ *Conjunto de vértices sin procesar hasta el momento*
- $I(v, v') =$ *Valor de la información en la arista* (v, v')
- $E :=$ *Conjunto de aristas en el grafo final*
- $R :=$ *Vértice raíz del grafo de dependencias*

Algoritmo 3.3 Construcción del grafo de dependencias para variables gaussianas según Chow y Liu

Sean $V = \{1, 2, ..., n\}$, $A = V$, $E = \emptyset$, $B = A \times V$
for $i = 1, \ldots, n-1$ **do**
  if $B \neq \emptyset$ **then**
    Sea (v, v') aquella arista que satisfaga

$$(v, v') = \arg\max_{(u,w)\in B} I(u, w)$$

    Sea $A = V \setminus \{v, v'\}$
    Sea $B = A \times (V \setminus A)$
    Sea $E = E \cup \{(v, v')\}$
  end if
end for
Sea R el único vértice que no forma parte de alguna arista en E

Al igual que en el caso discreto, la simulación de la siguiente generación de individuos a partir de μ y Σ en el método de Chow y Liu para variables gaussianas requiere el recorrido en *entreorden* de un árbol $n-ario$, que difiere del primero referido solamente en las distribuciones de las variables involucradas. Se presenta la forma de realizar dicha simulación en el Algoritmo 3.4

Algoritmo 3.4 Recorrido del grafo de dependencias en el método de Chow y Liu para variables gaussianas

Input: padre $\equiv X_{j(i)}$, nodo $\equiv X_i$
if $X_i = R$ **then** // El nodo es la raíz
  Simule m valores distribuidos $N(\widehat{\mu}_{X_i}, \widehat{\Sigma}_{X_i})$
else // El nodo X_i está condicionado por $X_{j(i)}$
  Calcule $\widehat{\mu}_{X_i|X_{j(i)}}$ y $\widehat{\Sigma}_{X_i|X_{j(i)}}$
  Simule m valores distribuidos $N(\widehat{\mu}_{X_i|X_{j(i)}}, \widehat{\Sigma}_{X_i|X_{j(i)}})$
end if
while X_i tenga hijos **do**
  Ejecute el Algoritmo 3.4 con nodo, hijo
end while

La forma simple que adquieren las expresiones para la información mutua cuando se trata de variables gaussianas facilita el cálculo del máximo. Recordemos del Teorema 32 que la expresión para la información mutua cuando se trata de variables gaussianas es

$$I(X, Y) = -\frac{1}{2}\log\left(1 - \rho^2\right).$$

Para obtener el máximo, se pueden utilizar herramientas de cálculo diferencial para

observar que $I(X,Y)$ tiene un mínimo en $\rho = 0$, y asíntotas verticales (véase [3, p. 232]) en $\rho = -1$ y $\rho = 1$, puntos en los cuales, $I \to \infty$. De modo que la información es máxima cuando el valor absoluto del coeficiente de correlación lo es, a saber, cuando existe una mayor relación entre las variables involucradas.

3.5. El mejor árbol de dependencia, caso continuo

En esta sección se extenderán los resultados presentados en la sección 2.5 para el caso en que se trabaja con variables continuas. Esto se hará mediante la demostración del siguiente resultado:

Afirmación 34 *Sea f acotada. La distribución de probabilidad definida por el árbol t, $f_t(\mathbf{x})$, es una aproximación óptima a la función de densidad de probabilidad $f(\mathbf{x})$ si y sólo si t es un árbol de dependencia de peso máximo.*

Prueba. *Se puede considerar que la distribución f_t aproxima a f de manera óptima en el sentido de proporcionar el mínimo valor para la divergencia de Kullback-Liebler, si es la solución de*

$$\begin{aligned} f_t &= \arg\min_t D(f\|f_t) \\ &= \arg\min_t \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{f_t(\mathbf{x})} d\mathbf{x} \end{aligned}$$

Se demostrará que

$$\min_t \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{f_t(\mathbf{x})} d\mathbf{x}$$

equivale a

$$\max_t \sum_{i=1}^{n} I\left(x_i, x_{j(i)}\right)$$

Para encontrar el árbol que produce f_t tal que minimice $D(f\|f_t)$ obsérvese que

$$\begin{aligned} D(f\|f_t) &= \int_{\mathbf{x}} f(\mathbf{x}) \log \frac{f(\mathbf{x})}{f_t(\mathbf{x})} d\mathbf{x} \\ &= \int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \log \frac{f(x_1, ..., x_n)}{f_t(x_1, ..., x_n)} dx_n \cdots dx_1. \end{aligned}$$

Luego

$$\begin{aligned} D(f\|f_t) &= \int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \left[\log f(x_1, ..., x_n) - \log f_t(x_1, ..., x_n)\right] dx_n \cdots dx_1 \\ &= \int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \log f(x_1, ..., x_n)\, dx_n \cdots dx_1 \\ &\quad - \int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \log f_t(x_1, ..., x_n)\, dx_n \cdots dx_1 \end{aligned}$$

$$
\begin{aligned}
D\left(f\|f_t\right) &= -H\left(X\right) \\
&\quad -\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log f_t\left(x_1,...,x_n\right)dx_1\cdots dx_n
\end{aligned}
$$

$$
\begin{aligned}
D\left(f\|f_t\right) &= -H\left(X\right) \\
&\quad -\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log\prod_{i=1}^{n} f_{X_i}\left(x_i\right) f_{X_i|X_{j(i)}}\left(x_i \mid x_{j(i)}\right)dx_n\cdots dx_1 \quad (3.10)\\
&= -H\left(X\right) \quad (3.11)\\
&\quad -\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log f\left(x_{i_1}\right)dx_n\cdots dx_1 \quad (3.12)\\
&\quad -\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\sum_{\substack{i=1\\ j(i)\neq 0}}^{n}\log f\left(x_i \mid x_{j(i)}\right)dx_n\cdots dx_1 \quad (3.13)
\end{aligned}
$$

Nótese que:

1. *$H\left(X\right)$ no depende de t.*

2. *Puede demostrarse que (3.12) tampoco depende de t*

Por tanto es posible omitir ambas expresiones en el proceso de minimización de $D\left(f\|f_t\right)$. La expresión (3.13), por otro lado, no se comporta como se asume en el artículo de Chow y Liu [14], sin embargo se logra probar la afirmación de interés.

Se procede inicialmente a probar que (3.12) no depende de t. Se tiene entonces

$$\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log f\left(x_{i_1}\right)dx_n\cdots dx_1$$

donde i_1 puede ser $1,2,...,n$. Supóngase que $i_1=1$, entonces

$$
\begin{aligned}
&\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log f\left(x_{i_1}\right)dx_n\cdots dx_1\\
=\ &\int_{x_1}\cdots\int_{x_n} f\left(x_1,...,x_n\right)\log f\left(x_1\right)dx_n\cdots dx_1\\
=\ &\int_{x_1}\log f\left(x_1\right)\left[\int_{x_2}\cdots\int_{x_n} f\left(x_1,...,x_n\right)dx_n\cdots dx_2\right]dx_1
\end{aligned}
$$

en donde puede cambiarse el orden de integración porque las f son funciones de densidad de probabilidad, por tanto sus distribuciones de probabilidad correspondientes son medidas finitas y puede entonces aplicarse el Teorema de Tonelli (véase [9, p. 118]).

Pero por la definición de distribución marginal, se tiene que $\int_{x_2} \cdots \int_{x_n} f(x_1, ..., x_n)\, dx_n \cdots dx_2 = f(x_1)$, *luego*

$$\begin{aligned} & -\int_{x_2} \cdots \int_{x_n} f(x_1, ..., x_n) \log f(x_1)\, dx_n \cdots dx_1 \\ = & -\int_{x_1} \log f(x_1) [f(x_1)]\, dx_1 = -\int_{x_1} f(x_1) \log f(x_1)\, dx_1 \\ = & H(x_1) = H(x_{i_1}). \end{aligned}$$

Si $i_1 = 2, 3, ..., n$ *el resultado es análogo, así que (3.12) no depende de* t *y se le puede omitir del proceso de minimización.*

Aquí, sin embargo, cabe hacer la siguiente consideración:

El valor verdadero de $H(x_{i_1})$ *no depende de* t, *pero si se aproxima* H *mediante* $\widehat{H}$, *obtenida de la distribución que proporciona da el árbol* t, *entonces sí resulta de interés en la minimización. En particular, se quiere que sea mínima, para que así* $\widehat{H}(x_{i_1})$ *proporcione la mínima contribución a* D. *Recuérdese que* x_{i_1} *es la variable que se coloca en la raíz del árbol, y que en el algoritmo ADNMB (el que se está desarrollando) el criterio para elegir dicha variable es que su entropía sea mínima.*

Hasta aquí, todo bien; pero se procede ahora a pasar a la parte complicada, en donde, tal parece, no se cumple completamente lo que afirman Chow y Liu. Se trabajará ahora con la expresión (3.13), que es:

$$\int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1$$

Extendiendo a variables continuas la afirmación que se hace en el artículo de Chow y Liu [14], se debería cumplir que

$$\int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1 \tag{3.14}$$

$$= \int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{i=1}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f(x_i) f\left(x_{j(i)}\right)} dx_n \cdots dx_1 \tag{3.15}$$

$$= \sum_{i=1}^{n} I\left(x_i, x_{j(i)}\right) \tag{3.16}$$

Sin embargo, se demostrará que lo que se cumple es:

1. $\int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{i=1}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f(x_i) f\left(x_{j(i)}\right)} dx_n \cdots dx_1 = \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right)$, *es decir (3.15)=(3.16)*

2. $\int_{x_1} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1 \leq \sum_{\substack{i=1 \\ j(i)\neq 0}} I\left(x_i, x_{j(i)}\right) + M_2$, *o bien (3.14)* $\leq$ *(3.15)* $+M_2$, *donde* M_2 *es una constante. No obstante esto, se*

consigue el objetivo inicial de minimizar D maximizando I, como se explicará más adelante.

A continuación se considera ambas expresiones por separado. Primero se probará que (3.15) y (3.16) son equivalentes. Comenzando con

$$\int_{x_1}\cdots\int_{x_n} f(x_1,...,x_n) \sum_{\substack{i=1\\ j(i)\neq 0}}^{n} \log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}dx_n\cdots dx_1$$

debe notarse que

$$\sum_{\substack{i=1\\ j(i)\neq 0}}^{n} \log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}$$

es una suma en donde aparecen todas las variables, excepto la que funge como raíz del árbol. Se supondrá que dicha raíz es x_1; entonces queda

$$\begin{aligned}&\int_{x_1}\cdots\int_{x_n} f(x_1,...,x_n)\sum_{i=2}^{n}\log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}dx_n\cdots dx_1\\ =\;&\sum_{i=2}^{n}\left[\int_{x_1}\cdots\int_{x_n} f(x_1,...,x_n)\log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}dx_n\cdots dx_1\right],\end{aligned}$$

suponiendo que $\log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)f\left(x_{j(i)}\right)}$ *sea integrable según Lebesgue (véase [9, p. 32]) o según Riemann(véase [3, p. 438]).*

$$\begin{aligned}=\;&\sum_{i=2}^{n}\left[\int_{x_i}\int_{x_{j(i)}}\log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}dx_i dx_{j(i)}\int_{x_1}\cdots\int_{x_n} f(x_1,...,x_n)\,dx_n\cdots dx_1\right]\\ =\;&\sum_{i=2}^{n}\left\{\int_{x_i}\int_{x_{j(i)}}\log\frac{f\left(x_i,x_{j(i)}\right)}{f(x_i)\,f\left(x_{j(i)}\right)}\left[f\left(x_i,x_{j(i)}\right)\right]dx_i dx_{j(i)}\right\}\end{aligned}$$

donde la última igualdad se cumple porque la integral múltiple se realiza sobre todas las x desde 1 hasta n, excepto sobre x_i y $x_{j(i)}$ y debido a la definición de distribución marginal. De lo anterior se sigue que

$$\begin{aligned}&\int_{x_i}\cdots\int_{x_n} f(x_1,...,x_n)\sum_{\substack{i=1\\ j(i)\neq 0}}^{n}\log\frac{f\left(x_i,x_{j(i)}\right)}{f\left(x_{j(i)}\right)f(x_i)}dx_n\cdots dx_1\\ =\;&\sum_{i=2}^{n}\left\{\int_{x_i}\int_{x_{j(i)}}\log\frac{f\left(x_i,x_{j(i)}\right)}{f\left(x_{j(i)}\right)P(x_i)}\left[f\left(x_i,x_{j(i)}\right)\right]dx_i dx_{j(i)}\right\}\\ =\;&\sum_{i=2}^{n} I\left(x_i,x_{j(i)}\right).\end{aligned}$$

Si la raíz fuera otra variable distinta de x_1, el razonamiento sería análogo al presentado, con lo cual se ha probado que

$$\begin{aligned} & \int_{x_i} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f(x_i)} dx_n \cdots dx_1 \\ = & \sum_{\substack{i=1 \\ j(i)\neq 0}} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Se probará a continuación que la expresión (3.14) ***no es igual*** *a (3.15), sino que (3.14) $\leq$(3.15)+M_2.*

Nuevamente se simplifica $x_{i_1} = x_1$. Obsérvese que la igualdad no se cumple porque por la definición de probabilidad condicional:

$$\begin{aligned} & \int_{x_i} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{i=2}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1 \\ = & \int_{x_i} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} dx_n \cdots dx_1 \\ \neq & \int_{x_i} \cdots \int_{x_n} f(x_1, ..., x_n) \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f(x_i)} dx_n \cdots dx_1. \end{aligned}$$

