CS 487 / $\cdots$
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 ω_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

$$O\left((3n)^{\omega_{\text{inverse}}}\right) = O\left(n^{\omega_{\text{inverse}}}\right).$$

Multiplication $\Longrightarrow$ inverse

Suppose we want to invert a matrix A of size $\boldsymbol{n = 2^k}$. We cut A into blocks of size $\boldsymbol{m = n/2}$:

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

and do as if we invert a $\mathbf{2 \times 2}$ 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

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

Multiplication $\implies$ inverse

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

$$F = X_1 X_2 - X_3 X_4,$$

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.

Not only in symbolic computation

An important advantage of the reverse mode is that it is significantly less costly to evaluate (in terms of operation count) than the forward mode for functions with a large number of inputs. In the extreme case of $f : \mathbb{R}^n \to \mathbb{R}$, only one application of the reverse mode is sufficient to compute the full gradient $\nabla f = \left(\frac{\partial y}{\partial x_1}, \ldots, \frac{\partial y}{\partial x_n}\right)$, compared with the n passes of the forward mode needed for populating the same. Because machine learning practice principally involves the gradient of a scalar-valued objective with respect to a large number of parameters, this establishes the reverse mode, as opposed to the forward mode, as the mainstay technique in the form of the backpropagation algorithm.

12. Also called *adjoint* or *cotangent linear* mode.

12

Journal of Machine Learning Research 18 (2018) 1-43

A naive solution

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

Example :

$$\begin{aligned} G_1 &= X_1 - X_2 \\ G_2 &= G_1^2 \\ G_3 &= G_2 X_3 \end{aligned}$$

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

A naive solution

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

Example :

$$\begin{aligned} G_1 &= X_1 - X_2 \\ G_2 &= G_1^2 \\ G_3 &= G_2 X_3 \end{aligned}$$

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

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

G_i	$\partial G_i / \partial X_1$	$\partial G_i / \partial X_2$	$\partial G_i / \partial X_3$
$G_1 = X_1 - X_2$	1	-1	0
$G_2 = G_1^2$	$2G_1 \partial G_1 / \partial X_1$	$2G_1 \partial G_1 / \partial X_2$	$2G_1 \partial G_1 / \partial X_3$
$G_3 = X_3 G_2$	$X_3 \partial G_2 / \partial X_1$	$X_3 \partial G_2 / \partial X_2$	$X_3 \partial G_2 / \partial X_3 + G_2$

Total: $O(NL)$

The reverse mode

Setup.

- Let $G_1, \ldots, G_L$ be the polynomials computed by Γ.
- Let Δ the program in variables $\boldsymbol{X_1, \ldots, X_N, Y}$ obtained by removing the first line of Γ and replacing G_1 by Y. Let $D_2, \ldots, D_L$ be the polynomials it computes.

Example: with Γ given by

$$\begin{array}{l|l} G_1 = X_1 \times X_2 & G_1 = X_1 X_2 \\ G_2 = G_1 + X_1 & G_2 = X_1 X_2 + X_1 \\ G_3 = G_1 \times G_2 & G_3 = X_1^2 X_2^2 + X_1^2 X_2 \end{array}$$

We get Δ given by

$$\begin{array}{l|l} D_2 = Y + X_1 & D_2 = Y + X_1 \\ D_3 = Y \times D_2 & D_3 = Y^2 + Y X_1 \end{array}$$

The reverse mode

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

$$X_a + X_b, \quad X_a X_b, \quad \lambda X_a, \quad \lambda + X_a.$$

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

$$X_a + X_b,\ \ X_a X_b,\ \ \lambda X_a,\ \ \lambda + X_a.$$

For $i \notin \{a, b\}$,

$$\frac{\partial G_L}{\partial X_i} = \frac{\partial D_L}{\partial X_i}.$$

The reverse mode

For $i = a$ (same for b)

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1) \quad \text{(first - fourth cases)}$$

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1)X_b \quad \text{(second case)}$$

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1)\lambda \quad \text{(third case)}$$

At most **2** new operations for $\frac{\partial G_L}{\partial X_a}$ and **2** new operations for $\frac{\partial G_L}{\partial X_b}$ (if there is a b).

The reverse mode

For $i = a$ (same for b)

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1) \quad \text{(first - fourth cases)}$$

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1)X_b \quad \text{(second case)}$$

$$\frac{\partial G_L}{\partial X_a} = \frac{\partial D_L}{\partial X_a} + \frac{\partial D_L}{\partial Y}(X_1, \ldots, X_N, G_1)\lambda \quad \text{(third case)}$$

At most **2** new operations for $\frac{\partial G_L}{\partial X_a}$ and **2** new operations for $\frac{\partial G_L}{\partial X_b}$ (if there is a b).

