ASUPRA COMPLEXITATII MATEMATICE A FILTRARII PE STRUCTURI DE SEMIGRUP

Dinu COLTUC
FIE, Universitatea Valahia Targoviste,
email: coltuc@valahia.ro

CEA-UPB – 15 mai 2008

PLAN

- Introduction
- Complexité mathématique
- Filtrage max/min
- Moyenneur & lissage exponentiel
- Conclusions

INTRODUCTION

Monoïde: structure algébrique

- ensemble E
- loi de composition interne associative $\diamond$
- (élément neutre)

Exemples (E, $\diamond$):

- (ensemble des entiers naturels, addition)
- (ensemble des parties d'un ensemble E, union ensembliste)
- (ensemble des entiers naturels, max/min)
- (ensemble des entiers naturels, multiplication)

Le filtrage:

- $x_i \in E$, $i = 0, \ldots, N$ - l'entrée
- $y_i \in E$, $i = 0, \ldots, N - p + 1$ - la sortie
- p: la fenêtre du filtre

$$y_i = x_i \diamond x_{i+1} \diamond \ldots \diamond x_{i+p-1}$$

Le but

- la complexité mathématique du filtrage
- l'existance d'algorithmes optimaux
- l'extension pour d'autre structures algébriques

COMPLEXITE MATHEMATIQUE (*IEEE TSP'08*)

On va examiner le calcul pour toute la séquence par groups de n

- $O(n)$ - le nombre d'operations ($\diamond$) pour calculer n résultats
- $C(n)$ - la complexite mathematique per échantillon:

$$C(n) = \frac{O(n)}{n} \text{ operations/échantillon}$$

- C la complexité du filtrage pour toute la sequence:

$$C = \min_n \frac{O(n)}{n} \text{ operations/échantillon}$$

- $n = 1$

$$y_0 = x_0 \diamond x_1 \diamond \ldots \diamond x_{p-1}$$

  - $O(1) = p - 1$
  - $C(1) = p - 1$

- $n = 2$

$$\begin{aligned} y_0 &= x_0 \diamond \ (x_1 \diamond x_2 \ldots \diamond x_{p-1}) \\ y_1 &= \quad (x_1 \diamond x_2 \ldots \diamond x_{p-1}) \diamond x_p \end{aligned}$$

  - $p-1$ échantillons d'entré sont communs $\Rightarrow p-2$ operations
  - $O(2) = p - 2 + 1 + 1$
  - $C(2) = p/2$

$$C(1) > C(2)$$

- $n \leq p$

$$x_0 x_1 \ldots x_{p-1} | x_p \ldots x_{p+n-1}$$

  – les échantillons d'entré divisé en 2 groupes

  – le minimum d'opérations:

    * calculer y_0 (1er group) avec $p-1$ opérations.
    * calculer le resultat partiel du 2eme group avec $n-2$ op.
    * et calculer les $n-1$ résultats finaux avec $n-1$ op.

  – $O(n) = (p-1) + (n-2) + (n-1)$

  – $C(n) = \frac{p+2n-4}{n} = 2 + \frac{p-4}{n}$

- $C(n)$ décroît avec $n \Rightarrow$ le minimum pour $n = p$

$$C(p) = \frac{3p-4}{p}$$

- **L'algorithme:**

$$x_0 x_1 \dots x_{p-2} x_{p-1} | x_p x_{p+1} \dots x_{p+n-1}$$

Group I	**Group II**
$R_{p-1} = x_{p-1}$	$S_0 = x_p$
$R_{p-2} = x_{p-2} \diamond R_{p-1}$	$S_1 = S_0 \diamond x_{p+1}$
$\vdots$	$\vdots$
$R_i = x_i \diamond R_{i+1}$	$S_i = S_{i-1} \diamond x_{p+i}$
$\vdots$	$\vdots$
$R_0 = x_0 \diamond R_1$	$S_{n-2} = S_{n-3} \diamond x_{n+p-2}$

Les résultats finaux:

$$y_0 = R_0,\ y_i = R_i \diamond S_{i-1},\ i = 1, \dots, n-1$$

- le cas $n = p \Rightarrow$ Gil-Werman (PAMI'93), Van Herk (PRL'91)

