Algoritmos Basados en Poblaciones

Jose Antonio Lozano

Intelligent Systems Group
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad del País Vasco–Euskal Herriko Unibertsitatea

UPV - EHU

Organización del tema

Organización del tema

UPV - EHU

Algoritmos poblacionales

Características

- A cada paso del algoritmo mantienen un conjunto de soluciones (*población*)
- Algoritmos inspirados en la naturaleza
- Pueden utilizar métodos de búsqueda local
- Cuestión clave: ¿cómo sacar provecho de las poblaciones?

UPV - EHU

Organización del tema

UPV - EHU

Computación evolutiva

Ideas básicas

- Un conjunto de técnicas computacionales basadas en *imitar* el proceso de evolución de las especies en el mundo natural siguiendo los postulados de Darwin
- Engloba diferentes paradigmas estando la mayoría de ellos dedicados a la optimización
- Sus comienzos datan de los 60 y los 70 siendo las referencias más relevantes: Holland (1975), Goldberg (1989), Schwefel (1981), Fogel (1962,1964)

UPV - EHU

Computación evolutiva

Ideas básicas

- Un conjunto de técnicas computacionales basadas en *imitar* el proceso de evolución de las especies en el mundo natural siguiendo los postulados de Darwin
- Engloba diferentes paradigmas estando la mayoría de ellos dedicados a la optimización
- Sus comienzos datan de los 60 y los 70 siendo las referencias más relevantes: Holland (1975), Goldberg (1989), Schwefel (1981), Fogel (1962,1964)

Computación evolutiva

Ideas básicas

- Un conjunto de técnicas computacionales basadas en *imitar* el proceso de evolución de las especies en el mundo natural siguiendo los postulados de Darwin
- Engloba diferentes paradigmas estando la mayoría de ellos dedicados a la optimización
- Sus comienzos datan de los 60 y los 70 siendo las referencias más relevantes: Holland (1975), Goldberg (1989), Schwefel (1981), Fogel (1962,1964)

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

UPV - EHU

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

UPV - EHU

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

UPV - EHU

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

Características y algoritmos

Características principales

- Uso de un conjunto de soluciones a cada paso: población
- Se generan nuevas soluciones combinando y modificando las actuales mediante el uso de operadores:
  - Selección de los mejores
  - Operadores de reproducción: cruce y mutación

Algoritmos principales

- Algoritmos genéticos
- Estrategias evolutivas-Programación evolutiva
- Programación genética

Algoritmos genéticos

Esquema básico

- Cada solución al problema se denomina individuo o cromosoma y se representa mediante una cadena binaria
- A cada paso se mantiene un conjunto de soluciones denominado población
- Se aplican tres operadores evolutivos: selección, cruce y mutación
- La población en el tiempo t se sustituye por la población en el tiempo $t + 1$

UPV - EHU

Algoritmos genéticos

Esquema básico

- Cada solución al problema se denomina individuo o cromosoma y se representa mediante una cadena binaria
- A cada paso se mantiene un conjunto de soluciones denominado población
- Se aplican tres operadores evolutivos: selección, cruce y mutación
- La población en el tiempo t se sustituye por la población en el tiempo $t+1$

Algoritmos genéticos

Esquema básico

- Cada solución al problema se denomina individuo o cromosoma y se representa mediante una cadena binaria
- A cada paso se mantiene un conjunto de soluciones denominado población
- Se aplican tres operadores evolutivos: selección, cruce y mutación
- La población en el tiempo t se sustituye por la población en el tiempo $t + 1$

UPV - EHU

Algoritmos genéticos

Esquema básico

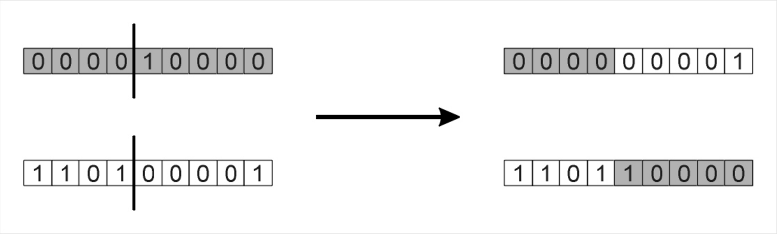
- Cada solución al problema se denomina individuo o cromosoma y se representa mediante una cadena binaria
- A cada paso se mantiene un conjunto de soluciones denominado población
- Se aplican tres operadores evolutivos: selección, cruce y mutación
- La población en el tiempo t se sustituye por la población en el tiempo $t+1$

UPV - EHU

Pseudocódigo

Hallar la población inicial P_0
hasta condición_parada = TRUE **hacer**
  repetir $\frac{n}{2}$ **veces**
    Elegir aleatoriamente dos individuos de la población P_k
    Cruzar los dos individuos con probablidad p_c
    Mutar los dos individuos resultantes con probabilidad p_m
    Introducir los dos nuevos individuos en la población P'_k
  Seleccionar n individuos de P'_k para obtener P_{k+1}
Devolver la mejor solución

