Neurocomputing, Volume 318, Issue November, 2018, pp. 55-64
DOI: 10.1016/j.neucom.2018.08.027

Bayesian calibration for multiple source regression model

Dmitry I. Ignatyev, Hyo-Sang Shin, Antonios Tsourdos
School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield, MK43 OAL, U.K.

Corresponding author e-mail: d.ignatyev@cranfield.ac.uk

In large variety of practical applications, using information from different sources or different kind of data is a reasonable demand. The problem of studying multiple source data can be represented as a multi-task learning problem, and then the information from one source can help to study the information from the other source by extracting a shared common structure. From the other hand, parameter evaluations obtained from various sources can be confused and conflicting. This paper proposes a Bayesian based approach to calibrate data obtained from different sources and to solve nonlinear regression problem in the presence of heteroscedastisity of the multiple-source model. An efficient algorithm is developed for implementation. Using analytical and simulation studies, it is shown that the proposed Bayesian calibration improves the convergence rate of the algorithm and precision of the model. The theoretical results are supported by a synthetic example, and a real-world problem, namely, modeling unsteady pitching moment coefficient of aircraft, for which a recurrent neural network is constructed.

Keywords: Multiple source data, multitask learning, heteroscedastisity, Bayesian calibration, regularization, nonlinear regression.

1. Introduction

In many engineering, economics and data analytics applications it is often difficult to get reliable evaluations of the critical parameters. If additional data of the same source are unavailable, a possible way is to combine the information from different kind of data or different sources [1-6]. In marketing, the study of the effect of interest rates on a demand for durable goods obtained from different communities [1], or modeling the preferences of many people, for example, with similar demographics, are also common practice [2, 3]. In addition, we can consider the prediction of student test results for a collection of schools, based on school demographics [4], and survival of patients in different clinics [5]. For engineering application, combination of numerical and experimental studies are commonly used to get the data. All of the data sources should be used to improve a model comprehensiveness. As an example, one can consider a development of a prognostic model on crack growth [6] or design of an aerodynamics model simultaneously based on the solution of the Navier-Stokes equations and experimental data [7]. Furthermore, numerous identification problems deal with solving models via combining data obtained at the tests conducted at different time instances and/or different maneuvers [8-12]. In this case, data obtained from different time frames or maneuvers could also be considered as data obtained from different sources because the test conditions corresponding to different frames could change.

Suppose that we have *s* data sources: $D_p = \left\{ \mathbf{x}_j^p, \mathbf{y}_j^p \right\}$ denotes given data for source *p*, where $j = 1 \ldots N_p$ is the number of examples for the data set *j*, $\mathbf{x}_j^p$ denotes the predictors and $\mathbf{y}_j^p$ denotes the responses. Our model assumption is that the response $\mathbf{y}_j^p$ is the output of a function *f* with additional Gaussian noise of standard deviation σ_p.

This problem can be considered as a multitask learning problem [13-23]. Any multitask learning model exploits the fact that the tasks are somehow related, and makes use of shared common structure among the tasks to obtain improved estimations. Multi-task neural network [13] learns the hidden layer representation as a common data representation for all tasks. Hierarchical Bayesian inference as a model for studying multitask learning was proposed in [15]. Information-theoretic considerations demonstrate that multitask learning can be highly advantageous. The necessary assumption to arrive at these theoretical results is that the tasks are indeed related, i.e., that they are drawn from the same (hyper)distribution. Multi-task feature learning [16] also learns a common data representation but under the regularization framework. Regularized multi-task support vector machine (SVM) [17] assumes that all tasks are similar and incorporates this assumption into the objective function of conventional SVM as a regularization term. Task clustering methods [18; 19] partition all tasks into clusters and learn a common data or model representation for all tasks in each cluster.

Under the assumption of linearity of the function *f*, we come to the multivariate linear regression problem [24]. Recently, significant efforts have been undertaken in designing a variety of techniques based on different assumptions. If the regression coefficients across different tasks are assumed to be coupled by some shared common factors, then the low rank structure of regression coefficient matrix can be obtained. Under such conditions, one can develop a good estimator of the regression coefficient matrix by adopting either a non-convex rank constraint [25-27], or a convex relaxation using the nuclear norm

regularization [21, 28-40]. Prior knowledge about task "features" can facilitate multitask learning problem [19]. For the case of multiple source learning problem, the higher-level task-dependent characteristics can be introduced naturally, namely, through dependency of the characteristics on the source type.

While developing the regression model based on multiple sources, one should bear in mind that the data could be collected in different parameter ranges, with different number of data points and with different accuracies. Considering such data leads us to a heteroscedastic model. In the processes of fitting the model to the data, it is unavoidable that some fitting of the model to data of a noise source will occur, because some components of the noise are indistinguishable from real data. The data obtained from one of the "noisy" sources could have a negative effect on the model predictions for the other source data.

There are two solutions to handle this problem. The first solution is to diagnose the outliers, which can be seen as special noise with long tail [41], then the training samples processed by removing the detected outliers will be fed into the regression models [42]. The second solution is to construct the regression model, which is robust to outliers directly [43].

There are a lot of techniques that follow the first solution [44-50]. The final regression accuracy depends largely on the goodness of outliers detection results. Note that those outlier detection methods have the risks of identifying normal points as outliers. In this case, certain information in training samples will lose. This fact will have great effects on regression performance, especially for the small size training samples. In our study we follow the second solution, i.e., designing the robust regression models.

The common strategy to enhance the robustness of the regression model is to add weights to different sources. In [49], authors showed that samples with large simulation residuals should be given small weights. In [50], authors claimed that the relatively smaller weights should be given to the sample points with large distance to others. In [51], maximum correntropy criterion that comes from information theoretic learning is selected as a loss function, while a truncated least squares loss function is employed in [52]. This loss function is non-convex, which leads to a difficult optimization task. Another method to obtain a robust regression model is to model the noise comprehensively by mixture distributions [52-60]. However, the limitation of such nonlinear regression models is that the mixture distributions can fit the noise in nonlinear regression models.

When one data source is more reliable or costly than another, a cost-sensitive learning [61] could be used. However this type of learning requires additionally specified labeling of data concerning what types are more preferable [62-64] and this is not always suitable for the multiple-source data regression.

In our approach, we follow Mackay evidence [65] for the single-source problem and extend it to the multiple-source problem to design a Bayesian Calibration for Multiple Source data (BCMS) approach. Using Bayesian method, we automatically and quantitatively embody penalty terms for "non-reliable" data source, which arise from the data itself, maximizing the evidence of the model. Regardless of the cost-sensitive learning, we do not impose any external constraints or modification of the loss function, for example, different weights for over-prediction and under-prediction errors. Using the proposed technique, we can both extract common structure and solve heteroscedasticity of the model. The noisy data from different sources are self-penalizing under Bayes' rule. Based on the proposed approach the effective algorithm, which improves the precision of the model, is elaborated. The algorithm can be applied for both linear and nonlinear problems, including neural network training. Using theoretical analysis and simulations, we show that the BCMS improves the algorithm convergence.

The paper organized in the following way. In Section 2, we briefly review the single task problem. In Section 3, Bayesian approach for development of nonlinear single-source model will be extended to the multiple-source case, and equations for optimization of hyperparameters will be derived. In Section 4, algorithms for implementation for both linear and nonlinear cases are considered. In Section 5, the algorithm convergence is analyzed. Section 6 deals with experiments. And, finally, Section 7 concludes the paper.

