Collected matrix derivatives (AD for NLA)

Mike Giles

mike.giles@maths.ox.ac.uk

Oxford University Mathematical Institute

Oxford-Man Institute for Quantitative Finance

Outline

This is the topic of my paper in the conference proceedings.

- collection of mathematical results for forward and reverse mode AD for matrices
- highlights contribution by Dwyer & Macphail in 1948
- relevant for those using highly-tuned high-level software packages (e.g. LAPACK, MATLAB) for which it is inappropriate to apply black-box AD

Friday's talk is on opportunities and challenges for AD in computational finance.

Matrix Derivative

If $f(C)$ is a scalar output of a matrix input A, then define

$$\overline{C}_{ij} = \frac{\partial f}{\partial C_{ij}}$$

and so

$$\dot{f} = \sum_{ij} \overline{C}_{ij} \dot{C}_{ij} = \text{tr}\left(\overline{C}^T \dot{C}\right)$$

Note: for any A, B (with A and B^T of same dimensions),

$$\text{tr}(AB) \;=\; \sum_{ij} A_{ji} B_{ij} \;=\; \text{tr}(BA)$$

Key steps

If C is a function of matrices A, B,

$$C = g(A, B)$$

we use standard perturbation analysis to compute $\dot{C}$ as a function of $\dot{A}, \dot{B}$, and then use the identity

$$\text{tr}\left(\overline{C}^T \dot{C}\right) = \text{tr}\left(\overline{A}^T \dot{A} + \overline{B}^T \dot{B}\right), \quad \forall\, \overline{C}, \dot{A}, \dot{B}$$

to determine $\overline{A}, \overline{B}$ as a function of $\overline{C}$.

Once we have results for a range of elementary matrix operations, we can combine them in the usual way to construct forward or reverse mode derivatives for "programs" composed of these.

Matrix multiply

For example,

$$C = A\,B \implies \dot{C} = \dot{A}\,B + A\,\dot{B}$$

and so

$$\text{tr}\left(\overline{C}^T \dot{C}\right) = \text{tr}\left(\overline{C}^T \dot{A}\,B + \overline{C}^T A\,\dot{B}\right) = \text{tr}\left(B\,\overline{C}^T \dot{A} + \overline{C}^T A\,\dot{B}\right)$$

and hence

$$\begin{aligned} \overline{A} &= \overline{C}\,B^T \\ \overline{B} &= A^T\,\overline{C} \end{aligned}$$

Other basics

Addition:

$$C = A + B, \qquad \dot{C} = \dot{A} + \dot{B}, \qquad \overline{A} = \overline{C}, \quad \overline{B} = \overline{C}$$

Inverse:

$$C = A^{-1}, \qquad \dot{C} = -C\,\dot{A}\,C, \qquad \overline{A} = -C^T\,\overline{C}\,C^T$$

Determinant:

$$C = \mathrm{det}(A), \qquad \dot{C} = C\ \mathrm{tr}(A^{-1}\dot{A}), \qquad \overline{A} = \overline{C}\,C\,A^{-T}$$

Maximum Likelihood Estimation

We can build on these elementary results to tackle harder applications.

In Maximum Likelihood Estimation, if $p(x)$ is defined as

$$p(x) = \frac{1}{\sqrt{\det \Sigma}\ (2\pi)^{d/2}}\ \exp\Big(-\tfrac{1}{2}(x-\mu)^T\,\Sigma^{-1}(x-\mu)\Big)$$

then given a set of N data points x_n, their joint probability density function is

$$P = \prod_{n=1}^{N} p(x_n) \quad \Longrightarrow \quad \log P = \sum_{n=1}^{N} \log p(x_n)$$

Maximum Likelihood Estimation

The derivatives w.r.t. μ and Σ are

$$\frac{\partial \log P}{\partial \mu} = -\sum_{n=1}^{N} \Sigma^{-1}(x_n - \mu),$$

$$\frac{\partial \log P}{\partial \Sigma} = -\tfrac{1}{2} \sum_{n=1}^{N} \left\{ \Sigma^{-1} - \Sigma^{-1}(x_n - \mu)\,(x_n - \mu)^T \Sigma^{-1} \right\}.$$

and equating these to zero gives the maximum likelihood estimates

$$\mu = N^{-1} \sum_{n=1}^{N} x_n, \qquad \Sigma = N^{-1} \sum_{n=1}^{N} (x_n - \mu)\,(x_n - \mu)^T.$$

Dwyer and Macphail

This MLE result was derived by Dwyer in 1967, building on an earlier paper by Dwyer and Macphail in 1948 on “Symbolic matrix derivatives” in *The Annals of Mathematical Statistics*.

