UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Transmisia datelor multimedia in retele de calculatoare

6. Codificare Huffman

Codari Prefix Optimale

- **Codari optimale**
  1. Simbolurile care apar mai frecvent vor avea coduri mai scurte
  2. Ultimele 2 cele mai putin frecvente simboluri vor avea coduri de lungimi egale
- Demonstratie:
  1. Vrem sa demonstram faptul ca codul este clar suboptimal
  2. Presupunem opusul
     - Fie X, Y cele mai putin frecvente simboluri si
     - |cod(X)| = k, |cod(Y)| = k+1

     Atunci
     - Datorita decodificarii unice(UD), cod(X) nu poate fi prefix pt cod(Y)
     - Deasemenea toate celelalte coduri sunt mai scurte

  ➔ Eliminand ultim bit al lui|cod(Y)| ar genera un nou cod scurt unic decodificabil, eea ce contrazice presupunerea initiala de optimalitate

Codificare Huffman

- Algoritmul Huffman ia ca intrare o lista de ponderi ne-negative {$w(1)$, ... ,$w(n)$ } si construieste un *arbore binar complet* ale carui frunze sunt numerotate cu ponderi
  - un arbore binar este complet daca fiecare nod are zero sau 2 ramificatii
- Ponderile reprezinta probabilitatile asociate simbolurilor sursei
- Initial arborele are numai doua noduri, cele corespunzatoare ponderilor celor mai mici
- La fiecare pas in algoritm, cele mai mici ponderi definesc un nou nod cu ponderea $w(i)+w(j)$ si a carui radacina (*root*) are doi sub-arbori, reprezentati de *w(i)* si *w(j)*
- Ponderile *w(i)* si *w(j)* sunt indepartate din lista si locul lor este preluat de *w(i)+w(j)*
- Procesul continua pana cand se obtine o lista cu o singura valoare.

Codificare Huffman

- Algoritmul Huffman (1952) constituie un algoritm optimal, în sensul că nici un alt algoritm nu asigură o mai mică lungime medie a cuvintelor
  - Sunt situaţii în care şi alţi algoritmi pot da o lungime medie egală cu cea dată de algoritmul Huffman, dar niciodată mai mică

- Exemplu:

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2		
b		0.4		
c		0.2		
d		0.1		
e		0.1		

Codificare Huffman

- Initializare: Creeaza o multime din fiecare litera

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	a	0.2
b		0.4	b	0.4
c		0.2	c	0.2
d		0.1	d	0.1
e		0.1	e	0.1

Codificare Huffman

1. Sorteaza multimile dupa probabilitate

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	d	0.1
b		0.4	e	0.1
c		0.2	a	0.2
d		0.1	c	0.2
e		0.1	b	0.4

Codificare Huffman

2. Insereaza prefixul ‘1’ in codurile literelor din prima multime

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	d	0.1
b		0.4	e	0.1
c		0.2	a	0.2
d	1	0.1	c	0.2
e		0.1	b	0.4

Codificare Huffman

3. Insereaza prefixul ‘0’ in codurile literelor din a doua multime

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	d	0.1
b		0.4	e	0.1
c		0.2	a	0.2
d	1	0.1	c	0.2
e	0	0.1	b	0.4

Codificare Huffman

4. Uneste primele 2 multimi

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	de	0.2
b		0.4	a	0.2
c		0.2	c	0.2
d	1	0.1	d	0.4
e	0	0.1		

Codificare Huffman

1. Sorteaza crescator multimile dupa probabilitate

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	de	0.2
b		0.4	a	0.2
c		0.2	c	0.2
d	1	0.1	b	0.4
e	0	0.1		

Codificare Huffman

2. Insereaza prefixul ‘1’ in codurile literelor din prima multime

Litera	Cod	Probabilitate	Set	Prob Set
a		0.2	de	0.2
b		0.4	a	0.2
c		0.2	c	0.2
d	11	0.1	b	0.4
e	10	0.1		