2. Preliminaries

Let us first give a short overview of a single-source regression problem. The problem of interpolating noisy data is formulated in the following way [65]. The data set *D* is a pair of vectors $(\mathbf{x},\mathbf{y})$, with $\mathbf{x}=\left(x_1...x_N\right)^T$ $\mathbf{y}=\left(y_1...y_N\right)^T$. The problem is to find a function *f* that models the data set with some noise

$$y_j = f\left(\mathbf{x}_j\right) + \nu,\ j=1...N,$$

where ν is the independent zero-mean Gaussian noise with standard deviations σ.

f is an interpolation model with a given functional $\mathbb{F}$ and interpolation parameters $\mathbf{w}=(w_1..w_k)\in R^k$. The problem is to find the interpolation parameters **w**.

Mackay [65] exploited Bayesian approach to introduce a regularization technique for model development. He showed that at the first level of Bayesian inference, the most probable values **w** of the approximation function corresponds to minimizing the following objective function

$$F = \frac{1}{2}\eta E_W + \frac{1}{2}\rho E_D, \tag{1}$$

where $E_D = \mathbf{e}^T\mathbf{e}$, $\mathbf{e} = \left(e_1 \ldots e_N\right)^T$, $e_i = \left(f(x_i) - y_i\right)$ is an error vector, and $E_W = \mathbf{w}^T\mathbf{w}$ is a penalizing term that implements regularization preventing overfitting. The hyperparameters η and ρ are evaluated at the second level of inference by maximizing an evidence for them. The equations for updating the parameters are given as:

$$\eta = \frac{\gamma}{E_W}, \tag{2}$$

$$\rho = \frac{N - \gamma}{E_D}, \tag{3}$$

where $\gamma = \mathrm{k} - \eta\,\mathrm{Trace}(\mathbf{H}^{-1})$ is called the effective number of parameters**, H** is a Hessian matrix, specified by the following equation

$$\mathbf{H} = \nabla^2 F = \eta\nabla^2\left(E_W\right) + \rho\nabla^2\left(E_D\right) = \eta\mathbf{I} + \rho\mathbf{B}, \tag{4}$$

where **I** and **B** denote $\mathbf{I} = \nabla^2\left(E_W\right)$, $\mathbf{B} = \nabla^2\left(E_D\right)$, ∇^2 is the Laplace operator.

3. Bayesian approach for development of multi-source regression model.

For the multi-source problem, we have *s* sources. The overall data set *D* can be divided into *s* subsets $D_1, \ldots, D_s$ of the pairs obtained from *s* different experiments, namely, $\left(\mathbf{x}^1, \mathbf{y}^1\right), \ldots, \left(\mathbf{x}^s, \mathbf{y}^s\right)$, with $\mathbf{x}^p = \left(\mathbf{x}_1^p \ldots \mathbf{x}_{N_p}^p\right)^T$, $\mathbf{y}^p = \left(\mathbf{y}_1^p, \ldots, \mathbf{y}_{N_p}^p\right)^T$, and $p = 1 \ldots s$. The data set is modeled as deviation from a mapping *f* under some additive noise processes $\mathbf{v} = \left(v^1 \ldots v^s\right)^T$

$$\begin{aligned} \mathbf{y}_{j_1}^1 &= f\left(\mathbf{x}_{j_1}^1\right) + v^1,\ j_1 = 1 \ldots N_1, \\ &\ldots \\ \mathbf{y}_{j_s}^s &= f\left(\mathbf{x}_{j_s}^s\right) + v^s,\ j_s = 1 \ldots N_s. \end{aligned} \tag{5}$$

where $\mathbf{x}_{j_p}^p = (x_{j_p 1} \ldots x_{j_p n})^T$, $j_p = 1..N_p$ is the *n*-dimensional state vector, N_p is a number of the pairs from the *p-th* data subset, $N = \sum_{p=1}^{s} N_p$ is the total number of observations and v^p are the independent zero-mean Gaussian noises with standard deviations σ_p.

Suppose that the responses $\mathbf{y}^1, \ldots, \mathbf{y}^s$ are independent related to the same function *f* and the number of observations for each subset is not necessarily the same.

Then, the likelihood function is

$$P(D \mid \mathbf{w}, \boldsymbol{\rho}) = \frac{\sqrt{\det \boldsymbol{\rho}}}{(2\pi)^{N/2}} \exp\left(-\frac{1}{2}\mathbf{e}^T\boldsymbol{\rho}\mathbf{e}\right) = \frac{1}{Z_D}\exp\left(-\frac{1}{2}\mathbf{e}^T\boldsymbol{\rho}\mathbf{e}\right), \tag{6}$$

where the constant $Z_D = \frac{(2\pi)^{N/2}}{\sqrt{\det \boldsymbol{\rho}}}$ is the normalizing coefficient. The noise matrix $\boldsymbol{\rho} = \mathrm{diag}\left(\rho_1, \ldots, \rho_N\right)$ is a $N \times N$ diagonal matrix, where $\rho_i = \sigma_i^{-2}$, and $\sigma_i, \sigma_j, i \neq j$ related to the observations x_i, x_j from the same data subset D_p are equal.

We also introduce a prior knowledge about expected smoothness of the interpolant *f* in order to improve the model prediction performance

$$P(\mathbf{w}\,|\,\eta) = \frac{1}{Z_w}\exp\left(-\eta E_w(\mathbf{w})\right). \tag{7}$$

The parameter η is a measure of expected smoothness of the function *f* and $Z_w = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \exp\left(-\eta E_w\right) dw_1 .. dw_k = \left(\frac{2\pi}{\eta}\right)^{k/2}$ is a normalizing coefficient.

At the first level of Bayesian inference, the hyperparameters η and $\rho_i, i = 1...s$ are considered to be known and the posterior probability of the interpolation parameters **w** is given by

$$P(\mathbf{w}\,|\,D,\ \eta, \boldsymbol{\rho}) = \frac{P(D\,|\,\mathbf{w}, \eta, \boldsymbol{\rho})P(\mathbf{w}\,|\,\eta)}{P(D\,|\,\eta, \boldsymbol{\rho})}\ , \tag{8}$$

or

$$P(\mathbf{w}\,|\,D, \eta, \boldsymbol{\rho}) = \frac{1}{Z_F}\exp\left(-F(\mathbf{w})\right)\ , \tag{9}$$

here Z_F is the normalizing constant, and the objective function for maximization of the posterior (9) has the following form

$$F(\mathbf{w}) = \frac{1}{2}\eta \mathbf{w}^T\mathbf{w} + \frac{1}{2}\mathbf{e}^T\boldsymbol{\rho}\mathbf{e}\,, \tag{10}$$

Note that the objective function in form (10) contains the weighted sum of errors $\mathbf{e}^T\boldsymbol{\rho}\mathbf{e}$.

Usually, the parameters η and even $\rho_i, i = 1...s$ are unknown a priori. Nevertheless, the smoothness of the interpolant are significantly determined by η. Let us use the Bayesian rule to infer the values of η and $\rho_i, i = 1...s$ from the data, namely,

$$P(\eta, \boldsymbol{\rho}\,|\,D) = \frac{P(D\,|\,\eta, \boldsymbol{\rho})P(\eta, \boldsymbol{\rho})}{P(D)}\,, \tag{11}$$

Similar to [65], we do not take any prior considerations about values of η and $\boldsymbol{\rho}$, and $P(\eta, \boldsymbol{\rho})$ is taken as a flat prior. Thus, the most probable values of η and $\boldsymbol{\rho}$ are obtained by maximizing the evidence $P(D\,|\,\eta, \boldsymbol{\rho})$. At the same time, this function is the normalizing constant in the equation (8). Expressing it in terms of the normalizing constants, one can obtain

$$P(D\,|\,\eta, \boldsymbol{\rho}) = \frac{Z_F(\eta, \boldsymbol{\rho})}{Z_W(\eta)\ Z_D(\boldsymbol{\rho})}\,. \tag{12}$$

The constants Z_D and Z_W are determined earlier in the equations (6) and (7). The constant Z_F is the following integral

$$Z_F = \int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} \exp\left(-F\right) dw_1 .. dw_k\ . \tag{13}$$

