Numerical Reliability of Randomized Algorithms

Inner Product – Two Norm

Ilse Ipsen

North Carolina State University
Raleigh, NC, USA
Randomized Matrix Multiplication

Sarlós 2006
Drineas, Kannan & Mahoney 2006
Belabbas & Wolfe 2008

Goal:

Algorithm behaviour for moderate matrix dimensions
Numerical properties of algorithms

Outline

Randomized inner product – squared two norm
Relative error due to randomization
Repeated sampling of same elements
“Stability” of algorithm
Randomized Inner Product – Squared Two Norm

[Drineas, Kannan & Mahoney 2006]

**Input:** real vector \( a = (a_1 \ldots a_n)^T \)
- probabilities \( p_k > 0, \sum_{k=1}^{n} p_k = 1 \)
- number \( c \) where \( 1 \leq c \leq n \)

**Output:** Approximation \( X \) to \( a^Ta \)
from \( c \) randomly sampled elements \( a_k \)

\[
X = 0 \\
\text{for } t = 1 : c \text{ do} \\
\quad \text{Sample } k_t \text{ from } \{1, \ldots, n\} \text{ with probability } p_{k_t} \text{ independently and with replacement} \\
\quad X = X + \frac{a_{k_t}^2}{c p_{k_t}}
\]
end for
Unbiased estimator

$$E[X] = a^T a$$

Uniform probabilities: $p_k = 1/n$, $1 \leq k \leq n$

Absolute error bound
For every $\delta > 0$ with probability at least $1 - \delta$

$$|a^T a - X| < \frac{n \|a\|_2^2}{\sqrt{c}} \sqrt{8 \ln(2/\delta)}$$
Relative Error due to Randomization
Relative Error Bound

For every $\delta > 0$ with probability at least $1 - \delta$

$$\frac{|X - a^T a|}{a^T a} < \epsilon$$

where

$$\epsilon \geq \frac{1}{\sqrt{c} \delta} \sqrt{\sum_{k=1}^{n} \frac{a_k^4}{p_k \|a\|_2^4} - 1}$$

Proof: Chebyshev inequality

Uniform probabilities $p_k = 1/n$

$$\epsilon \geq \frac{1}{\sqrt{c} \delta} \sqrt{n \left(\frac{\|a\|_4}{\|a\|_2}\right)^4 - 1}$$
$n = 10^6$, $a_k$ are independent uniform $[0, 1]$

Relative errors $|X - a^T a| / a^T a$ for every $c$

Chebyshev bound with probability $.99$
Relative Error for Uniform Probabilities

**Uniform vectors**

\[ a_k \text{ iid uniform } [0, 1], \ n = 10^6 \]

Relative error: \( 10^{-2} - 10^{-1} \)

**Weakly graded vectors**

\[ a = \begin{pmatrix} 1 & 2 & \ldots & n \end{pmatrix}^T \]

*With probability* \( 1 - \delta \): Relative error \( \geq \frac{.8}{\sqrt{\delta \ c}} \)

*With 99 percent probability:*

Relative error \( \approx 10^{-8} \) for \( c \geq 10^{20} \)
Weakly Graded Vectors

\[ a = (1 \ 2 \ \ldots \ n)^T, \ n = 10^4 \]

Relative errors \( \left| X - a^T a \right| / a^T a \) for every \( c \)

Chebyshev bound with probability .99
Strongly Graded Vectors

\[ a = (1 \ 2^{-1} \ \ldots \ 2^{-n+1})^T, \ n = 10^4 \]

Relative errors \(|X - a^T a| / a^T a\) for every \(c\)

Relative error \(\geq \sqrt{.6n - 1} / \sqrt{\delta c}\) grows with \(n\)
Non-Uniform Probabilities

Sample \( a_k \) with probability \( p_k = \frac{|a_k|}{\|a\|_1} \)

- For every \( \delta > 0 \) with probability at least \( 1 - \delta \)
  
  \[
  \frac{|X - a^T a|}{a^T a} < \epsilon
  \]

  where

  \[
  \epsilon \geq \frac{1}{\sqrt{c \delta}} \sqrt{\frac{\|a\|_1 \|a\|_3^3}{\|a\|_2^4}} - 1
  \]

  Smaller than relative error for uniform probabilities

- Weakly and strongly graded vectors
  
  Relative error \( \geq 0.3/\sqrt{\delta \cdot c} \) independent of \( n \)
Strongly Graded Vectors

\[ a = (1 \ 2^{-1} \ \ldots \ 2^{-n+1})^T, \ n = 10^4 \]

non uniform probabilities \( p_k = \frac{|a_k|}{\|a\|_1} \)

Relative errors \(|X - a^T a|/a^T a\) for every \( c \)

Chebyshev bound with probability .99
Relative Errors: Summary

- **Moderate dimensions**
  - For $n \leq 10^6$: relative error $\approx 10^{-2} - 10^{-1}$
  - Output of algorithm has 1-2 correct decimal digits

- **Larger dimensions**
  - For relative error of $10^{-8}$ need dimension $n \geq 10^{20}$

- **Uniformly distributed and weakly graded vectors**
  - Uniform probabilities suffice

- **Strongly graded vectors**
  - Need non-uniform probabilities

- **Probability bounds**
  - Hoeffding’s bound is tighter by only factor of 10 compared to Chebyshev bound
Repeated Sampling of Same Elements
Maximal Number of Times Same Element Is Sampled

\( n = 10^3, a_k \text{ iid uniform } [0, 1], \text{ uniform probabilities} \)

