Preconditioning of Nonlinear Mixed Effect models for Stabilization of the Covariance Matrix Computation

Yasunori Aoki, Rikard Nordgren and Andrew C. Hooker

Department of Pharmaceutical Biosciences
Uppsala University
Sweden
## Motivating example

<table>
<thead>
<tr>
<th>Estimation in NONMEM 7.3</th>
<th>OFV -2 ln (likelihood)</th>
<th>Estimated Parameter Value (V1)</th>
<th>Estimated SE (V1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux Cluster</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel Xeon E5645</td>
<td>-2,346.78706</td>
<td>7.913632283</td>
<td>1.2618562800</td>
</tr>
<tr>
<td>GCC 4.4.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MacBook Pro</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel Core i7</td>
<td>-2,346.78706</td>
<td>7.913632354</td>
<td>1.7525223004</td>
</tr>
<tr>
<td>GCC 4.9.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Difference (%)</strong></td>
<td>0.0000000001%</td>
<td>0.0000008939%</td>
<td><strong>32.6%</strong></td>
</tr>
</tbody>
</table>

# Motivating example

<table>
<thead>
<tr>
<th>Estimation in NONMEM 7.3</th>
<th>OFV -2 ln (likelihood)</th>
<th>Estimated Parameter Value (V1)</th>
<th>Estimated SE (V1)</th>
<th>Estimated SE with preconditioning (V1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux Cluster</strong></td>
<td>-2,346.78706</td>
<td>7.913632283</td>
<td>1.2618562800</td>
<td>1.38730205594</td>
</tr>
<tr>
<td>Intel Xeon E5645</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCC 4.4.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MacBook Pro</strong></td>
<td>-2,346.78706</td>
<td>7.913632354</td>
<td>1.7525223004</td>
<td>1.38466752752</td>
</tr>
<tr>
<td>Intel Core i7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCC 4.9.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Difference (%)</strong></td>
<td>0.0000000001%</td>
<td>0.0000008939%</td>
<td><strong>32.6%</strong></td>
<td><strong>0.190%</strong></td>
</tr>
</tbody>
</table>

Why does this happen?

Sandwich estimator of the Covariance matrix of parameter estimates (M)

\[ M = R^{-1} S R^{-1} \]

R: Hessian of the -2 ln likelihood
S: Sum of the cross products of the gradient vectors of the -2 ln individual-likelihood

**Reminder**: computation on a computer is not exact
- For example: machine epsilon gives an upper bound on the relative rounding error in floating point arithmetic (double precision ~ $10^{-15}$)
Why does this happen?

Sandwich Estimator of the Covariance matrix

\[ M = R^{-1}SR^{-1} \]

Computational error from rounding error

\[ \Rightarrow \] Condition number of R matrix

\[ \times \] Small perturbation in S and R matrix

\[ \times \] Condition number of R matrix
Why does this happen?

Sandwich Estimator of the Covariance matrix

\[ M = R^{-1}SR^{-1} \]

Computational error from rounding error

\[ 4.623879 \times 10^6 \times \]

Small perturbation in S and R matrix

\[ 4.623879 \times 10^6 \]
Why does this happen?

Sandwich Estimator of the Covariance matrix

\[ M = R^{-1}SR^{-1} \]

Computational error from rounding error

\[ 4.623879 \times 10^6 \times 10^{-15} \times 4.623879 \times 10^6 \]
Why does this happen?

Sandwich Estimator of the Covariance matrix

\[ M = R^{-1}SR^{-1} \]

\[ \sqrt{10^{-3}} = 10^{-1.5} \]
How does preconditioning work?

Linearly re-parameterize the model by matrix $P$

$$\theta = P\hat{\theta}$$

to reduce the condition number of the $\hat{R}$ matrix

$$M = P\hat{R}^{-1}\hat{S}\hat{R}^{-1}P^T = P\hat{M}P^T$$

Computational error from rounding error

$\approx$ Condition number of $\hat{R}$ matrix 

$x$

Small perturbation in $P$, $\hat{S}$ and $\hat{R}$ matrix

$x$

Condition number of $\hat{R}$ matrix
Choosing P

Assuming we are at the maximum likelihood with \( \theta \) then we can obtain an eigen decomposition of \( R \) such that:

\[
R = V \Lambda V^T
\]

\( V \) : Normalized eigenvectors
\( \Lambda \) : Eigenvalues

then the \( \hat{R} \) matrix condition number will be one if:

\[
P = V \Lambda^{-1/2}
\]
How does preconditioning work?