Integrating (13) is a rather complicated problem, however, we can estimate the integral (13), assuming the objective function to be quadratic in a small area in the vicinity of a minimum point (mp). Taylor-expanding of the objective function near the minimum point $\mathbf{w}_{mp}$ of the posterior density, where the gradient is zero, gives

$$F \ \approx\ F(\mathbf{w}_{mp})\ + \frac{1}{2}(\mathbf{w} - \mathbf{w}_{mp})^{\mathrm{T}}\mathbf{H}(\mathbf{w} - \ \mathbf{w}_{mp})\,,$$

and now we can evaluate the integral (13)

$$Z_F \ \approx\ \exp\ (-F_{mp}\ )\ (2\pi)^{k/2}\ (\det \mathbf{H}_{mp})^{-1/2}\,. \tag{14}$$

Substitution of the obtained expression (14) into (12) yields

$$P(D\,|\,\eta,\boldsymbol{\rho}) \;\approx \frac{\exp\,(-F_{mp}\,)\,(2\pi)^{k/2}\;(\det \mathbf{H}_{mp})^{-1/2}}{\left(\frac{2\pi}{\eta}\right)^{k/2}\left(\frac{2\pi}{\rho_1}\right)^{N_1/2}\ldots\left(\frac{2\pi}{\rho_s}\right)^{N_s/2}}\,, \tag{15}$$

The log evidence for η and $\boldsymbol{\rho}$ is

$$\ln P(D\,|\,\eta,\boldsymbol{\rho}) = -F_{mp} - \frac{1}{2}\ln\det\mathbf{H}_{mp} + \frac{k}{2}\ln\eta + \frac{N_1}{2}\ln\rho_1 + \ldots + \frac{N_s}{2}\ln\rho_s - \frac{N}{2}\ln 2\pi\,.$$

To get the value η maximizing the objective function *F*, we should set the derivative of the log evidence with respect to η equal to zero

$$\frac{d}{d\eta}\log P(D\,|\,\eta,\boldsymbol{\rho}) = 0\,.$$

Taking into account that

$$\mathbf{H} = \eta\nabla^2 E_W + \sum_{i=1}^{s}\rho_i\nabla^2\left(E_{D_i}\right) = \eta\mathbf{I} + \sum_{i=1}^{s}\rho_i\mathbf{B}_i\,, \tag{16}$$

where $E_{D_i} = \frac{1}{2}\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i$ is the sum of error squares on subset *i*, $\mathbf{B}_i = \nabla^2\left(E_{D_i}\right)$, and

$$\frac{d}{d\eta}\log\det\mathbf{H} = \frac{d}{d\eta}\mathrm{Trace}\left(\log\mathbf{H}\right) = \mathrm{Trace}\left(\mathbf{H}^{-1}\frac{d\,\mathbf{H}}{d\eta}\right) = \mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{I}\right) = \mathrm{Trace}\left(\mathbf{H}^{-1}\right),$$

we have the expression for η

$$\eta = \frac{k\;-\eta\,\mathrm{Trace}(\mathbf{H}^{-1})}{2E_w}\,. \tag{17}$$

In the similar way, differentiating the log evidence with respect to $\rho_i, i=1...s$ and setting the corresponding derivatives being equal to zero

$$\frac{d}{d\rho_i}\log P(D\,|\,\eta,\boldsymbol{\rho}) = 0,\; i=1...s\,,$$

one can obtain

$$2E_{D_i}\rho_i = N_i - \gamma_i\;. \tag{18}$$

Here $\gamma_i = \rho_i\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}_i\right)$ is a measure of the effective number of parameters that are well determined by the subset *i* of the data. The expectation of the χ_D^2 misfit between the true interpolant and the data is *N*. In our case, we do not know the true interpolant but only the inferred one, and we can evaluate only the misfit between the inferred interpolant and the data, namely, $\chi_D^2 = 2E_D\rho$ [65]. According to the equations (18) the most probable noise estimates for each subset $\rho_i, i=1...s$ lead to misfit determined for each subset $\chi_{D_i}^2 = N_i - \gamma_i$.

As opposite to the single-source approximation problem [65], where the objective function is (1), and the global parameters are calculated through the equations (2) and (3), in the framework of the proposed approach, the global parameters ρ_i are adjusted, taking into account the corresponding set error.

It could be proved theoretically that the maximum of the evidence evaluation (12) exists. The proof is provided in Appendix A.

In the linear case the statement of the problem (5) could be rewritten as follows:

$$\mathbf{y}^p = \mathbf{\Theta}^T \mathbf{x}^p + \nu^p,\ p = 1...s. \tag{19}$$

where $\mathbf{\Theta} = \left[\Theta_1 \ldots \Theta_n\right]^T$ is a *n*-dimensional state vector of parameters common to all sources. In this case the likelihood function is

$$P(D \mid \mathbf{\Theta}, \boldsymbol{\rho}) = \frac{1}{Z_D} \exp\left(-\frac{1}{2} \sum_{p=1}^{s} \rho^p \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right)^T \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right) \right). \tag{20}$$

Using the prior in the form (7), one can obtain the following optimization problem, which should be solved:

$$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \left\{ \sum_{p=1}^{s} \rho^p \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right)^T \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right) + \eta\, \mathbf{\Theta}^T \mathbf{\Theta} \right\}. \tag{21}$$

4. Algorithms for multiple source regression model

Let us start with design of an algorithm for evaluation of the nonlinear model weights. The Levenberg-Marquardt (LM) is the effective minimization algorithm for solving ill-possessed problems [66, 67]:

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \left(\mathbf{H} + \mu \mathbf{I} \right)^{-1} \mathbf{g}, \tag{22}$$

where $\mathbf{g} = \nabla \mathbf{F}$ is the gradient of the objective function, μ is a parameter of the algorithm.

The Gauss-Newton method applied to approximate the Hessian matrix within the LM algorithm leads us to $\mathbf{H} \approx \mathbf{J}^{\mathrm{T}} \boldsymbol{\rho} \mathbf{J} + \eta \mathbf{I}$, where $\mathbf{J}$ is the Jacobian. The gradient is calculated through the following equation $\mathbf{g} \approx \mathbf{J}^{\mathrm{T}} \boldsymbol{\rho} \mathbf{e} + \eta \mathbf{w}_{i-1}$. Substituting the expressions for $\mathbf{H}$ and $\mathbf{g}$ into (22), one can obtain the following equations for the adjustment of weights:

$$\mathbf{w}_i = \mathbf{w}_{i-1} - \left(\mathbf{J}^T \boldsymbol{\rho} \mathbf{J} + (\eta + \mu) \mathbf{I} \right)^{-1} \left(\mathbf{J}^T \boldsymbol{\rho} \mathbf{e} + \eta \mathbf{w}_{i-1} \right) \tag{23}$$

The derived modification of the LM algorithm improves the location of the minimum point in the case of heteroscedastic data.

The linear problem (21) is well-studied and, for example, the ridge penalized least squares estimate can be used. After introducing the Bayesian calibration weights, the equation for estimating of a model coefficients $\hat{\mathbf{\Theta}}$ becomes as follows

$$\hat{\mathbf{\Theta}} = \left(\mathbf{x}^T \boldsymbol{\rho} \mathbf{x} + \eta \mathbf{I} \right)^{-1} \mathbf{x}^T \boldsymbol{\rho} \mathbf{y}\ . \tag{24}$$

The Table 1 summarizes the results of the previous sections and gives guidelines for using the BCMS.

