Tema 3: Algoritmos de Estimación de Distribuciones

Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga

Departamento de Ciencias de la Computación e Inteligencia Artificial

Universidad del País Vasco

http://www.sc.ehu.es/isg/

Introducción

Motivación

- Algoritmos Evolutivos
  - varios parámetros a determinar
  - dificultad en la predicción de las poblaciones a través de las generaciones
  - *building blocks*
  - relación entre las variables (*linkage learning*)
  - problemas engañosos

Introducción

Nueva aproximación a la Computación Evolutiva

- Basada en poblaciones
- Sin operadores de cruce ni mutación
- En cada generación se estima de los individuos seleccionados, la distribución de probabilidad subyacente a los mismos
- Muestreando esta distribución se obtiene la siguiente población
- Se repiten los dos pasos anteriores hasta el criterio de terminación

EDA (Estimation of Distribution Algorithms) Mühlenbein y Paaβ (1996), Larrañaga y Lozano (2002)

EDAs por medio de un ejemplo

$$\max\ h(\boldsymbol{x}) = \sum_{i=1}^{6} x_i \text{ con } x_i = 0, 1$$

(a) D_0 $\qquad p_0(X_i = 1) = 0{,}5$ para $i = 1, \ldots, 6$

EDAs por medio de un ejemplo

	X_1	X_2	X_3	X_4	X_5	X_6	$h(\boldsymbol{x})$
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

EDAs por medio de un ejemplo

(b) $|D_0^{Se}| = 10$ truncación

	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

EDAs por medio de un ejemplo

(c)

$$p_1(\boldsymbol{x}) = p_1(x_1, \ldots, x_6) = \prod_{i=1}^{6} p(x_i|D_0^{Se})$$

modelo a aprender

$$\hat{p}(X_1 = 1|D_0^{Se}) = 0{,}7$$

$$\hat{p}(X_2 = 1|D_0^{Se}) = 0{,}7$$

$$\hat{p}(X_3 = 1|D_0^{Se}) = 0{,}6$$

$$\hat{p}(X_4 = 1|D_0^{Se}) = 0{,}6$$

$$\hat{p}(X_5 = 1|D_0^{Se}) = 0{,}8$$

$$\hat{p}(X_6 = 1|D_0^{Se}) = 0{,}7$$

EDAs por medio de un ejemplo

(d) muestreando $p_1(\boldsymbol{x})$

D_1

	X_1	X_2	X_3	X_4	X_5	X_6	$h(\boldsymbol{x})$
1	1	1	1	1	1	1	6
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

EDAs por medio de un ejemplo

(e) $|D_1^{Se}| = 10$ truncación

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	1	1	1	1	1
2	1	0	1	0	1	1
3	1	1	1	1	1	0
5	1	1	1	1	0	1
6	1	0	0	1	1	1
8	1	1	1	0	1	0
9	1	1	1	0	0	1
15	1	1	1	1	1	1
17	1	1	1	1	1	0
20	1	1	0	1	1	1

EDAs por medio de un ejemplo

(f) repetir

- Seleccionar Se individuos de D_l obteniendo D_l^{Se}
- Aprender la distribución de probabilidad de los seleccionados

$$p_l(\boldsymbol{x}) = \prod_{i=1}^{6} p(x_i | D_{l-1}^{Se})$$

- Muestrear $p_l(\boldsymbol{x})$ obteniendo D_l

Aproximación general

D_0

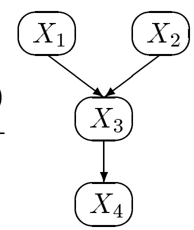
	X_1	X_2	X_3	X_4	eval
1	1	0	1	1	13.25
2	0	0	1	1	32.45
⋮	⋮	⋮	⋮	⋮	⋮
N	1	1	0	0	34.12

Selection of Se<N individuals

D_{l-1}^{Se}

	X_1	X_2	X_3	X_4	eval
1	0	0	1	1	32.25
2	1	1	1	1	33.78
⋮	⋮	⋮	⋮	⋮	⋮
Se	1	1	0	0	34.12

Induction of the probability model



$\rho_l(\boldsymbol{x}) = \rho_l(\boldsymbol{x}|D_{l-1}^{Se})$

Sampling from $\rho_l(\boldsymbol{x})$

D_l

	X_1	X_2	X_3	X_4	eval
1	1	1	1	1	33.78
2	0	0	1	1	32.45
⋮	⋮	⋮	⋮	⋮	⋮
N	0	0	0	0	37.26

Selection of Se<N individuals

EDAs Aproximación general

$D_0 \leftarrow$ Generar N individuos (la población inicial) al az

Repetir para $l = 1, 2, \ldots$ hasta la condición de parada

$D_{l-1}^{Se} \leftarrow$ Seleccionar $Se \leq N$ individuos de D_{l-1} siguiendo un método de selección

$p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D_{l-1}^{Se}) \leftarrow$ Estimar la distribución de probabilic de los individuos seleccionados

$D_l \leftarrow$ Muestrear N individuos (la nueva población) de p

UMDA

UMDA (Univariate Marginal Distribution Algorithm) Mühlenbein, 1998

$D_0 \leftarrow$ Generar M individuos (la población inicial) al azar

Repeat for $l = 1, 2, \ldots$ hasta que se verifique el criterio de parada

$D_{l-1}^{Se} \leftarrow$ Seleccionar $N \leq M$ individuos de D_{l-1} de acorde con un método de selección

$p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D_{l-1}^{Se}) = \prod_{i=1}^{n} p_l(x_i) = \prod_{i=1}^{n} \frac{\sum_{j=1}^{N} \delta_j(X_i = x_i|D_{l-1}^{Se})}{N} \leftarrow$ Estimar la distribución de probabilidad conjunta

$D_l \leftarrow$ Muestrear M individuos (la nueva población) de $p_l(\boldsymbol{x})$

PBIL

PBIL (Population Based Incremental Learning) Baluja, 1994

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

---

cGA

cGA (compact Genetic Algorithm) Harik et al., 1998

```
Paso 1. Inicializar el vector de probabilidades p_0(x)
        p_0(x) = p_0(x_1,...,x_i,...,x_n) = (p_0(x_1),...,p_0(x_i),...,p_0(x_n)) =
         (0,5,...,0,5,...,0,5)
Paso 2. l = l + 1. Muestrear p_l(x) with l = 0,1,2,... obteniendo dos
        individuos: x_1^l, x_2^l
Paso 3. Evaluar y ordenar x_1^l and x_2^l obteniendo:
        x_{1:2}^l (el mejor) y x_{2:2}^l (el peor)
Paso 4. Actualizar el vector de probabilidades p_l(x) hacia x_{1:2}^l
        for i = 1 to n
         if x_{i,1:2}^l ≠ x_{i,2:2}^l entonces
           if x_{i,1:2}^l = 1 entonces p_l(x_i) = p_{l-1}(x_i) + 1/K
           if x_{i,1:2}^l = 0 entonces p_l(x_i) = p_{l-1}(x_i) - 1/K
Paso 5. Chequear si el vector de probabilidades p_l(x) ha convergido
        for i = 1 to n do
         if p_l(x_i) > 0 y p_l(x_i) < 1 entonces volver al Paso 2
Paso 6. p_l(x) representa la solución final
```

Sin dependencias