Differentiation of functions of covariance matrices

or: Why you can forget you ever read this

Richard Turner

Covariance matrices are symmetric, but we often conveniently forget this when we differentiate functions of them. When is this amnesia useful and when is it problematic?

1 Differentiation of $\log|X|$

Let's take differentiation of $\log|X|$ as an example, where X is a 2×2 matrix and it can be non-symmetric (B) or symmetric (C).

$$B = \begin{pmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \end{pmatrix} \tag{1}$$

$$C = \begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix} \tag{2}$$

The inverses are thus:

$$B^{-1} = \frac{1}{x_{11}x_{22} - x_{12}x_{21}} \begin{pmatrix} x_{22} & -x_{21} \\ -x_{12} & x_{11} \end{pmatrix} \tag{3}$$

$$C^{-1} = \frac{1}{x_{11}x_{22} - x_{12}^2} \begin{pmatrix} x_{22} & -x_{12} \\ -x_{12} & x_{11} \end{pmatrix} \tag{4}$$

And the derivatives:

$$\frac{d\log|B|}{dB} = \frac{d\log(x_{11}x_{22} - x_{12}x_{21})}{dB} \tag{5}$$

$$= \frac{1}{x_{11}x_{22} - x_{12}x_{21}} \begin{pmatrix} x_{22} & -x_{12} \\ -x_{21} & x_{11} \end{pmatrix} \tag{6}$$

$$= B^{-T} \tag{7}$$

$$\frac{d\log|C|}{dC} = \frac{d\log(x_{11}x_{22} - x_{12}^2)}{dC} \tag{8}$$

$$= \frac{1}{x_{11}x_{22} - x_{12}^2} \begin{pmatrix} x_{22} & -2x_{12} \\ -2x_{12} & x_{11} \end{pmatrix} \tag{9}$$

$$= 2C^{-1} - I \circ C^{-1} \tag{10}$$

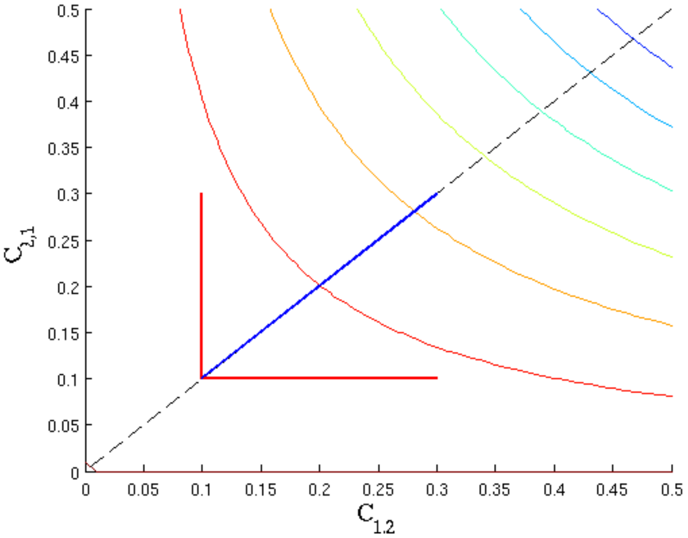

Figure 1: The contours of the log $|C|$ as a function of C_{12} and C_{21}. The manifold (line/hypersurface) upon which covariance matrices must lie is shown by a dashed black line. The direction of the partial derivatives (given by eqn. 5) are in red. The derivative along the symmetric direction (given by eqn. 6) is shown by the solid blue line. As the function is invariant under $C \rightarrow C^T$ the derivative of a symmetric matrix will itself be a symmetric matrix.

Where the Hadamard or entry-wise product is defined as:

$$(X \circ Y)_{ij} = x_{ij} y_{ij} \tag{11}$$

These results can be understood geometrically (see Fig. 1).

Equations (7) and (8) are actually correct for general B and C. A simple way for deriving this, and other derivatives with respect to constrained matrices, uses the chain rule, where the important quantity is the derivative of a matrix with respect to itself (a fourth order tensor):

$$\frac{df}{dA} = \sum_{i,j} \frac{df}{dA_{ij}} \frac{dA_{ij}}{dA} \tag{12}$$

2 When can you forget about this result?

Why can you sometimes forget about (6) and when do you really need to understand what's going on? Well, it depends upon what you are using the derivatives for.

2.1 Stationary points

Strictly you should **not** just use (5) to find the stationary points of functions of the determinant of a covariance matrix - a Lagrange multiplier is required to ensure you obey the symmetry constraint. (6) incorporates this constraint automatically, can be used directly, and is therefore favourable. Let's look at a common example:

To find the maximum likelihood covariance of a multivariate Gaussian we have to solve:

$$argmin \log |C| + Tr[C^{-1}\langle \mathbf{x}\mathbf{x}^T\rangle] \tag{13}$$

where C is symmetric, and we should use the above result (6), combined with:

$$\frac{dTr[C^{-1}\langle \mathbf{x}\mathbf{x}^T\rangle]}{dC} = -2C^{-1}\langle \mathbf{x}\mathbf{x}^T\rangle C^{-1} + I \circ (C^{-1}\langle \mathbf{x}\mathbf{x}^T\rangle C^{-1}) \tag{14}$$

to locate the stationary point. However, most derivations seem to forget this and come up with the correct answer. This begs the question: "Why does the wrong method work for finding the ML covariance of a multivariate Gaussian?". This seems a particularly strange paradox as the covariance matrix is just one particular choice of parameterisation. The quadratic form in the cost function above: $\sum_{ij}(C^{-1})_{ij}\langle x_i x_j\rangle$ is invariant along the line $(C^{-1})_{ij} + (C^{-1})_{ij} = \alpha$, where α is a constant. Therefore the family of matrices with identical diagonal elements, and off diagonal elements which sum to the same value, specify equivalent quadratic forms. As such the minima of our cost function should lie on a line in C_{ij} space. We expect the naïve approach of forgetting about the constraint to return this line (Fig. 2). We should then have to use the symmetric constraint, corresponding to our particular parameterisation of the covariance, to pick out the point on the line which we desire. Why does this happen automatically?

FIX: I should really describe all this by differentiating wrt C^{-1} as that is what is discussed in the next section.

The resolution of this paradox comes from considering the normalising constant of the multivariate Gaussian: $|C|$. This is correct for symmetric C, but if C takes some other form it should be replaced by: $|(\frac{1}{2}(C^{-1}+C^{-T}))^{-1}|$. The later is invariant along the line $(C^{-1})_{ij} + (C^{-1})_{ji} = \alpha$ but the former is not. In fact:

$$|C| \geq |[\frac{1}{2}(C^{-1} + C^{-T})]^{-1}| \tag{15}$$

With equality when C is symmetric [as can be verified by writing down the determinants and using $C^{-1}_{ij}C^{-1}_{ji} = C^{-1}_{ij}(C^{-1}_{ij} - \alpha)$ is minimised when $C^{-1}_{ij} = \frac{1}{2}\alpha$]. Therefore, as Fig. 3 shows, using the wrong expression for the determinant *boosts* the likelihood of non-symmetric C^{-1}. This means that

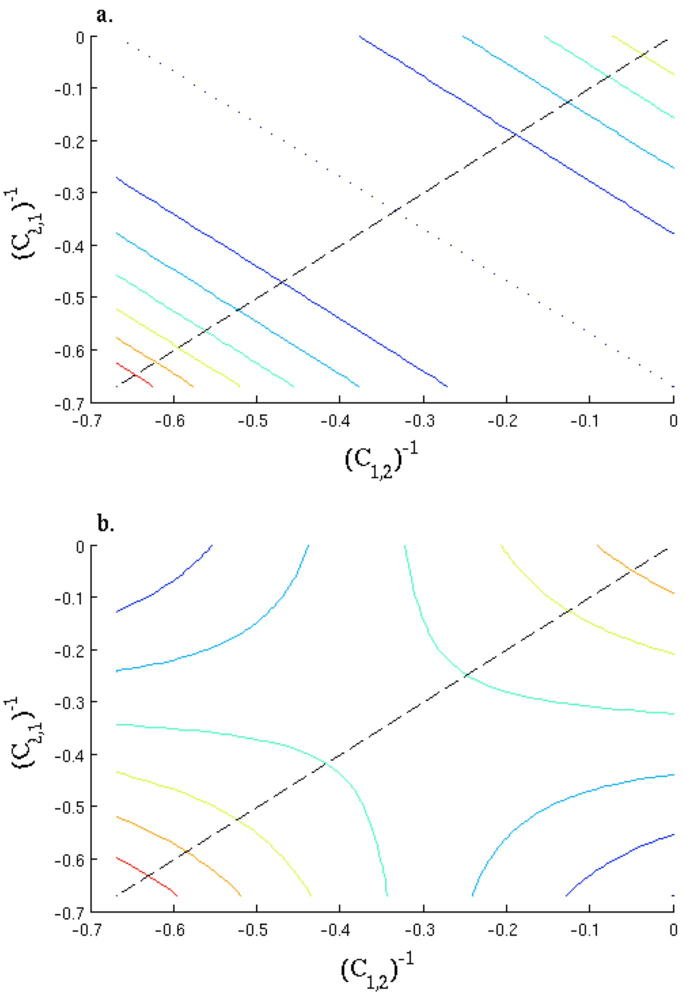

Figure 2: a. What we expected the contours of the cost function to look like as a function of $(C_{12})^{-1}$ and $(C_{21})^{-1}$. b. What it actually looks like. Note the saddle point.