Table 1: Summary of BCMS algorithm applications

BCMS details	Problem statement	
	Linear	Nonlinear
Optimization problem	$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \left\{ \sum_{p=1}^{s} \rho^p \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right)^T \left(\mathbf{y}^p - \mathbf{\Theta}^T \mathbf{x}^p \right) + \eta\, \mathbf{\Theta}^T \mathbf{\Theta} \right\}$	$\hat{\mathbf{\Theta}} = \arg\min_{\mathbf{\Theta}} \left\{ \mathbf{e}^T \boldsymbol{\rho} \mathbf{e} + \eta \mathbf{w}^T \mathbf{w} \right\}$
Estimator	Ridge $\hat{\mathbf{\Theta}} = \left(\mathbf{x}^T \boldsymbol{\rho} \mathbf{x} + \eta \mathbf{I} \right)^{-1} \mathbf{x}^T \boldsymbol{\rho} \mathbf{y}$	LM: $\mathbf{w}_i = \mathbf{w}_{i-1} - \left(\mathbf{J}^T \boldsymbol{\rho} \mathbf{J} + (\eta + \mu) \mathbf{I} \right)^{-1} \left(\mathbf{J}^T \boldsymbol{\rho} \mathbf{e} + \eta \mathbf{w}_{i-1} \right)$

The detailed steps for implementation of the BCMS for a nonlinear regression problem are given in Algorithm 1.

Algorithm 1 Bayesian Calibration for Multi-Source Data (BCMS)

1: Initialize the parameters η, ρ_i ,where *i*=1...*s*. We choose to set $\eta = 0,\ \rho_i = 1/s$, $\mu = 0.005$. We use the Nguyen-Widrow method of initializing the weighs [68].
2: Take one step of the LM algorithm (23) to minimize the objective function *F* (10).
3: Compute the sum of error squares $E_{D_i} = \frac{1}{2}\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i$ for each subset *i*, *i=1...s* and the regularizing function $E_W = \mathbf{w}^T\mathbf{w}$.
4: Compute the parameters of the objective function $\eta, \rho_1..\rho_s$ using (17), (18).
5: Iterate steps 2 through 4 until convergence.

The best results are obtained if the training data is first mapped into the range [-1,1] (or some similar region). Typically both inputs and outputs are scaled.

5. Convergence analysis

Convergence of the algorithms of the type (23) is determined with the condition number $\mu = \left|\max_k \lambda_{\mathbf{H}}^k\right| / \left|\min_k \lambda_{\mathbf{H}}^k\right|$ of the Hessian **H** [69], where $\lambda_{\mathbf{H}}^k$ is the eigen values of it. A lesser condition number leads to a higher convergence rate. We show bellow that iterations (16-18), and (23) improves the condition number and speeds up the algorithm convergence as compared to the GNBR [67]. Let us analyze the convergence in case of two subsets, the extension to the multiple-subset case is straight-forward. In this case the data obtained for the first data set is more reliable, namely, $\sigma_1 < \sigma_2$ and, hence, $E_{D_1} < E_{D_2}$.

If we apply the BCMS iterations instead of GNBR iterations we can get the value of ρ_1 :

$$\rho_1 = \frac{N_1 - \rho\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}_1\right)}{2E_{D_1}}.$$

where $\mathbf{B}_1 = \nabla^2\left(E_{D_1}\right)$.

Under the estimations $E_D \cdot \frac{N_1}{N} \geq E_{D_1}$, and $\mathbf{B}\cdot\frac{N_1}{N} \geq \mathbf{B}_1$, where $\mathbf{B} = \nabla^2\left(E_D\right)$, the following evaluations for the global parameter ρ_1 can be obtained

$$\rho_1 = \frac{N_1 - \rho\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}_1\right)}{2E_{D_1}} \geq \frac{N - \rho\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}\right)}{2E_D} = \rho_{GNBR}. \tag{25}$$

where ρ_{GNBR} is obtained within the GNBR technique through Eq. (3). In the similar way, we can evaluate:

$$\rho_2 = \frac{N_2 - \rho\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}_2\right)}{2E_{D_2}} \leq \frac{N - \rho\,\mathrm{Trace}\left(\mathbf{H}^{-1}\mathbf{B}\right)}{2E_D} = \rho^{GNBR}, \tag{26}$$

where $\mathbf{B}_2 = \nabla^2\left(E_{D_2}\right)$.

Thus, the algorithm assigns for the data from the more reliable source the higher weight and the smaller weight to lesser reliable source. The algorithm prevents growth of the model parameters **w** caused by fitting the noisy data penalizing the less reliable source with smaller weights, while increasing the weights for more reliable data.

The highest value of the eigen value $\max_k \lambda_{\mathbf{H}}^k$ determine the eigen vector aligned with the direction that is well determined by the data, whilst the smallest value $\min_k \lambda_{\mathbf{H}}^k$ specifies the vector that lies in the direction that is poorly determined by the

data. In the case of domination of noisy data in the overall data set the highest eigen value is determined by the noise source and the smallest one is determined by the more reliable source. Assigning higher weights for the more reliable data, and lower weights for the less reliable data one can decrease the condition number of the problem $\mu = \left|\max_k \lambda_{\mathbf{H}}^k\right| / \left|\min_k \lambda_{\mathbf{H}}^k\right|$, through decreasing of $\max_k \lambda_{\mathbf{H}}^k$ and increasing of $\min_k \lambda_{\mathbf{H}}^k$. Thus, the Bayesian calibration improves the ill-conditioned matrix **H** and increases the robustness of the algorithm in case of combining the data from multiple sources.

It could be also analyzed through the numbers of the effective parameters (18) and the misfit criterion determined for each subset $\chi_{D_i}^2$. Increasing the weights for the less noisy data the algorithm automatically increases the number of parameters used for the reliable source. It helps to prevent excessive fitting the data as the misfit of the noisy data $\chi_{D_2}^2 = N_2 - \gamma_2$ is decreased.

6. Experiments

Simple example

Let us consider an example of using the proposed approach. In order to demonstrate the main advantages of the BCMS a simple example will be considered. We try to estimate the following test function from the noisy data:

$$y = \begin{cases} \sqrt{|x|}, & x < 0; \\ \sin 3x, & x \geq 0. \end{cases} \tag{27}$$

We constructed two subsets using function (27). Subset 1 consists of 12 points generated using function (27) at $x < 0$, with an addition of a noise with standard normal distribution with zero mean and variance 0.01. The Subset 2 is 500 points generated using the function (27) at $x > 0$, with an addition of a noise with standard normal distribution with zero mean and variance 0.9. This example artificially simulates a problem of solving the regression model based on data obtained from two sources. For example, one of the sources is more precise but more expensive ($x < 0$). The other ($x > 0$) is less precise but more affordable, so we can get more data. The simulated data (markers) and the noiseless function (solid line) are shown in Fig.1.

In order to evaluate performances of the algorithms solely, we selected the polynomial regression model *f* of the tenth order. The BCMS applied for this data is compared with GNBR. The results obtained for the linear regression (LR) technique are also compared.

