CS 487 /...

Introduction to Symbolic Computation

University of Waterloo
Éric Schost
eschost@uwaterloo.ca
The exponent of linear algebra
Main idea

All problems of linear algebra are more or less equivalent.

More precisely

• the exponent of a problem $P$ (multiplication, inverse, ... ) is a number $\omega_P$ such that one can solve problem $P$ for matrices of size $n$ in time $O(n^{\omega_P})$.

• then

$$\omega_{\text{product}} = \omega_{\text{inverse}} = \omega_{\text{determinant}} = \cdots$$
Inverse $\implies$ multiplication

Suppose we want to multiply two matrices $A$ and $B$, but all that we have is an algorithm for inverse.

Define

$$D = \begin{bmatrix} I_n & A & 0 \\ 0 & I_n & B \\ 0 & 0 & I_n \end{bmatrix}$$

Then

$$D^{-1} = \begin{bmatrix} I_n & -A & AB \\ 0 & I_n & -B \\ 0 & 0 & I_n \end{bmatrix}$$

So product in size $n$ can be done using inverse in size $3n$, so in time

$$O\left( (3n)^{\omega_{\text{inverse}}} \right) = O\left( n^{\omega_{\text{inverse}}} \right).$$
Suppose we want to invert a matrix $A$ of size $n = 2^k$. We cut $A$ into blocks of size $m = n/2$:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}.$$ 

and do as if we invert a $2 \times 2$ matrix.

$$\begin{bmatrix} I_m & 0 \\ -A_{2,1}A_{1,1}^{-1} & I_m \end{bmatrix} A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & S \end{bmatrix}, \quad S = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2},$$

so

$$A^{-1} = \begin{bmatrix} A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}S^{-1} \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I_m & 0 \\ -A_{2,1}A_{1,1}^{-1} & I_m \end{bmatrix}.$$
Complexity:

\[ I(n) \leq 2I(n/2) + Cn^{\omega_{\text{product}}} \]

implies

\[ I(n) \leq C'n^{\omega_{\text{product}}} \]

**Proof:** some form of the master theorem.

**Remark 1:** we need our matrices to be “nice for this to work. \( A_{1,1} \) may be not invertible, even if \( A \) is.

**Remark 2:** this also gives the determinant.
Automatic differentiation
**Partial derivatives**

**Def:** if $F(X_1, \ldots, X_N)$ is a polynomial in $N$ variables, we define the partial derivatives

\[
\frac{\partial F}{\partial X_1}, \ldots, \frac{\partial F}{\partial X_N},
\]

where

\[
\frac{\partial F}{\partial X_i}
\]

is obtained by keeping all other $X_j$ constant, and differentiating in $X_i$.

**Example:** with

\[
F = X_1X_2 - X_3X_4,
\]

we get

\[
\frac{\partial F}{\partial X_1} = X_2, \quad \frac{\partial F}{\partial X_2} = X_1, \quad \frac{\partial F}{\partial X_3} = -X_4, \quad \frac{\partial F}{\partial X_4} = -X_3.
\]
Automatic differentiation

**Prop.**

- If $F$ can be computed using $L$ operations $+,-,\times$, then **all** partial derivatives

$$\frac{\partial F}{\partial X_1}, \ldots, \frac{\partial F}{\partial X_N},$$

can be computed using $4L$ operations.

- Independent of $N$.

**Remarks**

- widely used for optimization (using Newton’s iteration in several variables)

- some polynomials (such as $(X - 1)^k$) can be computed using few operations ($L = O(\log(k))$), even though they have a lot of monomials.
A naive solution

We are given a program $\Gamma$ with input variables $X_1, \ldots, X_N$.

**Example:**

\[
G_1 = X_1 - X_2 \\
G_2 = G_1^2 \\
G_3 = G_2X_3
\]

computes $(X_1 - X_2)^2X_3$, with $L = 3$. 
A naive solution

We are given a program $\Gamma$ with input variables $X_1, \ldots, X_N$.

**Example:**

$$
G_1 = X_1 - X_2 \\
G_2 = G_1^2 \\
G_3 = G_2X_3
$$

computes $(X_1 - X_2)^2X_3$, with $L = 3$.

