

Revista Electrónica Nova Scientia

Una extensión del algoritmo MIMIC mediante
Cópulas
A extension for MIMIC algorithm through Co-
pulas

**Rogelio Salinas Gutiérrez¹, Arturo Hernández Aguirre¹,
Enrique Raúl Villa Diharce¹**

¹Centro de Investigación en Matemáticas, Guanajuato.

Artículo por Invitación

México

Resumen

Una nueva manera de modelar dependencias probabilísticas en el algoritmo de Maximización de Información Mutua mediante Clústeres de Entrada (MIMIC) es presentada. Mediante cópulas es posible separar la estructura de dependencia de las distribuciones marginales en una distribución conjunta. El uso de cópulas como un mecanismo para modelar distribuciones y su aplicación a MIMIC es ilustrado en la función de prueba Rosenbrock.

Palabras Clave: Optimización global, Cómputo evolutivo, Algoritmos de Estimación de Distribuciones, Cópulas.

Recepción: 10-08-09

Aceptación: 03-09-09

Abstract

A new way of modeling probabilistic dependencies in Mutual Information Maximization for Input Clustering (MIMIC) algorithm is presented. By means of copulas it is possible to separate the dependence structure from marginal distributions in a joint distribution. The use of copulas as a mechanism for modeling distributions and its application to MIMIC is illustrated on Rosenbrock test function.

Keywords: Global optimization, Evolutionary computing, Estimation of Distribution Algorithms, Copulas.

Introducción

El algoritmo de Maximización de Información Mutua mediante Clústeres de Entrada (MIMIC, por sus siglas en inglés) (De Bonet 1997) pertenece a una clase más general de algoritmos evolutivos llamados Algoritmos de Estimación de Distribuciones (EDAs, por sus siglas en inglés) (Larrañaga 2002). Los EDAs son utilizados para resolver problemas de optimización mediante búsqueda estocástica. A diferencia de otros algoritmos evolutivos, los EDAs usan modelos probabilísticos en lugar de operadores genéticos tales como cruza y mutación. El uso de modelos probabilísticos permite representar explícitamente dependencias entre las variables de decisión así como su estructura. Un pseudocódigo para EDAs se muestra en el Algoritmo 1.

Algoritmo 1. Pseudocódigo para EDAs

1. asignar $t \leftarrow 0$
generar aleatoriamente la población inicial P_0 con M individuos
 2. seleccionar un conjunto de N soluciones S_t de P_t , con $N < M$
 3. estimar un modelo probabilístico M_t de S_t
 4. generar una población nueva por muestreo de la distribución de S_t
asignar $t \leftarrow t + 1$
 5. si no se ha cumplido el criterio de paro ir al paso 2
-

Como puede observarse en el paso 3, las interacciones entre las variables de decisión son tomadas en cuenta a través del modelo estimado. La posibilidad de incorporar las dependencias entre variables dentro de la nueva población modifica favorablemente el desempeño de un EDA. En la actualidad, varios EDAs han sido propuestos para resolver problemas de optimización en dominios discretos y continuos. En particular, el algoritmo MIMIC utiliza un modelo probabilístico que solo considera dependencias en una secuencia lineal de variables. En este trabajo presentamos una generalización del algoritmo MIMIC en dominios continuos mediante el uso de funciones de cópula.

Algoritmo MIMIC

Al igual que otros EDAs, el algoritmo $MIMIC_C^G$ para dominios continuos (Larrañaga, 1999) propone el uso de un modelo probabilístico $f_\pi(x)$ para aproximar la verdadera función de

densidad $f(x)$ subyacente a un determinado problema de optimización. En particular el modelo $f_\pi(x)$ propuesto por $MIMIC_C^G$ es de la forma:

$$f_\pi(x) = f(x_{i_1} | x_{i_2}) \cdot f(x_{i_2} | x_{i_3}) \cdots f(x_{i_{n-1}} | x_{i_n}) \cdot f(x_{i_n}), \quad (1)$$

donde las funciones de densidad condicional y univariadas son distribuciones Gaussianas. El subíndice $\pi = (i_1, i_2, \dots, i_n)$ representa una permutación de índices entre 1 y n .