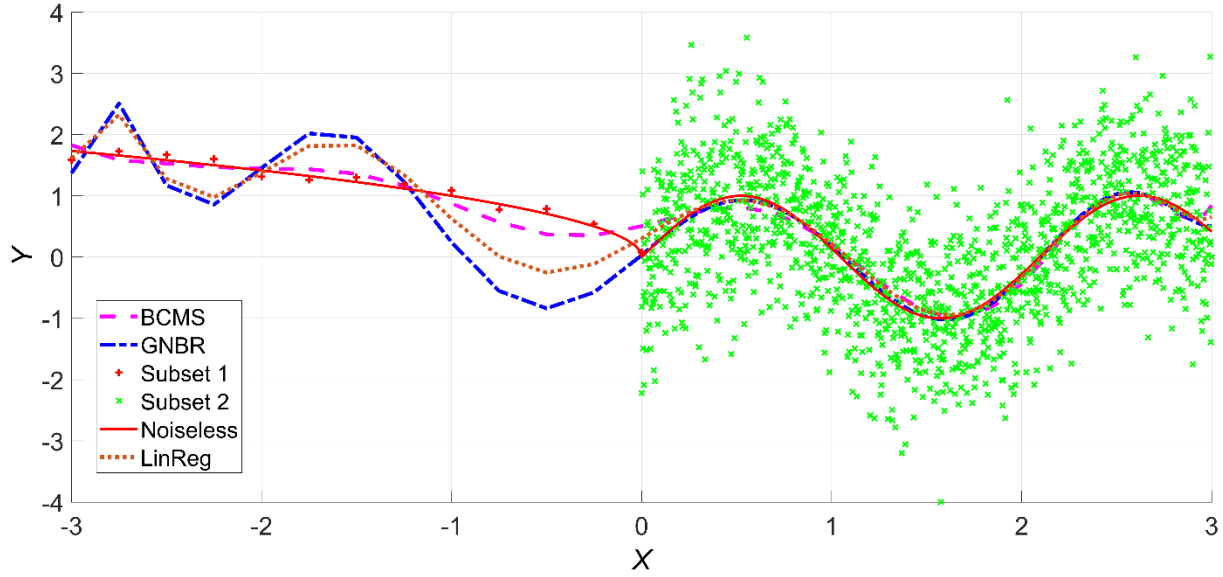


Fig. 1. Estimation of the synthetic example function corrupted by noise. The Subset 1 is the noisy function (27) at $x < 0$, the noise has the standard normal distribution with zero mean and variance 0.01. The Subset 2 is the noisy function (23) at $x > 0$, the noise has standard normal distribution with zero mean and variance 0.9.

From the figure one can see that the GNBR as well as the LR techniques are significantly determined by the data extracted from the "second" experiment and do not catch the "Noiseless" dependency determined in the "first" experiment. The GNBR and LR models oscillate with respect to the sought function ($\sqrt{|x|}$) since the data obtained in the "second" experiment ($\sin 3x$) dominate in the training set. As compared to the other models, the BCMS does not perform significantly worse on the "second" subset but it fits the initial model more better on the "first" subset. The regularization factors ρ_1, ρ_2, inferring the "reliability" of the data subset from the data itself, prevent the model from studying intensively the "less reliable" data.

Development of the regression model of (27) is an ill-possessed problem with poor conditional number. For example, usage of the LR gives $\mu = 1.55\,e{+}11$. Usage of the proposed Bayesian calibrated weight matrix $\boldsymbol{\rho}$ improves the condition number while calculating the Hessian **H** through scaling the eigen values. The condition number for the Hess matrix **H** for the GNBR algorithm in this example is $\mu = 1.55\,e{+}11$, but for the BCMS algorithm we can get $\mu = 4.3\,e{+}10$. In addition, the tests showed that the BCMS requires lesser time for convergence, compared with the GNBR, in spite of the fact that BCMS requires additional calculation of the values of calibration weights according to (18) instead of calculating only the single value for GNBR (3). Thus, introduction of the calibration matrix $\boldsymbol{\rho}$ into the objective function (10) amends the Hessian that leads to improved convergence and reduced calculation time.

It should be noted that the BCMS should give the same results as the GNBR in case of similar error for the different subsets. For example, if we consider the same function (27) with the noise with standard normal distribution with zero mean and variance 0.01 added to each subset (500 points for each subset), the performance of both methods will be quite similar (see Figure 2).

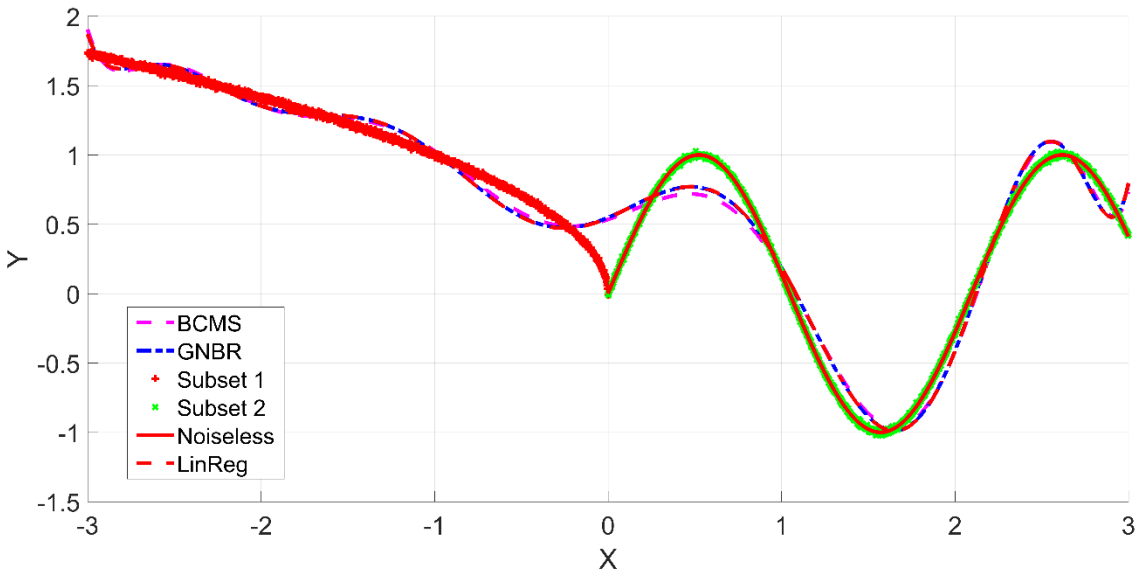


Fig. 2. Estimation of the synthetic example function corrupted by noise that has the standard normal distribution with zero mean and variance 0.01.

Thereby, these examples demonstrate that the proposed approach reduces fitting of the noisy data in the case when the noisy data prevails over the less noisy data. On the other hand, if both sources have the same level of noise, the BCMS algorithm yields the results not worse than results that could be obtained using the conventional approaches.

A real-world problem

The BCMS algorithm is also tested for the development of the model of the unsteady pitching moment coefficient of aircraft C_m derived from two sources, the data could be found in [70]. The data are obtained in wind tunnel experiments in two different studies, representing the two data sources. The results of the first experiments are given in the form of tabular model and the results of the second experiment are represented in the form of pitching moment coefficient evolutions through time during the oscillation of the aircraft model inside a wind tunnel. The first source, which is the tabular model of the unsteady pitching moment coefficient, is suitable for description only within the normal flight envelope. The additionally conducted experiment

is used to support the investigations of the flight dynamics and control systems beyond the normal flight envelope. The resulting model should fit the data from both sources.

The tabular model specifies the unsteady pitching moment coefficient as Taylor series expansion with respect to state parameters up to linear terms:

$$C_m = C_{m_0}(\alpha_0) + C_{m_\alpha}(\alpha_0)\cdot(\alpha - \alpha_0) + C_{m_q}(\alpha_0)\cdot\bar{q}\,, \tag{28}$$

where the α_0 is a trimming angle of attack, α is a current angle of attack, $\bar{q}$ is an undimensional pitching rate. The coefficient of the model $C_{m_0}(\alpha_0), C_{m_\alpha}(\alpha_0)$ and $C_{m_q}(\alpha_0)$ are the tabular data, which are usually determined by wind tunnel experiments with small-amplitude pitch forced oscillation of the aircraft model

$$\begin{aligned} \alpha &= \alpha_0 + A_\alpha \sin(2\pi ft), \\ q &= 2\pi f\ A_\alpha \cos(2\pi ft). \end{aligned} \tag{29}$$

$C_{m_0}, C_{m_\alpha}, C_{m_q}$ are two-dimensional matrices, describing the dependency of coefficients on two parameters, namely, mean angle of attack α_0 and the frequency of oscillation *f*. The dependency of the coefficients of the model (28) not only on the trimming angle α_0, but also on the frequency of oscillations *f* significantly complicates the analysis of flight dynamics and control system design, since such parameters could not be determined for an arbitrary aircraft manoeuvre. The model of pitch moment coefficient should implicitly incorporate the effects obtained in the experiments, namely, the dependence on trimming angle α_0, frequency of oscillations *f*, but these experimental parameters cannot be the predictors of the model. The coefficients of the model (28) are given for frequencies *f* = 0.5, 1 and 1.5 Hz for angles of attack α_0 from -10 deg to 40 deg with a step 2 deg. Thus, there are 78 test cases for the first source (26 cases for each oscillation frequency).