Linearly re-parameterize the model by matrix $P$

$$\theta = P\theta$$

to reduce the condition number of the $\hat{R}$ matrix

$$M = P\hat{R}^{-1}\hat{S}\hat{R}^{-1}P^T = P\hat{M}P^T$$

Computational error from rounding error

$\approx 1.125299 \times 10^0$

x

Small perturbation in $P$, $\hat{S}$ and $\hat{R}$ matrix

x

$1.125299 \times 10^0$
## Motivating example

<table>
<thead>
<tr>
<th>Estimation in NONMEM 7.3</th>
<th>OFV -2 ln (likelihood)</th>
<th>Estimated Parameter Value (V1)</th>
<th>Estimated SE (V1)</th>
<th>Estimated SE with preconditioning (V1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linux Cluster</strong></td>
<td>-2,346.78706</td>
<td>7.913632283</td>
<td>1.2618562800</td>
<td>1.38730205594</td>
</tr>
<tr>
<td>Intel Xeon E5645</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCC 4.4.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MacBook Pro</strong></td>
<td>-2,346.78706</td>
<td>7.913632354</td>
<td>1.7525223004</td>
<td>1.38466752752</td>
</tr>
<tr>
<td>Intel Core i7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCC 4.9.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Difference (%)</strong></td>
<td>0.0000000001%</td>
<td>0.0000008939%</td>
<td>32.6%</td>
<td>0.190%</td>
</tr>
</tbody>
</table>

Preconditioning can be done using the `precond` command available in PsN 4.4
precond model1.mod
precond model1.mod
$R = V \Lambda V^T$

$P = V \Lambda^{-1/2}$

$\theta = P\hat{\theta}$
\[
R = V \Lambda V^T \\
\hat{P} = V \Lambda^{-1/2}
\]

\[\theta = P \hat{\theta}\]

precond model1.mod

$PK$

IF (NEWIND == 0) THEN

THE_1 = \(p_{11} \cdot \text{THETA}(1) + p_{12} \cdot \text{THETA}(2)\)

THE_2 = \(p_{21} \cdot \text{THETA}(1) + p_{22} \cdot \text{THETA}(2)\)

END IF

model1_repara.mod

\[\hat{\theta}\]

model1_repara.lst

\[\hat{R}^{-1} \hat{S} \hat{R}^{-1}\]

model1_repara.cov
$\theta = P\hat{\theta}$

$R = V\Lambda V^T$

$P = V\Lambda^{-1/2}$

$\hat{\theta} = \theta$

$M = P\hat{R}^{-1}\hat{S}\hat{R}^{-1}P^T$

model1.mod

model1.rmt

$\text{precond model1.mod}$

$\text{model1_repara.mod}$

$\text{model1_repara.lst}$

$\text{model1_repara.cov}$

raw_results_model1.csv

model1.cov

$\$PK$

IF (NEWIND == 0) THEN
THE_1 = p_{11} * THETA(1) + p_{12} * THETA(2)
THE_2 = p_{21} * THETA(1) + p_{22} * THETA(2)
END IF
Limitations with the ‘precond’ tool in PsN

• Neglects constraints on the parameter search space (i.e., boundaries of the fixed-effect parameters set in $\text{THETA}$ record).
  – Use “abs(\text{THETA}(X))” to account for non-negative parameter spaces.

• Only preconditions on the fixed effect portion of the model.
  – Re-parameterization of the model so that the standard deviation of random effects can be estimated with a fixed effect.

• Cannot precondition mixture models
Preconditioning can

• Reduce computational environment dependencies

• Recover failed covariance matrix computations

• Aid in revealing model parameter non-identifiability
Numerical Experiment 1
Reduce computational environment dependencies
Initial parameter estimates set to final parameter estimates from previous analysis
Initial parameter estimates set to final parameter estimates from previous analysis

Linux Cluster

MacBook Pro
Linux Cluster

Model1.mod  data.csv

MacBook Pro

Model1.mod  data.csv

Model1.lst  Compare  Model1.lst
Original model results

<table>
<thead>
<tr>
<th>Model</th>
<th>R Condition Number</th>
<th>Difference in OFV</th>
<th>Ave. Difference of SE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1 [1]</td>
<td>$4.623879 \times 10^6$</td>
<td>$0.000000028020$</td>
<td>21.2%</td>
</tr>
<tr>
<td>Model2 [2]</td>
<td>$3.674548 \times 10^{11}$</td>
<td>$0.000000203632$</td>
<td>5.47%</td>
</tr>
<tr>
<td>Model3 [3]</td>
<td>$4.795944 \times 10^6$</td>
<td>$0.000000858280$</td>
<td>11.2%</td>
</tr>
</tbody>
</table>

