Gaussian graphical model inversion

Oliver K. Ernst

December 18, 2020

1 Problem statement

Find B given a mix of constraints on B and $\Sigma = B^{-1}$ as follows:

$$\begin{aligned} B_{ij} &= 0 \text{ from graphical model} \\ \Sigma_{kl} &= (B^{-1})_{kl} = \text{given numerically} \end{aligned} \tag{1}$$

Note that there are as many equations as unknowns.

2 Approaches

1. Solve analytically for small matrices.
2. Minimize the L_2 loss:
$$L_2(\{B_{ij}\}|\{\sigma_{kl}\}) = \sum_{kl} \left[\sigma_{kl} - (B^{-1})_{kl}\right]^2 \tag{2}$$
   where $\sigma_{kl} = ((B^*)^{-1})_{kl}$ are the given numerical values.

   In this case, we are learning only the unknown elements of B.
3. Non-linear root finding with Newtons method of:
$$\boldsymbol{F} = \text{upperTriangleToVector}(B\Sigma - I) = \mathbf{0} \tag{3}$$
   where upperTriangleToVector constructs a vector from the upper triangle of the matrix, since the matrices are symmetric.

   In this case, we are learning both the unknown elements of B and the elements in Σ.

If an initial guess sufficiently close to the inverse is available, then the third method is preferred. Minimizing the L_2 loss is slower but more robust if such a guess is not available.

3 Root finding with Newton's method

Apply Newton's root finding method to the matrix equation:

$$\boldsymbol{F} = \text{vec}(B\Sigma - I) = 0 \tag{4}$$

The parameters are split between elements of B and elements of Σ. Define:

$$\begin{aligned} \boldsymbol{b} &= \text{vecOfLearnableParams}(B) \\ \boldsymbol{\sigma} &= \text{vecOfLearnableParams}(\Sigma) \\ \boldsymbol{x} &= \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{\sigma} \end{pmatrix} \end{aligned} \tag{5}$$

where $\boldsymbol{x}$ are the params to be learned. Here vecOfLearnableParams extracts the learnable parameters, i.e. those **not** given in (1).

Newton's method has updates $\boldsymbol{h}_x$ of the form:

$$\begin{aligned} F_x \boldsymbol{h}_x &= -\boldsymbol{F} \\ \boldsymbol{x} &\to \boldsymbol{x} + \boldsymbol{h}_x \end{aligned} \tag{6}$$

where F_x denotes the Jacobian. The gradients are simply:

$$\begin{aligned} \frac{\partial}{\partial B_{ij}}(B\Sigma) &= I_{ij}\Sigma \\ \frac{\partial}{\partial \Sigma_{ij}}(B\Sigma) &= BI_{ij} \end{aligned} \tag{7}$$

where I_{ij} is all zeros except 1 at (i,j) **and** at (j,i).

4 Minimize the L_2 loss

Minimize the L_2 loss:

$$L_2(\{B_{ij}\}|\{\sigma_{kl}\}) = \sum_{kl} \left[\sigma_{kl} - (B^{-1})_{kl}\right]^2 \tag{8}$$

where $\sigma_{kl} = ((B^*)^{-1})_{kl}$ are the given numerical values.

In this case, we are learning only the unknown elements of B.

The first order gradients are:

$$\frac{\partial L_2}{\partial B_{ij}} = -2\sum_{kl} \left[\sigma_{kl} - (B^{-1})_{kl}\right] \frac{\partial (B^{-1})_{kl}}{\partial B_{ij}} \tag{9}$$

$$\begin{aligned} \frac{\partial (B^{-1})_{kl}}{\partial B_{ij}} &= -\left(B^{-1}\frac{\partial B}{\partial B_{ij}}B^{-1}\right)_{kl} \\ &= -\left(B^{-1}I_{ij}B^{-1}\right)_{kl} \\ &= -(B^{-1})_{ki}(B^{-1})_{jl} - (1-\delta_{ij})(B^{-1})_{kj}(B^{-1})_{il} \end{aligned} \tag{10}$$

5 Alternative approaches

1. Non-linear root finding with Newtons method of:

$$\boldsymbol{F} = \text{toVec}(B^{-1} - \Sigma) = \begin{pmatrix} (B^{-1})_{kl} - \sigma_{kl} \\ \dots \end{pmatrix} = \mathbf{0} \tag{11}$$

   The gradients can be computed as before.

2. Non-linear root finding with Newtons method of:

$$\boldsymbol{F} = \text{toVec}((B\Sigma)^{-1} - I) = \mathbf{0} \tag{12}$$

   This is somewhat popular for generalized inverses, see e.g. "On the Computation of a Matrix Inverse Square Root" by N. Sherif in Computing 46, 295-305 (1989).