Para el algoritmo $MIMIC_C^G$ es importante determinar la permutación π que permita una mejor aproximación de $f_\pi(x)$ a $f(x)$. Para tal fin, se busca minimizar la divergencia Kullback-Liebler:

$$D_{KL}(f(x), f_\pi(x)) = E_{f(x)} \left[\log \frac{f(x)}{f_\pi(x)} \right]. \quad (2)$$

La divergencia Kullback-Liebler siempre es mayor o igual a cero y puede interpretarse como una distancia entre dos funciones. Mediante conceptos de entropía y de información mutua es posible expresar de manera equivalente la Ec. (2) como:

$$D_{KL}(f(x), f_\pi(x)) = -H(X) + \sum_{k=1}^n H(X_{i_k}) - \sum_{k=1}^{n-1} I(X_{i_k}, X_{i_{k+1}}), \quad (3)$$

donde $H(X) = -E_{f(x)}[\log f(x)]$ denota la entropía de una variable aleatoria continua X con

densidad $f(x)$ y $I(X_{i_k}, X_{i_{k+1}}) = E_{f(x_{i_k}, x_{i_{k+1}})} \left[\log \frac{f(x_{i_k}, x_{i_{k+1}})}{f(x_{i_k}) \cdot f(x_{i_{k+1}})} \right]$ representa la información

mutua entre las variables aleatorias X_{i_k} y $X_{i_{k+1}}$. Como puede observarse, los dos primeros términos de la Ec. (3) no dependen de la permutación π , por lo que minimizar la divergencia Kullback-Liebler es equivalente a maximizar la suma de información mutua entre pares de variables. Determinar el orden en que la suma de información mutua es máxima permite encontrar la permutación óptima π . Debido a razones de eficiencia computacional se utiliza un algoritmo voraz con el cual se escoge una permutación adecuada.

Algoritmo 2. Algoritmo voraz para seleccionar una permutación π

1. escoger $(i_n, i_{n-1}) = \operatorname{argmax}_{j \neq k} \hat{I}(X_j, X_k)$, donde $\hat{I}(\cdot, \cdot)$ es una estimación de la información mutua entre dos variables
-

2. escoger $i_k = \operatorname{argmax}_j \hat{I}(X_{k+1}, X_j)$, donde $j \neq i_{k+1}, \dots, i_n$ y $k = n-1, n-2, \dots, 2, 1$

De esta manera, el Algoritmo 2 determina el ordenamiento de las variables de decisión a través de una medida de dependencia como lo es la información mutua. El algoritmo $MIMIC_C^G$ fue propuesto bajo el supuesto de que las distribuciones entre pares de variables son Normales bivariadas, lo que permite encontrar la permutación π a través de fórmulas cerradas en términos de varianzas y covarianzas. En este trabajo se propone el uso de cópulas para generalizar el tipo de dependencia entre variables y separar esas asociaciones de las distribuciones marginales. De esta manera no es necesario restringirse al uso de distribuciones Normales.

Funciones cópula

El concepto de cópula (Sklar, 1959) fue introducido para separar el efecto de dependencia del efecto de distribuciones marginales en una distribución conjunta. La separación entre distribuciones marginales y una estructura de dependencia explica la flexibilidad de modelado que tienen las cópulas y, por esta razón, han sido ampliamente usadas en muchas áreas de aplicación e investigación tales como Finanzas (Trivedi, 2007), Clima (Schölzel, 2008), Oceanografía (De-Waal, 2005), Hidrología (Genest, 2007), Geodesia (Bacigál, 2006) y Confiabilidad (Monjardin, 2007).