Cruce basado en un punto



Operador de mutación

- Cada bit modifica su valor con una probabilidad p_m

1	0	1	0	0	0	0	1	0

$\longrightarrow$

1	0	0	1	0	0	0	0	0

Operador de selección

Selección proporcional al valor de función objetivo

- Los individuos se seleccionan de forma aleatoria
- La probabilidad de seleccionar un individuo x es:

$$p_{sel}(x) = \frac{f(x)}{\sum_{y \in P_t} f(y)}$$

Operador de selección

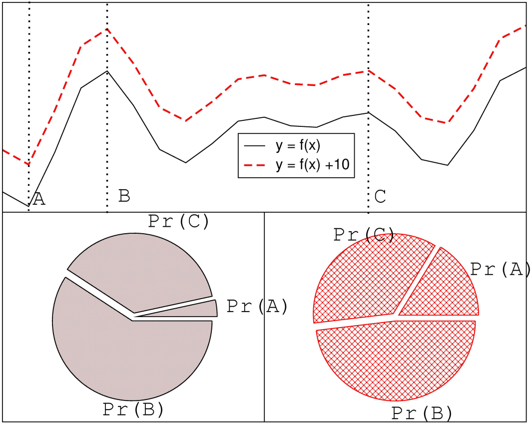
Selección proporcional al valor de función objetivo

- Los individuos se seleccionan de forma aleatoria
- La probabilidad de seleccionar un individuo x es:

$$p_{sel}(x) = \frac{f(x)}{\sum_{y \in P_t} f(y)}$$

UPV - EHU

Operador de selección



UPV - EHU

Ejemplo de aplicación

Ejemplo: Máximo de $f(x) = x^2$ sobre los enteros $\{1, 2, \ldots, 32\}$

	Población inicial (fenotipos)	x valor genotipo	$f(x)$ valor (función adaptación)	$f(x)/\sum f(x)$ (probabilidad selección)	Probabilidad de selección acumulada
1	01101	13	169	0.14	0.14
2	11000	24	576	0.49	0.63
3	01000	8	64	0.06	0.69
4	10011	19	361	0.31	1.00
Suma			1170		
Media			293		
Mejor			576		

Ejemplo de aplicación

Emparejamiento de los individuos seleccionados	Punto de cruce	Descen-dientes	Nueva población descendientes mutados	x valor genotipo	$f(x)$ función adaptación
11000	2	11011	11011	27	729
10011	2	10000	10000	16	256
01101	3	01100	11100	28	784
11000	3	11001	11101	29	841
Suma					2610
Media					652.5
Mejor					841

Extensiones y modificaciones del algoritmo genético simple

Extensiones

- Codificación: numéricas, reales, permutaciones
- Operadores de selección: torneo, truncación, Boltzmann, elitismo
- Operadores de cruce: basado en *n* puntos, uniforme, específicos para el problema y la codificación

Extensiones y modificaciones del algoritmo genético simple

Extensiones

- Codificación: numéricas, reales, permutaciones
- Operadores de selección: torneo, truncación, Boltzmann, elitismo
- Operadores de cruce: basado en *n* puntos, uniforme, específicos para el problema y la codificación

Extensiones y modificaciones del algoritmo genético simple

Extensiones

- Codificación: numéricas, reales, permutaciones
- Operadores de selección: torneo, truncación, Boltzmann, elitismo
- Operadores de cruce: basado en *n* puntos, uniforme, específicos para el problema y la codificación

Aplicación de los algoritmos genéticos

Aspectos a determinar para resolver un problema mediante algoritmos genéticos

- Codificación más adecuada
- Generación de la población inicial
- Operadores a utilizar
- Valores para los parámetros: tamaño de la población, probabilidad de cruce, probabilidad de mutación, criterio de parada

Aplicación de los algoritmos genéticos

Aspectos a determinar para resolver un problema mediante algoritmos genéticos

- Codificación más adecuada
- Generación de la población inicial
- Operadores a utilizar
- Valores para los parámetros: tamaño de la población, probabilidad de cruce, probabilidad de mutación, criterio de parada

Aplicación de los algoritmos genéticos

Aspectos a determinar para resolver un problema mediante algoritmos genéticos

- Codificación más adecuada
- Generación de la población inicial
- Operadores a utilizar
- Valores para los parámetros: tamaño de la población, probabilidad de cruce, probabilidad de mutación, criterio de parada

Aplicación de los algoritmos genéticos

Aspectos a determinar para resolver un problema mediante algoritmos genéticos

- Codificación más adecuada
- Generación de la población inicial
- Operadores a utilizar
- Valores para los parámetros: tamaño de la población, probabilidad de cruce, probabilidad de mutación, criterio de parada

Organización del tema

Introducción a los EDAs

Características de los EDAs

- Algoritmos de optimización basados en poblaciones
- Se eliminan los operadores de reproducción de un algoritmo genético:
  - Se aprende una distribución de probabilidad a partir de los individuos seleccionados
  - Se muestrea la distribución aprendida para obtener la nueva población

Introducción a los EDAs

Características de los EDAs

- Algoritmos de optimización basados en poblaciones
- Se eliminan los operadores de reproducción de un algoritmo genético:
  - Se aprende una distribución de probabilidad a partir de los individuos seleccionados
  - Se muestrea la distribución aprendida para obtener la nueva población