No obstante lo anterior, se puede probar que (3.14)$\leq$(3.15)+M_2, con M_2 constante. Para ello, recuérdese que f es acotada, por tanto existe $M \in \mathbb{R}^+$ tal que para todo x_i en el dominio de f

$$0 \leq f(x_i) \leq M$$

de donde

$$\frac{1}{f(x_i)} \geq \frac{1}{M} \geq 0$$

por tanto

$$\frac{f\left(x_i, x_{j(i)}\right)}{f(x_i) f\left(x_{j(i)}\right)} \geq \frac{f\left(x_i, x_{j(i)}\right)}{M f\left(x_{j(i)}\right)}.$$

Además, como la función log *es creciente,*

$$\log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f(x_i)} \geq \log \frac{f\left(x_i, x_{j(i)}\right)}{M f\left(x_{j(i)}\right)}$$

luego

$$\log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f(x_i)} \geq \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} - \log M,$$

y

$$\begin{aligned}
\sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f\left(x_i\right)} &\geq \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} - \sum_{i=2}^{n} \log M \\
\sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f\left(x_i\right)} &\geq \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} - (n-1) M \\
\sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f\left(x_i\right)} &\geq \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} - M_2,
\end{aligned}$$

donde $M_2 = (n-1) M$ *es una constante que no depende de* t*, el árbol de dependencia utilizado.*

De lo anterior, se tiene que

$$\begin{aligned}
&\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right)} dx_n \cdots dx_1 - \int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) M_2 dx_n \cdots dx_1 \\
\leq &\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log \frac{f\left(x_i, x_{j(i)}\right)}{f\left(x_{j(i)}\right) f\left(x_i\right)}
\end{aligned}$$

es decir,

$$\begin{aligned}
&\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1 - M_2 \\
\leq &\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log \frac{f\left(A_i, A_{j(i)}\right)}{f\left(A_{j(i)}\right) f\left(A_i\right)} dx_n \cdots dx_1 \qquad (3.17) \\
\equiv &\sum_{i=2}^{n} I\left(x_i, x_{j(i)}\right).
\end{aligned}$$

Luego

$$\begin{aligned}
&-\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_1 \cdots dx_n \\
\geq &-\int_{x_i} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{i=2}^{n} \log \frac{f\left(A_i, A_{j(i)}\right)}{f\left(A_{j(i)}\right) f\left(A_i\right)} dx_1 \cdots dx_n - M_2 \qquad (3.18) \\
\equiv &-\sum_{i=2}^{n} I\left(x_i, x_{j(i)}\right) - M_2.
\end{aligned}$$

Recuérdese de la expresión (3.10) que la divergencia de Kullback-Leibler entre la densidad f *y la aproximación proporcionada por el árbol* t *tiene la forma*

$$D\left(f \| P_t\right) = A - B - \int_{x_1} \cdots \int_{x_n} f\left(x_1, ..., x_n\right) \sum_{\substack{i=1 \\ j(i) \neq 0}}^{n} \log f\left(x_i \mid x_{j(i)}\right) dx_n \cdots dx_1$$

donde A y B no dependen de t. Luego, por (3.18),

$$D\left(P\|P_t\right) \geq A - B - \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right) - M_2$$

Pero M_2 tampoco depende de t, y entonces

$$\begin{aligned} \min_t D\left(P\|P_t\right) &\geq \min\left[-\sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right)\right] \\ \min_t D\left(P\|P_t\right) &\geq \max \sum_{\substack{i=1 \\ j(i)\neq 0}}^{n} I\left(x_i, x_{j(i)}\right). \end{aligned}$$

Esto implica que si se obtiene el árbol de dependencia t de peso máximo, la distribución de probabilidad que define dicho árbol, f_t, es la que minimiza el valor de la divergencia de Kullback-Leibler con respecto a f, con lo cual queda concluida la demostración.
■

Capítulo 4

ADNMB

En este trabajo Tesis se presenta una propuesta de Algoritmo de Estimación de Distribución Continuo, basado en relaciones bivariadas, por lo cual se han revisado previamente los principales algoritmos de este tipo, como MIMIC [11], BMDA [35] y la obtención de un bosque de dependencias que aproxime a una distribución de probabilidad conjunta según Chow y Liu [14]. Se llama a este algoritmoque se propone aquí ADNMB, por *Aproximación de una Distribución Normal Multivariada mediante un Bosque* de dependencias bivariadas.

4.1. Algoritmo ADNMB

Las características principales del algoritmo que se propone son:

1. Supone que la función de densidad de probabilidad conjunta se puede factorizar mediante un bosque de relaciones bivariadas, al estilo de BMDA (véase la Sección 2.3), o bien, mediante un conjunto de árboles de relaciones bivariadas al estilo de los producidos por el Algoritmo de Chow y Lui (véase la Sección 3.4).

2. Utiliza distribuciones normales bivariadas en donde las variables involucradas son dependientes entre sí lo cual, tratándose de variables gaussianas equivale a tener que el coeficiente de correlación, ρ, tiene un valor distinto de cero.

3. Construye un bosque de dependencias de la siguiente forma:

   a) La raíz de cada árbol es la que tiene la mínima entropía de entre todas las variables que aún no han sido procesadas, como se hace en MIMIC.

   b) Se agregan solamente relaciones significativas, considerando como tales aquellas en las que el valor absoluto del coeficiente de correlación es significativamente (en el sentido estadístico) mayor que un cierto valor establecido. Se crean uno o más árboles o grafos de dependencia en donde las relaciones son bivariadas. Esto es similar a lo realizado en BMDA

c) Las variables que se van agregando a cada árbol del bosque maximizan la suma de la información mutua en las ramas de dicho árbol, del mismo modo en que se hace en el algoritmo de Chow-Liu.

4. Aproxima los cinco parámetros del modelo presentes en una distribución normal bivariada con correlación mediante sus estimadores de máxima verosimilitud.

Para establecer la forma general de las funciones de densidad de probabilidad estimadas que proporciona ADNMB, recordemos que en el algoritmo de Chow y Liu las fdps tienen la forma (véase p. 48):

$$f(x_1, x_2, ..., x_n) = f(x_{i_1}) f\left(x_{i_2}|x_{j(i_2)}\right) ... f\left(x_{i_n}|x_{j(i_n)}\right)$$

en donde $i_1, i_2, ..., i_n$ es una permutación de los índices $1, ..., n$; y $j(i_l) \in \{1, ..., n\} \backslash \{i_l\}$ es la variable que condiciona a i_l, para $l = 2, ..., n$. En ADNMB lo que se tiene son k árboles de este estilo:

$$\text{Árbol 1: } f\left(x_{i_{11}}, x_{i_{12}}, ..., x_{i_{1n_1}}\right) = f(x_{i_{11}}) f\left(x_{i_{12}}|x_{j(i_{12})}\right) ... f\left(x_{i_{1n_1}}|x_{j\left(i_{1n_1}\right)}\right)$$

$$\text{Árbol 2: } \quad f\left(x_{i_{21}}, x_{i_{22}}, ..., x_{i_{2n_1}}\right) = f(x_{i_{21}}) f\left(x_{i_{22}}|x_{j(i_{22})}\right) ... f\left(x_{i_{2n_1}}|x_{j\left(i_{2n_1}\right)}\right)$$

$$\vdots$$

$$\text{Árbol } k\text{: } f\left(x_{i_{k1}}, x_{i_{k2}}, ..., x_{i_{kn_k}}\right) = f(x_{i_{k1}}) f\left(x_{i_{k2}}|x_{j(i_{k2})}\right) ... f\left(x_{i_{kn_k}}|x_{j\left(i_{kn_k}\right)}\right)$$

con lo que la función de densidad de probabilidad conjunta asociada a las n variables $X_1, X_2, ...X_n$ es

$$f(x_1, ..., x_n) = \prod_{l=1}^{k} f(x_{i_l 1}) f\left(x_{i_l 2} \,|\, x_{j(i_l 2)}\right) \dots f\left(x_{i_{l_{n_l}}}|x_{j\left(i_{l_{n_l}}\right)}\right) \tag{4.1}$$

donde $A_1 := \{i_{l1}, ..., i_{k_1}\} \subset \{1, ..., n\}$, $A_2 =: \{1, ..., n\} \setminus A_1$ y para todo $l \in \{1, ..., n\}$ y $l_2 \in A_2$, se cumple que $j(i_{ll_2}) \in \{1, ..., n\} \setminus \{j(i_{ll_2})\}$ es el subíndice de la variable que condiciona la densidad de la variable i_{ll_2}.

A manera de ejemplo, consideremos la Figura 4.1. La fdp conjunta es

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = [f(x_1) f(x_2|x_1) f(x_6|x_1) f(x_5|x_1)] [f(x_3) f(x_4|x_3)]$$

es decir el árbol 1 es

$$f(x_1) f(x_2|x_1) f(x_6|x_1) f(x_5|x_1)$$

y el árbol 2 es

$$f(x_3) f(x_4|x_3)$$

El resultado más importante de este trabajo de Tesis tiene que ver con que el bosque definido por ADNMB es el mejor estimador de la fdp conjunta. Esto se demuestra en la siguiente

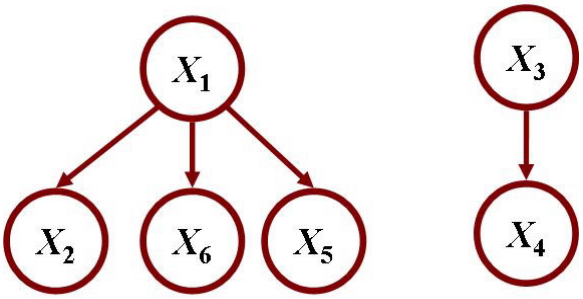


Figura 4.1: Forma típica de un grafo obtenido mediante ADNMB para aproximar una distribución conjunta continua

Afirmación 35 *El método ADNMB es el mejor bosque para estimar la densidad conjunta, suponiendo que esta se puede expresar mediante densidades condicionales marginales y bivariadas según la expresión (4.1).*

Prueba. *Sea la función de densidad de probabilidad conjunta f. Supongamos que esta fdp se puede expresar en la forma dada por la expresión (4.1), es decir,*

$$f(x_1, ..., x_n) = \prod_{l=1}^{k} f(x_{i_{l1}}) f\left(x_{i_l 2} \mid x_{j(i_l 2)}\right) \dots f\left(x_{i_{ln_l}} | x_{j\left(i_{ln_l}\right)}\right)$$

Entonces se puede escribir de la forma

$$f(x_1, ..., x_n) = \prod_{l=1}^{k} t_l$$

con

$$t_l = f(x_{i_{l1}}) f\left(x_{i_l 2} \mid x_{j(i_l 2)}\right) \dots f\left(x_{i_{ln_l}} | x_{j\left(i_{ln_l}\right)}\right); \quad l = 1, ..., k$$

y dado que las funciones de densidad de probabilidad marginal y las de densidad de probabilidad conjunta son no negativas

$$\begin{aligned} \max f(x_1, ..., x_n) &= \max \prod_{l=1}^{k} t_l \\ &= \prod_{l=1}^{k} \max t_l. \end{aligned}$$

Ahora, para cada $l = 1, ..., k$, t_l es un árbol de Chow y Liu, así que por la Afirmación 34 de la p. 51, la fdp conjunta aproximada por ADNMB es el mejor bosque para aproximar a la verdadera fdp conjunta f. ■

La idea general de ADNMB se presenta en el Algoritmo 4.1:

La construcción del grafo de dependencias se explica mediante el Algoritmo 4.2, en donde se utilizan las siguientes definiciones y notación:

Algoritmo 4.1 Aproximación de una distribución Normal Multivariada mediante un Bosque de Chow y Liu, ADNMB

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime $\widehat{\mu}_{MV}$ = Estimador de máxima verosimilitud del vector de medias, μ, y $\widehat{\Sigma}_{MV}$ = Estimador de máxima verosimilitud de la matriz de covarianzas, Σ
5: Construya el grafo de dependencias $G = (V, E, R)$ a partir de $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
6: $hijos$ = Genere aleatoriamente una población de tamaño m con base en G, $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
7: $pob_{previa} = padres \cup hijos$
8: pob = Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
9: **end while**

- **Notación 36** *Para ADNMB*

  - *Vértice := Variable*
  - *Arista := Relación de dependencia entre un par de variables*
  - D := *Conjunto de aristas tales que las variables participantes son dependientes entre sí*
  - V := *Conjunto de vértices*
  - A := *Conjunto de vértices sin procesar hasta el momento*
  - E := *Conjunto de aristas en el grafo final*
  - R := *Conjunto de vértices raíz del grafo de dependencias*
  - $I(X_u, X_w)$:= *Información entre las variables* X_u *y* X_w.

Inicialmente, se toma la variable que tiene la mínima entropía y se le coloca como raíz de un árbol de dependencias. Posteriormente, se van agregando aristas a la estructura actual, de modo que cada arista recién agregada sea aquella que proporcione, de entre todas las posibles, la máxima información mutua entre un vértice de alguno de los árboles existentes y un vértice que no ha sido procesado. Cuando no es posible agregar más aristas al árbol actual, el siguiente nodo o vértice con la mínima entropía de los que no han sido procesados pasa a formar parte del conjunto de raíces R.

La construcción del conjunto de aristas dependientes, D, es una parte importante en la obtención del grafo de dependencias. Recuérdese que en este conjunto, ADNMB incluye solamente las relaciones que sean significativamente mayores, en valor absoluto, que un valor de referencia ρ_0. En la puesta a prueba del algoritmo se utilizaron los valores ρ_0 = -0.50, -0.20, 0.20, 0.50.

