Simultaneous Determination of Tuning and Calibration Parameters for Computer Experiments

Gang Han\(^1\), Thomas Santner\(^1\), and Jeremy Rawlinson\(^2\)

\(^1\) Department of Statistics, The Ohio State University
\(^2\) College of Veterinary Medicine, Cornell University

Joint Statistical Meetings
Denver, Colorado
August 06, 2008
Outline

I. Introduction and a motivating example
II. A Bayesian/distance method for simultaneous tuning and calibration
III. Conclusion
I. Introduction

- Computer experiments are used as surrogates for many physical experiments.
- Computer codes used for computer experiments have running times that can range from minutes to days.
- Some types of inputs to computer experiments
  – Control variables: design inputs for both the computer and the physical experiments; e.g., engineering design inputs.
  – Tuning parameters and calibration parameters.
• Tuning parameters

1. Tuning parameters are the **numerical values** used by the algorithms implementing a mathematical model.

2. Tuning parameters have *no* physical meaning in an associated physical experiment.

3. The goal of tuning a computer code is to select values of the tuning parameter(s) to “best” match the physical experiment outcomes.

Calibration parameters

1. Calibration parameters are controllable inputs to a computer code and are unknown or unmeasured during the running of an associated physical experiment.

2. Calibration parameters typically have a meaning in the physical experiment.

3. The goal of calibration is to make inference about the true calibration parameter value(s).

4. Example: friction at the bone-prosthesis interface in a biomechanics application.
Goal

To propose a method for simultaneous setting calibration and tuning parameters.
A Motivating Example

- A finite-element analysis computer code simulating wear/movement of knees in a mechanical testing device.
• Response: anterior-posterior displacement (APD) of tibial tray relative to femoral component.

• Control variable: percentile of the gait cycle.

• Tuning parameter: discretization of loading curve for gait.

• Calibration parameter: initial position of femoral component with respect to tibial tray.
• Computer experiment: 438 runs; physical experiment: 36 runs.

**APD** from the knee simulator (triangles) and the FEA computer code (dots) vs **Gait cycle percentile**.
• One approach: treat both load discretization and initial position as calibration parameter; e.g., using the Bayesian approach of Higdon et. al. (2004).
Example: using Truncated Normal$(0.5, 10^2)$ ($\text{TN}(0.5, 10^2)$) priors on load discretization and initial position gives posteriors.

Marginal and joint simulated posterior distributions of load discretization (upper left) and initial position (lower right).
II. A Bayesian/distance method for simultaneous tuning and calibration

Notation

- $x$ $==$ control variable; $t$ $==$ tuning parameter; $c$ $==$ calibration parameter; $\theta_c$ $==$ the true value of $c$.
- $\{(x_i^s, c_i, t_i), y^s(x_i^s, c_i, t_i)\}_{i}^{n_s}$ $==$ training data from the computer experiment.
- $\{(x_j^p), y^p(x_j^p)\}_{j}^{n_p}$ $==$ training data from physical experiment
- $(n_s, n_p)$ $==$ number of runs of computer and physical experiments.
- $y^s = \left( y^s(x_1^s, c_1, t_1), \ldots, y^s(x_{n_s}^s, c_{n_s}, t_{n_s}) \right)^\top$.
- $y^p = \left( y^p(x_1^p), \ldots, y^p(x_{n_p}^p) \right)^\top$. 
The idea

1. Model the computer experiment $y^s(x, c, t)$ as a draw from a Gaussian process $Y^s(x, c, t)$ and the discrepancy $\delta(x, c, t) = E_\epsilon(Y^p(x)) - y^s(x, c, t)$ as a draw from a Gaussian process $\Delta(x, c, t)$. 
• The idea

1. Model the computer experiment $y^s(x, c, t)$ as a draw from a Gaussian process $Y^s(x, c, t)$ and the discrepancy $\delta(x, c, t) = E_\epsilon(Y^p(x)) - y^s(x, c, t)$ as a draw from a Gaussian process $\Delta(x, c, t)$.

2. 

\[
Y^p(x) = Y^s(x, \theta_c, t^*) + \Delta(x, \theta_c, t^*) + \epsilon(x)
\]

3. The three processes $Y^s(\cdot)$, $\Delta(\cdot)$, and the noise are independent.
Modeling Computer Experiment Output

• $Y^s(x, c, t) = \beta_Z^\top f_Z(x, c, t) + Z(x, c, t)$.

• $\beta_Z^\top f_Z(x, c, t)$: $f_Z(x, c, t)$ is the known regression coefficient vector and $\beta_Z$ is the unknown regression parameter vector.