Introducción a los EDAs

Características de los EDAs

- Algoritmos de optimización basados en poblaciones
- Se eliminan los operadores de reproducción de un algoritmo genético:
  - Se aprende una distribución de probabilidad a partir de los individuos seleccionados
  - Se muestrea la distribución aprendida para obtener la nueva población

Introducción a los EDAs

Características de los EDAs

- Algoritmos de optimización basados en poblaciones
- Se eliminan los operadores de reproducción de un algoritmo genético:
    - Se aprende una distribución de probabilidad a partir de los individuos seleccionados
    - Se muestrea la distribución aprendida para obtener la nueva población

UPV - EHU

Un ejemplo sencillo

Optimización de OneMax con EDAs

$$max\ f(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

con $x_i = 0, 1$

Un ejemplo sencillo

Optimización de OneMax con EDAs

$$max\ f(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

$$\text{con } x_i = 0, 1$$

	X_1	X_2	X_3	X_4	X_5	X_6	$f(\boldsymbol{x})$
1	1	0	1	0	1	0	3
2	0	1	0	0	1	0	2
3	0	0	0	1	0	0	1
4	1	1	1	0	0	1	4
5	0	0	0	0	0	1	1
6	1	1	0	0	1	1	4
7	0	1	1	1	1	1	5
8	0	0	0	1	0	0	1
9	1	1	0	1	0	0	3
10	1	0	1	0	0	0	2
11	1	0	0	1	1	1	4
12	1	1	0	0	0	1	3
13	1	0	1	0	0	0	2
14	0	0	0	0	1	1	2
15	0	1	1	1	1	1	5
16	0	0	0	1	0	0	1
17	1	1	1	1	1	0	5
18	0	1	0	1	1	0	3
19	1	0	1	1	1	1	5
20	1	0	1	1	0	0	3

Un ejemplo sencillo

Optimización de OneMax con EDAs

$$max\ f(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

$$\text{con } x_i = 0, 1$$

	X_1	X_2	X_3	X_4	X_5	X_6	$f(\boldsymbol{x})$
1	1	0	1	0	1	0	3
2	0	1	0	0	1	0	2
3	0	0	0	1	0	0	1
4	1	1	1	0	0	1	4
5	0	0	0	0	0	1	1
6	1	1	0	0	1	1	4
7	0	1	1	1	1	1	5
8	0	0	0	1	0	0	1
9	1	1	0	1	0	0	3
10	1	0	1	0	0	0	2
11	1	0	0	1	1	1	4
12	1	1	0	0	0	1	3
13	1	0	1	0	0	0	2
14	0	0	0	0	1	1	2
15	0	1	1	1	1	1	5
16	0	0	0	1	0	0	1
17	1	1	1	1	1	0	5
18	0	1	0	1	1	0	3
19	1	0	1	1	1	1	5
20	1	0	1	1	0	0	3

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10}$$

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \tfrac{7}{10} \qquad p(X_2 = 1) = \tfrac{7}{10} \qquad p(X_3 = 1) = \tfrac{6}{10}$$

$$p(X_4 = 1) = \tfrac{6}{10} \qquad p(X_5 = 1) = \tfrac{8}{10} \qquad p(X_6 = 1) = \tfrac{7}{10}$$

UPV - EHU

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \qquad p(X_2 = 1) = \frac{7}{10} \qquad p(X_3 = 1) = \frac{6}{10}$$

$$p(X_4 = 1) = \frac{6}{10} \qquad p(X_5 = 1) = \frac{8}{10} \qquad p(X_6 = 1) = \frac{7}{10}$$

UPV - EHU

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1)$$

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

$$p(\mathbf{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$p(X_1 = 1) \cdot p(X_2 = 0) \cdot p(X_3 = 1) \cdot p(X_4 = 0) \cdot p(X_5 = 1) \cdot p(X_6 = 1) =$$

UPV - EHU

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

$$p(\mathbf{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$p(X_1 = 1) \cdot p(X_2 = 0) \cdot p(X_3 = 1) \cdot p(X_4 = 0) \cdot p(X_5 = 1) \cdot p(X_6 = 1) =$$

$$\downarrow$$

$$\frac{7}{10} \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

Un ejemplo sencillo

Aprendiendo la distribución de probabilidad

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$\begin{array}{ccccccccccccc}
p(X_1 = 1) & \cdot & p(X_2 = 0) & \cdot & p(X_3 = 1) & \cdot & p(X_4 = 0) & \cdot & p(X_5 = 1) & \cdot & p(X_6 = 1) & = \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
\frac{7}{10} & \cdot & \frac{3}{10} & \cdot & \frac{6}{10} & \cdot & \frac{4}{10} & \cdot & \frac{8}{10} & \cdot & \frac{7}{10} &
\end{array}$$

Un ejemplo sencillo

Obteniendo la nueva población

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$0.23 \qquad p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$$

$$0.85 \qquad p(X_2 = 1) = \frac{7}{10} \longrightarrow 0$$

$$0.89 \qquad p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$$

$$0.12 \qquad p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$$

$$0.98 \qquad p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$$

$$0.54 \qquad p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$$

UPV - EHU

Un ejemplo sencillo

Obteniendo la nueva población

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23	$p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$
0.85	$p(X_2 = 1) = \frac{7}{10} \longrightarrow 0$
0.89	$p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$
0.12	$p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$
0.98	$p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$
0.54	$p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$