Algoritmo 4.2 Construcción del grafo de dependencias para ADNMB

1: Sean $V = \{1, 2, ..., n\}$, $A = V$, $E = \emptyset$, $R = \emptyset$, $B = A \times V$
2: **while** $B \neq \emptyset$ **do**
3: Sea $Xv = \arg\min_{X \in A} Var(X)$
4: Sea $A = A \setminus \{v\}$
5: Sea $R = R \cup \{v\}$
6: **while** $B \cap D \neq \emptyset$ **do**
7: Sea
$$(v, v') = \arg\max_{(u,w) \in B \cap D} I(X_u, X_w)$$
8: Sea $E = E \cup \{(v, v')\}$
9: Sea $A = A \setminus \{v, v'\}$
10: Sea $B = A \times (V \setminus A)$
11: **end while**
12: **end while**

Para determinar si existía evidencia estadística con respecto al valor de ρ en alguno de los siguientes dos sentidos:

$$\rho > \rho_0 > 0$$

o bien

$$\rho < \rho_0 < 0,$$

se utilizó el contraste de hipótesis para comparar el coeficiente de correlación contra cualquier valor específico (véase [42, pp. 413-415]), a saber,

$$\begin{aligned} H_0 &: \rho = \rho_0 \\ H_a &: \rho < \rho_0 \end{aligned}$$

para ρ_0 = -0.50, -0.20 y

$$\begin{aligned} H_0 &: \rho = \rho_0 \\ H_a &: \rho > \rho_0 \end{aligned}$$

para $\rho_0 = 0.20, 0.50$.

Puede probarse [42, pp. 413-415] que si $(X, Y) \sim N_2(\mu, \Sigma)$ y $\widehat{\rho}$ es el estimador del coeficiente de correlación entre X y Y, calculado a partir de una muestra de tamaño m, entonces

$$Z = \frac{1}{2} \log\left(\frac{1 + \widehat{\rho}}{1 - \widehat{\rho}}\right)$$

se distribuye aproximadamente $N\left(\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right), \frac{1}{m-3}\right)$; así que un estadístico de contraste que puede usarse es

$$Z_{Calc} = \frac{\sqrt{m-3}}{2} \log\left(\frac{[1 + \widehat{\rho}]\,[1 - \rho]}{[1 - \widehat{\rho}]\,[1 + \rho]}\right)$$

el cual se compara contra los cuantiles de la distribución normal estándar. En los experimentos conducidos, se comparó Z_{Calc} contra $Z_{Tabla} = Z_\alpha = Z_{0.025} = 1.96$.

En el Algoritmo 4.3 se describe cómo se lleva a cabo la construcción de D. Se denota la $ij - ésima$ entrada de la matriz $\widehat{\Sigma}_{MV}$ por

$$\begin{cases} \widehat{\sigma}_{ij}, \text{ si } i \neq j \\ \widehat{\sigma}_i^2, \text{ si } i = j \end{cases}$$

Algoritmo 4.3 Construcción del conjunto de aristas dependientes en ADNMB

1: Sea $D = \emptyset$
2: **for** $i = 1, \ldots, n$ **do**
3: **for** $j = i + 1, \ldots, n$ **do**
4: Calcule el estimador $\widehat{\rho} = \widehat{\sigma}_{ij} / (\widehat{\sigma}_i \widehat{\sigma}_j)$
5: Calcule $Z_{Calc} = \frac{1}{2}\sqrt{n-3}\log\left(\frac{[1+\widehat{\rho}][1-\rho_0]}{[1-\widehat{\rho}][1+\rho_0]}\right)$
6: Si $|Z_{Calc}| > Z_{Tabla}$, haga $D = D \cup \{(i, j)\}$
7: **end for**
8: **end for**

El Algoritmo 4.4 presenta la forma de realizar la simulación de la siguiente generación en ADNMB, la cual dada la forma en que se concibió combina las características de Chow y Liu para variables gaussianas (se requiere estimar μ y Σ) y de BMDA (puede haber más de una raíz).

Algoritmo 4.4 Recorrido del grafo de dependencias en el ADNMB

Input: padre $\equiv X_{j(i)}$, nodo $\equiv X_i$
1: **if** $X_i \in R$ **then** // El nodo es una raíz
2: Simule m valores distribuidos $N(\widehat{\mu}_{X_i}, \widehat{\Sigma}_{X_i})$
3: **else** // El nodo X_i está condicionado por $X_{j(i)}$
4: Calcule $\widehat{\mu}_{X_i|X_{j(i)}}$ y $\widehat{\Sigma}_{X_i|X_{j(i)}}$
5: Simule m valores distribuidos $N(\widehat{\mu}_{X_i|X_{j(i)}}, \widehat{\Sigma}_{X_i|X_{j(i)}})$
6: **end if**
7: **while** X_i tenga hijos **do**
8: Ejecute el Algoritmo 4.4 con nodo, hijo
9: **end while**

4.2. Funciones de prueba

Se tomó como referencia el texto de Pedro Larrañaga y José A. Lozano de 2001, *"Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation"* [25], en el cual se reportan los resultados de someter a prueba siete Algoritmos de Estimación de Distribución y una Estrategia Evolutiva:

- UMDA$_c$, versión continua de UMDA, en donde la densidad conjunta se calcula como el producto de las densidades marginales que son normales univariadas [32], [24].
- MIMIC$_c$, versión continua de MIMIC. La densidad conjunta se factoriza mediante un modelo de tipo "cadena" utilizando medias y varianzas [11], [25].
- EGNA$_{BIC}$, "la factorización se realiza mediante una red Gaussiana, se utiliza el criterio de máxima verosimilitud penalizada para la búsqueda del modelo en cada generación y la heurística de búsqueda del modelo es una búsqueda local" [25].
- EGNA$_{BGe}$, "similar al anterior, pero hace uso de una métrica Bayesiana" [25].
- EGNA$_{ee}$, utiliza contrastes de hipótesis para seleccionar arcos en la estructura de una red Gaussiana [24].
- EMNA$_{global}$, hace uso de una distribución normal multivariada [24].
- EMNA$_a$, también utiliza una distribución normal multivariada [24].
- ES, una estrategia evolutiva $(\mu + \lambda)$ con recombinación [6].

Se probó el algoritmo ADNMB con las siguientes funciones (véase también [37], [2]):

1. **Esfera**

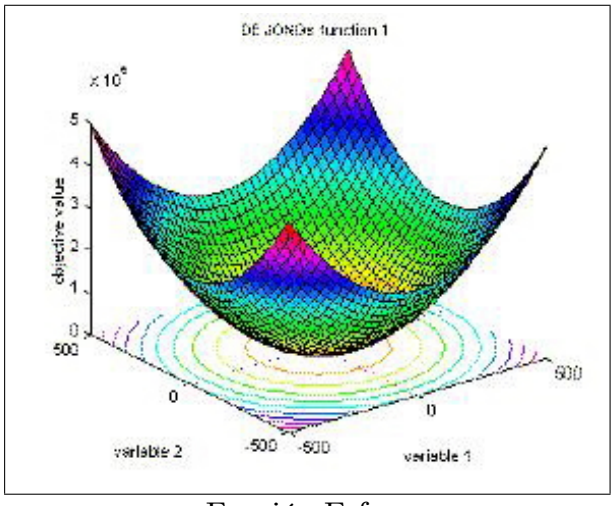
$$f_{esfera}(x) = \sum_{i=1}^{n} x_i^2; \qquad -600 \leq x_i \leq 600, \quad i = 1, ..., n$$

Es un problema muy simple de resolver, es una función convexa y unimodal. El óptimo es

$$f_{esfera}(x_{\text{mín}}) = 0$$

y lo alcanza en

$$x_{\text{mín}} = (0, 0, ..., 0)^T.$$



Función Esfera

2. **Rosenbrock**

$$\begin{aligned} f_{rosen}(x) &= \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (1 - x_i)^2 \right]; \\ -10 &\leq x_i \leq 10, \quad i = 1, ..., n \end{aligned}$$

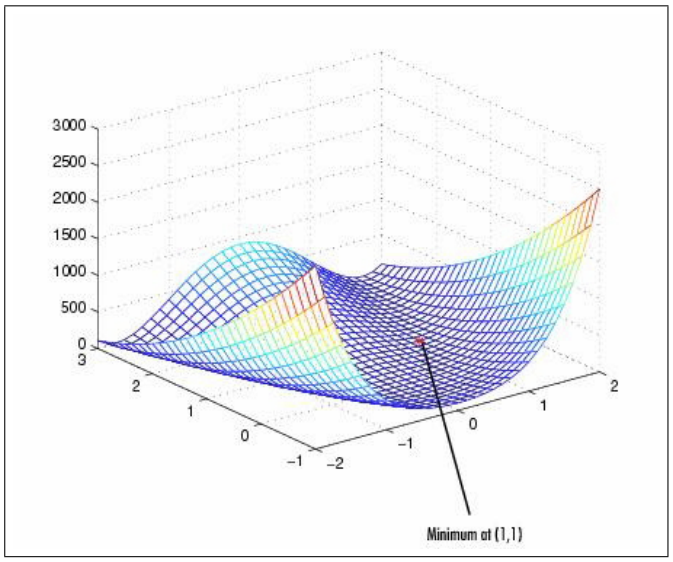
Es un problema originalmente propuesto, para dos dimensiones, en 1960 por Rosenbrock [38]; posteriormente, en 1998, Salomon lo generalizó a n dimensiones. El óptimo se encuentra dentro de un valle parabólico plano, largo y angosto. Alcanzar dicho valle es simple, pero después la consecución del óptimo es complicada, razón por la cual este es un problema clásico de optimización, generalmente utilizado para medir el desempeño de algoritmos a este fin.

El óptimo es

$$f_{rosen}(x_{\text{mín}}) = 0$$

en

$$x_{\text{mín}} = (1, 1, ..., 1)^T .$$



3. **Cancelación de suma**

$$\begin{aligned} f_{sumCan}(x) &= \frac{1}{10^{-5} + \sum_{i=1}^{n} |y_i|}; \quad y_1 = x_1,\ y_i = x_i + y_{i-1},\ i = 2, ..., n \\ -0{,}16 &\leq x_i \leq 0{,}16, \quad i = 1, ..., n \end{aligned}$$

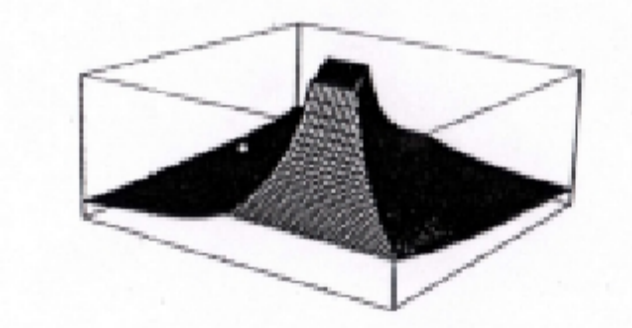
Es una función introducida por Baluja y Davies en 1997 [8], de difícil resolución, ya que pequeños cambios en los valores de entrada producen grandes cambios en la salida.

El óptimo es

$$f_{sumCan}\left(x_{\text{máx}}\right) = 10^5$$

en

$$x_{\text{máx}} = (0, 0, ..., 0)^T .$$



4. **Griewangk**

$$\begin{aligned} f_{griewagk}\left(x\right) &= 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right); \\ -600 &\leq x_i \leq 600, \quad i = 1, ..., n \end{aligned}$$

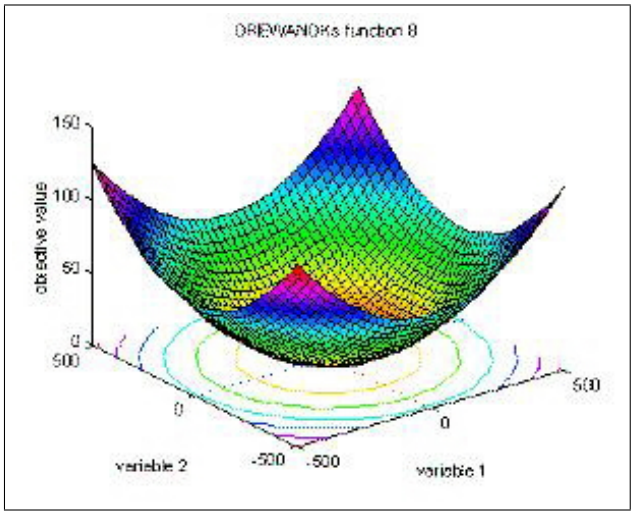
Propuesto por Törn y Žilinskas en su texto de 1989 [41], este problema tiene un gran número de mínimos locales, los cuales están distribuidos regularmente en la región de búsqueda.