• $Z(\cdot)$ is a stationary Gaussian stochastic process with mean 0 and variance $\sigma^2_Z$. The correlation between $Z(x_1, c_1, t_1)$ and $Z(x_2, c_2, t_2)$.

$$R_Z((x_1, c_1, t_1), (x_2, c_2, t_2)) = \prod_i 4\times(x_{1,i} - x_{2,i})^2 \times \prod_j 4\times(c_{1,j} - c_{2,j})^2 \times \prod_k 4\times(t_{1,k} - t_{2,k})^2,$$

where all $\rho \in [0, 1]$. 
Modeling the bias and the random noise processes

- \( \Delta(\cdot) \) is a Gaussian stochastic process with mean \( \beta_\Delta^\top f_\Delta(x, \theta_c, t) \) and variance \( \sigma^2_\Delta \).

- The correlation between \( \Delta(x_1, \theta_c, t_1) \) and \( \Delta(x_2, \theta_c, t_2) \) is

\[
R_\Delta \left((x_1, \theta_c, t_1), (x_2, \theta_c, t_2)\right) = \prod_i \rho_{\Delta,x,i}^{4 \times (x_{1,i} - x_{2,i})^2} \times \prod_k \rho_{\Delta,t,k}^{4 \times (t_{1,i} - t_{2,i})^2},
\]

where \( \{\rho_{\Delta,x,i}\}_i \) and \( \{\rho_{\Delta,t,k}\}_k \) are correlation parameters.

- Model measurement error \( \epsilon(x) \) as independent white noise process \((0, \sigma^2_\epsilon)\).
## Priors and Implementations

<table>
<thead>
<tr>
<th>Model parameter</th>
<th>Prior distribution</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>All elements of $\theta_c$</td>
<td>TN(0.5, 2$^2$)</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>All elements of $\rho_z$ and $\rho_\Delta$</td>
<td>Beta(1, 0.5)</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>$\sigma^2_z$</td>
<td>IG(10, 0.1/$\hat{\sigma}^2_s$)</td>
<td>(0, $+\infty$)</td>
</tr>
<tr>
<td>$\sigma^2_D$, if $\hat{\sigma}^2_s &gt; \hat{\sigma}^2_p$</td>
<td>IG(1, 100/$\hat{\sigma}^2_s$)</td>
<td>(0, $+\infty$)</td>
</tr>
<tr>
<td>$\sigma^2_D$, if $\hat{\sigma}^2_s &lt; \hat{\sigma}^2_p$</td>
<td>IG(10, 0.1/($\hat{\sigma}^2_p - \hat{\sigma}^2_s$))</td>
<td>(0, $+\infty$)</td>
</tr>
<tr>
<td>$\sigma^2_\epsilon$</td>
<td>IG(1, 100/$\hat{\sigma}^2_s$)</td>
<td>(0, $+\infty$)</td>
</tr>
</tbody>
</table>

We use a Metropolis Hastings algorithm to make draws from the posterior distribution of $\theta_c$ given the data and a fixed $t$. 
Let $\eta(\mathbf{x}) = E_\epsilon(Y^p(\mathbf{x}))$.

Define the ideal value of $t$

$$t^* = \arg \min_{t} \int \int (\eta(\mathbf{x}) - y^s(\mathbf{x}, \theta_c, t))^2 \theta_c | y^s, y^p, t] d\theta_c d\mathbf{x}.$$
• Let $\eta(x) = E_\epsilon(Y^p(x))$.

• Define the ideal value of $t$

$$ t^* = \arg \min_t \int \int (\eta(x) - y^s(x, \theta_c, t))^2 \left[ \theta_c | y^s, y^p, t \right] d\theta_c dx. $$

We estimate $(\eta(x) - y^s(x, \theta_c, t))^2 = \delta^2(x, \theta_c, t)$ by its BLUP

$$ E \left( \Delta^2(x, \theta_c, t) | y^s, y^p, t, \theta_c \right) = 
E_{[\phi | y^s, y^p, t, \theta_c]} \left[ E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \phi, \theta_c) \right], $$

where $\phi$ denotes all the remaining model parameters.
• We estimate $t^*$ by

$$\hat{t}^* = \arg\min_t \int \int E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \theta_c) \theta_c | y^s, y^p, t] d\theta_c dx$$

$$= \arg\min_t \int E[\phi, \theta_c | y^s, y^p, t] [E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \phi, \theta_c)] dx,$$

where we approximate the latter by draws from an MCMC algorithm.