Un ejemplo sencillo

Obteniendo la nueva población

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$
\begin{array}{ll}
0.23 & p(X_1 = 1) = \frac{7}{10} \longrightarrow 1 \\
0.85 & p(X_2 = 1) = \frac{7}{10} \longrightarrow 0 \\
0.89 & p(X_3 = 1) = \frac{6}{10} \longrightarrow 0 \\
0.12 & p(X_4 = 1) = \frac{6}{10} \longrightarrow 1 \\
0.98 & p(X_5 = 1) = \frac{8}{10} \longrightarrow 0 \\
0.54 & p(X_6 = 1) = \frac{7}{10} \longrightarrow 1
\end{array}
$$

Un ejemplo sencillo

Obteniendo la nueva población

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23	$p(X_1 = 1) = \frac{7}{10}$	$\longrightarrow 1$
0.85	$p(X_2 = 1) = \frac{7}{10}$	$\longrightarrow 0$
0.89	$p(X_3 = 1) = \frac{6}{10}$	$\longrightarrow 0$
0.12	$p(X_4 = 1) = \frac{6}{10}$	$\longrightarrow 1$
0.98	$p(X_5 = 1) = \frac{8}{10}$	$\longrightarrow 0$
0.54	$p(X_6 = 1) = \frac{7}{10}$	$\longrightarrow 1$

UPV - EHU

Un ejemplo sencillo

Obteniendo la nueva población

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$\begin{array}{ll}
0.23 & p(X_1 = 1) = \frac{7}{10} \longrightarrow 1 \\
0.85 & p(X_2 = 1) = \frac{7}{10} \longrightarrow 0 \\
0.89 & p(X_3 = 1) = \frac{6}{10} \longrightarrow 0 \\
0.12 & p(X_4 = 1) = \frac{6}{10} \longrightarrow 1 \\
0.98 & p(X_5 = 1) = \frac{8}{10} \longrightarrow 0 \\
0.54 & p(X_6 = 1) = \frac{7}{10} \longrightarrow 1
\end{array}$$

Un ejemplo sencillo

Obteniendo la nueva población

	X_1	X_2	X_3	X_4	X_5	X_6	$f(\boldsymbol{x})$
1	1	1	0	1	0	1	5
2	1	0	1	0	1	1	4
3	1	1	1	1	1	0	5
4	0	1	0	1	1	1	4
5	1	1	1	1	0	1	5
6	1	0	0	1	1	1	4
7	0	1	0	1	1	0	3
8	1	1	1	0	1	0	4
9	1	1	1	0	0	1	4
10	1	0	0	1	1	1	4
11	1	1	0	0	1	1	4
12	1	0	1	1	1	0	4
13	0	1	1	0	1	1	4
14	0	1	1	1	1	0	4
15	1	1	1	1	1	1	6
16	0	1	1	0	1	1	4
17	1	1	1	1	1	0	5
18	0	1	0	0	1	0	2
19	0	0	1	1	0	1	3
20	1	1	0	1	1	1	5

Pseudocódigo para los EDAs

Obtener una población inicial de individuos D_0
Repetir hasta que se cumpla un criterio de parada

Seleccionar de D_i un conjunto de individuos D_i^S
Aprender una distribución de prob. $p_i(\boldsymbol{x})$ a partir de D_i^S
Muestrear $p_i(\boldsymbol{x})$ para obtener $D_{i+1/2}$
Crear la nueva población D_{i+1} a partir de D_i y $D_{i+1/2}$

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

UPV - EHU

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

UPV - EHU

Diseño de un EDA

¿Qué componentes hay que establecer en los EDAs?

- Tamaño de población
- Criterio de selección
- Determinación del modelo probabilístico a utilizar
- Algoritmo de aprendizaje de la distribución de probabilidad
- Algoritmo de muestreo de la distribución de probabilidad

UPV - EHU

Algunas ideas generales

¿Por qué surgen los EDAs?

- Pésimo comportamiento de los AGs en algunos problemas sencillos (*deceptive*, separables)
- Necesidad de diseñar muchos componentes y establecer el valor de muchos parámetros
- Extender los AGs de cara a considerar interacciones entre las variables (generalización de los AGs)

Algunas ideas generales

¿Por qué surgen los EDAs?

- Pésimo comportamiento de los AGs en algunos problemas sencillos (*deceptive*, separables)
- Necesidad de diseñar muchos componentes y establecer el valor de muchos parámetros
- Extender los AGs de cara a considerar interacciones entre las variables (generalización de los AGs)

Algunas ideas generales

¿Por qué surgen los EDAs?