El óptimo es

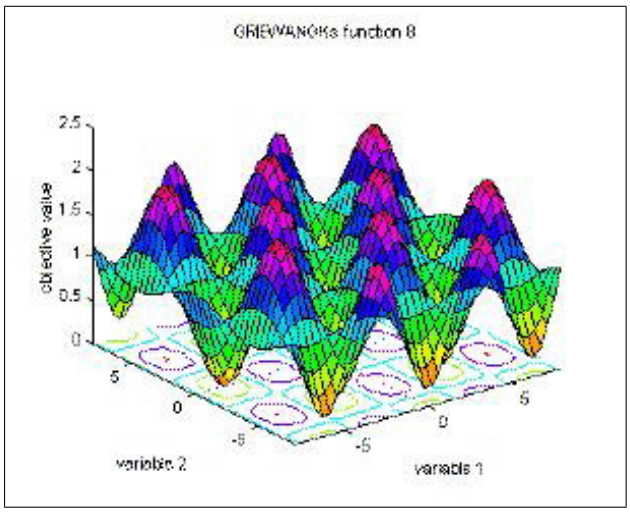
$$f_{griewangk}\left(x_{\text{mín}}\right) = 0$$

en

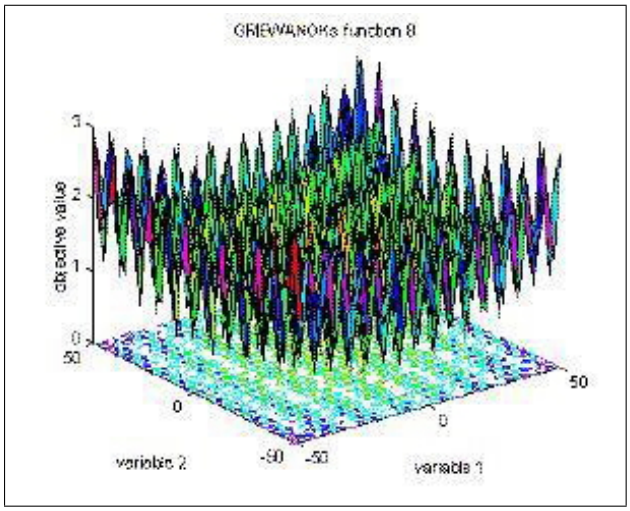
$$x_{\text{mín}} = (0, 0, ..., 0)^T .$$



Función de Griewangk, vista lejana



Griewangk, acercamiento 1



Griewangk, acercamiento 2

5. **Ackley**

$$\begin{aligned} f_{ackley}(x) &= -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) \\ &\quad +20 + \exp(1); \\ -10 &\leq x_i \leq 10, \quad i = 1, ..., n \end{aligned}$$

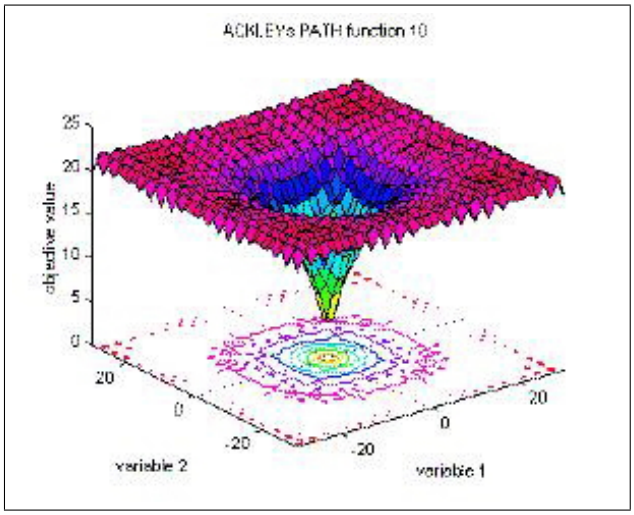
Presentada por primera vez por Ackley en su texto de 1987 [1] para dos dimensiones y posteriormente generalizada por Thomas Bäck en su libro de 1996 [5], esta función es un problema de minimización con múltiples mínimos locales.

El óptimo es

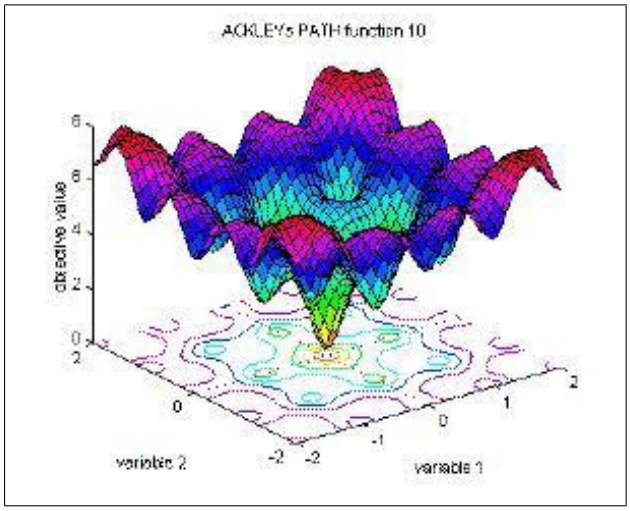
$$f_{ackley}(x_{\text{mín}}) = 0$$

en

$$x_{\text{mín}} = (0, 0, ..., 0)^T .$$



Ackley, vista 1



Ackley, acercamiento

4.3. Experimentos realizados y resultados obtenidos

Siguiendo el esquema dado en el texto citado de Larrañaga y Lozano [25]:

- Cada una una de estas funciones de prueba se resolvió en R^n con $n = 2, 10, 20$ y 50
- Se usaron los siguientes valores máximos para el esquema evolutivo: 200 individuos, 100 padres seleccionados, a lo más 301,850 evaluaciones de la función de aptitud

- La ejecución se detuvo cuando la diferencia entre el óptimo verdadero y el valor de aptitud alcanzado por el individuo de élite fuera menor que 10^{-6}, o bien se hubiera alcanzado el máximo permitido de evaluaciones de la función de aptitud (fijado en 301,850), o bien no se hubiera observado una mejora de al menos 10^{-6} en el individuo de élite en las últimas 25 generaciones

- Se hicieron 30 repeticiones de cada caso, con base en cuyos resultados se calcularon valores de media y desviación estándar del mejor individuo y del número de evaluaciones de la función objetivo

Los resultados obtenidos se presentan a continuación en el formato *media* ± *desviación estándar*. Se comparan en cada tabla los resultados producidos por ADNMB con aquellos obtenidos por el mejor EDA reportado en [25].

4.3.1. Esfera

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
2	ADNMB1	$2.94(10^{-7}) \pm 3.12(10^{-7})$	3,107 ± 155
10	ADNMB1	$7.10(10^{-7}) \pm 1.72(10^{-7})$	10,633 ± 183
	$UMDA_c$	$6.74(10^{-6}) \pm 1.26(10^{-6})$	74,164 ± 1,759
20	ADNMB1	$8.28(10^{-7}) \pm 1.12(10^{-7})$	16,807 ± 170
50	ADNMB1	$2.61(10^{-6}) \pm 3.81(10^{-6})$	34,040 ± 3,486
	$UMDA_c$	$8.91(10^{-6}) \pm 8.41(10^{-7})$	211,495 ± 1,264

4.3.2. Rosenbrock

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
2	ADNMB1	$4.05(10^{-7}) \pm 2.47(10^{-7})$	12,000 ± 1,886
10	ADNMB1	$7.49 \pm 2.04\ (10^{-1})$	301,850 ± 0
	$EGNA_{ee}$	$8.74 \pm 2.23(10^{-2})$	301,850 ± 0
20	ADNMB1	$1.90(10^{1}) \pm 6.89(10^{-1})$	14,840 ± 1,613
50	ADNMB1	$7.03(10^{1}) \pm 7.23(10^{1})$	27,633 ± 3,159
	$EMNA_{global}$	49.7588 ± 0.52	296,253 ± 7,287

4.3.3. Cancelación de suma

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
2	ADNMB1	$1.00(10^5) \pm 2.74(10^{-7})$	10,567 ± 217
10	ADNMB1	$2.52(10^4) \pm 2.28(10^4)$	301,850 ± 0
	$EGNA_{ee}$	$9.99992(10^4) \pm 1.73(10^{-1})$	195,903 ± 1,632
20	ADNMB1	$1.16\ (10^3) \pm 1.62(10^3)$	244,460 ± 120,895
50	ADNMB1	$3.87(10^{-1}) \pm 5.52(10^{-2})$	12,860 ± 4,648
	$EGNA_{ee}$	$8.62(10^4) \pm 8.91(10^3)$	301,850 ± 0

4.3.4. Griewangk

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
2	ADNMB1	$4.55(10^{-7}) \pm 3.10(10^{-7})$	7,407 ± 1,056
10	ADNMB1	$7.57(10^{-7}) \pm 1.81(10^{-7})$	11,413 ± 292
	$EGNA_{BIC}$	$3.93(10^{-2}) \pm 2.43(10^{-2})$	301,850 ± 0
20	ADNMB1	$8.11(10^{-7}) \pm 1.35(10^{-7})$	13,753 ± 239
50	ADNMB1	$8.77(10^{-7}) \pm 8.69(10^{-8})$	23,186 ± 458
	$EGNA_{BGe}$	$8.65(10^{-6}) \pm 7.71(10^{-7})$	173,514 ± 1,264

4.3.5. Ackley

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
2	ADNMB1	$6.03(10^{-7}) \pm 2.57(10^{-7})$	4,180 ± 177
10	ADNMB1	$8.40(10^{-7}) \pm 1.21(10^{-7})$	13,113 ± 155
	$EGNA_{ee}$	$7.50(10^{-6}) \pm 1.72(10^{-6})$	118,452 ± 2,318
20	ADNMB1	$8.93(10^{-7}) \pm 6.91(10^{-8})$	20,113 ± 180
50	ADNMB1	$1.02(10^{-3}) \pm 3.81(10^{-3})$	38,022 ± 4,086
	$EGNA_{BGe}$	$8.65(10^{-6}) \pm 3.79(10^{-7})$	282,060 ± 632

En general se observó que el método ADNMB:

- Supera los resultados de $UMDA_c^g$ en todas las funciones de prueba, excepto en la fución Cancelación de Suma. Esto era esperable, puesto que la forma de estimar la distribución de probabilidad conjunta que se utiliza en ADNMB es superior a la suposición de independencia entre todas las variables.

- Supera a MIMIC en todas las funciones de prueba, excepto también en Cancelación de Suma. Este resultado también era esperable dado que la forma de construir el grafo de dependencias en ADNMB es más general que la que permite MIMIC, de modo que ADNMB podría captar relaciones entre las variables que no pueden representarse con el modelo planteado por MIMIC.

- Es superior a todos los algoritmos examinados, de acuerdo con lo reportado en el texto de referencia de Larrañaga y Lozano [25], para $n = 10$, para todas las funciones consideradas (excepto Cancelación de Suma), en cuanto a:
  - Valor promedio del mínimo alcanzado
  - Precisión del valor mínimo alcanzado
  - Número de evaluaciones de la función de aptitud requerido para alcanzar el mínimo
  - Precisión del número de evaluaciones de la función aptitud requerido para alcanzar el mínimo
- Es superior a todos los algoritmos examinados en el texto de referencia de Larrañaga y Lozano [25], para $n = 50$, para todas las funciones consideradas (excepto Cancelación de Suma), en cuanto a:
  - Valor promedio del mínimo alcanzado
  - Precisión del valor mínimo alcanzado
  - Número de evaluaciones de la función de aptitud requerido para alcanzar el mínimo
  - Precisión del número de evaluaciones de la función aptitud requerido para alcanzar el mínimo
- No obstante el punto anterior, se observó que el algoritmo puede llegar a experimentar un "estancamiento" en un óptimo local en cierto momento de la ejecución, principalmente en dimensiones altas

4.4. Conclusiones sobre el desempeño de ADNMB

- Se observó que, en general, el algoritmo produce buenos resultados puesto que supera los resultados alcanzados por los algoritmos con una estructura más limitada en cuanto al tipo de relaciones entre variables que pueden considerar. También supera en casi todas las funciones a otros métodos que utilizan estructuras más complejas para modelar las relaciones entre las variables objetivo
- Examinando las soluciones producidas por el algoritmo se determinó que el estancamiento era producido por una reducción de la varianza de las mejores soluciones. Considerando lo anterior, se determinó que se requería de dotarlo de una forma de mantener la diversidad o de aumentarla en ciertos puntos de la ejecución. Las modificaciones que se realizaron al algoritmo, así como los resultados obtenidos se explican en el siguiente capítulo

Capítulo 5

ADNMB modificado

5.1. Modificaciones planteadas

En el Capítulo anterior se indicó que aunque el ADNMB básico resuelve de manera adecuada la mayoría de los problemas de prueba abordados, se observó que existían ciertas situaciones mejorables en su desempeño.

En particular se observó que, en algunas de las ejecuciones, la búsqueda del óptimo llegaba a un punto de "estancamiento" del que ya no le era posible recuperarse, provocando que se obtuviera como solución un mínimo local. Examinando el comportamiento de las distribuciones que el método estima, se hizo evidente que dicha falta de movilidad del método era debida a la pérdida de diversidad producida por la reducción extrema de la varianza. Como se sabe, dicha varianza (al igual que las medias) se estima de los individuos más aptos, por lo cual puede ocurrir que disminuya con mayor celeridad que la de el común de los individuos en la población. Algunas investigaciones presentadas en artículos de publicación relativamente reciente ya apuntan en el sentido de la pérdida de diversidad en la población de soluciones, principalmente por la contracción extrema de la varianza (véase por ejemplo [12] o [16]). Se consideró entonces la modificación del algoritmo para retrasar la contracción de la varianza y mantener la diversidad en consecuencia.

Un enfoque distinto para mejorar la localización del óptimo mediante EDAs consiste en proporcionar al método alguna forma de orientar la búsqueda de las mejores soluciones actuales. Por ejemplo, el EDA Direccional, introducido en el artículo de Mayorga y Hernández Aguirre [29], utiliza los individuos de las dos generaciones más recientes para proveer al EDA de dicho método de orientación durante la búsqueda. Este direccionamiento mejora el desempeño del EDA. Por tanto, además de controlar la contracción de la varianza, se decidió dotar al método de un procedimiento de direccionamiento que guiara el recorrido de las mejores soluciones.