• Posterior of $\theta_c$: $[\theta_c \mid y^p, y^s, \hat{t}^*]$. 
The result from Bayesian calibration program

Simulated posterior distributions of load discretization (the left panel) and initial position (the right panel).
• The result from our method

Simulated posterior distribution of initial position with estimated load discretization $\hat{t}^* = 19$. 
• **Prediction** for $\eta(x)$

$$\hat{\eta}(x) \approx \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E(Y_s(x, \hat{\theta}_{c,l}, \hat{t}^*)) + \Delta(x, \hat{\theta}_{c,l}, \hat{t}^* | y^s, y^p, \hat{t}^*, \hat{\theta}_{c,l}, \hat{\phi}_l).$$

• **Predictive interval**

$$\hat{\eta}(x) \pm z^{\alpha/2} \sqrt{\text{Var}(Y_s(x, \theta_c, \hat{t}^*)) + \Delta(x, \theta_c, \hat{t}^* | y^s, y^p, \hat{t}^*)},$$

where $z^{\alpha/2}$ is the upper $\alpha/2$ critical point of the standard normal distribution.
Toy example

\[ y^p(x) = \exp(-x) + ((x - 0.5)^2 - 0.125) + N(0, 0.01^2) \]

\[ y^s(x, (c_1, c_2), t) = c_1 \times \exp(-c_2 \times x) + (t - 0.5)^2 \times 10 \]

\[ \{n_s, n_p, N_{MC}\} = \{50, 20, 300\} \]

\[ \eta(x) = \exp(-x) + ((x - 0.5)^2 - 0.125) \]

\[ (\theta_{c_1}, \theta_{c_2}, t^*) = (1, 1, 0.5) \]
The prediction error (RMSPE) over 101 points of $\eta(x)$ from Bayesian calibration: 0.1662.

The training data (solid circles), the true response curve (the solid line), and the predictions (pluses) obtained by the Bayesian calibration program.
The prediction error (RMSPE) of \( \eta(x) \) from the Simultaneous Tuning and Calibration (STaC) program: 0.0445. Improvement rate = 73.23\% \left(= \frac{0.1662 - 0.0445}{0.1662} \times 100\% \right).

The training data (solid circles), true response curve (the solid line), predictions (pluses), and 99\% prediction bands (dashes) using the STaC program.
III. Discussion and Conclusions

- STaC performs well for tuning, calibration, and prediction of the true input-output relationship.

- We recommend that tuning parameters should be set using a discrepancy measure.

- Future work
  - Speeding up the computation.
  - Estimating the uncertainty in $\hat{t}^*$.  
  - Extending the current method to other settings; e.g., multivariate outputs, quantitative and qualitative inputs.
Thanks for your attention!
Given \( t^* = \arg\min_t \int \int \delta^2(x, \theta_c, t) [\theta_c | y^s, y^p, t] d\theta_c dx \) and

\[
\hat{\delta}^2(x, \theta_c, t) = E[\Delta^2(x, \theta_c, t) | y^s, y^p, t, \theta_c] = E[\phi | y^s, y^p, t, \theta_c] [E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \phi, \theta_c)],
\]
we define

\[
\hat{t}^* = \arg\min_t \int \int E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \theta_c) [\theta_c | y^s, y^p, t] d\theta_c dx
\]

\[
= \arg\min_t \int E[\phi, \theta_c | y^s, y^p, t] [E(\Delta^2(x, \theta_c, t) | y^s, y^p, t, \phi, \theta_c)] dx
\]

\[
\approx \arg\min_t \int \frac{1}{N_{MC}} \sum_{l=1}^{N_{MC}} E[\Delta^2(x, \theta_{c,l}, t) | \theta_{c,l}, \phi_l, y^s, y^p, t] dx
\]

\[
\approx \arg\min_t \frac{1}{N_x} \frac{1}{N_{MC}} \sum_{i=1}^{N_x} \sum_{l=1}^{N_{MC}} E[\Delta^2(x_i, \theta_{c,l}, t) | \theta_{c,l}, \phi_l, y^s, y^p, t].
\]
\[ E[\Delta^2(x, \theta_c, t)|y^s, y^p, t, \theta_c, \phi] = \\
[E(\Delta(x, \theta_c, t)|y^s, y^p, t, \theta_c, \phi)]^2 + \text{Var}[\Delta(x, \theta_c, t)|y^s, y^p, t, \theta_c, \phi] \]

- \( E(\Delta(x, \theta_c, t)|y^s, y^p, t, \theta_c, \phi) \) is a measure of the magnitude.

- \( \text{Var}[\Delta(x, \theta_c, t)|y^s, y^p, t, \theta_c, \phi] \) is a measure of the variation.

- Our method favors a \( t \) that gives small, stable discrepancy between the computer code and the physical experiment.