The second source of the data is the pitching moment coefficient evolution through time obtained at the large-amplitude pitch forced oscillation of the aircraft model inside the wind tunnel. These motions are more aggressive, compared with the small-amplitude oscillations, and dedicated to investigate high-angle-of-attack departures such as aircraft stall, Cobra maneuvers, etc. This source of data gives the evolution of the total pitching moment coefficient C_m with respect to time. The examples of the pitching moment coefficient evolutions are given in Fig. 3.

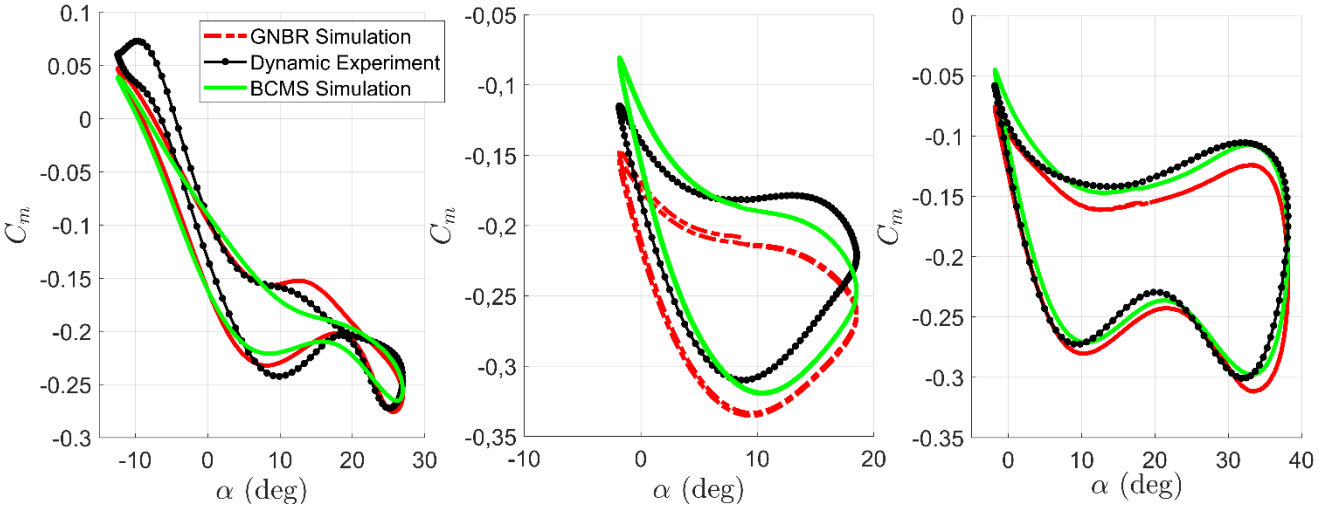


Fig. 3. Estimation of the pitch moment coefficient obtained at the forced large-amplitude oscillation results. Left figure is the pitch moment coefficient evolution during angle-of-attack oscillation, the mean angle of attack $\alpha_0 = 8$ deg and the oscillation amplitude $A_\alpha = 20$ deg. Center figure is the oscillation case with the mean angle of attack $\alpha_0 = 8$ deg and the oscillation amplitude $A_\alpha = 10$ deg. Right figure is the oscillation case with the mean angle of attack $\alpha_0 = 18$ deg and the oscillation amplitude $A_\alpha = 20$ deg.

In order to combine the data represented in the form of look-up table and pitching moment coefficient evolutions we generated the pitching moment coefficient evolution through the time with Eqs. (28-29), i.e., we obtained the pitching moment coefficient during the small-amplitude oscillations with three frequencies *f* for each angle of attack α_0 available in the look-up table. This is our first data subset D_1. The second data subset D_2 is the pitch moment evolutions during the large amplitude oscillations. Small-amplitude oscillation cases are 78, large-amplitude oscillation cases are 12; 36 from 78 small amplitude test cases and 8 from 12 large amplitude test cases were randomly selected for training, the rest of the data were used for testing. Evolutions of the pitch moment coefficient and kinematic parameters during each oscillating case (both for small and large amplitude oscillation tests) are discretized in time into 128 steps.

In the present study, we use the Recurrent Neural Network (NN) [71], which has the NARX configuration, since such type suites better for modelling of dynamics observed in the unsteady aerodynamic coefficients [72]. In the Recurrent NN an output depends on a current input to the network, as well as on the current or previous inputs and outputs of the network.

For the considered problem one hidden layer containing 12 neurons is used. The activation function of hidden layer neurons is the sigmoid function $f_k(x) = 1/(1+\exp(-x))$. The input layer has 7 neurons; the output layer has only one neuron.

While modeling, the NN inputs are the input vectors at the time steps *i*, *i-1*, *i-2*, as well as the NN simulation results at the previous time step *i-1*. The resulting NN model [71] takes the following form:

$$y(t) = M\left(\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), y(t-1)\right),$$

where $\mathbf{x}(i) = \left(\alpha(i), q(i)\right)$ is the state vector, $y(t)$ is the pitching moment coefficient C_m.

The number of delay steps for the state vector is $D_{in} = 2$ and the number of delay steps for the pitch moment is $D_{out} = 1$. Such configuration is selected as providing better generalization ability. Smaller numbers of delay steps do not allow the NN to describe the data. On the other hand, the larger numbers of delay steps introduce additional information in NN that disturbs the output signal.

A series-parallel configuration can be used to train the RNN [73]. Because the true output is available during the training of the network, it is possible to create a feed-forward architecture, in which the true output is used instead of feeding back the estimated output. This has two advantages. The first is that the input to the feed-forward network is more accurate. The second is that the resulting network has a purely feed-forward architecture, and static backpropagation can be used for training. Application of the BCMS is quite straightforward in this case. After training the series-parallel configuration was switched to the recurrent configuration. For training, the patterns are composed of the records of pitch moment coefficient $C_m(i)$ at the current step *i*, and of the pitch moment coefficient at the step *i*-1 $C_m(i-1)$, together with the state vector $\mathbf{x}(i) = \left(\alpha(i), q(i)\right)$ at the steps *i*, *i*-1, *i*-2.

At the modelling stage, predicting the pitch moment coefficient $C_m(i)$, RNN uses results computed at the previous time step $C_m(i-1)$, along with the current and two previous steps of the state vector. Hereby, the model is the nonlinear regression on seven parameters.

As with the simple synthetic test described above, the proposed algorithm converges faster than GNBR. The prediction abilities of the models are compared on the testing set in Fig. 3 through coplotting the C_m values measured in the experiment and predicted by the models. From the above results, one can conclude that the NN model, trained with the BCMS algorithm, has better agreement with the experiments.