- $n = p + 1$ (Coltuc, ICECS'96)

$$x_0 x_1 \ldots x_{p-1} | x_p \ldots x_{2p-1}$$

  – les échantillons d'entré divisé en 2 groupes

  – le minimum d'opérations:

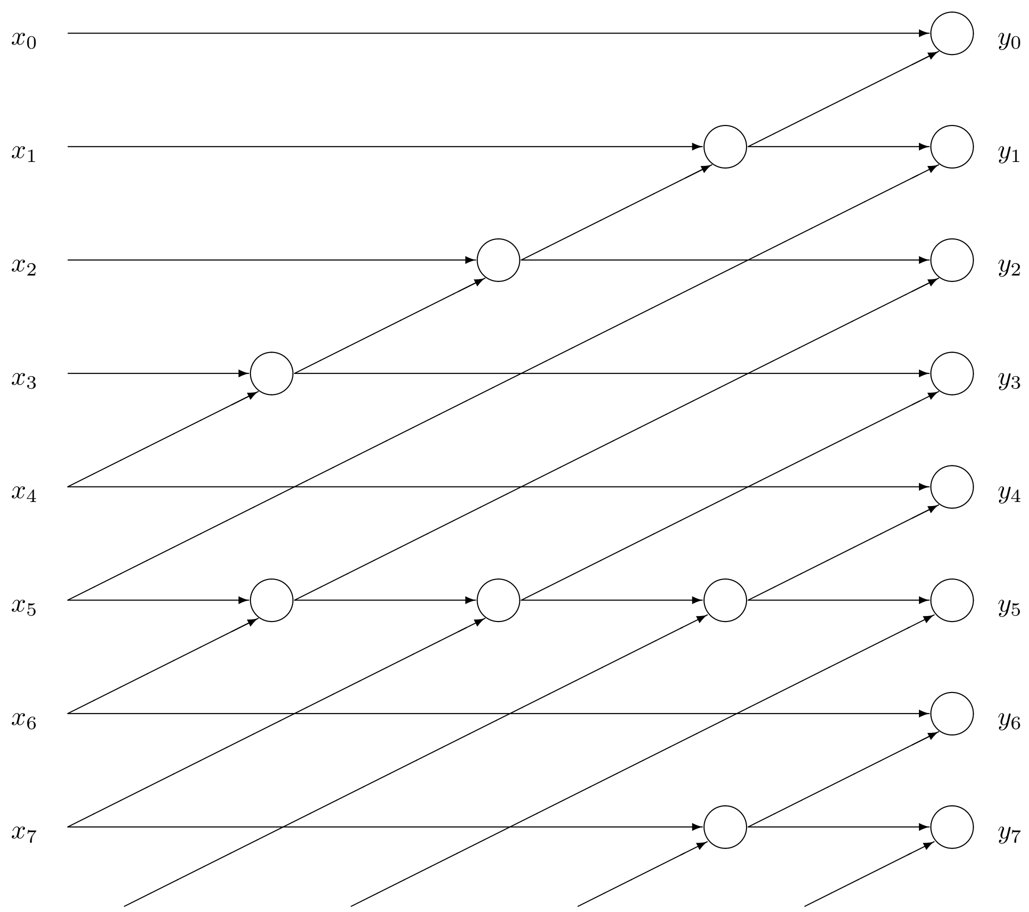
    * calculer y_0 (group I) avec $p - 1$ opérations
    * calculer y_p (group II) avec $p - 1$ opérations
    * calculer les autre $p - 1$ résultats finaux avec $p - 1$ op.