En las secciones siguientes se examinan los métodos utilizados como base para la modificación de ADNMB.

5.2. IDEA con Escalamiento Adaptativo de Varianza Propiciado por Correlación (CT-AVS-IDEA)

En su artículo de 2006, Jörn Grahl, Peter A. N. Bosman y Franz Rothlauf [16] introducen el *Escalamiento Adaptativo de Varianza Propiciado por Correlación* (CT-AVS, por *Correlation Triggered Adaptive Variance Scaling*) aplicado al *Algoritmo Evolucionario de Estimación Iterada de Densidad* (IDEA, *Iterated Density Estimation Evolutionary Algorithm*), que denominan CT-AVS-IDEA.

En dicho artículo, Grahl *et al.* comentan sobre las dificultades que experimenta UMDA_c para resolver ciertos problemas de optimiazación, incluso algunos relativamente sencillos, y mencionan que en otros estudios se ha mostrado que el UMDA_c es capaz de alcanzar el óptimo solamente si comienza su búsqueda suficientemente cerca de aquél. Los autores establecen entonces que para que un modelo de probabilidad funcione adecuadamente como distribución de búsqueda en un EDA requiere tener dos características:

1. Ser adecuado, lo cual implica que pueda aproximar adecuadamente los contornos de la función objetivo con precisión arbitraria, con el fin de que pueda realizar la búsqueda del óptimo.

2. Ser competente, que significa que debe de hecho ser capaz de obtener una estimación adecuada de la disposición de las soluciones y simular los integrantes de la siguiente generación.

Grahl, Bosman y Rothlauf también separan en dos tipos las regiones que forman la apariencia de una función objetivo: picos y pendientes. También afirman que una fdp normal multivariada es adecuada para aproximar las curvas de nivel de la función de aptitud en torno a un pico u óptimo, de modo que si la búsqueda comienza cerca de este, entonces las selecciones sucesivas de individuos en cada generación son capaces de mover la media hacia el óptimo. Esto hace que el uso de estimaciones de los parámetros de la distribución normal multivariada sea un procedimiento competente para generar nuevas soluciones candidatas. Por otro lado, también apuntan Grahl *et al.*, las curvas de nivel de una región de tipo pendiente no concuerdan con una fdp normal multivariada, de modo que esta representación resulta inadecuada. Las soluciones obtenidas basándose en estimaciones de los parámetros de dicha distribución podrían estar desviadas hacia la media de esta, lo cual tendría como resultado la convergencia prematura de la búsqueda y, consecuentemente, que tal representación no resulte competente en las regiones de tipo pendiente.

En el artículo que se discute se propone una solución simple para evitar la convergencia prematura de la varianza, consistente en un coeficiente adaptativo de escalamiento de la varianza, c^{AEV} o c^{AVS} por *Adaptive Variance Scaling*. Para producir las nuevas soluciones se utiliza $c^{AEV}\Sigma$, en vez de Σ. Si se obtiene una mejora en la aptitud del individuo de élite, entonces el tamaño actual de la varianza permite que exista un progreso,

por tanto se aplica un mayor aumento en dicha varianza. Para evitar la contracción de la variabilidad debida a la selección se multiplica c^{AEV} por un factor $\eta^{inc} > 1$. Por otro lado, si no hubo mejoras en la aptitud, posiblemente el escalamiento de la varianza fue excesivo, por lo cual se le disminuye multiplicando c^{AEV} por un factor $\eta^{dism} \in [0, 1]$, el cual se elige (simplemente por simetría) como $\eta^{dism} = 1/\eta^{inc}$. Se definen también valores para acotar tanto por arriba como por abajo al coeficiente adaptativo de escalamiento: $c^{AEV\ \text{mín}} \leq c^{AEV} \leq c^{AEV\ \text{máx}}$, con $c^{AEV\ \text{mín}} = 1/c^{AEV\ \text{máx}}$. Si llegara a ocurrir que $c^{AEV} < c^{AEV\ \text{mín}}$, se fuerza $c^{AEV} = c^{AEV\ \text{mín}}$ para favorecer la exploración del espacio de búsqueda. De manera similar, si ocurre que $c^{AEV} > c^{AEV\ \text{máx}}$, se fuerza $c^{AEV} = c^{AEV\ \text{máx}}$ a fin de evitar que la búsqueda se disperse en una región demasiado amplia.

El escalamiento de la varianza es propiciado por el valor de r, la correlación existente entre la dispersión de la densidad estimada y la ubicación de los elementos de élite. Los autores del artículo presentan un valor umbral denominado θ^{corr} que determina el momento en que se requiere un escalamiento de la varianza. Si $r \leq \theta^{corr}$, entonces la estimación de la varianza de la distribución de los individuos de élite se utiliza sin modificación. Por el contrario, si $r > \theta^{corr}$, entonces se utiliza la varianza escalada, $c^{AEV}\Sigma$. Este procedimiento se presenta en el Algoritmo 5.1.

5.3. IDEA con Escalamiento Adaptativo de Varianza Propiciado por Tasa de Desviación Estándar (SDR-AVS-IDEA)

En 2007, Peter A. N. Bosman, Jörn Grahl, y Franz Rothlauf presentaron una modificación de su método previo, CT-AVS-IDEA, que se discutió en la sección anterior. Denominaron a este nuevo método *SDR-AVS-IDEA: Standard Deviation Ratio - Adaptive Variance Scaling - Iterated Density Estimation Evolutionary Algorithm*, es decir, *Algoritmo Evolucionario de Estimación Iterada de Densidad con Escalamiento Adaptativo de Varianza por Tasa de Desviación Estándar* [12]. Las principales modificaciones con respecto al método original son dos: la primera es que no se permite al multiplicador de la varianza ser menor que uno (debido a que esta característica les resultaba necesaria solamente si se desea encontrar nichos para optimizar funciones multimodales, y en este caso están interesados solamente en funciones unimodales). La segunda, y más importante, es que modifican el modo de establecer en qué momento se requiere el escalamiento de la varianza. Ahora la varianza se escala considerando la tasa o razón con respecto la desviación estándar de la ubicación de los individuos de élite en la población.

La observación fundamental de Bosman *et al.* consiste en lo siguiente: si las mejoras suceden principalmente lejos de la media actual de la población, entonces esto implica que se requiere un desplazamiento de la media hacia la posición de las mejoras. Por otro lado, si las mejoras no están demasiado lejos de la media global, entonces no se requiere este movimiento de la media.

Algoritmo 5.1 CT-AVS-IDEA

1: Sea $c^{AEV} = 1$
2: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
3: Sea el contador de generaciones $t = 1$
4: **while** no se cumpla el criterio de finalización **do**
5: $padres$ = Seleccione los $s_1 \leq m$ individuos más aptos
6: $elMejor(t)$ = Mejor aptitud del grupo seleccionado de $padres$
7: **if** $elMejor(t) = elMejor(t-1)$ **then**
8: $c^{AEV} = c^{AEV} \cdot \eta^{dism}$
9: **else**
10: $c^{AEV} = c^{AEV} \cdot \eta^{inc}$
11: **end if**
12: Calcule el coeficiente de correlación jerárquico, r
13: **if** $r > \theta^{corr}$ **then**
14: Sea $\Sigma = c^{AEV}\Sigma$
15: **end if**
16: Estime la distribución de probabilidad de los individuos seleccionados, P, mediante $\widehat{P}$
17: $hijos$ = Simule $s_2 \leq m$ individuos distribuidos según $\widehat{P}$
18: $pob_{previa} = padres \cup hijos$
19: pob = Los mejores m individuos de entre los $s_1 + s_2$ contenidos en pob_{previa}
20: Sea $t = t + 1$
21: **end while**

Hecha esta observación, los autores proponen calcular un valor umbral θ^{SDR} (*Standard Deviation Ratio*) o θ^{TDE} (Tasa de Desviación Estándar), tal que $\theta^{TDE} \in [0, \infty)$, mediante el cual se propicia el escalamiento de la varianza siempre y cuando exista algún $j \in \{1, ..., n\}$ tal que

$$TDE_j := \frac{\left|\overline{x}_j^{mejoras} - \mu_j\right|}{\sigma_j} > \theta^{TDE} \tag{5.1}$$

donde $\overline{x}^{mejoras}$ es la media aritmética de los individuos de élite y μ es la media de toda la población de soluciones.

El Algoritmo 5.2 resume este procedimiento.

Algoritmo 5.2 SDR-AVS-IDEA

Sea $c^{AEV} = 1$
$pob =$ Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
while no se cumpla el criterio de finalización **do**
  $padres =$ Seleccione los $s \leq m$ individuos más aptos
  $elMejor =$ Mejor aptitud en la población
  Calcule las estimaciones de los parámetros de la distribución, $\widehat{\mu}$ y $\widehat{\Sigma}$
  $hijos =$ Simule m individuos a partir de $\widehat{\mu}$ y $\widehat{\Sigma}$
  $n^{mejoras} =$ Número de individuos en $hijos$ con mejor valor de aptitud que $elMejor$
  if $n^{mejoras} > 0$ **then**
    Calcule el vector de medias de las mejoras, $\bar{x}^{mejoras}$
    Sea $SDR = \max\limits_{j \in \{1,...,n\}} \frac{|\bar{x}_j^{mejoras} - \mu_j^{global}|}{\sigma_j}$
    if $SDR > \theta^{TDE}$ **then**
      $c^{AEV} = c^{AEV} \cdot \eta^{inc}$
    end if
  else
    $c^{AEV} = c^{AEV} \cdot \eta^{dism}$
  end if
  if $c^{AEV} < 1$ **then**
    $c^{AEV} = 1$
  end if
  $pob_{previa} = padres \cup hijos$
  $pob =$ Los mejores m individuos de entre los $s + m$ contenidos en pob_{previa}
end while

Observación 37 *Aún cuando el artículo de Bosman et al. refiere literalmente que "se requiere un desplazamiento de la media", puede verse en el Algoritmo 5.2 que tal desplazamiento no se realiza, solamente se aumenta la varianza a través del valor de escalamiento.*

5.4. AED Direccional

Mayorga Álvarez y Hernández Aguirre [29] presentaron en 2008 un Algoritmo de Estimación de Distribución que introduce la idea de dotar la búsqueda de un método de direccionamiento. El enfoque consiste en combinar linealmente los padres más recientemente seleccionados con una población intermedia de hijos para obtener la nueva generación.

El direccionamiento de la búsqueda se consigue de la manera siguiente: sea t el contador de generaciones y considérese $u \in pob^t$, el $l-ésimo$ mejor vector de la población en el tiempo t. De la población en el tiempo $t+1$ tómese $v \in pob^{t+1}$, el correspondiente mejor vector, es decir, también aquel en la posición l. Con base en estos dos vectores se obtiene el vector modificado para la nueva generación:

$$v^* = u + \lambda_l^{t+1}(v - u)$$

donde el vector $\lambda_l^{t+1}(v-u)$ es la dirección de movimiento basada en los mejores individuos que buscan el óptimo. El tamaño de paso en el tiempo $t+1$, λ_l^{t+1}, se disminuye o aumenta teniendo en cuenta la evolución de la aptitud del $l-ésimo$ mejor individuo, $a_l := g(u)$, donde g es la función de aptitud, en las generaciones $t-2$, $t-1$ y t:

1. Si $a_l^{t-2} > a_l^{t-1} > a_l^t$ (asumiendo minimización), entonces $\lambda_l^t = 2\lambda_l^{t-1}$.
2. Si $a_l^{t-2} < a_l^{t-1} < a_l^t$ (asumiendo minimización), entonces $\lambda_l^t = \frac{1}{2}\lambda_l^{t-1}$.
3. En cualquier otro caso, $\lambda_l^t = \lambda_l^{t-1}$.

El pseudocódigo del EDA Direccional se presenta en el Algoritmo 5.3.

5.5. ADNMB Direccional con Desplazamiento de Media

Como se mencionó en la Sección 5.1, se dotó al *Algoritmo de Aproximación de una Distribución Normal Multivariada Mediante un Bosque* de dependencias bivariadas, ADNMB, de:

1. Un elemento direccional que permite guiar la búsqueda del óptimo. Esto se obtuvo modificando ideas presentadas en SDR-AVS y DEDA (Secciones 5.3 y 5.4).
2. Un método de mantener la diversidad, mediante la contención de la contracción extrema de la varianza. Esto se consiguió a través de un factor de escalamiento de la varianza inspirado en CT-AVS y SDR-AVS (véase las Secciones 5.2 y 5.3).

En esta sección se proporcionan los detalles sobre las modificaciones a ADNMB que fueron inspiradas en los métodos recién mencionados, a las cuales se agregó la aplicación de elementos de métodos estadísticos multivariados.