More thorough analysis of the obtained results is implemented to determine whether the BCMS algorithm helped to improve accuracy of the models derived from the multi-source data. A quantitative comparison of the training techniques is done by calculating the errors obtained for the models of pitch moment coefficient C_m and the aerodynamic derivative C_{mq} separately for the train and test subsets. The error measure is the mean square error divided by the entire range Δy of the measured value y^{test}:

$$err_i = \frac{\sqrt{\frac{1}{N_i - 1}\sum_{j=1}^{N_i}(y_j^{test} - y_j^{sim})^2}}{\Delta y}. \qquad (30)$$

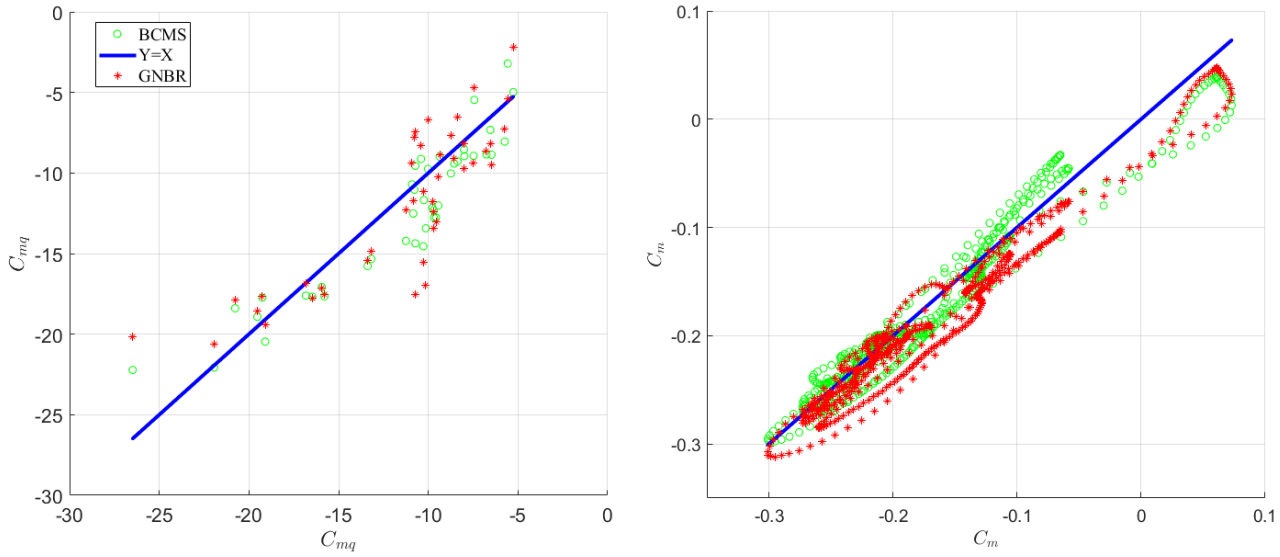
The calculated performance according to (30) as well as the standard deviation (Std) over 30 realizations are given in Table 2.

Table 2 Errors of models

Name	Variable	err_i			
		Train, %	Std	Test, %	Std
GNBR	C_{mq} (small amplitudes)	7.09	236.32	8.58	316.31
	C_m (large amplitudes)	5.59	185.44	8.3	228.66
BCMS	C_{mq} (small amplitudes)	5.65	92.73	5.77	174.87
	C_m (large amplitudes)	4.53	60.09	6.34	41.18
FFNN (GNBR)	C_{mq} (small amplitudes)	3.88	00.07	3.96	65.90
	C_m (large amplitudes)	15.25	232.84	27.42	229.85

In order to compare the performance of the proposed algorithm solely let us firstly compare the results obtained for the RNNs, trained with GNBR and BCMS algorithms. The comparison reveals the accuracy improvement of the model developed with BCMS. The errors for C_m decreased by 19% and 24% for the train and test subsets, respectively. The errors for C_{mq} decreased by 20% and 33% for the train and test subsets, respectively. The results for standard deviation calculated for 30 realizations of the models are also presented in Table 2. One can see from it that the Bayesian calibration significantly reduced the standard deviation, thereby improving the robustness of the algorithm.

In addition, the scatterograms plotted for the test subsets of C_{mq} and C_m are shown in Figs. 4a and 4b. The pattern obtained for BCMS is less scattered.



a) b)

Fig. 4. Scattering diagrams obtained for the test subsets. (Left) Small-amplitude subset; (Right) Large-amplitude subset.

Let us compare the results obtained for the RNN networks with the results obtained using a NN of the feed-forward configuration (FFNN). The designed FFNN has 2 hidden layers, with 12 neurons being in the first layer and 7 neurons being in the second layer. The neuron activation function of the both layers is also chosen sigmoid function. Patterns for the training of the FFNN are composed of the records of pitch moment coefficient C_m, together with the state input determined for the oscillation case

$$\mathbf{x} = \left(\alpha(t), q(t), t, \alpha_0, A_\alpha, f\right)$$

Each oscillation case (both for small and large amplitude oscillation tests) is discretized in time into 128 steps. Subdivision of the data on the training and testing subsets is similar to the RNN case. To simulate the pitch moment coefficient at the time *t* of an experiment case, the input vector should be fed in the NN.

While training of the FFNN, we used the Bayesian regularization approach designed for the single-source application, minimizing the objective function (1), with objective function updating law in the form (2) and (3). LM algorithm (22) with Gauss-Newton approximation of the Hessian GNBR [67] was used. The model performance calculated via Eq. (30) as well as standard deviation calculated over 30 realizations is presented in the Table 2.

From the comparison results, one can conclude that the FFNN error for the small amplitude test is approximately the same as for the training and testing subsets. The error for the large amplitude test is very high, approximately four times higher for the train subset and seven times higher for the test subset as compared with the small-amplitude results. The similar tendency is observed for the standard deviation, it is much higher for large-amplitude test. Thus, the FFNN provides worse performance for the large amplitude subset. The reason behind this results is that the small-amplitude training examples are dominant in the overall training set, namely, 36 from 44 cases. The FFNN trained better to model the small-amplitude behavior shows poor performance for the large-amplitude subset. This is not satisfactory from the point of view of flight dynamics because a model should guarantee an equivalent level of the model precision in overall simulation domain for proper design a control system [74, 75].

7. Conclusion

In many practical applications, usage of information from different sources or different kind of data helps to improve a model comprehensiveness. On the one hand, using the multiple sources, we can extract a shared common structure among different sources. On the other hand, parameter evaluations obtained from certain different sources can be confused and conflicting. We applied the Bayesian approach to calibrate data obtained from different sources and to solve the nonlinear regression problem in the presence of the model heteroscedastisity. The sources with larger simulation residuals have smaller weights while minimizing the objective function. The proposed technique infers the calibration weights from the data according to the source reliability and prevents overfitting over the data obtained from the noisy source. We implemented two levels of the Bayesian inference: at the first level, we infer what the model's parameters might be given the data. At the second level we infer from the data the values of the model hyperparameters, namely, the calibration weights and the regularization parameter. We designed the algorithm for the effective implementation of the proposed technique. The algorithm uses the Gauss-Newton approximation of the Hessian matrix within the Levenberg-Marquardt algorithm.

Significant benefits of using the algorithm are observed when the data from the noisy source dominating in the overall data set. Construction of the regression in this case could become ill-conditioned problem. The proposed Bayesian calibration decreases the condition number of the problem improving the robustness and convergence rate of the algorithm. Increasing the calibration weights for the less noisy data, the algorithm automatically increases the number of the model parameters used for the more reliable source. Such reallocation of the parameters helps to prevent excessive fitting of the noisy data. When the level of noise is equal in all sources the algorithm performance is similar to conventional algorithms.

Two experiments based on the synthetic and the real-world datasets are conducted. The real-world experiment was the development of the neural network models describing the pitching moment coefficient of aircraft based on wind tunnel data. The results revealed that the proposed approach deals effectively in the presence of the data obtained from different sources. Usage of the Bayesian calibration improves the precision of the models. The tests supported the theoretical results on convergence speed-up.