Definición 1. Una cópula es una función de distribución conjunta de variables aleatorias uniforme estándar. Esto es,

$$C(u_1, \dots, u_n) = P[U_1 \leq u_1, \dots, U_n \leq u_n],$$

donde $U_i \sim U(0,1)$ para $i = 1, \dots, n$.

El lector interesado puede consultar (Joe, 1997) y (Nelsen, 2006) para una descripción más formal de cópulas. El siguiente teorema establece la relación entre distribuciones marginales, cópula y función de distribución.

Teorema 1. (Teorema de Sklar) Sea F una función de distribución n-dimensional con marginales F_1, F_2, \dots, F_n , luego existe una n-Cópula C tal que para todo x en \bar{R}^n ,

$$F(x_1, x_2, \dots, x_n) = C(F_1(x_1), F_2(x_2), \dots, F_n(x_n)),$$

donde \bar{R}^n denota la recta real extendida $[-\infty, \infty]$. Si F_1, F_2, \dots, F_n son todas continuas, entonces C es única. En caso contrario, C está únicamente determinada en $Ran(F_1) \times Ran(F_2) \times \dots \times Ran(F_n)$, donde Ran indica rango de la función.

De acuerdo al Teorema de Sklar, la densidad n-dimensional f puede ser representada como

$$f(x_1, x_2, \dots, x_n) = f_1(x_1) \cdot f_2(x_2) \cdot \dots \cdot f_n(x_n) \cdot c(F_1(x_1), F_2(x_2), \dots, F_n(x_n)),$$

donde c es la densidad de la cópula C . Este resultado permite la elección de marginales diferentes y una estructura de dependencia dada por la cópula para luego ser usadas en la construcción de una distribución multivariada. Esto contrasta con la manera usual de construir distribuciones multivariadas, que tiene la restricción de que las marginales son usualmente del mismo tipo.

Hay muchas familias de cópulas y cada una de ellas está caracterizada por un parámetro o vector de parámetros. Estos parámetros miden la dependencia entre las marginales y son llamados parámetros de dependencia θ . En este artículo usamos cópulas bivariadas con un parámetro de dependencia θ . La Tabla 1 muestra información sobre las cópulas Frank y Gaussiana. Puede observarse la relación que hay entre el parámetro de dependencia y la medida de concordancia Tau de Kendall.

Cópula	Descripción
Frank	Distribución: $C(u_1, u_2; \theta) = \frac{-1}{\theta} \ln \left(1 + \frac{(e^{-\theta u_1} - 1)(e^{-\theta u_2} - 1)}{e^{-\theta} - 1} \right)$ Parámetro: $\theta \in (-\infty, \infty)$ Tau de Kendall: $\tau = 1 - \frac{4}{\theta} [1 - D_1(\theta)]$, donde $D_1(\theta) = \frac{1}{\theta} \int \frac{t}{e^t - 1} dt$
Gaussiana	Distribución: $C(u_1, u_2; \theta) = \Phi_G(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$, donde Φ_G es la distribución normal bivariada con correlación θ Parámetro: $\theta \in (-1, 1)$

<p>Tau de Kendall: $\tau = \frac{2}{\pi} \arcsin(\theta)$</p> <p>Entropía: $H(U_1, U_2) = \frac{1}{2} \log(1 - \theta^2)$</p>

Tabla 1. Cópulas bivariadas utilizadas en este artículo.

El parámetro de dependencia θ de una cópula bivariada puede estimarse mediante máxima verosimilitud. Para ello debe maximizarse la función log-verosimilitud dada en la Ec. (4),

$$l(\theta) = \sum_{t=1}^T \ln c(F(x_{1t}), F(x_{2t}); \theta). \quad (4)$$

El valor de θ que maximiza la log-verosimilitud es llamado estimador de máxima verosimilitud $\hat{\theta}_{EMV}$. Una vez que el parámetro θ es estimado, la cópula bivariada queda completamente definida. Para maximizar la función de log-verosimilitud utilizamos como aproximación inicial de $\hat{\theta}_{EMV}$ a una estimación no paramétrica de θ dada por Tau de Kendall.