We can follow line-by-line and apply the rules for differentiation. This is called the **direct mode**.

<table>
<thead>
<tr>
<th>$G_i$</th>
<th>$\partial G_i/\partial X_1$</th>
<th>$\partial G_i/\partial X_2$</th>
<th>$\partial G_i/\partial X_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1 = X_1 - X_2$</td>
<td>1</td>
<td>$-1$</td>
<td>0</td>
</tr>
<tr>
<td>$G_2 = G_1^2$</td>
<td>$2G_1 \partial G_1/\partial X_1$</td>
<td>$2G_1 \partial G_1/\partial X_2$</td>
<td>$2G_1 \partial G_1/\partial X_3$</td>
</tr>
<tr>
<td>$G_3 = X_3G_2$</td>
<td>$X_3 \partial G_2/\partial X_1$</td>
<td>$X_3 \partial G_2/\partial X_2$</td>
<td>$X_3 \partial G_2/\partial X_3 + G_2$</td>
</tr>
</tbody>
</table>

**Total:** $O(NL)$
The reverse mode

Setup.

- Let \( G_1, \ldots, G_L \) be the polynomials computed by \( \Gamma \).
- Let \( \Delta \) the program in variables \( X_1, \ldots, X_N, Y \) obtained by removing the first line of \( \Gamma \) and replacing \( G_1 \) by \( Y \). Let \( D_2, \ldots, D_L \) be the polynomials it computes.

**Example:** with \( \Gamma \) given by

\[
\begin{align*}
G_1 &= X_1 \times X_2 \\
G_2 &= G_1 + X_1 \\
G_3 &= G_1 \times G_2
\end{align*}
\]

\[
\begin{align*}
G_1 &= X_1X_2 \\
G_2 &= X_1X_2 + X_1 \\
G_3 &= X_1^2X_2^2 + X_1^2X_2
\end{align*}
\]

We get \( \Delta \) given by

\[
\begin{align*}
D_2 &= Y + X_1 \\
D_3 &= Y \times D_2
\end{align*}
\]

\[
\begin{align*}
D_2 &= Y + X_1 \\
D_3 &= Y^2 + YX_1
\end{align*}
\]
Prop. \( G_L = D_L(X_1, \ldots, X_N, G_1(X_1, \ldots, X_N)) \)
The reverse mode

**Prop.** \( G_L = D_L(X_1, \ldots, X_N, G_1(X_1, \ldots, X_N)) \)

**Corollary** For all \( i = 1, \ldots, N \),

\[
\frac{\partial G_L}{\partial X_i} = \frac{\partial D_L}{\partial X_i}(X_1, \ldots, X_N, G_1) + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1) \frac{\partial G_1}{\partial X_i}.
\]
The reverse mode

**Prop.** \( G_L = D_L(X_1, \ldots, X_N, G_1(X_1, \ldots, X_N)) \)

**Corollary** For all \( i = 1, \ldots, N \),

\[
\frac{\partial G_L}{\partial X_i} = \frac{\partial D_L}{\partial X_i}(X_1, \ldots, X_N, G_1) + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1) \frac{\partial G_1}{\partial X_i}.
\]

**Key remark.** \( G_1 \) has one of the following shapes

\( X_a + X_b, \ X_aX_b, \ \lambda X_a, \ \lambda + X_a, \ \lambda. \)

So for \( i \not\in \{a, b\}, \ \frac{\partial G_L}{\partial X_i} = \frac{\partial D_L}{\partial X_i}. \) For \( i \in \{a, b\}, \ \frac{\partial G_L}{\partial X_i} \) can be deduced from \( \frac{\partial D_L}{\partial X_i} \) and \( \frac{\partial D_L}{\partial Y} \) in at most 4 operations.

**Conclusion.** Suppose we know a program \( \Delta' \) that augments \( \Delta \) by computing all partial derivatives of \( D_L \) in \( X_1, \ldots, X_N, Y \). Then we can deduce a program \( \Gamma' \) of length \( \leq L(\Delta') + 4 \), that computes all partial derivatives of \( G_L \).
Corollary. Continuing inductively to remove the first lines, we finally obtain a program of length 1.