  – $O(p + 1) = (p - 1) + (p - 2) + (p - 1)$

  – $C(p + 1) = \frac{3p-3}{p+1} = 3 - \frac{6}{p+1}$

$$C(1) > C(2) > \ldots > C(p) > C(p + 1)$$

- **L'algorithme:** exemple $p = 5 \Rightarrow C(6) = 2$ opérations/échantillon



- $n = p + 2$
  - calculer les y_j qui n'ont pas des x_i commun ($2p - 2$ op.)
  - calculer les y_k qui restent en utilisant des résultats partiels

$$x_0 x_1 \ldots x_{p-1} | x_p x_{p+1} \ldots x_{2p-1} | x_{2p}$$

$$x_0 | x_1 \ldots x_{p-1} x_p | x_{p+1} \ldots x_{2p-1} x_{2p} |$$

$$x_0 x_1 \ldots x_{p-1} | x_p | x_{p+1} \ldots x_{2p-1} x_{2p}$$

  - $O(p+2) = 4p - 4 \Rightarrow C(p+2) = 4 - \frac{12}{p+2}$
  - $C(1) > C(2) > \ldots > C(p) > C(p+1) < C(p+2)$

- $n = p + 3, p + 4, ..., 2p + 2$
  - $\ldots > C(p+1) < C(p+2) > ... > C(2p+2) = C(p+1)$

Commentaires

- la complexité du filtrage sur monoïdes dans une fenêtre p:

$$C_p \geq 3 - \frac{6}{p+1}$$

- C_p est croissante en fonction de p
- $C_p < 3$ opérations/échantillon, $\forall p$
- algorithmes optimaux existent $\forall p$
- unicité? non (pour certain p impaires)
- seulement l'associativité de $\diamond$ a été consideré
- si $\diamond$ a d'autres propriétés, la complexité peut diminuer

FILTRAGE MAX/MIN

- applications - signal et image (morphologie, filtrage d'ordre)
- E: l'ensemble des entiers
- (E,max), (E,min) monoïdes $\Rightarrow$ complexité du filtrage:

$$C_p = 3 - \frac{6}{p+1}$$

- C_p **independant** de la statistique des données
- le gain, par raport a l'algorithme de Gil-Werman:

$$\Delta C_{OGW} = \frac{2}{p+1} - \frac{4}{p(p+1)} \approx \frac{2}{p}$$

- max, min $\Rightarrow$ d'autres propriétés (treillis ordonée)

MAX/MIN - Gevorkian et al. (*IEEE Trans on PAMI'96*)

- algorithme dépendent de la statistique
- basé sur l'algorithme de Gil-Werman +
  - l'information aquise à l'iteration anterieure - la position du max: le dernier ou avant-dernier: tous R_i^{k+1} disponible ($p-1$ comparaisons eliminé), etc.
- complexité (iid): $C_G = 2.5 - \frac{3.5}{p} + \frac{1}{p^2}$;
- **avec l'algorithme optimal:** $C_{OG} = 2.5 - \frac{4}{p+1} - \frac{1}{p(p+1)}$
- **le gain:**

$$\Delta C_G = \frac{1}{2(p+1)} - \frac{3}{2p(p+1)} + \frac{1}{p^2(p+1)} \approx \frac{1}{2p}$$

MAX/MIN Gil-Kimmel (*IEEE Trans on PAMI*, dec. 2002)

- algorithme independent de données
- le plus rapide algorithme publié
- basé sur l'algorithm de Gil-Werman +
  - étape calcul preliminaire: la demi-fenêtre où est le max
    - 1 comparison pour éliminer $\frac{p-1}{2}$
  - étape calcul final: observe que les R_i et S_j sont ordonnées
    - $\log p$ a la place de p

MAX/MIN Gil-Kimmel

- complexité:

$$C_{GK} = 1.5 + \frac{\lceil \log_2(p-1) \rceil}{p} - \frac{p \bmod 2}{2p}$$

- avec **l'algorithm optimal:**

$$C = 1.5 - \frac{3}{2(p+1)} + \frac{\lceil \log_2 p \rceil}{p+1} - \frac{p \bmod 2}{2(p+1)}$$

- **le gain:**

$$\Delta C(p = 2^k + 1) = \frac{1}{2(p+1)} + \frac{\log_2(p-1) + 1}{2p(p+1)} \approx \frac{1}{2p}$$

$$\Delta C(p \neq 2^k + 1) = \frac{3}{2(p+1)} + \frac{2\lceil \log_2 p \rceil - p \bmod 2}{2p(p+1)} \approx \frac{3}{2p}$$