Algoritmo MIMIC con cópulas

Un resultado que utilizaremos para determinar la información mutua entre dos variables y que involucra la entropía de una cópula (Davy, 2005) es el siguiente

$$H(U_1, U_2) = -I(X_1, X_2),$$

donde $U_1 = F(X_1)$ y $U_2 = F(X_2)$.

Para el algoritmo MIMIC usaremos dos funciones de dependencia diferentes: cópula Frank y cópula Gaussiana. Estas cópulas son seleccionadas debido a que su parámetro de dependencia tiene asociado todo el rango de valores de la medida Tau de Kendall. Esto significa que dependencias positivas y negativas entre marginales son consideradas en ambas cópulas. Sin embargo, las cópulas Frank y Gaussiana difieren en la manera en que ellas modelan valores extremos y centrados. Por ejemplo, una cópula Frank es más adecuada para datos con débil dependencia en los extremos y fuerte dependencia entre valores centrados.

El algoritmo MIMIC que se propone en este trabajo consiste en estimar la entropía de la cópula asociada a cada par de variables para calcular la información mutua. Las dos variables con la

información mutua más grande son seleccionadas como las primeras dos variables de la permutación π . Las siguientes variables de la permutación π son escogidas de acuerdo a la información mutua con respecto a la variable anterior (Algoritmo 2).

Para una cópula Gaussiana hay una manera directa para calcular su entropía e información mutua; para una cópula Frank estimamos su entropía mediante una aproximación numérica.

Una vez que una permutación π es encontrada, la generación de muestras sigue el orden establecido por la Ec. (1). Para realizarlo, primero se muestrea la variable U_{i_n} de una distribución uniforme estándar y posteriormente se muestrean las variables U_{i_k} de la distribución de cópula condicionada de U_{i_k} dado el valor de $U_{i_{k+1}}$ para $k = n-1, \dots, 1$. Después de lo anterior, se utilizan los valores simulados de U_i para determinar los cuantiles X_i mediante la ecuación $X_i = F_{X_i}^{-1}(U_i)$.

Es importante mencionar que mediante el uso de cópulas es posible escribir la Ec. (1) como

$$f_{\pi}(x) = \prod_{i=1}^n f(x_i) \cdot \prod_{k=1}^{n-1} c(u_{i_k}, u_{i_{k+1}}), \quad (5)$$

donde $u_{i_k} = F(x_{i_k})$ y $u_{i_{k+1}} = F(x_{i_{k+1}})$. Esto significa que $MIMIC_C^G$ puede obtenerse como un caso particular de la Ec. (5) al utilizar cópulas Gaussianas y marginales Gaussianas.

La función de prueba que utilizamos en este trabajo tiene un espacio de búsqueda acotado. Cada valor de la variable X_i del espacio de búsqueda es transformado a un valor del intervalo $(0,1)$ a través de una transformación lineal. Por lo anterior usamos distribuciones Beta como distribuciones marginales.

Para estimar los parámetros de las densidades que aparecen en la Ec. (5) usamos el método de Inferencia para Marginales (Cherubini, 2004). Este método está basado en el método de Máxima

Verosimilitud y consiste en estimar los parámetros de las distribuciones marginales y después utiliza esas estimaciones para estimar los parámetros de las cópulas.

Experimentos

Se implementan cuatro algoritmos para optimizar la función de prueba Rosenbrock de 10 dimensiones. Una descripción de la función Rosenbrock se muestra en la Tabla 2. Uno de los algoritmos es $MIMIC_C^G$ y los otros tres son extensiones del mismo:

- un algoritmo con cópulas Frank y marginales Beta $MIMIC_{Beta}^{Frank}$
- un algoritmo con cópulas Frank y marginales Gaussianas $MIMIC_{Gaussiana}^{Frank}$
- un algoritmo con cópulas Gaussianas y marginales Beta $MIMIC_{Beta}^{Gaussiana}$

De acuerdo a la expresión dada por la Ec. (5) el algoritmo $MIMIC_C^G$ puede verse como un modelo probabilístico que utiliza marginales Gaussianas y cópulas Gaussianas, es decir, puede verse como $MIMIC_{Gaussiana}^{Gaussiana}$.