- Pésimo comportamiento de los AGs en algunos problemas sencillos (*deceptive*, separables)
- Necesidad de diseñar muchos componentes y establecer el valor de muchos parámetros
- Extender los AGs de cara a considerar interacciones entre las variables (generalización de los AGs)

Clasificación de los EDAs

Varias clasificaciones

- Basada en si la estructura del modelo se mantiene fija a lo largo de todo el algoritmo o no
- Basada en el tipo de dependencias que consideran los modelos probabilísticos utilizados

Clasificación de los EDAs

Varias clasificaciones

- Basada en si la estructura del modelo se mantiene fija a lo largo de todo el algoritmo o no
- Basada en el tipo de dependencias que consideran los modelos probabilísticos utilizados

Clasificación en función de las dependencias

- Algoritmos que suponen que las variables son independientes
- Algoritmos que consideran relaciones de orden dos entre las variables
- Algortimos que no restrigen las relaciones entre las variables

UPV - EHU

Clasificación en función de las dependencias

- Algoritmos que suponen que las variables son independientes
- Algoritmos que consideran relaciones de orden dos entre las variables
- Algortimos que no restrigen las relaciones entre las variables

Clasificación en función de las dependencias

- Algoritmos que suponen que las variables son independientes
- Algoritmos que consideran relaciones de orden dos entre las variables
- Algortimos que no restrigen las relaciones entre las variables

Univariate Marginal Distribution Algorithm (UMDA)

Obtener una población inicial de individuos D_0

Repetir hasta que se cumpla un criterio de parada

Seleccionar de D_l un conjunto de individuos D_l^S

Aprender una distribución de prob.:
$p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D_l^S) = \prod_{i=1}^n p_l(x_i) = \prod_{i=1}^n \frac{\sum_{j=1}^N \delta_j(X_i=x_i|D_l^S)}{N}$ *a partir de D_l^S*

Muestrear $p_l(\boldsymbol{x})$ para obtener $D_{l+1/2}$

Crear la nueva población D_{l+1} a partir de D_l y $D_{l+1/2}$

UPV - EHU

Univariate Marginal Distribution Algorithm (UMDA)

Obtener una población inicial de individuos D_0

Repetir hasta que se cumpla un criterio de parada

Seleccionar de D_l un conjunto de individuos D_l^S

Aprender una distribución de prob.: $p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D_l^S) = \prod_{i=1}^{n} p_l(x_i) = \prod_{i=1}^{n} \frac{\sum_{j=1}^{N} \delta_j(X_i = x_i | D_l^S)}{N}$ a partir de D_l^S

Muestrear $p_l(\boldsymbol{x})$ para obtener $D_{l+1/2}$

Crear la nueva población D_{l+1} a partir de D_l y $D_{l+1/2}$

UPV - EHU

Population Based Incremental Learning (PBIL)

Obtener un vector de probabilidad inicial $p_0(\boldsymbol{x})$

while no convergencia **do**
begin

Usando $p_l(\boldsymbol{x})$ obtener M individuos: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Evaluar y ordenar $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Seleccionar los N ($N \leq M$) mejores individuos: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$

Actualizar el vector de probabilidades $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$

for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1-\alpha)p_l(x_i) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{i,k:M}^l$

end

Population Based Incremental Learning (PBIL)

Obtener un vector de probabilidad inicial $p_0(\boldsymbol{x})$

while no convergencia **do**
begin

Usando $p_l(\boldsymbol{x})$ obtener M individuos: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Evaluar y ordenar $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Seleccionar los N ($N \leq M$) mejores individuos: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$

Actualizar el vector de probabilidades $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$

for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1-\alpha)p_l(x_i) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{i,k:M}^l$

end

Population Based Incremental Learning (PBIL)

Obtener un vector de probabilidad inicial $p_0(\boldsymbol{x})$

while no convergencia **do**
begin

Usando $p_l(\boldsymbol{x})$ obtener M individuos: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Evaluar y ordenar $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Seleccionar los N ($N \leq M$) mejores individuos: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$

Actualizar el vector de probabilidades $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$

for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1-\alpha)p_l(x_i) + \alpha\frac{1}{N}\sum_{k=1}^{N} x_{i,k:M}^l$

end

Population Based Incremental Learning (PBIL)

Obtener un vector de probabilidad inicial $p_0(\boldsymbol{x})$

while no convergencia **do**
begin

Usando $p_l(\boldsymbol{x})$ obtener M individuos: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Evaluar y ordenar $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Seleccionar los N ($N \leq M$) mejores individuos: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$

Actualizar el vector de probabilidades $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$

for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1-\alpha)p_l(x_i) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{i,k:M}^l$

end

UPV - EHU

Population Based Incremental Learning (PBIL)

Obtener un vector de probabilidad inicial $p_0(\boldsymbol{x})$

while no convergencia **do**
begin

Usando $p_l(\boldsymbol{x})$ obtener M individuos: $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Evaluar y ordenar $\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_k^l, \ldots, \boldsymbol{x}_M^l$

Seleccionar los N ($N \leq M$) mejores individuos: $\boldsymbol{x}_{1:M}^l, \ldots, \boldsymbol{x}_{k:M}^l, \ldots, \boldsymbol{x}_{N:M}^l$

Actualizar el vector de probabilidades $p_{l+1}(\boldsymbol{x}) = (p_{l+1}(x_1), \ldots, p_{l+1}(x_n))$

for $i = 1, \ldots n$ **do**
$p_{l+1}(x_i) = (1-\alpha)p_l(x_i) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{i,k:M}^l$

end

UPV - EHU

Algoritmo TREE

Características generales

- Modelo probabilístico: $p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | x_{j(i)})$
- Aprendizaje estructural: algoritmo de Chow y Liu
- Aprendizaje paramétrico: máxima verosimilitud
- Muestreo: muestreo lógico probabilístico