- The gradient of such a program is easy to compute.
- Then we can go backward to recover the gradient of $G_L$, adding a bounded number of operations (at most 4) at each step.

So the gradient of $G_L$ can be computed using $4L$ operations.
We detail the previous example. Removing the first instruction in $\Delta$ gives the program

$$
\Phi \ E_3 = Y \times Z \quad | \quad E_3(X_1, X_2, Y, Z) = YZ.
$$

Hence,

$$
\frac{\partial E_3}{\partial X_1} = \frac{\partial E_3}{\partial X_2} = 0, \quad \frac{\partial E_3}{\partial Y} = Z, \quad \frac{\partial E_3}{\partial Z} = Y
$$

So the program $\Phi'$ computes $E_3$ and its gradient:

$$
\Phi' \quad | \quad \begin{align*}
E_3 &= Y \times Z \\
E_{3,x_{12}} &= 0 \quad \text{(gives $\frac{\partial E_3}{\partial X_1}$ and $\frac{\partial E_3}{\partial X_2}$)} \\
E_{3,Y} &= Z \quad \text{(gives $\frac{\partial E_3}{\partial Y}$)} \\
E_{3,Z} &= Y \quad \text{(gives $\frac{\partial E_3}{\partial Z}$)}
\end{align*}
$$
Example

Recall that \( D_3(X_1, X_2, Y) = E_3(X_1, X_2, Y, Y + X_1) \), so

\[
\frac{\partial D_3}{\partial X_1, X_2, Y} = \frac{\partial E_3}{\partial X_1, X_2, Y}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1) \frac{\partial (Y + X_1)}{\partial X_1, X_2, Y}
\]
Recall that $D_3(X_1, X_2, Y) = E_3(X_1, X_2, Y, Y + X_1)$, so

$$\frac{\partial D_3}{\partial X_1, X_2, Y} = \frac{\partial E_3}{\partial X_1, X_2, Y}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1) \frac{\partial (Y + X_1)}{\partial X_1, X_2, Y}$$

and thus

$$\frac{\partial D_3}{\partial X_1} = \frac{\partial E_3}{\partial X_1}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1)$$

$$\frac{\partial D_3}{\partial X_2} = \frac{\partial E_3}{\partial X_2}(X_1, X_2, Y, Y + X_1)$$

$$\frac{\partial D_3}{\partial Y} = \frac{\partial E_3}{\partial Y}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1)$$
Example

Recall that \( D_3(X_1, X_2, Y) = E_3(X_1, X_2, Y, Y + X_1) \), so

\[
\frac{\partial D_3}{\partial X_1, X_2, Y} = \frac{\partial E_3}{\partial X_1, X_2, Y}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1) \frac{\partial (Y + X_1)}{\partial X_1, X_2, Y}
\]

and thus

\[
\frac{\partial D_3}{\partial X_1} = \frac{\partial E_3}{\partial X_1}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1) \\
\frac{\partial D_3}{\partial X_2} = \frac{\partial E_3}{\partial X_2}(X_1, X_2, Y, Y + X_1) \\
\frac{\partial D_3}{\partial Y} = \frac{\partial E_3}{\partial Y}(X_1, X_2, Y, Y + X_1) + \frac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1)
\]

yielding the program \( \Delta' \)

\[
\begin{align*}
D_2 &= Y + X_1 \\
D_3 &= Y \times D_2 \\
E_{3,X_{12}} &= 0 \\
E_{3,Y} &= D_2 \\
E_{3,Z} &= Y \\
D_{3,X_1} &= E_{3,X_{1,2}} + E_{3,Z} \\
D_{3,Y} &= E_{3,Y} + E_{3,Z}
\end{align*}
\]

(gives \( \frac{\partial D_3}{\partial X_2} \))

(gives \( \frac{\partial D_3}{\partial X_1} \))