Descripción: $F(x) = \sum_{i=1}^9 \left[100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right]$

Dominio: $-10 \leq x_i \leq 10$, con $i = 1, \dots, 10$.
--

Valor mínimo: $F(\mathbf{1}) = \mathbf{0}$
--

Tabla 2. Descripción de la función de prueba Rosenbrock.

Cada algoritmo es ejecutado 30 veces utilizando un tamaño de población de 200 individuos. El número máximo de evaluaciones es de 300,000. Sin embargo, cuando es detectada convergencia a un mínimo local la ejecución es detenida. Una mejoría menor a 1×10^{-6} en 25 iteraciones es considerada como convergencia.

Resultados

En la Tabla 3 se muestra un resumen descriptivo de los valores de aptitud logrados por los cuatro algoritmos para la función de prueba Rosenbrock. La información acerca del número de evaluaciones requeridas por cada algoritmo se reporta en la Tabla 4.

Algoritmo	Mejor valor	Mediana	Media	Peor valor	Desviación estándar
$MIMIC_{Beta}^{Frank}$	6.57E+00	7.78E+00	7.72E+00	8.54E+00	4.16E-01
$MIMIC_{Gaussiana}^{Frank}$	6.31E+00	7.74E+00	7.70E+00	8.28E+00	4.51E-01
$MIMIC_{Beta}^{Guassiana}$	7.25E+00	7.68E+00	7.80E+00	8.59E+00	3.32E-01
$MIMIC_C^G$	7.43E+00	8.03E+00	7.98E+00	8.27E+00	2.42E-01

Tabla 3. Resultados descriptivos de aptitud para la función de prueba Rosenbrock.

Como puede observarse en la Tabla 3, el algoritmo $MIMIC_{Gaussiana}^{Frank}$ obtiene en promedio el mejor valor de aptitud aunque tiene asociada una mayor dispersión respecto a los demás algoritmos. En cuanto al número de evaluaciones de función puede verse en la Tabla 4 que $MIMIC_C^G$ es el algoritmo que en promedio requiere menos evaluaciones de la función Rosenbrock, además de ser el más consistente, pues obtiene la desviación estándar más pequeña.

Algoritmo	Media	Desviación estándar
$MIMIC_{Beta}^{Frank}$	172627.70	75183.61
$MIMIC_{Gaussiana}^{Frank}$	102333.40	30150.18
$MIMIC_{Beta}^{Guassiana}$	59661.20	18532.65
$MIMIC_C^G$	22182.87	2591.34

Tabla 4. Resultados descriptivos de evaluaciones para la función de prueba Rosenbrock.

Para determinar si existen diferencias significativas en los resultados se realiza una comparación estadística entre el algoritmo $MIMIC_C^G$ y cada una de las tres extensiones utilizadas en este trabajo. Para ello se utiliza una prueba de hipótesis basada en el método Bootstrap para diferencias de medias. De igual manera se determina mediante Bootstrap los intervalos de confianza del 95% para la diferencia de medias entre algoritmos comparados.

Algoritmos comparados	Intervalo 95%		valor-p
$MIMIC_{Beta}^{Frank}$ vs. $MIMIC_C^G$	-3.98E-01	-1.13E-01	6.75E-03
$MIMIC_{Gaussiana}^{Frank}$ vs. $MIMIC_C^G$	-4.34E-01	-1.34E-01	4.95E-03
$MIMIC_{Beta}^{Guassiana}$ vs. $MIMIC_C^G$	-2.98E-01	-5.71E-02	2.18E-02

Tabla 5. Resultados para la diferencia de aptitud entre medias para la función de prueba Rosenbrock. Un intervalo de confianza del 95% y un valor-p son obtenidos mediante bootstrap.