Conclusion. If we know a program Δ' that augments Δ by computing all partial derivatives of D_L in $X_1, \ldots, X_N, Y$, we can deduce a program Γ' of length $\leqslant L(\Delta') + 4$, that computes all partial derivatives of G_L.

Complexity

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.

Example

We detail the previous example. Removing the first instruction in Δ gives the program

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

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 Φ' computes E_3 and its gradient:

$$\Phi' \left| \begin{array}{ll} E_3 = Y \times Z & \\ E_{3,X_{12}} = 0 & \left(\text{gives} \frac{\partial E_3}{\partial X_1} \text{ and } \frac{\partial E_3}{\partial X_2}\right) \\ E_{3,Y} = Z & \left(\text{gives} \frac{\partial E_3}{\partial Y}\right) \\ E_{3,Z} = Y & \left(\text{gives} \frac{\partial E_3}{\partial Z}\right) \end{array} \right.$$

Example

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

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

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
$$\begin{aligned}
\tfrac{\partial D_3}{\partial X_1} &= \tfrac{\partial E_3}{\partial X_1}(X_1, X_2, Y, Y + X_1) + \tfrac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1) \\
\tfrac{\partial D_3}{\partial X_2} &= \tfrac{\partial E_3}{\partial X_2}(X_1, X_2, Y, Y + X_1) \\
\tfrac{\partial D_3}{\partial Y} &= \tfrac{\partial E_3}{\partial Y}(X_1, X_2, Y, Y + X_1) + \tfrac{\partial E_3}{\partial Z}(X_1, X_2, Y, Y + X_1)
\end{aligned}$$

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

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

yielding the program Δ'

$$\left|\begin{array}{ll}
D_2 = Y + X_1 & \\
D_3 = Y \times D_2 & \\
E_{3,X_{12}} = 0 & \left(\text{gives } \frac{\partial D_3}{\partial X_2}\right) \\
E_{3,Y} = D_2 & \\
E_{3,Z} = Y & \\
D_{3,X_1} = E_{3,X_{1,2}} + E_{3,Z} & \left(\text{gives } \frac{\partial D_3}{\partial X_1}\right) \\
D_{3,Y} = E_{3,Y} + E_{3,Z} & \left(\text{gives } \frac{\partial D_3}{\partial Y}\right)
\end{array}\right.$$

Example

Recall that $G_3(X_1, X_2) = E_3(X_1, X_2, X_1X_2)$, so

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

Example

This finally yields

$$\Gamma' \quad \left|\ \begin{array}{ll} G_1 = X_1 \times X_2 & \\ G_2 = G_1 + X_1 & \\ G_3 = G_1 \times G_2 & \\ E_{3,X_{1,2}} = 0 & \\ E_{3,Y} = G_2 & \\ E_{3,Z} = G_1 & \\ D_{3,X_1} = E_{3,X_{1,2}} + E_{3,Z} & \\ D_{3,Y} = E_{3,Y} + E_{3,Z} & \\ \mathrm{tmp}_1 = D_{3,Y} \times X_2 & \\ G_{3,X_1} = D_{3,X_1} + \mathrm{tmp}_1 & \left(\text{gives}\, \frac{\partial G_3}{\partial X_1}\right) \\ \mathrm{tmp}_2 = D_{3,Y} \times X_1 & \\ G_{3,X_2} = E_{3,X_{1,2}} + \mathrm{tmp}_2 & \left(\text{gives}\, \frac{\partial G_3}{\partial X_2}\right) \end{array}\right.$$

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

$$\begin{aligned} \det(A) &= a_{1,1}a_{2,2}a_{3,3} - a_{1,1}a_{2,3}a_{3,2} + a_{2,1}a_{3,2}a_{1,3} \\ &\quad -a_{2,1}a_{1,2}a_{3,3} + a_{3,1}a_{1,2}a_{2,3} - a_{3,1}a_{2,2}a_{1,3}. \end{aligned}$$

Example with $n = 3$

Take the partial derivatives:

$$\begin{aligned}
\frac{\partial A}{\partial a_{1,1}} &= a_{2,2}a_{3,3} - a_{2,3}a_{3,2} \\
\frac{\partial A}{\partial a_{1,2}} &= a_{3,1}a_{2,3} - a_{1,2}a_{3,3} \\
\frac{\partial A}{\partial a_{1,3}} &= a_{2,1}a_{3,2} - a_{3,1}a_{2,2}, \text{ etc} \ldots
\end{aligned}$$

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

$$\begin{aligned}
b_{1,1} &= \frac{a_{2,2}a_{3,3} - a_{2,3}a_{3,2}}{\det(A)} \\
b_{2,1} &= \frac{a_{3,1}a_{2,3} - a_{1,2}a_{3,3}}{\det(A)} \\
b_{3,1} &= \frac{a_{2,1}a_{3,2} - a_{3,1}a_{2,2}}{\det(A)}, \text{ etc} \ldots
\end{aligned}$$

Determinant $\implies$ inverse

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).