Preconditioned model results

<table>
<thead>
<tr>
<th>Model</th>
<th>R Condition Number</th>
<th>Difference in OFV</th>
<th>Ave. Difference of SE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1 [1]</td>
<td>4.623879×10^6</td>
<td>0.0000000028020</td>
<td>21.2%</td>
</tr>
<tr>
<td>preconditioned model1</td>
<td>1.125299×10^0</td>
<td>0.000000140300</td>
<td>0.101%</td>
</tr>
<tr>
<td>Model2 [2]</td>
<td>3.674548×10^{11}</td>
<td>0.000000203632</td>
<td>5.47%</td>
</tr>
<tr>
<td>preconditioned model2</td>
<td>2.630800×10^2</td>
<td>0.000000071808</td>
<td>0.504%</td>
</tr>
<tr>
<td>Model3 [3]</td>
<td>4.795944×10^6</td>
<td>0.000000858280</td>
<td>11.2%</td>
</tr>
<tr>
<td>preconditioned model3</td>
<td>1.533025×10^0</td>
<td>0.000000640840</td>
<td>0.297%</td>
</tr>
</tbody>
</table>

Numerical Experiment 2
Recover failed variance-covariance matrix computations
model1.mod  data.csv
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Original Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>100/100</td>
</tr>
<tr>
<td>Model 2</td>
<td>68/100</td>
</tr>
<tr>
<td>Model 3</td>
<td>62/100</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Original Model</th>
<th>Preconditioned Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>100/100</td>
<td>100/100</td>
</tr>
<tr>
<td>Model 2</td>
<td>68/100</td>
<td>100/100</td>
</tr>
<tr>
<td>Model 3</td>
<td>62/100</td>
<td>98/100* **</td>
</tr>
</tbody>
</table>

*Estimation of the original model failed
**S-matrix not obtained for original or preconditioned model
Numerical Experiment 3
Aid in revealing model parameter non-identifiability
## Original model results [4]

<table>
<thead>
<tr>
<th></th>
<th>OFV</th>
<th>CL</th>
<th>V1</th>
<th>Q2</th>
<th>V2</th>
<th>Q3</th>
<th>V3</th>
<th>RUV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original model</td>
<td>186.145</td>
<td>0.36</td>
<td>3.44</td>
<td>1.94</td>
<td>0.69</td>
<td>1.22</td>
<td>3.25</td>
<td>0.045</td>
</tr>
</tbody>
</table>

---

**No covariance step:**

"R MATRIX ALGORITHMICALLY SINGULAR AND ALGORITHMICALLY NON-POSITIVE-SEMIDEFINITE"

---

Parameters obtained using preconditioning

<table>
<thead>
<tr>
<th></th>
<th>OFV</th>
<th>CL</th>
<th>V1</th>
<th>Q2</th>
<th>V2</th>
<th>Q3</th>
<th>V3</th>
<th>RUV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original model</strong></td>
<td>186.145</td>
<td>0.36</td>
<td>3.44</td>
<td>1.94</td>
<td>0.69</td>
<td>1.22</td>
<td>3.25</td>
<td>0.045</td>
</tr>
<tr>
<td><strong>Preconditioned model</strong></td>
<td>186.145</td>
<td>19.36</td>
<td>187.32</td>
<td>105.53</td>
<td>37.34</td>
<td>66.70</td>
<td>177.10</td>
<td>0.045</td>
</tr>
</tbody>
</table>

- Same OFV, very different parameter values
  - Used the parameters obtained with preconditioning as the initial estimate of the original model and obtained the same OFV.

- The smallest eigenvalue of the preconditioned R-matrix is $-5.95 \times 10^{-10}$ indicating R-matrix to be singular.

- Hence this model is highly likely to be unidentifiable.
SSE Study

- In 27/100 cases variance-covariance matrix was obtainable for the original model and RSE of all parameters were less than 50%.
- Typical RSE with preconditioning were orders of magnitude larger:

<table>
<thead>
<tr>
<th></th>
<th>CL</th>
<th>V1</th>
<th>Q2</th>
<th>V2</th>
<th>Q3</th>
<th>V3</th>
<th>PropErr.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original RSE</td>
<td>7.68%</td>
<td>6.49%</td>
<td>21.87%</td>
<td>23.73%</td>
<td>31.99%</td>
<td>25.93%</td>
<td>5.78%</td>
</tr>
<tr>
<td>RSE with Preconditioning</td>
<td>4024%</td>
<td>4027%</td>
<td>4124%</td>
<td>4089%</td>
<td>3985%</td>
<td>4069%</td>
<td>5.93%</td>
</tr>
</tbody>
</table>
Preconditioning can

- Reduce computational environment dependencies
- Recover failed covariance matrix computations
- Aid in revealing model parameter non-identifiability

- *precond* available in PsN 4.4 ([psn.sf.net](psn.sf.net))

- Computational instability can also influence the parameter estimates and an investigation of this correlation using the preconditioning method is presented as a poster:

**Acknowledgement:** This work was supported by the DDMoRe ([www.ddmore.eu](http://www.ddmore.eu)) project.