(gives \( \frac{\partial D_3}{\partial Y} \)
Recall that $G_3(X_1, X_2) = E_3(X_1, X_2, X_1X_2)$, so

$$\frac{\partial G_3}{\partial X_1} = \frac{\partial D_3}{\partial X_1}(X_1, X_2, X_1X_2) + \frac{\partial D_3}{\partial Y}(X_1, X_2, X_1X_2) \frac{\partial X_1X_2}{\partial X_1}$$

$$= \frac{\partial D_3}{\partial X_1}(X_1, X_2, X_1X_2) + X_2 \frac{\partial D_3}{\partial Y}(X_1, X_2, X_1X_2)$$

$$\frac{\partial G_3}{\partial X_2} = \frac{\partial D_3}{\partial X_2}(X_1, X_2, X_1X_2) + \frac{\partial D_3}{\partial Y}(X_1, X_2, X_1X_2) \frac{\partial X_1X_2}{\partial X_2}$$

$$= \frac{\partial D_3}{\partial X_2}(X_1, X_2, X_1X_2) + X_1 \frac{\partial D_3}{\partial Y}(X_1, X_2, X_1X_2)$$
This finally yields

\[
\Gamma' = \begin{align*}
G_1 &= X_1 \times X_2 \\
G_2 &= G_1 + X_1 \\
G_3 &= G_1 \times G_2 \\
E_{3,X_{1,2}} &= 0 \\
E_{3,Y} &= G_2 \\
E_{3,Z} &= G_1 \\
D_{3,X_1} &= E_{3,X_{1,2}} + E_{3,Z} \\
D_{3,Y} &= E_{3,Y} + E_{3,Z} \\
tmp_1 &= D_{3,Y} \times X_2 \\
G_{3,X_1} &= D_{3,X_1} + tmp_1 \quad (\text{gives } \frac{\partial G_3}{\partial X_1}) \\
tmp_2 &= D_{3,Y} \times X_1 \\
G_{3,X_2} &= E_{3,X_{1,2}} + tmp_2 \quad (\text{gives } \frac{\partial G_3}{\partial X_2})
\end{align*}
\]
Back to matrix computations
Differentiating the determinant

Using automatic differentiation, an algorithm for the **determinant** gives an algorithm for **inverse**.

**Prop.** Let \( A = [a_{i,j}] \) be a matrix of size \( n \), whose entries are variables.

- The derivatives of the determinant of \( A \) w.r.t. \( a_{1,1}, \ldots, a_{n,n} \) are (almost) the entries of \( A^{-1} \).

**“Proof” (on an example):** \( n = 3 \). Take

\[
A = \begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} \\
a_{2,1} & a_{2,2} & a_{2,3} \\
a_{3,1} & a_{3,2} & a_{3,3}
\end{bmatrix}
\]

so

\[
\text{det}(A) = a_{1,1}a_{2,2}a_{3,3} - a_{1,1}a_{2,3}a_{3,2} + a_{2,1}a_{3,2}a_{1,3} \\
- a_{2,1}a_{1,2}a_{3,3} + a_{3,1}a_{1,2}a_{2,3} - a_{3,1}a_{2,2}a_{1,3}.
\]
Example with $n = 3$

Take the partial derivatives:

\[
\frac{\partial A}{\partial a_{1,1}} = a_{2,2}a_{3,3} - a_{2,3}a_{3,2}
\]
\[
\frac{\partial A}{\partial a_{1,2}} = a_{3,1}a_{2,3} - a_{1,2}a_{3,3}
\]
\[
\frac{\partial A}{\partial a_{1,3}} = a_{2,1}a_{3,2} - a_{3,1}a_{2,2}, \text{ etc} \ldots
\]

whereas the entries of $B = A^{-1}$ are

\[
b_{1,1} = \frac{a_{2,2}a_{3,3} - a_{2,3}a_{3,2}}{\det(A)}
\]
\[
b_{2,1} = \frac{a_{3,1}a_{2,3} - a_{1,2}a_{3,3}}{\det(A)}
\]
\[
b_{3,1} = \frac{a_{2,1}a_{3,2} - a_{3,1}a_{2,2}}{\det(A)}, \text{ etc} \ldots
\]
Suppose we have a program using $L$ additions / subtractions / multiplications that computes the determinant of $A$.

(No division because I don’t want to bother with the issues of division by zero)

Then we can turn it into a program that computes all entries of $A^{-1}$ using $O(L)$ additions / subtractions / multiplications, and 1 division (by the determinant).