El valor-p reportado en la Tabla 5 para cada comparación de algoritmos indica que existe diferencia significativa entre los promedios de aptitud. Esto también puede apreciarse a través de los intervalos de confianza construidos para la diferencia de promedios, pues en todos los casos se excluye al cero. Dado que en la comparación de algoritmos los intervalos de confianza excluyen valores positivos, puede afirmarse que el algoritmo $MIMIC_C^G$ no obtiene una aptitud similar a la de los otros algoritmos. Esto quiere decir que en los algoritmos $MIMIC_{Beta}^{Frank}$, $MIMIC_{Gaussiana}^{Frank}$ y $MIMIC_{Beta}^{Guassiana}$ los promedios de aptitud son mejores que los de $MIMIC_C^G$. Lo anterior significa que modificar la estructura de dependencia o modificar la distribución de las marginales en el algoritmo MIMIC puede ayudar a obtener mejores resultados en un problema de optimización.

Conclusiones

Se ha presentado una modificación del algoritmo MIMIC mediante el uso de cópulas. Las funciones cópulas permiten separar la estructura de dependencia de las distribuciones marginales, lo que permite una mayor flexibilidad de modelado en distribuciones conjuntas. Aunque la estructura de dependencia usada en los experimentos fue fijada al igual que las distribuciones marginales, esto no necesariamente debe ocurrir para todas las variables. Combinaciones de cópulas y de distribuciones marginales pueden ser utilizadas en un mismo modelo de probabilidad.

Los tres algoritmos que modifican su estructura de dependencia o la distribución de sus marginales obtienen mejores desempeños respecto al algoritmo $MIMIC_C^G$. Esto sugiere que el uso de cópulas puede mejorar el desempeño de los EDAs, sin embargo, debe tenerse cuidado de

escoger las cópulas y las distribuciones marginales más adecuadas para cada problema de optimización.

Referencias

Bacigál, T. y Komorníková, M. (2006). Fitting archimedean copulas to bivariate geodetic data. *Compstat 2006 Proceedings in Computational Statistics*. Physica-Verlag HD.

Cherubini, U., Luciano, E. y Vecchiato, W. (2004). *Copula Methods in Finance*. Wiley.

Davy, M. y Doucet, A. (2005). Copulas: a new insight into positive time-frequency distributions. *Signal Processing Letters, IEEE*, 10(7): 215-218.

De Bonet, J.S., Isbell, C.L. y Viola, P. (1997). MIMIC: Finding optima by estimating probability densities. *Advances in Neural Information Processing Systems* (9): 424-430. The MIT Press.

De-Waal, D.J. y Van-Gelder, P.H.A.J.M. (2005). Modelling of extreme wave heights and periods through copulas. *Extremes* 8(4): 345-356.

Genest, C. y Favre, A.C. (2007). Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering* 12(4): 347-368.

Joe, H. (1997). *Multivariate models and dependence concepts*. Chapman and Hall.

Larrañaga, P., Etxeberria, R., Lozano, J.A. y Peña, J.M. (1999). Optimization by learning and simulation of bayesian and gaussian networks. Technical Report KZZA-IK-4-99, University of the Basque Country.

Larrañaga, P. y Lozano, J.A. (2002). Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers.

Monjardin, P.E. (2007). Análisis de dependencia en tiempo de falla. Tesis de maestría. Centro de Investigación en Matemáticas. Guanajuato, México.

Nelsen, R.B. (2006). An introduction to Copulas. Springer Series in Statistics.

Schölzel, C. y Friederichs, P. (2008). Multivariate non-normally distributed random variables in climate research – introduction to the copula approach. *Nonlinear Processes in Geophysics* 15(5): 761-772.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris* (8): 229-231.

Trivedi, P.K. y Zimmer, D.M. (2007). Copula Modeling: An Introduction for Practitioners. *Foundations and Trends in Econometrics* (1). Now Publishers.