Algoritmo 5.3 EDA Direccional

1: Haga $t = 1$
2: Sea g la función de aptitud
3: Genere $pob = \{u_l^t \mid l = 1, \ldots, m\}$ un conjunto de m individuos distribuidos uniformemente en la región de búsqueda
4: Calcule la aptitud de cada individuo en la población, $g(u_l^t),\ l = 1, \ldots, m$
5: Sea $\mathcal{F}_l^t = \{g(u_l^k) \mid u_l^k \in pob^k;\ k = t, t-1, t-2\};\ l = 1, \ldots, m$
6: $\lambda_l^1 = 1;\ l = 1, 2, \ldots, m$
7: **while** no se cumpla el criterio de finalización **do**
8: Ordene pob^t ascendentemente con respecto a $g(u_l^t)$
9: $padres =$ Seleccione los $s \leq m$ individuos más aptos
10: Calcule las estimaciones de los parámetros de la distribución de la población a partir de $padres$
11: $hijos =$ Simule m individuos a partir de los parámetros estimados
12: **for** $l = 1, \ldots, m$ **do**
13: $u = pob_l^t,\ v = hijos_l$
14: Sea $v^* = u + \lambda_l^t(v - u)$
15: $pob_i^{t+1} = v^*$
16: **if** el conjunto $\mathcal{F}_i^t$ está en orden descendente (se asume minimización) **then**
17: $\lambda_l^{t+1} = 2\lambda_l^t$
18: **end if**
19: **if** el conjunto $\mathcal{F}_i^t$ está en orden ascendente (se asume minimización) **then**
20: $\lambda_l^{t+1} = \frac{1}{2}\lambda_l^t$
21: **end if**
22: **end for**
23: $t = t + 1$
24: **end while**

5.5.1. Elemento direccional en el ADNMB modificado

Recuérdese que en el algoritmo SDR-AVS los autores proponen escalar, por un valor mayor que 1, la varianza estimada siempre que la media de las mejores soluciones esté *suficientemente lejos* de la media global. A partir de la expresión (5.1) tenemos que se considera que ambas medias se encuentran lejos la una de la otra si

$$SDR \equiv \max_{j\in\{1,\ldots,n\}} TDE_j = \max_{j\in\{1,\ldots,n\}} \frac{\left|\overline{x}_j^{mejoras} - \mu_j\right|}{\sigma_j} > \theta^{TDE}.$$

Aunque los autores del artículo en cuestión no lo mencionan, nótese que cada TDE_j, $j \in \{1, ..., n\}$ es proporcional al estadístico de razón de verosimilitud de una prueba uniformemente más potente para $H_0 : \mu_j^{mejoras} = \mu_j$ contra $H_1 : \mu_j^{mejoras} \neq \mu_j$, puesto que dicho estadístico es

$$T_{Calc} = \frac{\overline{x}_j^{mejoras} - \mu_j}{\widehat{\sigma}_j/\sqrt{n}}$$

que bajo H_0 se distribuye T de Student con $n-1$ grados de libertad, es decir, $T_{Calc} \sim T(n-1)$. Por tanto, el valor de θ^{TDE} propuesto por Bosman *et al.* es igual a $\frac{1}{\sqrt{n}} t_{n-1,\alpha/2}$, donde $t_{n-1,\alpha/2}$ es el $(1-\alpha/2)-$ cuantil de la distribución $T(n-1)$ y α es la probabilidad de cometer un error tipo I, rechazar una hipótesis nula verdadera [31, pp. 428 - 431].

La observación anterior nos permite generalizar el planteamiento de Bosman *et al.* y dotar al método de un criterio estadístico para determinar el tamaño de $t_{n-1,\alpha/2}$: dependerá de α, el tamaño del error que se esté dispuesto a cometer al decir que la media de las mejoras está lejos de la media global cuando en realidad no lo está. Entonces el criterio SDR para una dimensión, generalizando estadísticamente, sería que se requiere aumentar la varianza en la dirección de X_j si

$$T_{Calc\,j} = \frac{\left|\overline{x}_j^{mejoras} - \mu_j\right|}{\widehat{\sigma}_j/\sqrt{n}} > t_{n-1,\alpha/2}. \tag{5.2}$$

Ahora bien, nótense dos puntos importantes:

1. El criterio SDR se usa para aumentar la varianza, pero en realidad lo que nos indica el contraste de hipótesis recién planteado en caso de rechazar H_0 es que los elementos de élite se encuentran lejos de la media de la distribución, por tanto esto lo que sugiere es que *hay que desplazar la media* global a la posición de la media de los élites para examinar una región más promisoria, y

2. Aunque el planteamiento dado en la expresión (5.2) resulta correcto en una dimensión, utilizar $\max_{j\in\{1,\ldots,n\}} T_{Calc\,j}$ para establecer si $\overline{x}^{mejoras}$ está alejada de μ no es lo más adecuado, puesto que tales medias podrían estar distantes en términos multivariados aún cuando no lo estuvieran en cada una de sus coordenadas consideradas por separado (véase, por ejemplo [28, pp. 120 - 122]).

Por tanto, para establecer estadísticamente de una manera más adecuada si las medias involucradas están suficientemente lejanas entre sí, se requiere tener en cuenta no cada dimensión por separado, sino todas en conjunto mediante un contraste de hipótesis multivariado.

Dicho contraste de hipótesis podemos basarlo en el siguiente resultado:

Afirmación 38 *Si $\widehat{\mu}$ y $\widehat{\Sigma}$ son, respectivamente, los estimadores del vector de medias y la matriz de covarianzas de una muestra de tamaño m de una distribución $N_n(\mu, \Sigma)$, entonces el estadístico*

$$F_{Calc} := \frac{m-n}{n} (\widehat{\mu} - \mu)' \widehat{\Sigma}^{-1} (\widehat{\mu} - \mu)$$

sigue la distribución F de Fisher-Snedecor con $m-n$ grados de libertad en el numerador y n grados de libertad en el denominador [28, p. 75].

Luego el contraste de hipótesis que plantea ADNMB modificado para establecer si se requiere desplazar la media global es:

$$\begin{aligned} H_0 &: \mu^{mejoras} = \mu \\ H_1 &: \mu^{mejoras} \neq \mu \end{aligned}$$

Rechazar H_0 implica que la media de las mejoras está alejada de la media global y por tanto se requiere desplazar esta última hacia la primera para examinar una región que podría reportar individuos con mejores aptitudes. Se rechazará H_0 al nivel α siempre que

$$F_{Calc} = \frac{m-n}{n} (\widehat{\mu} - \mu)' \widehat{\Sigma}^{-1} (\widehat{\mu} - \mu) > F_{m-n,n,\alpha}$$

donde $F_{m-n,n,\alpha}$ es el $(1-\alpha)-$ cuantil de la distribución F de Fisher-Snedecor con $m-n$ y n grados de libertad y α es la probabilidad de cometer un error tipo I, es decir, el riesgo que se desea correr de decir, con base en las observaciones de la muestra, que la media de las mejoras está alejada de la media global cuando en realidad no lo está. El desplazamiento de la media global depende del valor crítico, $F_{m-n,n,\alpha}$, y este depende de α. Si se escoge α cercano a cero, será difícil permitir modificar el valor de la media global, mientras que, por el contrario, si se elige el valor de α cercano a 1 se favorecerá el desplazamiento de la ubicación de las soluciones candidatas.

El procedimiento para desplazar la media en el ADNMB modificado se ilustra en el Algoritmo 5.4

5.5.2. Escalamiento de la matriz de covarianzas en el ADNMB modificado

En el apartado anterior se describió la forma de establecer si se requiere desplazar la media de la distribución hacia la media de los individuos más aptos. Ahora se discutirá el modo de escalar la varianza para favorecer la diversidad de las soluciones.

Algoritmo 5.4 Desplazamiento de la media en el ADNMB Direccional

Input: $\overline{x}^{mejoras}$, $\widehat{\mu}_{MV}$, $\widehat{\Sigma}_{MV}$, α

1: Calcule

$$F_{Calc} = \frac{n^{mejoras} - n}{n}(\overline{x}^{mejoras} - \widehat{\mu}_{MV})'\widehat{\Sigma}_{MV}^{-1}(\overline{x}^{mejoras} - \widehat{\mu}_{MV})$$

2: Sea $F_{Tabla} = F_{n^{mejoras}-n,n,\alpha}$

3: **if** $F_{Calc} - F_{Tabla}$ **then** // Las mejoras están lejos de la media global

4: $\quad \widehat{\mu}_{MV} = \overline{x}^{mejoras}$

5: **end if**

6: **Return:** $\widehat{\mu}_{MV}$

En los artículos de Bosman, Grahl y Rothlauf [12], [16] se define c^{AEV}, el factor que multiplica a la matriz de covarianzas Σ cuando se identifica que tal acción es requerida. Se utiliza entonces $c^{AEV}\Sigma$ como matriz de covarianzas para la siguiente generación.

Obsérvese que al multiplicar Σ por la constante de escalamiento se obtiene:

$$\begin{aligned} c^{AEV}\Sigma &= c^{AEV}\begin{bmatrix} \sigma^2_{X_1} & \sigma_{X_1X_2} & \cdots & \sigma_{X_1X_n} \\ \sigma_{X_1X_2} & \sigma^2_{X_2} & \cdots & \sigma_{X_2X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X_1X_n} & \sigma_{X_2X_n} & \cdots & \sigma^2_{X_n} \end{bmatrix} \\ &= \begin{bmatrix} c^{AEV}\sigma^2_{X_1} & c^{AEV}\sigma_{X_1X_2} & \cdots & c^{AEV}\sigma_{X_1X_n} \\ c^{AEV}\sigma_{X_1X_2} & c^{AEV}\sigma^2_{X_2} & \cdots & c^{AEV}\sigma_{X_2X_n} \\ \vdots & \vdots & \ddots & \vdots \\ c^{AEV}\sigma_{X_1X_n} & c^{AEV}\sigma_{X_2X_n} & \cdots & c^{AEV}\sigma^2_{X_n} \end{bmatrix}. \end{aligned}$$

Con lo anterior, cuando $c^{AEV} > 1$, se consigue aumentar el tamaño de las varianzas, lo cual debería favorecer la diversidad. Una cuestión susceptible de mejora en este método es que tal aumento se realiza por igual a las varianzas en todas las direcciones, pero existe la posibilidad de que resulte más provechoso hacer dicha ampliación en mayor medida en algunas direcciones que en otras.

Haciendo uso de la idea presentada en DEDA [29] de utilizar la generación actual y una intermedia para obtener la generación siguiente, para ADNMB se planteó lo siguiente:

1. Selecciónense los mejores individuos de la población actual para fungir como *padres*.

2. Estímense los parámetros de la distribución a partir de los *padres*.

3. Genérese una primera camada, $hijos^{interiores}$, con base en las estimaciones anteriores.

4. Obténgase la diferencia entre la población de hijos interiores y la población actual.

5. Determínese en qué direcciones crecen más tales diferencias y cuál es la magnitud de tal crecimiento.

6. Auméntese (o disminúyase) la varianza de cada dirección, proporcionalmente al tamaño del crecimiento de las diferencias en esa dirección.

7. Genérese la segunda camada, $hijos^{exteriores}$, con base en la media original y la nueva varianza.

8. Combínense $padres$, $hijos^{interiores}$ e $hijos^{exteriores}$ para obtener la nueva generación.

La parte medular de los pasos anteriores es la que encierran los puntos 4 a 6. Para la modificación de ADNMB, estos puntos se resolvieron según se explica en lo que resta de este apartado.

La diferencia entre la población de hijos interiores y la población actual se obtuvo calculando el vector diferencia entre cada elemento de $hijos^{interiores}$ y la media de la población actual, μ:

$$d_l = hijos_l^{interiores} - \mu; \quad l = 1, ..., m$$

Mientras mayores sean estas diferencias, querrá decir que más lejos estarán los élites del centro de la distribución, y esto nos hace pensar que entonces se debería favorecer la diversidad en las direcciones en las que ello ocurra. Tales direcciones son aquellas variables en las que la varianza de los d_l es mayor, por tanto, se requiere calcular la matriz de covarianzas de la matriz $D := [d_{lj}]$, Σ_d, para identificarlas.

A partir de resultados básicos de análisis multivariado, se sabe que un *componente principal* es una combinación lineal estandarizada y con varianza máxima de las variables originales. En este caso, el $j-ésimo$ componente principal de la matriz D, al cual se denotará por Y_j, es

$$Y_j = (D - 1\mu_d')\,\gamma_j; \quad j = 1, ..., n$$

donde μ_d es el vector de valores esperados de las columnas de D y γ_j es el *eigenvector* correspondiente al $j-ésimo$ mayor *eigenvalor* de la matriz de covarianza Σ_d. En particular, el primer componente principal representa la dirección de mayor variación de las variables originales [28, pp. 213 - 228], [27, pp. 75 - 90]. Se puede obtener fácilmente el *eigenvalor* dominante de una matriz definida positiva (como lo es una matriz de covarianzas), junto con su *eigenvector* asociado aplicando el método de Potencia, el método de Potencia Inverso o el método de Iteración del Cociente de Rayleigh (véase, por ejemplo, [33, pp. 407 - 418]). El primero de los mencionados se usó en la implementación del método y se describe en el Algoritmo 5.5.

Dado el primer componente principal, las variables afectadas por coeficientes positivos grandes contribuyen a aumentar el valor de aquel; por tanto, si se aumenta la varianza de dichas variables se aumentará la varianza del componente principal, y en

Algoritmo 5.5 Método de Potencia

Input: A

1: Sea $\gamma = \left(\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}}\right)'$
2: **while** no se cumpla el criterio de convergencia **do**
3: Calcule $\gamma = A\gamma$
4: Sea $\lambda = \underset{j=1,\ldots,n}{max}\{\gamma_j\}$
5: Sea $\gamma = \frac{1}{\lambda}\gamma$
6: **end while**
7: **Return:** *eigenvector* $= \gamma$, *eigenvalor* $= \lambda$

consecuencia favorecerá la diversidad en las dimensiones que sirven para construir la dirección de mayor dispersión.