MAX/MIN - plus rapide (*Coltuc, IEEE Trans on SP'08*)

- x_i ordonées $\Rightarrow$ complexité: 1 comparaison/échantillon
- exploiter l'ordre $\Rightarrow$ elargir le group de $p+1$ à $p+r+s+1$
  - $x_0, x_1, \ldots, x_{2p+r+s-1}$

$$x_{p-1} \geq x_p \geq x_{p+1} \geq \ldots \geq x_{p+r-1}$$

$$x_{p+r} \leq x_{p+r+1} \leq \ldots \leq x_{p+r+s-1} \leq x_{p+r+s}$$

  - soit $t = r + s$; la complexité devient:

$$C_{p,t} = 3 - \frac{2t+6}{p+t+1}$$

  - x_i **ordonées** $\Rightarrow C_{p,t} = 1$
  - x_i **IID** $\Rightarrow t_{moy} = 2$:

$$C_{p,IID} = 3 - \frac{10}{p+3}$$

Max/min: (*Coltuc, IEEE Trans on SP'08*)

Algorithm	Mathematical complexity [comparisons/sample]	Memory [operands/iteration]
Gil-Werman	$3 - \frac{4}{p}$	$2p - 1$
Optimal semi-group	$3 - \frac{6}{p+1}$	$2p$
Extended semi-group	$3 - \frac{6+2T}{p+1+T}$	$2p + T$
Gil-Kimmel	$1.5 + \frac{\lceil \log_2(p-1) \rceil}{p} - \frac{p \bmod 2}{2p}$	$2p + \lceil \frac{p+1}{2} \rceil - 1$
Improved Gil-Kimmel	$1.5 - \frac{1.5}{p+1} + \frac{\lceil \log_2 p \rceil}{p+1} - \frac{p \bmod 2}{2(p+1)}$	$2p + \lceil \frac{p+1}{2} \rceil$
Extended Gil-Kimmel	$\frac{1.5p - (p \bmod 2)/2 + \lceil \log_2 p \rceil + T}{p+1+T}$	$2p + \lceil \frac{p+1}{2} \rceil + T$

LE MOYENNEUR

- on élimine les divisions par p (x_i/p, y_i/p)

$$y_i = x_i + x_{i+1} + \ldots + x_{i+p-1}$$

- (E,+) monoïde $\Rightarrow$ avec l'algorithme optimal

$$C_M = 3 - \frac{6}{p+1} \rightarrow 3 \ \text{ additions/échantillon}$$

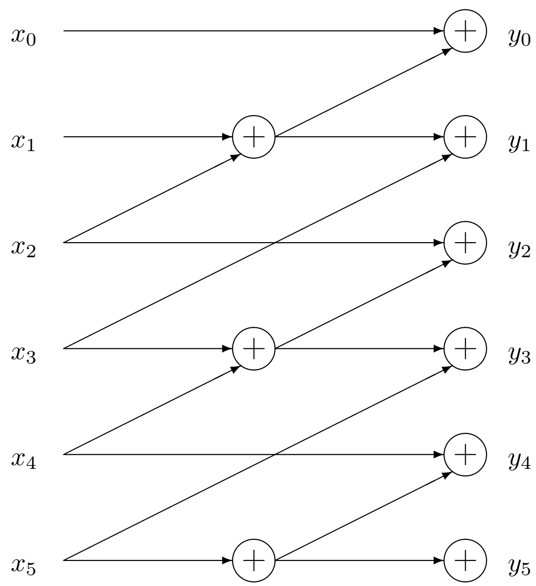
- (E,+) $\Rightarrow$ structure de group (x_i inversible)