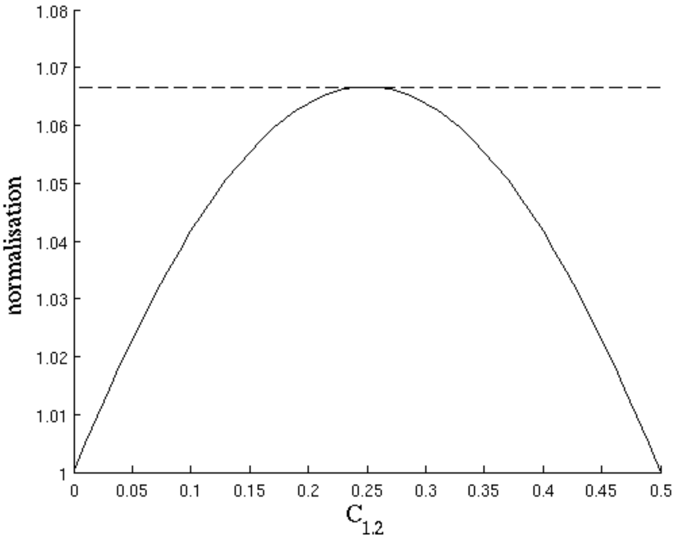

Figure 3: The contribution from $\log|C|$ (solid line) and $\log|[\frac{1}{2}(C^{-1} + C^{-T})]^{-1}|$ (dashed line) along the line $C^{-1}_{ij} + C^{-1}_{ji} = \alpha$. The later is constant. The former is the reciprocal of a quadratic (plus a constant) with a maximum at the symmetric point, and there it is equal to the correct formulation.

the solution we desire is at a *saddle point* and this is the unique stationary point of the cost function (as shown in Fig. 2).

In this particular case two wrongs made a right: we incorrectly used the unconstrained derivatives and an 'incorrect' normalising constant and everything worked out fine. In general, however, this is not a fail-safe method and the constrained derivatives should be used.

2.2 Differentials: Changes in height

Another use for derivatives is to calculate the differential of a function using the relation:

$$df = \sum_i \frac{df}{dx_i} dx_i \tag{16}$$

Where the sum is over the variables upon which f depends. Equations (5) and (6) can both be used *with care* for calculating differentials of $\log|C|$ where C is a symmetric matrix. Taking the two dimensional case as an example, in the first case we sum over the 4 variables, but constrain both A

and therefore dA to be symmetric.

$$df = \sum_{i=1}^{4} \frac{\log|B|}{dx_i} dx_i \tag{17}$$

with the constraints $x_{12} = x_{21}$ and $dx_{12} = dx_{21}$

In the second case, the matrix representation can be misleading as we only have *three* independent variables (the constraint has already been satisfied):

$$df = \sum_{i=1}^{3} \frac{\log|C|}{dx_i} dx_i \tag{18}$$

So, in this particular case, we had to understand what was going on.

2.3 Gradient Descent

The gradient descent learning rule updates parameters according to:

$$\theta^{(n+1)} = \theta^{(n)} - \eta \frac{dL(\theta^{(n)})}{d\theta^{(n)}} \tag{19}$$

If θ corresponds to a covariance matrix then we should use the constrained derivatives to compute this.

Fortunately, due to the symmetry of the function, if A is symmetric, then so is dA (see eqn. 5) and therefore $\frac{dL(\theta)}{d\theta}$ will be too. So we should not wander off the manifold of symmetric matrices using (5). As long as we kick off with a symmetric initialisation, we'll be fine. However, any peturbation (for example, due to numerical error) will push us off the manifold and then we will continue to diverge from it due to boosting of non-symmetric C (for example, we will not stay at the saddle point if we get perturbed). This instability can be corrected by some heuristic which ensures or steps are always in the symmetric direction. However, generally speaking, if we happen to be working with a cost function which is not invariant under $B \rightarrow B^T$, we need to use the constrained derivatives.

3 Parameterisation of covariance matrices

Much of the mire we have stumbled through here stems from the fact that a covariance matrix is over parameterised (N^2 elements for only $\frac{1}{2}N(N-1)$ independent parameters). Through this short document we have already hinted at three separate ways to parameterise the covariance:

$$\begin{aligned} C_{ij} = & \quad C_{ji} & \forall i \neq j \quad (20) \\ C_{ij} = & \quad 0 & \forall i < j \quad (21) \\ C = & \ \tfrac{1}{2}[A + A^T] & \quad (22) \end{aligned}$$

Where the first is the usual symmetric covariance matrix and one of the benefits of this parameterisation is that it is easy to transform into a rotated (or stretched) coordinate system. The second is an upper (or lower) triangular matrix and this has the advantage of not being over-parameterised. This is the parameterisation implicitly used in (6). The third uses a completely general matrix to specify a symmetric covariance. Which of these approaches is the most natural though?

In the two dimensional case two 'natural' parameterisations are:

$$C = c \begin{pmatrix} \sqrt{1 + a^2 + b^2} + a & b \\ b & \sqrt{1 + a^2 + b^2} - a \end{pmatrix} \quad (23)$$

Where a controls the oblateness of the contours and b the correlation between x_1 and x_2. Is there a natural generalisation of this into N dimensions though?

$$C = R^T \Sigma R \quad (24)$$

Where $R^T R = I$ and $\Sigma = diag(\sigma_1^2, \sigma_2^2)$. This is a 'PCA' like specification.

References

[1] Kaare Petersen and Michael Pedersen, "The matrix cook book" (2005)

[2] Tom Minka, "Old and new matrix algebra useful for statistics" (1997)