Sea Y_{1i} el primer componente principal de la matriz D, evaluado en el $i - ésimo$ individuo; entonces Y_{1i} está dado por

$$Y_{1i} = \gamma_{11}x_{i1}^c + \cdots + \gamma_{1n}x_{in}^c,$$

donde $x_i^c = (x_{i1}^c, \ldots, x_{in}^c) = (x_{i1} - \mu_{di}, ..., x_{in} - \mu_{di})$ es el $i-ésimo$ individuo "centrado" (que se le ha sustraído su media). En virtud de que los valores de los $\gamma_{1j}, \quad j = 1, ..., n$ están normalizados, es decir, cumplen la restricción

$$\sum_{j=1}^{n} \gamma_{1j}^2 = 1,$$

si se cumpliera que

$$\gamma_{1j} = \frac{1}{\sqrt{n}}; \quad j = 1, ..., n$$

entonces el primer componente principal sería un promedio de todas las variables X_j y en tal caso sería aplicable el criterio de multiplicar cada componente de Σ, la matriz de covarianzas de las X_j, por la misma constante, c^{AEV}. Si $\gamma_{1j} > \frac{1}{\sqrt{n}}$ para algún $j = 1, ..., n$, entonces la dirección j amerita la aplicación de un mayor incremento en la varianza, en comparación con las otras direcciones. Finalmente, si $\gamma_{1j} < 0$ para algún $j = 1, ..., n$, entonces la $j - ésima$ variable no contribuye a producir un valor alto del primer componente principal, sino al contrario, por tanto puede recibir una penalización, disminuyendo su varianza.

Para escalar los valores de c^{AEV}, en ADNMB modificado se usa el factor η introducido también por Bosman, Grahl y Rothlauf, pero se le multiplica por un factor de varianza, f_v:

$$\eta_j^{t+1} = f_v \cdot \eta_j^t,$$

donde f_v puede tomar uno de tres valores

$$\begin{cases} \frac{1}{f_{v\,\text{máx}}} \leq f_v \leq \frac{1}{f_{v\,\text{mín}}}; & \text{si } \gamma_{1j} \in [-1, 0] \\ f_v = 1; & \text{si } \gamma_{1j} \in \left(0, \frac{1}{\sqrt{n}}\right) \\ f_{v\,\text{mín}} \leq f_v \leq f_{v\,\text{máx}}; & \text{si } \gamma_{1j} \in \left[\frac{1}{\sqrt{n}}, 1\right] \end{cases},$$

En la implementación se tomaron $f_{v\,\text{mín}} = 2$ y $f_{v\,\text{máx}} = 10$. Para el primer y el tercer caso, el valor particular de f_v es directamente proporcional a la diferencia entre γ_{1i} y su valor promedio, $\frac{1}{\sqrt{n}}$. De este modo se consigue favorecer la exploración en las direcciones de mayor variación en las diferencias de los elementos de élite con respecto a la media global, y desfavorecer la exploración en las direcciones en las que tales diferencias tienen poca variación.

Lo anteriormente explicado se resume en el Algoritmo 5.6 y ADNMB modificado se encuentra en el Algoritmo 5.7.

Algoritmo 5.6 Escalamiento de la varianza en el ADNMB Direccional

Input: $pob^{mejoras}$, $\widehat{\mu}_{MV}$, $\widehat{\Sigma}_{MV}$, η, α

Calcule $dif^{mejoras} = pob^{mejoras} - \widehat{\mu}'_{MV}$
Calcule $A = \text{Cov}(dif^{mejoras})$
Obtenga $\gamma_1 =$ el *eigenvector* asociado al *eigenvalor* dominante de A
for $j = 1, \ldots, n$ **do**
  if $\gamma_{1j} > \sqrt{\frac{1}{n}}$ **then**
    $f_v \propto \gamma_{1j} - \sqrt{\frac{1}{n}}$
  else if $\gamma_{1j} > 0$ **then**
    $f_v = 1$
  else
    $f_v \propto \frac{1}{\sqrt{\frac{1}{n}} - \gamma_{1j}}$
  end if
  $\eta_j = f_v \cdot \eta_j$
end for
for $i = 1, \ldots, n$ **do**
  for $j = 1, \ldots, n$ **do**
    $\widehat{\Sigma}_{MVij} = \sqrt{\eta_i} \cdot \sqrt{\eta_j} \cdot \widehat{\Sigma}_{MVij}$
  end for
end for
Return: $\widehat{\Sigma}_{MV}$

Algoritmo 5.7 Algoritmo ADNMB2

1: pob = Genere m individuos distribuidos uniformemente en la región de búsqueda y calcule su aptitud
2: **while** no se cumpla el criterio de finalización **do**
3: $padres$ = Seleccione los $s \leq m$ individuos más aptos
4: A partir de $padres$, estime $\widehat{\mu}_{MV}$ = Estimador de máxima verosimilitud del vector de medias, μ, y $\widehat{\Sigma}_{MV}$ = Estimador de máxima verosimilitud de la matriz de covarianzas, Σ
5: Construya el grafo de dependencias $G = (V, E, R)$ a partir de $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
6: $hijos^{interiores}$ = Genere aleatoriamente una población de tamaño m con base en G, $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
7: $hijos = hijos^{interiores}$
8: Actualice el valor de $\widehat{\mu}_{MV}$ mediante el Algoritmo 5.4
9: **if** se realizó algún cambio en $\widehat{\mu}_{MV}$ o $\widehat{\Sigma}_{MV}$ **then**
10: Actualice el valor de $\widehat{\Sigma}_{MV}$ mediante el Algoritmo 5.6
11: $hijos^{exteriores}$ = Genere aleatoriamente una población de tamaño m con base en G, $\widehat{\mu}_{MV}$ y $\widehat{\Sigma}_{MV}$
12: $hijos = hijos^{interiores} \cup hijos^{exteriores}$
13: **end if**
14: $pob_{previa} = padres \cup hijos$
15: pob = Los mejores m individuos de entre los $s+m$ o $s+2m$ contenidos en pob_{previa}
16: **end while**

5.6. Experimentos realizados y comparación de resultados

Se siguió de manera general el mismo esquema presentado en la Sección 4.3:

- Cada una una de estas funciones de prueba se resolvió en R^n con $n = 10$ y 50
- Se usaron los siguientes valores máximos para el esquema evolutivo: 200 individuos, 100 padres seleccionados, a lo más 301,850 evaluaciones de la función de aptitud
- La ejecución se detuvo cuando la diferencia entre el óptimo verdadero y el valor de aptitud alcanzado por el individuo de élite fuera menor que 10^{-6}, o bien se hubiera alcanzado el máximo permitido de evaluaciones de la función de aptitud (fijado en 301,850), o bien no se hubiera observado una mejora de al menos 10^{-6} en el individuo de élite en las últimas 25 generaciones
- Se hicieron 30 repeticiones de cada caso, con base en cuyos resultados se calcularon valores de media y desviación estándar del mejor individuo y del número de evaluaciones de la función objetivo

Los resultados obtenidos se presentan a continuación en el formato *media* ± *desviación estándar*. Se comparan en cada tabla los resultados producidos por ADNMB1 y ADNMB2 con aquellos obtenidos por el mejor EDA reportado en [25].

5.6.1. Esfera

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
10	ADNMB1	$7.10(10^{-7}) \pm 1.72(10^{-7})$	10,633 ± 183
	ADNMB2	$4.784(10^{-7}) \pm 2.51(10^{-7})$	13,600 ± 245
	$UMDA_c$	$6.74(10^{-6}) \pm 1.26(10^{-6})$	74,164 ± 1,759
50	ADNMB1	$2.61(10^{-6}) \pm 3.81(10^{-6})$	34,040 ± 3,486
	ADNMB2	$8.668(10^{-7}) \pm 5.36(10^{-8})$	36,267 ± 2,701
	$UMDA_c$	$8.91(10^{-6}) \pm 8.41(10^{-7})$	211,495 ± 1,264

5.6.2. Rosenbrock

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
10	ADNMB1	$7.49 \pm 2.04\ (10^{-1})$	301,850 ± 0
	ADNMB2	$8.87 \pm 1.01(10^{-2})$	12,560 ± 841
	$EGNA_{ee}$	$8.74 \pm 2.23(10^{-2})$	301,850 ± 0
50	ADNMB1	70.3 ± 72.3	27,633 ± 3,159
	ADNMB2	52.2 ± 6.03	31,300 ± 9,299
	$EMNA_{global}$	49.7588 ± 0.52	296,253 ± 7,287

5.6.3. Cancelación de suma

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
10	ADNMB1	$2.52(10^4) \pm 2.28(10^4)$	301,850 ± 0
	ADNMB2	$1.00000(10^5) \pm 9.15(10^{-4})$	61,267 ± 2,861
	$EGNA_{ee}$	$9.99992(10^4) \pm 1.73(10^{-1})$	195,903 ± 1,632
50	ADNMB1	$3.78(10^{-1}) \pm 5.52(10^{-2})$	12,860 ± 4,648
	ADNMB2	$2.7 \pm 9.47(10^{-1})$	35,320 ± 18,026
	$EGNA_{ee}$	$8.62(10^4) \pm 8.91(10^3)$	301,850 ± 0

5.6.4. Griewangk

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
10	ADNMB1	$7.57(10^{-7}) \pm 1.81(10^{-7})$	11,413 ± 292
	ADNMB2	$7.894(10^{-7}) \pm 2.28(10^{-7})$	17,560 ± 6,552
	$EGNA_{BIC}$	$3.93(10^{-2}) \pm 2.43(10^{-2})$	301,850 ± 0
50	ADNMB1	$8.77\ (10^{-7}) \pm 8.69\ (10^{-8})$	23,186 ± 458
	ADNMB2	$7.380(10^{-7}) \pm 1.15(10^{-7})$	26,150 ± 2,744
	$EGNA_{BGe}$	$8.65(10^{-6}) \pm 7.71(10^{-7})$	173,514 ± 1,264

5.6.5. Ackley

Dimensión	Algoritmo	Mejor valor de la función de aptitud	Número de evaluaciones
10	ADNMB1	$8.40(10^{-7}) \pm 1.21(10^{-7})$	13,113 ± 155
	ADNMB2	$7.72(10^{-7}) \pm 1.27(10^{-7})$	16,600 ± 316
	$EGNA_{ee}$	$7.50(10^{-6}) \pm 1.72(10^{-6})$	118,452 ± 2,318
50	ADNMB1	$1.02(10^{-3}) \pm 3.81(10^{-3})$	38,022 ± 4,086
	ADNMB2	$9.04(10^{-7}) \pm 7.23(10^{-8})$	35,800 ± 980
	$EGNA_{BGe}$	$8.65(10^{-6}) \pm 3.79(10^{-7})$	282,060 ± 632

En general se observó que el método ADNMB Direccional:

- Obtiene soluciones del mismo orden que ADNMB para $n = 10$ en todas las funciones de prueba.
- Para $n = 50$, supera los resultados del ADNMB original en las funciones Esfera, Cancelación de Suma y Ackley; y obtiene resultados del mismo orden que ADNMB original en las funciones Rosenbrock y Griewangk.
- Obtuvo mejores resultados que los reportados en el texto de referencia por el mejor método para $n = 10$ en todas las funciones de prueba.

- Obtuvo resultados mejores o del orden de los reportados en el texto de referencia para $n = 50$ en Rosenbrock, Griewangk y Ackley. Es importante hacer notar que ADNMB utiliza solamente relaciones bivariadas, mientras que los métodos reportados en el texto de referencia utilizan la distribución multivariada completa.

5.7. Conclusiones sobre el desempeño de ADNMB modificado

La dotación de ADNMB de un método de desplazamiento de la media y un método de escalamiento de la varianza le permiten ser competitivo con algoritmos de mucha mayor complejidad.

Estas mejoras aún resultan insuficientes en el caso de funciones como Rosenbrock o Cancelación de Suma, las cuales se ha visto que pueden ser solamente abordadas con éxito por métodos que utilizan relaciones de alta complejidad. Esto lleva a pensar que para poder resolver con mayor éxito funciones de este tipo, que cuentan con relaciones de alto grado entre variables se requiere de dotar al método de una forma de incorporar tales relaciones al grafo que aproxima la distribución.

5.8. Aportaciones principales de este trabajo

1. Desarrollo de un Algoritmo de Estimación de Distribución basado en relaciones bivariadas del cual se demuestra que es le mejor bosque de relaciones bivariadas para aproximar una distribución conjunta multivariada.

2. Generalización y demostración de la afirmación de Chow y Liu para un bosque de variables binarias.

3. Generalización y demostración de la afirmación de Chow y Liu para variables continuas con fdp acotada.

4. Generalización y demostración de la afirmación de Chow y Liu para un bosque de variables continuas con fdp acotada.

5. Aplicación de resultados de Estadística Matemática univariada y multivariada al contexto de los Algoritmos de Estimación de Distribución para la formalización de resultados presentados en algunos de los artículos de referencia.

6. Propuesta de ADNMB Direccional, un EDA Direccional inspirado en ideas de

   - CT-AVS-IDEA (Correlation Triggered Adaptive Variance Scaling Iterative Distribution Estimation Evolutionary Algorithm)
   - SDR EDA (Standard Deviation Ratio Estimation of Distribution Algorithm)

- DEDA (Directional EDA)
- Propiedades de la distribución normal bivariada
- Contraste estadístico de hipótesis univariadas
- Contraste estadístico de hipótesis multivariadas
- Análisis de componentes principales

5.9. Trabajo adicional por realizar

Algunas líneas de investigación que se podrían seguir con base en lo presentado en este trabajo, son las siguientes:

1. Valoración de la influencia de distintos valores para α, la probabilidad de cometer un error tipo I, en los distintos contrastes de hipóteesis utilizados
2. Valoración de la influencia de valores adicionales de comparación en el contraste del coeficiente de correlación, ρ
3. Exploración de modelos de relación entre variables de orden mayor que 2 y menor que el número de variables, para la construcción del grafo de dependencias

Bibliografía

[1] D. H. Ackley. "A connectionist machine for genetic hillclimbing". Kluwer, Boston (1987).