Codificare Huffman

3. Insereaza prefixul ‘0’ in codurile literelor din a doua multime

Litera	Cod	Probabilitate	Set	Prob Set
a	0	0.2	de	0.2
b		0.4	a	0.2
c		0.2	c	0.2
d	11	0.1	b	0.4
e	10	0.1		

Codificare Huffman

4. Uneste primele 2 multimi

Litera	Cod	Probabilitate	Set	Prob Set
a	0	0.2	dea	0.4
b		0.4	c	0.2
c		0.2	b	0.4
d	11	0.1		
e	10	0.1		

Codificare Huffman

1. Sorteaza crescator multimile dupa probabilitate

Litera	Cod	Probabilitate	Set	Prob Set
a	0	0.2	c	0.2
b		0.4	dea	0.4
c		0.2	b	0.4
d	11	0.1		
e	10	0.1		

Codificare Huffman

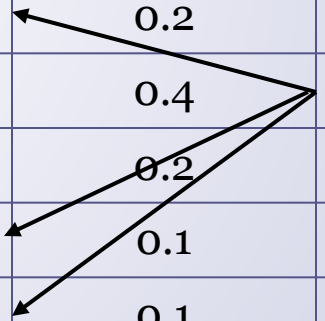
2. Insereaza prefixul ‘1’ in codurile literelor din prima multime

Litera	Cod	Probabilitate	Set	Prob Set
a	0	0.2	c	0.2
b		0.4	dea	0.4
c	1	0.2	b	0.4
d	11	0.1		
e	10	0.1		

Codificare Huffman

3. Insereaza prefixul ‘0 in codurile literelor din a doua multime

Litera	Cod	Probabilitate	Set	Prob Set
a	00	0.2	c	0.2
b		0.4	dea	0.4
c	1	0.2	b	0.4
d	011	0.1		
e	001	0.1		



Codificare Huffman

4. Uneste primele 2 multimi

Litera	Cod	Probabilitate	Set	Prob Set
a	00	0.2	cdea	0.6
b		0.4	b	0.4
c	1	0.2		
d	011	0.1		
e	010	0.1		

Codificare Huffman

1. Sorteaza crescator multimile dupa probabilitate

Litera	Cod	Probabilitate	Set	Prob Set
a	00	0.2	b	0.4
b		0.4	cdea	0.6
c	1	0.2		
d	011	0.1		
e	010	0.1		

Codificare Huffman

2. Insereaza prefixul ‘1’ in codurile literelor din prima multime

Litera	Cod	Probabilitate	Set	Prob Set
a	00	0.2	b	0.4
b	1	0.4	cdea	0.6
c	1	0.2		
d	011	0.1		
e	010	0.1		

Codificare Huffman

3. Insereaza prefixul ‘0’ in codurile literelor din a doua multime

Litera	Cod	Probabilitate	Set	Prob Set
a	000	0.2	b	0.4
b	1	0.4	cdea	0.6
c	01	0.2		
d	0011	0.1		
e	0010	0.1		

Codificare Huffman

4. Uneste primele 2 multimi

Litera	Cod	Probabilitate	Set	Prob Set
a	000	0.2	abcde	1.0
b	1	0.4		
c	01	0.2		
d	0011	0.1		
e	0010	0.1		

- Sfarsit

Codificare Huffman

- Statistici exemplu:
  - Lungime medie cod
    - $\boldsymbol{l}$ = 0.4x1 + 0.2x2 + 0.2x3 + 0.1x4 + 0.1x4
      = 2.2 bits/symbol
  - Entropie
    - $\boldsymbol{H} = \Sigma_{s=a..e} P(s) \log_2 P(s)$ = 2.122 bits/symbol
  - Redundanta
    - $\boldsymbol{l} - \boldsymbol{H}$ = **0.078** bits/symbol

- Peformantele sunt identice cu cele obtinute folosind codarea Shannon-Fano
  - Diferenta consta in faptul ca prin codarea Huffman se garanteaza obtinerea unui **cod optimal**