Algoritmo TREE

Características generales

- Modelo probabilístico: $p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | x_{j(i)})$
- Aprendizaje estructural: algoritmo de Chow y Liu
- Aprendizaje paramétrico: máxima verosimilitud
- Muestreo: muestreo lógico probabilístico

UPV - EHU

Algoritmo TREE

Características generales

- Modelo probabilístico: $p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_{j(i)})$
- Aprendizaje estructural: algoritmo de Chow y Liu
- Aprendizaje paramétrico: máxima verosimilitud
- Muestreo: muestreo lógico probabilístico

Algoritmo TREE

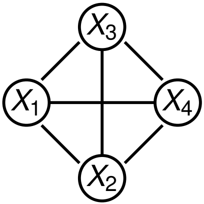
Características generales

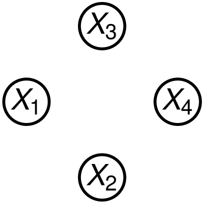
- Modelo probabilístico: $p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | x_{j(i)})$
- Aprendizaje estructural: algoritmo de Chow y Liu
- Aprendizaje paramétrico: máxima verosimilitud
- Muestreo: muestreo lógico probabilístico

Algoritmo de Chow y Liu

i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

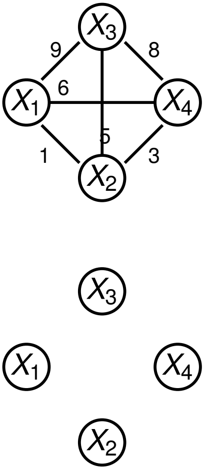




Algoritmo de Chow y Liu

i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

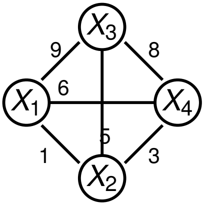


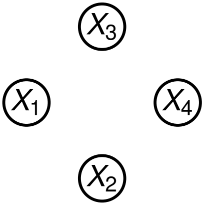
Algoritmo de Chow y Liu

i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir



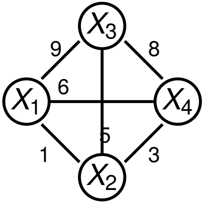


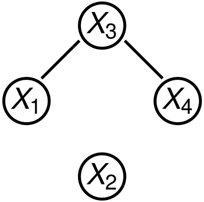
Algoritmo de Chow y Liu

i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir





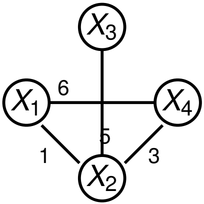
UPV - EHU

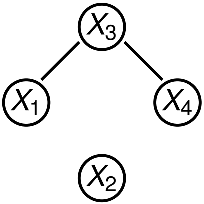
Algoritmo de Chow y Liu

i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir





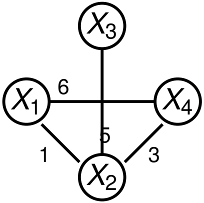
Algoritmo de Chow y Liu

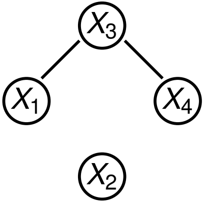
i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

iii) Repetir hasta tener $n - 1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla





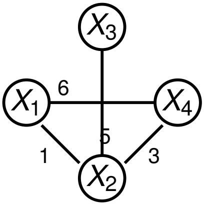
Algoritmo de Chow y Liu

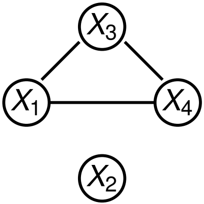
i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

iii) Repetir hasta tener $n - 1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla





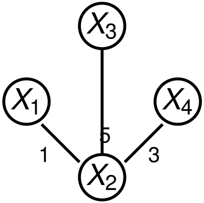
Algoritmo de Chow y Liu

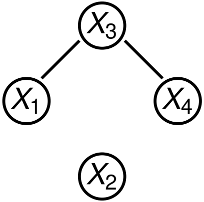
i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

iii) Repetir hasta tener $n-1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla





UPV - EHU

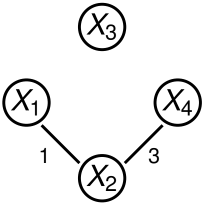
Algoritmo de Chow y Liu

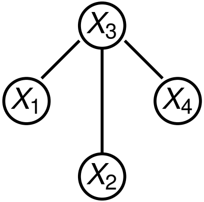
i) Para cada X_i, X_j calcular:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

iii) Repetir hasta tener $n - 1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla





UPV - EHU

Algoritmo de Chow y Liu

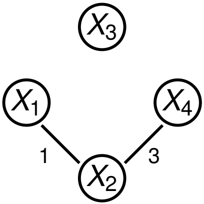
i) Para cada X_i, X_j calcular:

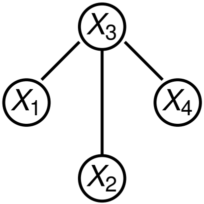
$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

iii) Repetir hasta tener $n - 1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla

iv) Orientar los arcos eligiendo cualquier nodo como raíz





UPV - EHU

Algoritmo de Chow y Liu

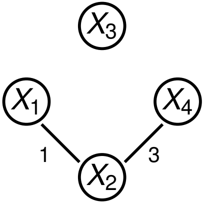
i) Para cada X_i, X_j calcular:

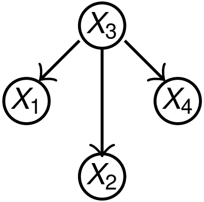
$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Asignar las dos aristas con mayor valor de MI al árbol a construir

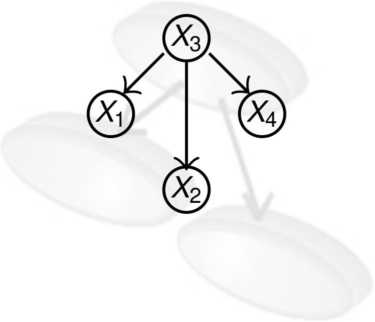
iii) Repetir hasta tener $n - 1$ aristas en el árbol:
Asignar la arista con mayor valor de MI a menos que forme un ciclo, en dicho caso eliminarla

iv) Orientar los arcos eligiendo cualquier nodo como raíz

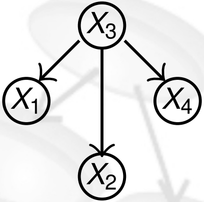




Algoritmo de Chow y Liu



Algoritmo de Chow y Liu



$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0)$$

$$p(X_1 = 1|X_3 = 1)$$

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0)$$

$$p(X_1 = 1|X_3 = 1)$$

UPV - EHU

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1|X_3 = 1)$$

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1|X_3 = 1)$$

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1|X_3 = 1) = \frac{0}{2}$$

Aprendizaje parámetrico en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2+\epsilon}{3+2\epsilon}$$

$$p(X_1 = 1|X_3 = 1) = \frac{0+\epsilon}{2+2\epsilon}$$

Muestreo en TREE

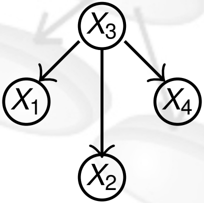
- Se ordenan las variables de forma que las variables padres esten antes que las variables hijo

- En nuestro caso: X_3, X_1, X_2, X_4
- Se muestrean las variables en dicho orden

UPV - EHU

Muestreo en TREE

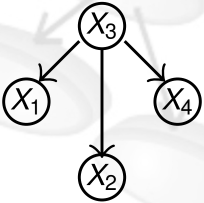
- Se ordenan las variables de forma que las variables padres esten antes que las variables hijo



- En nuestro caso: X_3, X_1, X_2, X_4
- Se muestrean las variables en dicho orden

Muestreo en TREE

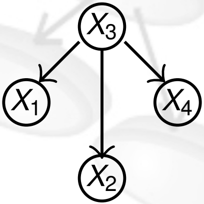
- Se ordenan las variables de forma que las variables padres esten antes que las variables hijo



- En nuestro caso: X_3, X_1, X_2, X_4
- Se muestrean las variables en dicho orden

Muestreo en TREE

- Se ordenan las variables de forma que las variables padres esten antes que las variables hijo



- En nuestro caso: X_3, X_1, X_2, X_4
- Se muestrean las variables en dicho orden

Muestreo en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Orden de las variables: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0.85	0.55	0.23	0.91
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$

Muestreo en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Orden de las variables: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0.85	0.55	0.23	0.91
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
		1	

UPV - EHU

Muestreo en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Orden de las variables: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0.85	0.55	0.23	0.91
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0		1	

UPV - EHU

Muestreo en TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

$p(X_1 = 1|X_3 = 0) = 0{,}3$ $\quad p(X_2 = 1|X_3 = 0) = 0{,}1$ $\quad p(X_3 = 1) = 0{,}7$ $\quad p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $\quad p(X_2 = 1|X_3 = 1) = 0{,}8$ $\quad p(X_4 = 1|X_3 = 1) = 0{,}6$

Orden de las variables: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0.85	0.55	0.23	0.91
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0	1	1	0

UPV - EHU

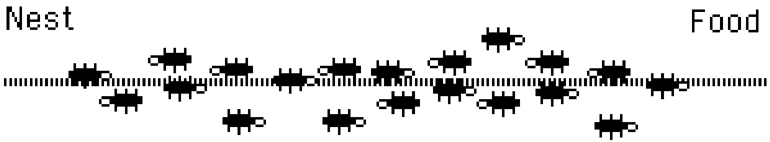
Organización del tema

Swarm Intelligence

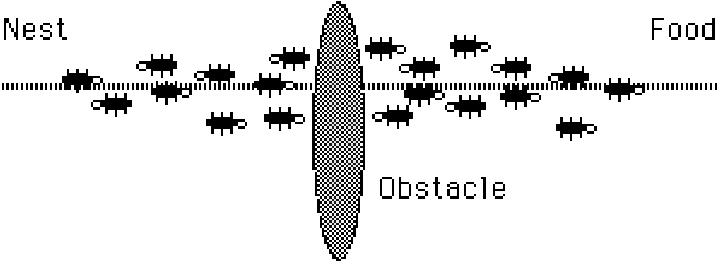
Características principales

- Inspirados en el comportamiento de los insectos: colonias de hormigas, enjambres, etc..
- Individuos con capacidades limitadas resuelven en grupo problemas muy complejos
- La adaptación para la resolución de problemas de optimización no es trivial