[2] E. Alba Torres, C. León Hernández, P. Isasi Viñuela, J. Gabarró Valles, and J. M. Sánchez Pérez. "Problemas de Prueba Del Proyecto TRACER". http://tracer.lcc.uma.es/problems/index.html, Málaga, España (2002).

[3] T. M. Apostol. "Cálculus. Volumen 2. Cálculo con Funciones de Varias Variables y Álgebra Lineal, con Aplicaciones a Las Ecuaciones Diferenciales y Las Probabilidades". Editorial Reverté, Barcelona (1985).

[4] T. Back. Optimization by means of genetic algorithms. In "Technical University of Ilmenau", pp. 163–169. Online]. Available: citeseer.ist.psu.edu/71967.html (1989).

[5] T. Bäck. "Evolutionary algorithms in theory and practice". Oxford University Press (1996).

[6] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of evolution strategies. In "Proceedings of the Fourth International Conference on Genetic Algorithms", pp. 2–9. Morgan Kaufmann (1991).

[7] S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. pp. 38–46. Morgan Kaufmann Publishers (1995).

[8] S. Baluja and S. Davies. Combining multiple optimization runs with optimal dependency trees. Technical Report, (1997).

[9] R. G. Bartle. "The Elements of Integration and Lebesgue Measure". John Wiley and Sons, Inc., Estados Unidos de América (1966).

[10] H.-G. Beyer and H.-P. Schwefel. Evolution strategies –a comprehensive introduction. *Natural Computing: an international journal* **1**(1), 3–52 (2002).

[11] J. D. Bonet, C. Isbell, and P. Viola. MIMIC: Finding Optima by Estimating Probability Densities. In "Advances in Neural Information Processing Systems (NIPS) 9", pp. 424–430 (1997).

[12] P. A. Bosman, J. Grahl, and F. Rothlauf. Sdr: a better trigger for adaptive variance scaling in normal edas. In "GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation", pp. 492–499, New York, NY, USA (2007). ACM.

[13] G. Casella and R. L. Berger. "Statistical Inference". Duxbury Press, EUA (1990).

[14] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on* **14**(3), 462–467 (1968).

[15] H. M. Deitel and P. J. Deitel. "Cómo Programar En C/C++". Prentice Hall Hispanoamericana, S. A., México (1995).

[16] J. Grahl, P. A. Bosman, and F. Rothlauf. The correlation-triggered adaptive variance scaling idea. In "GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation", pp. 397–404, New York, NY, USA (2006). ACM.

[17] R. M. Gray. "Entropy and Information Theory". Springer Verlag, Nueva York, EUA (1990).

[18] F. A. Graybill. "Theory and Applications of the Linear Model". Duxbury, Boston, Massachusetts (1976).

[19] N. Hansen. The cma evolution strategy: A tutorial (2005).

[20] J. H. Holland. "Adaptation in Natural and Artificial Systems". University of Michigan Press, Ann Arbor, EUA (1975).

[21] Institut National de Recherche en Informatique et Automatique (INRIA). "Scilab On Line Help". http://www.scilab.org/product/man/ Consultado 2007 - 2008, Francia (2008).

[22] I. Kang. "Crimson Editor Homepage". http://www.crimsoneditor.com Consultado 2007 - 2008 (2003).

[23] Y. Langsam, M. J. Augenstein, and A. M. Tenenbaum. "Estructuras de Datos con C y C++". Prentice Hall Hispanoamericana S. A., México (1996).

[24] P. Larrañaga. Algoritmos de estimación de distribuciones = computación evolutiva + modelos gráficos probabilísticos (2002).

[25] P. Larrañaga and J. A. Lozano. "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation". Kluwer Academic Publishers, Norwell, MA, USA (2001).

[26] P. Larrañaga, J. A. Lozano, and H. Mühlenbein. Estimation of distribution algorithms applied to combinatorial optimization problems. *Revista Iberoamericana de Inteligencia Artificial* **19**, 149 – 168 (2003).

[27] B. Manly. "Multivariate Statistical Methods. A Primer". Chapman and Hall CRC, EUA (2005).

[28] K. V. Mardia, J. T. Kent, and J. M. Bibby. "Multivariate Analysis". Academic Press, Reino Unido (1979).

[29] P. P. Mayorga-Alvarez and A. Hernández-Aguirre. The directional eda for global optimization. In "GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation", pp. 473–474, New York, NY, USA (2008). ACM.

[30] M. Mitchell. "An Introduction to Genetic Algorithms". MIT Press, Cambridge, MA, USA (1998).

[31] A. M. Mood, F. A. Graybill, and D. C. Boes. "Introduction to the Theory of Statistics". McGraw-Hill, EUA (1963).

[32] H. Mühlenbein and G. Paass. "From recombination of genes to the estimation of distributions I. binary parameters". Springer-Verlag (1996).

[33] B.Ñath Datta. "Numerical Linear Algebra and Applications". Brooks/Cole Publishing Company, EUA (1995).

[34] J.Ñocedal and S. J. Wright. "Numerical Optimization". Springer-Verlag, Nueva York, EUA (1999).

[35] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, "Advances in Soft Computing - Engineering Design and Manufacturing", pp. 521–535, London (1999). Springer-Verlag.

[36] K. B. Petersen and M. S. Pedersen. The matrix cookbook. http://matrixcookbook.com (2008).

[37] H. Pohlheim. "GEATbx: Genetic and Evolutionary Algorithm Toolbox for Use with Matlab". http://www.geatbx.com/docu/index.html (Consultado el 21 de agosto de 2006).

[38] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *Computer Journal* **3**, 175 – 184 (1960).

[39] C. E. Shannon. A mathematical theory of information. *Bell System Technical Journal* **27**, 379–423, 623–656 (1948).

[40] I. J. Taneja. "Generalized Information Measures and their Applications". http://www.mtm.ufsc.br/ taneja/book/node1.html (2001).

[41] A. Torn and A. Zilinskas. "Global optimization". Springer-Verlag New York, Inc., New York, NY, USA (1989).

[42] R. E. Walpole and R. H. Myers. "Probabilidad Y Estadística". McGraw-Hill Interamericana de México, México (1992).

[43] T. Weise. "Global Optimization Algorithms, Theory and Practice". http://www.it-weise.de/projects/book.pdf, Alemania (2008).

Apéndice A

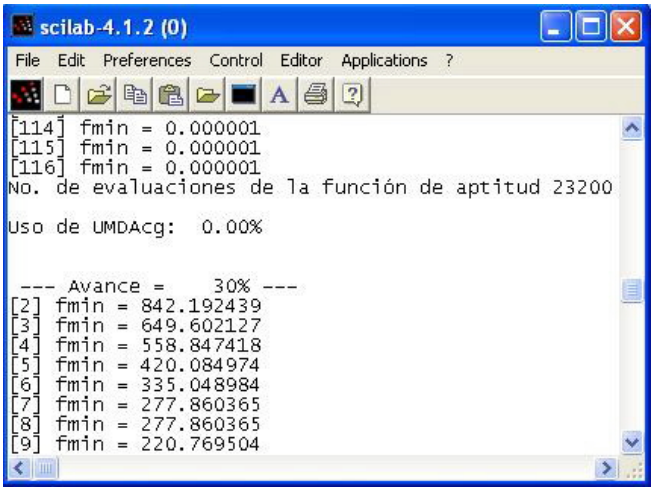
Software libre utilizado

En la realización de este trabajo de Tesis se hizo uso de algunos recursos de software libre. A continuación se describe brevemente cada uno.

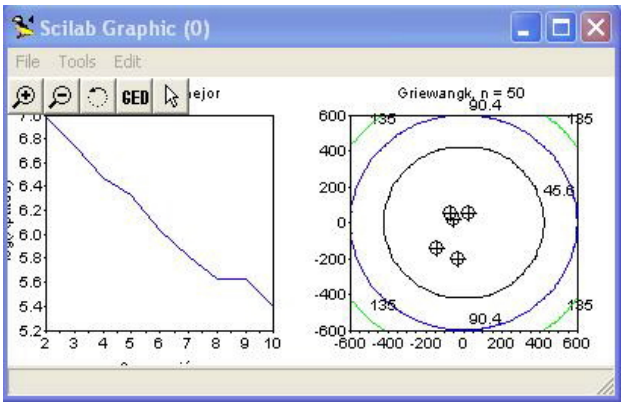
A.1. Scilab

Es una plataforma de código abierto para computación numérica [21], muy similar al software comercial *Matlab*. Se trata de un lenguaje intérprete de programación desarrollado por el *Scilab Consortium* dentro del Instituto Francés de Investigación en Informática y Control (*INRIA*, por sus siglas en francés), que proporciona una potente plataforma para desarrollo de aplicaciones de ingeniería y ciencias. La página de Internet de Scilab es http://www.scilab.org

Todos los programas desarrollados para efectos del presente trabajo, se escribieron en el lenguaje de Scilab versiones 4, 4.1.1 y 4.1.2.



Consola de Scilab

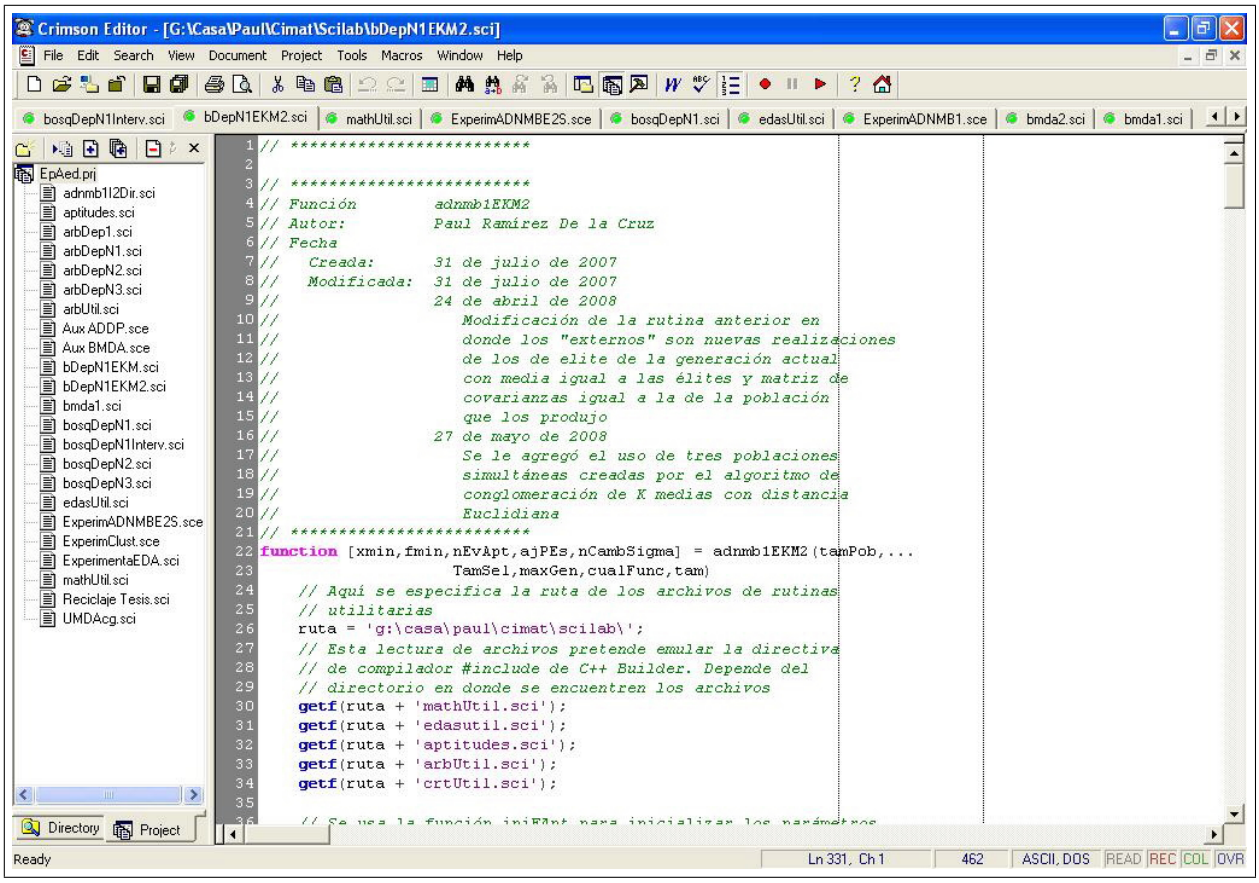


Ventana de gráficos de Scilab

A.2. Crimson Editor

Se trata de un editor de código que se puede ejecutar en el sistema operativo Windows. Contiene numerosas plantillas para resaltado de sintaxis y ofrece varias prestaciones que permiten manejar proyectos de programación de gran tamaño en muchos lenguajes de uso común, como HTML, C/C++, Perl, Java y Scilab, por mencionar algunos [22]. La página web del desarrollador es http://www.crimsoneditor.com/

Se utilizó Crimson Editor v. 3.70 para escribir todos los programas de este trabajo de Tesis.



Ventana principal de Crimson Editor