$$y_{i+1} = y_i + x_{i+p} - x_i$$

- complexité: 2 additions/échantillon

LE MOYENNEUR

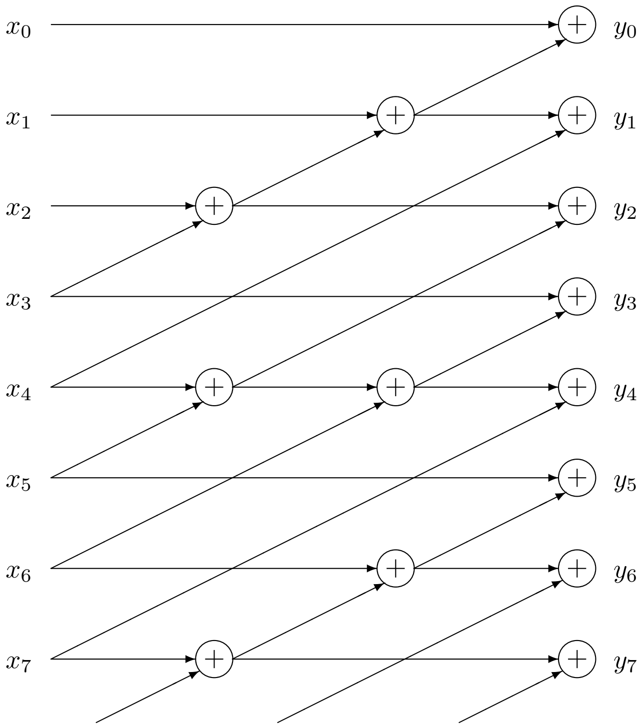
- $p = 3 \Rightarrow$ 1.5 additions/échantillon



- moyenneur $3 \times 3 \Rightarrow 3$ additions /échantillon

LE MOYENNEUR

- $p = 4 \Rightarrow 1.8$ additions/échantillon



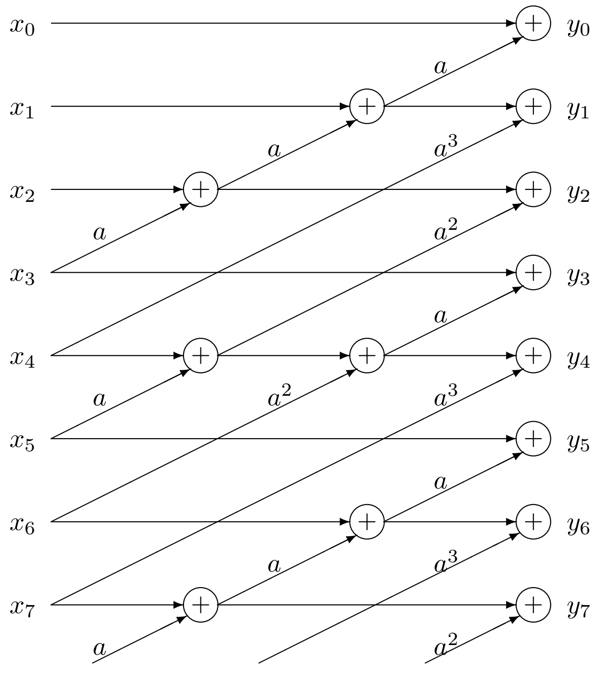
- $p = 5 \Rightarrow 2$ additions/échantillon
- $p > 5 \Rightarrow 2 < C_p < 3 \Rightarrow$ **sans soustractions**

LISSAGE EXPONENTIEL

- le même problème $y_i = \Sigma_{k=0}^{p-1} a^i x_{i+k}$
- calcul récursif $\Rightarrow$ 2 multiplications + 2 additions /ech.

$$y_{i+1} = (y_i - x_{i-p})/a + a^{p-1} x_{i+p}$$

- l'algorithme optimal $\Rightarrow$ moins complexe pour $p = 3, 4$



CONCLUSIONS

- nous avons montré que le filtrage sur un monoïde est de complexité $C(n) \geq 3 - \frac{6}{p+1}$ (p est la taille de la fenêtre)
- algorithmes optimaux existent pour $\forall$ fenetres (1D)
- si apart l'associativité, l'opération a d'autres propriétés, la complexité peut diminuer
- les performances des algorithmes max/min basés sur le schema de Gil-Werman sont ameliorées

  $-$ est-ce qu'on peut aller plus loin? $(1.25 < C_p < 1.5)$
- le moyenneur et le lissage exponentiel $\Rightarrow$ calcul moins complexe que par l'algorithm recursif pour fenêtres de petite taile