Probabilistic computation of the Smith normal form of a sparse integer matrix

Mark Giesbrecht

1996

Introduction

Nouvel algorithme pour calculer la forme normale de Smith de $A \in \mathcal{M}_{m,n}(\mathbb{Z})$:

- probabiliste de type Monte-Carlo (bon résultat avec probabilité $\geqslant \epsilon$)
- concept de boîte noire
- exploite la structure creuse

Introduction

Complexité temporelle : $O^{\sim}\left(m^2 \log\left(\|A\|\right) \log\left(\frac{1}{\epsilon}\right)\right)$ produits matrice-vecteur

Problème : convergence non démontrée

Plan

Première partie I

Préliminaires

Smith (1861)

$A \in \mathcal{M}_{m,n}(\mathbb{Z})$

$$S = PAQ = \begin{pmatrix} s_1 & 0 & \dots & \dots & \dots & 0 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & & & \vdots & \vdots & & \vdots \\ \vdots & \ddots & s_r & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & & \ddots & 0 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & & & \ddots & \ddots & 0 & \vdots & & \vdots \\ 0 & \dots & \dots & \dots & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

- P et Q unimodulaires
- $r = \mathrm{rg}(A)$
- S unique telle que $s_i | s_{i+1}$

Diviseurs déterminants

Définition : d_k, le $k^{\text{ème}}$ diviseur déterminant de A, est le pgcd de tous les mineurs de A de taille $k \times k$.

exemple

Propriété : On pose $d_0 = 1$. Pour tout i, $s_i = \dfrac{d_i}{d_{i-1}}$.

Présentation

On ne connaît pas directement la matrice A.

Deuxième partie II

Algorithme

Entrées-sorties

Entrées :

- boîte noire pour A
- tolérance $\epsilon > 0$

Sorties :

- $r = \mathrm{rg}(A)$
- $d_1, \ldots, d_r$

Les sorties sont correctes avec probabilité $1 - \epsilon$.

Initialisation et test

- Initialisation :
  $r, d_1, \ldots, d_m, count \leftarrow 0$
- Boucle principale, à répéter tant que
  $$count \leqslant \left(\log\left(\frac{1}{\epsilon}\right) + 3\log(m) + \log(\log(\|A\|)) \right)$$
- Test final :
  Si $d_r = d_1 \prod_{i=2}^{r} \frac{d_i}{d_{i-1}}$
  Alors renvoyer r et $(d_1, \ldots, d_r)$
  Sinon recommencer

Boucle principale (1/4)

Choisir au hasard :

$$U = \begin{pmatrix} 1 & \alpha_2 & \dots & \alpha_m \\ & 1 & \ddots & \vdots \\ & & \ddots & \alpha_2 \\ & & & 1 \end{pmatrix} ; \; L = \begin{pmatrix} 1 & & \\ \beta_2 & \ddots & \\ \vdots & \ddots & 1 \\ \vdots & & \beta_2 \\ \vdots & & \vdots \\ \beta_n & \dots & \beta_{n-m+1} \end{pmatrix}$$

$$D = \begin{pmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_m \end{pmatrix} \qquad \text{On pose : } B = UALD$$

Boucle principale (2/4)

- Calculer le polynôme minimal de B :

$$g = x^s + \sum_{k=0}^{s-1} b_k x^k$$

- Le polynôme caractéristique de B est *probablement* gx^{m-s}.

Boucle principale (3/4)

On pose $k = \text{Min}\{i \in \mathbb{N},\ b_i \neq 0\}$:

- si $k > 1$ ou $(k = 0 \wedge s < m)$ alors recommencer
- si $k = 0$ alors poser $\bar{r} = m$
- si $k = 1$ alors poser $\bar{r} = s - 1$

Boucle principale (4/4)

Si $\bar{r} = r$:

- incrémenter *count*
- pour i de 1 à r, poser $d_i = \text{pgcd}(d_i, b_{m-i})$

Si $\bar{r} > r$:

- poser $r = \bar{r}$ et $count = 1$
- pour i de 1 à r, poser $d_i = b_{m-i}$

Boucle principale (4/4)

Si $\bar{r} = r$:

- incrémenter *count*
- pour i de 1 à r, poser $d_i = \text{pgcd}(d_i, b_{m-i})$

Si $\bar{r} > r$:

- poser $r = \bar{r}$ et $count = 1$
- pour i de 1 à r, poser $d_i = b_{m-i}$

Théorème

Théorème : On utilise les notations de l'algorithme. On note $f = \sum_{i=0}^{n} a_i x^i$ le polynôme caractéristique de B.
Alors, pour tout $k \in [\![1, n]\!]$, $d_k | a_{n-k}$.

idée de la preuve

Complexité et convergence

Si convergence : $O^{\sim}\left(m^2 \log\left(\|A\|\right) \log\left(\frac{1}{\epsilon}\right)\right)$ produits matrice-vecteur (modulo un entier premier) et $O^{\sim}\left(\left(m^2 n \log\left(\|A\|\right) + m^3 \log^2\left(\|A\|\right)\right) \log\left(\frac{1}{\epsilon}\right)\right)$ opérations sur des bits

Mais problème de convergence.

Idée

Coefficients de U, L, D non plus choisis dans $\mathbb{Z}$ mais dans un anneau plus grand $\mathcal{R}$.

$\hookrightarrow$ Les coefficients du polynôme caractéristique de B sont polynomiaux.

Théorème : On reprend les notations précédentes.
Alors, pour tout $k \in [\![1, n]\!]$, a_{n-k} a pour contenu d_k.

Contenu d'un polynôme

Définition : Le contenu d'un polynôme à coefficients dans $\mathbb{Q}$ est le pgcd de ses coefficients.

exemple

Principe

Nouvel algorithme Monte-Carlo `FindContents` :

- Entrées : boîte noire évaluant un uplet de polynômes en un point de $\mathcal{R}$, ϵ
- Sorties : contenus de ces polynômes avec probabilité $\geqslant 1 - \epsilon$

Utilisation du pgcd comme précédemment.

Nouvel algorithme

1. Poser r le rang de A
2. Choisir au hasard des matrices U Toeplitz supérieure, L Toeplitz inférieure et D diagonale, à coefficients dans $\mathcal{R}$
3. Poser $B = UALD$ et définir une boîte noire qui évalue les r coefficients de plus grand ordre du polynôme caractéristique de B en $3n - 2$ points de $\mathcal{R}$
4. Appliquer `FindContents` en utilisant cette boîte noire

Nouvel algorithme

1. Poser r le rang de A
2. Choisir au hasard des matrices U Toeplitz supérieure, L Toeplitz inférieure et D diagonale, à coefficients dans $\mathcal{R}$
3. Poser $B = UALD$ et définir une boîte noire qui évalue les r coefficients de plus grand ordre du polynôme caractéristique de B en $3n - 2$ points de $\mathcal{R}$
4. Appliquer `FindContents` en utilisant cette boîte noire

Pourquoi se placer dans $\mathcal{R}$?

Exemple : $f(x) = 5x^4 + 10x^3 - 5x^2 - 10x$.

On peut se placer dans $\mathcal{R} = \mathbb{Z}[z]/\Gamma$ avec $\Gamma = z^3 + 2z^2 + 2z + 3$.

algorithme

Conclusion

- un algorithme probabiliste pour calculer la forme normale de Smith : meilleure complexité que le meilleur algorithme déterministe alors connu
- concept de boîte noire
- problème de convergence

Remarques

- problèmes soulignés (rang de A . . .)
- article hésitant
- implantation pas évidente

Des questions ?