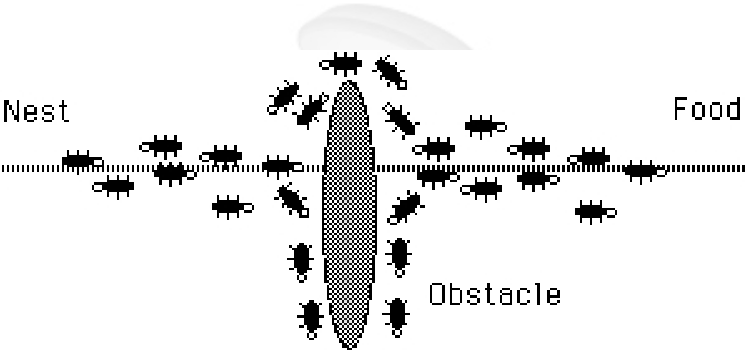
Algoritmo basado en colonias de hormigas



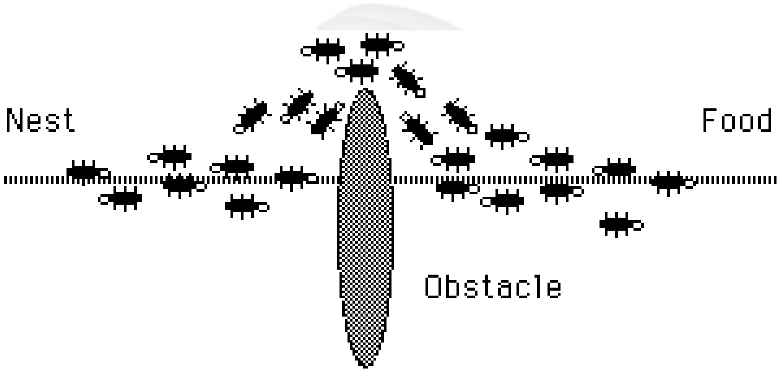
Algoritmo basado en colonias de hormigas



Algoritmo basado en colonias de hormigas



Algoritmo basado en colonias de hormigas



ACH aplicado al TSP

Aspectos generales

- Disponemos inicialmente una hormiga en cada ciudad
- A cada paso cada hormiga elige una ciudad no visitada anteriormente
- Tras n pasos todas las hormigas han completado una ruta
- Se procede a la modificación de la feromona

ACH aplicado al TSP

Método constructivo

- A cada paso se elige una ciudad
- ¿Cómo se elige la ciudad?

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in \mathcal{N}_i^k} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \quad \text{si} \quad j \in \mathcal{N}_i^k$$

probabilidad de que la k-ésima hormiga situada en la ciudad i-ésima elija la j-ésima ciudad

UPV - EHU

ACH aplicado al TSP

Método constructivo

$$p_{ij}^k = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in \mathcal{N}_i^k} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \quad \text{si} \quad j \in \mathcal{N}_i^k$$

- τ_{ij} es la feromona de positada entre las ciudades (i, j)
- η_{ij} es el valor del heurístico entre las ciudades (i, j)
- $\mathcal{N}_i^k$ ciudades a las que puede ir la hormiga k-ésima que se encuentra en la ciudad i-ésima
- α y β son parámetros del algoritmo

ACH aplicado al TSP

Método constructivo

$$p_{ij}^k = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in \mathcal{N}_i^k} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \quad \text{si} \quad j \in \mathcal{N}_i^k$$

- Posible valor para $\eta_{ij} = \frac{1}{d_{ij}}$
- Si $\alpha = 0$, el algoritmo es equivalente a un algoritmo greedy aleatorio
- Si $\beta = 0$, solo se tiene en cuenta la feromona, el algoritmo converge rápidamente a una mala solución
- Sería posible utilizar un único parámetro mezclando ambos

ACH aplicado al TSP

Modificación de la feromona

- Disipación de la feromona $\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \quad (0 < \rho < 1)$
- Incremento de feromona:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad \forall (i,j)$$

donde

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k} & \text{si el arco } (i,j) \text{ está en } T^k \\ 0 & \text{en otro caso} \end{cases}$$

siendo T^k la ruta de la hormiga k-ésima

ACH aplicado al TSP. Mejoras

Modelo elitista

- Se incrementa la feromona de los arcos en la mejor ruta encontrada hasta la fecha T^{bs}:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + e\Delta\tau_{ij}^{bs} \quad \forall(i,j)$$

con

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{C^{bs}} & \text{si el arco } (i,j) \text{ está en } T^{bs} \\ 0 & \text{en otro caso} \end{cases}$$

donde e es un nuevo parámetro

UPV - EHU

ACH aplicado al TSP. Mejoras

Modelo basado en ranking

- Cada hormiga deposita una cantidad de feromona que decrece con su ranking

Modelo Max-Min

- Sólo se utiliza la mejor ruta de cada iteración y la mejor globalmente para modificar la feromona
- La feromona se encuentra en un rango $[\tau_{min}, \tau_{max}]$

UPV - EHU

Organización del tema

UPV - EHU