The statistics/econometrics community know and use these results, but aren’t apparently aware of AD and the fact that one can systematically apply these techniques to much larger problems.

Key reference: *Matrix differential calculus with applications in statistics and econometrics*, J. Magnus & H. Neudecker, John Wiley & Sons (1988)

Matrix Polynomial

Suppose

$$C \;=\; p(A) \;=\; \sum_{n=0}^{N} a_n A^n.$$

Pseudo-code for the evaluation of C is as follows:

$C := a_N I$
for n from $N\!-\!1$ to 0
    $C := A\,C + a_n I$
end

where I is the identity matrix.

Matrix Polynomial

The forward mode sensitivity is given by the pseudo-code:

```
Ċ := 0
C := a_N I
for n from N−1 to 0
    Ċ := Ȧ C + A Ċ
    C := A C + a_n I
end
```

Matrix Polynomial

Similarly, the reverse mode pseudo-code to compute $\overline{A}$ is:

$C_N := a_N I$

for n from $N\!-\!1$ to 0

$\quad C_n := A\, C_{n+1} + a_n I$

end

$\overline{A} := 0$

for n from 0 to $N\!-\!1$

$\quad \overline{A} := \overline{A} + \overline{C}\, C_{n+1}^T$

$\quad \overline{C} := A^T\, \overline{C}$

end

Matrix Exponential

In MATLAB, the matrix exponential

$$\exp(A) \equiv \sum_{n=0}^{\infty} \frac{1}{n!} A^n,$$

is approximated through a scaling and squaring method as

$$\exp(A) \approx \left(p_1(A)^{-1} p_2(A) \right)^m,$$

where m is a power of 2, and p_1 and p_2 are polynomials such that $p_2(x)/p_1(x)$ is a Padé approximation to $\exp(x/m)$

Forward and reverse mode derivatives are obtained by combining addition, multiplication, inverse and polynomial results.

Eigenvalues/eigenvectors

An expanded technical report treats the eigenvalue/eigenvector problem.

Why is this important? In engineering, sometimes want to ensure that natural vibration frequencies are well away from forcing frequencies to minimise vibration.

Given a square matrix A with distinct eigenvalues, the eigenvector matrix U and diagonal eigenvalue matrix D satisfy

$$A\,U = U\,D$$

with the ordering of the eigenvalues and the scaling of the eigenvectors undefined.

Eigenvalues/eigenvectors

Defining the Hadamard product $A \circ B$ to be an element-wise product (i.e. $(A \circ B)_{ij} = A_{ij} B_{ij}$), one can prove that for a certain choice of eigenvector normalisation

$$\begin{aligned} \dot{D} &= I \circ (U^{-1} \dot{A}\, U), \\ \dot{U} &= U \left(F \circ (U^{-1} \dot{A}\, U) \right). \end{aligned}$$

where $F_{ij} = (d_j - d_i)^{-1}$ for $i \neq j$, and zero otherwise.

In reverse mode, we get

$$\overline{A} = U^{-T} \left(\overline{D} + F \circ (U^T\, \overline{U}) \right) U^T.$$

Other results

Other results in the expanded technical report:

- singular value decomposition `(svd(A))`
- Choleksy factorisation `(chol(A))`
- Frobenius and spectral norms `(norm(A))`
- a MATLAB code uses "the complex variable trick" (a form of operator overloading) to verify the forward mode sensitivities, and the identity

$$\mathrm{tr}\left(\overline{C}^T \dot{C}\right) = \mathrm{tr}\left(\overline{A}^T \dot{A} + \overline{B}^T \dot{B}\right), \quad \forall\, \overline{C}, \dot{A}, \dot{B}$$

  to check the reverse mode sensitivities

Conclusions

- Very few novel results, but hopefully the collection will be a useful reference
- Probably most relevant to those using high-level packages (e.g. LAPACK, MATLAB)
- Should give Dwyer & Macphail due credit for their 1948 paper

Acknowledgements: Andreas Griewank, Shaun Forth and Nick Trefethen for key references

Further information

M.B. Giles, “An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation”, Oxford University Computing Laboratory Numerical Analysis report 08/01.

- `people.maths.ox.ac.uk/~gilesm/`
- Email: `mike.giles@maths.ox.ac.uk`