Repeated sampling increases with \( c \)
Elements that are Repeatedly Sampled

Expected value of \# distinct elements sampled \textit{more than once}

\[ n \left( 1 - \left( 1 - \frac{1}{n} \right)^{c-1} \left( 1 + \frac{c-1}{n} \right) \right) \approx n - (n + c)e^{-c/n} \approx 0.27n \quad \text{for } c = n \]
Elements that are Never Sampled

Expected value of \# elements never sampled

\[ n \left( 1 - \frac{1}{n} \right)^c \approx n e^{-c/n} \approx 0.37n \text{ for } c = n \]
Relative errors $|X - a^T a|/|a^T a|$ for every $c$

Relative errors still around $10^{-2} - 10^{-1}$
Repeated Sampling

**Uniform probabilities**
- Number of times an element can be sampled increases with $c$
- About 27% elements sampled *more than once*
- About 37% elements *never sampled*
- Repeated sampling does not seem to hurt accuracy

**Non-uniform probabilities**
- Preliminary conjecture: repeated sampling occurs at same rate as for uniform probabilities
“Stability” of Randomized Algorithm
What is Stability?

- **Stability of deterministic algorithms:**
  
  *How does a perturbation of the input change the output of the algorithm?*

- **Difficulty with randomized algorithms:**
  
  We don’t know the output with certainty

- **Exception:**
  
  Constant vector \( a_k = \alpha, \ 1 \leq k \leq n \)

  Uniform probabilities:

  \[
  X = \frac{n}{c} \alpha^2 + \cdots + \frac{n}{c} \alpha^2 = n \alpha^2 = a^T a
  \]

  Randomized algorithm gives **exact result** for any \( c \)
Stability of Randomized Algorithm

- Relative perturbations of constant vector
  \[ \tilde{a}_k = \alpha (1 + \epsilon \rho_k) \]
  \(0 < \epsilon \ll 1\), \(\rho_k\) are iid random variables

- Perturbed approximation
  \[ \tilde{X} = \frac{n}{c} (\tilde{a}_{k_1}^2 + \cdots + \tilde{a}_{k_c}^2) \]

- Algorithm is numerically stable if
  \[ \left| \frac{\tilde{X} - n\alpha^2}{n\alpha^2} \right| = \mathcal{O}(\epsilon) \]
\( \alpha = 1, \ n = 10^4 \)

Perturbations: \( \epsilon = 10^{-14}, \ \rho_k \text{ iid uniform } [0, 1] \)

Forward errors (\( \tilde{X} - n\alpha^2 \))\(/(n\alpha^2) \) for every \( c \)

Forward errors bounded by \( \epsilon \) \( \Rightarrow \) algorithm stable
Expected Value of Forward Error

- First and second moments
  \[ E_\rho [\rho_k] = \mu_1 \quad E_\rho [\rho_k^2] = \mu_2 \]

- Expected value of forward error
  \[ E_\rho \left[ \frac{\tilde{X} - n\alpha^2}{n\alpha^2} \right] = 2\epsilon \mu_1 + \epsilon^2 \mu_2 \]

- If perturbations \( \rho_k \) are iid uniform \([\beta_1, \beta_2]\) then
  \[ E_\rho \left[ \frac{\tilde{X} - n\alpha^2}{n\alpha^2} \right] = \epsilon (\beta_1 + \beta_2) + \frac{\epsilon^2}{3} (\beta_1^2 + \beta_1\beta_2 + \beta_2^2) \]

Expected value of forward error is \( O(\epsilon) \)
Perturbations $\rho_k$ are iid uniform $[\beta_1, \beta_2]$

Probability that

$$\left| \frac{\bar{X} - n\alpha^2}{n\alpha^2} - E_{\rho} \left[ \frac{\bar{X} - n\alpha^2}{n\alpha^2} \right] \right| < \tau$$

is at least

$$1 - 2 \exp\left( -\frac{-\tau^2 c}{2 \left( \epsilon (\beta_2 - \beta_1) + \epsilon^2 \max\{\beta_1^2, \beta_2^2\} \right)^2} \right)$$

Proof: Azuma’s inequality
Bound on Forward Error

- Perturbations $\rho_k$ are iid uniform $[\beta_1, \beta_2]$

$$\left| \frac{\hat{X} - n\alpha^2}{n\alpha^2} \right| < \epsilon \left( 1 + |\beta_1 + \beta_2| \right) + \frac{\epsilon^2}{3} \left| \beta_1^2 + \beta_1\beta_2 + \beta_2^2 \right|$$

holds with probability at least $1 - \delta$ for

$$c \geq 2 \ln \left( \frac{2}{\delta} \right) \left( (\beta_2 - \beta_1) + \epsilon \max\{\beta_1^2, \beta_2^2\} \right)^2$$

- Perturbations $\rho_k$ are iid uniform $[0, 1]$

$$\left| \frac{\hat{X} - n\alpha^2}{n\alpha^2} \right| < 3 \epsilon$$

holds with probability at least .99 for $c \geq 22$
Randomized algorithm for inner product $a^T a$
from [Drineas, Kannan & Mahoney 2006]

- Low relative accuracy
  1-2 correct decimal digits for dimensions $n \leq 10^6$

- Repeated sampling of elements occurs frequently but does not seem to hurt accuracy

- Preliminary definition of numerical stability
  
  \textit{Change in output when constant vector perturbed by iid random variables}

- Randomized algorithm is stable w.r.t. perturbations by iid uniform $[\beta_1, \beta_2]$ variables