8. Appendix A. Analysis of existence of evidence maximum evaluation

Let us consider an existence of evidence maximum evaluation. For this, we should analyze the accessibility of the maximum of the evidence function in the form of the equation (12). Substituting the expressions for the constants Z_W, Z_D and Z_F from (6), (7) and (13) into the equation (12), we have

$$\begin{aligned} P(D\,|\,\eta,\boldsymbol{\rho}) = \\ &= \left(\frac{\eta}{2\pi}\right)^{k/2} \prod_{i=1}^{s}\left(\frac{\rho_i}{2\pi}\right)\int \exp\left(-\frac{1}{2}\left(\eta \mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathbf{e}^{\mathrm{T}}\boldsymbol{\rho}\mathbf{e}\right)\right) d\mathbf{w} = \\ &= (2\pi)^{-\frac{k+N}{2}} \int \exp\left(-\frac{1}{2}\left(\eta \mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathbf{e}^{\mathrm{T}}\boldsymbol{\rho}\mathbf{e} - \sum_{i=1}^{s}\log\left(\rho_i^{N_i}\right) - \log\left(\eta^k\right)\right)\right) d\mathbf{w}. \end{aligned} \tag{31}$$

To show the accessibility of the maximum of the function $P(D\,|\,\eta,\boldsymbol{\rho})$ we should prove the existence of the maximum of the integral (31). Let us use the following property of the integrals.

If integrals $\int_a^b f(x)dx$ *and* $\int_a^b g(x)dx$ *converge, and for all* $x \in (a,b)$ *the following inequality is true*

$$f(x) \le g(x),$$

then the following inequality is also true

$$\int_a^b f(x)\,dx \le \int_a^b g(x)\,dx.$$

Using the property stated above one can conclude that the existence of the maximum of $P(D\,|\,\eta,\boldsymbol{\rho})$ is equivalent to existence of the minimum of the following expression

$$E = \eta \mathbf{w}^{\mathrm{T}}\mathbf{w} + \mathbf{e}^{\mathrm{T}}\boldsymbol{\rho}\mathbf{e} - \sum_{i=1}^{s}\log\left(\rho_i^{N_i}\right) - \log\left(\eta^k\right),$$

defined on the set

$$\Lambda = \left\{\Theta = (\eta,\boldsymbol{\rho}) : \eta > 0, \rho_i > 0, i = 1...s\right\}.$$

The following Lemma is of use in our purposes. The proof is straightforward.

Lemma. *Function* $g(x) = Bx - A\log x,\ A, B, x > 0$ *has the following properties:*

1). $\lim_{x\to+0} g(x) = +\infty$, since $\lim_{x\to+0}\log x = -\infty$;

2). $\lim_{x\to+\infty} g(x) = +\infty$;

3). $\min_{x>0} g(x) = g(x_*) = A\left(1 + \log\frac{B}{A}\right);\ x_* = \frac{A}{B}$.

For the sake of simplicity, let us designate $\mathbf{w}^{\mathrm{T}}\mathbf{w} = Q_0,\ \mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i = Q_i$.

Statement. *If the following condition is true, namely,* $Q_i > 0, i = 0...s$, *then E is bounded below on the set* Λ, *and the lower boundary of E on the boundary of the set* Λ *is equal to infinity; hereby, the infinum of E is achieved*

Proof. Boundedness of E on Λ follows from the Lemma.

1) If Θ_m is not a boundary point, then

$$E = \eta_m Q_0 - k\log\left(\eta_m\right) + \sum_{i=1}^{s}\left(\rho_{i_m}\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i^{\mathrm{T}} - N_i\log\left(\rho_{i_m}\right)\right) \ge k\left(1 + \log\frac{Q_0}{k}\right) + \sum_{i=1}^{s} N_i\left(1 + \log\frac{Q_i}{N_i}\right).$$

2) If $\Theta_m \to \Theta^*$, where Θ^* is the boundary point of Λ, then $E(\Theta^*) \to +\infty$.

There are two possible cases, namely,

a). At least one of the parameters tends to zero, $\eta, \rho_i \to 0$, where i=1...s,

b). At least one of the parameters tends to infinity, $\eta, \rho_i \to \infty$, where i=1...s.

According to Lemma, $E(\Theta^*) \to +\infty$ in both cases. The statement is proven.

9. References

1. G. E. P. Box and G. C. TjaO, Bayesian inference in statistical analysis, 588 p., Addison-Wesley Publishing Company, 1973.
2. G. M. Allenby and P. E. Rossi, Marketing models of consumer heterogeneity, Journal of Econometrics, *89, p. 57–78*, 1999.
3. N. Arora G.M Allenby, and J. Ginter, A hierarchical Bayes model of primary and secondary demand, Marketing Science, 17,1, p. 29–44, 1998.
4. M. Aitkin and N. Longford, Statistical modelling issues in school effectiveness studies, Journal of the Royal Statistical Society A, 149, 1–43, 1986.
5. M. Daniels and C. Gatsonis. Hierarchical generalized linear models in the analysis of variations in healthcare utilization, Journal of the American Statistical Association, 94, 29–38, 1999.
6. A. Makeev, Y. Nikishkov, R. Cross, and E. Armanios, Empirical Modeling Based on Neural Networks and Bayesian Learning, 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference, USA, AIAA 2004-1830.
7. E. Groen, W. Ledegang, J. Field, H. Smaili, M. Roza, L. Fucke, S. Nooij, M. Goman, M. Mayrhofer, L. Zaichik, M. Grigoryev, and V. Biryukov, SUPRA - Enhanced Upset Recovery Simulation, AIAA Modeling and Simulation Technologies Conference, Guidance, Navigation, and Control and Co-located Conferences, AIAA 2012-4630, 2012.
8. E.A. Morelli, V. Klein, Application of system identification to aircraft at NASA Langley Research Center, Journal of Aircraft, 42, pp. 12-25, 2005
9. J. Shin, H. Jin Kim, S. Park, Y. Kim Model predictive flight control using adaptive support vector regression, Neurocomputing, Vol. 73, Is. 4–6, pp. 1031-1037, 2010.
10. J. Pattinson, M. H. Lowenberg, and M. G. Goman, Multi-Degree-of-Freedom Wind-Tunnel Maneuver Rig for Dynamic Simulation and Aerodynamic Model Identification, Journal of Aircraft, Vol. 50, No. 2, pp. 551-566, 2013.
11. S. De Jesus Mota and R. M. Botez, New identification method based on neural network for helicopters from flight test data, AIAA Atmospheric Flight Mechanics Conference, 2009, Chicago, USA, AIAA 2009-5938.
12. J.V. Caetano, C.C. de Visser, G.C.H.E. de Croon, B. Remes, C. De Wagter, J. Verboom, and M. Mulder, Linear aerodynamic model identification of a flapping wing mav based on flight test data. International Journal of Micro Air Vehicles, 5(4), pp. 273–286, 2013.
13. R. Caruana. Multitask learning, Machine Learning, 28, pp. 41–75, 1997.
14. L. Pratt and B. Jennings, A survey of transfer between connectionist networks. Connection science, 8, pp.163-184, 1996.
15. J. Baxter, A model of inductive bias learning, Journal of Artificial Intelligence Research, 12, pp. 149-198, 2000.
16. A. Argyriou, T. Evgeniou, and M. Pontil, Convex multi-task feature learning. Machine Learning, 73, p. 243-272, 2008.
17. T. Evgeniou and M. Pontil, Regularized Multi-Task Learning, Proc. 10th ACM SIGKDD International Conf. Knowledge Discovery and Data Mining, pp. 109-117, 2004.
18. S. Thrun and J. O'Sullivan, Discovering structure in multiple learning tasks: The TC algorithm. In International Conference on Machine Learning , pages 489–497, 1996.
19. B.Bakker and T.Heskes, Task clustering and gating for Bayesian multitask learning. Journal of machine learning research, 4, pp. 83-99, 2003.
20. R. K. Ando and T. Zhang, A framework for learning predictive structures from multiple tasks and unlabeled data, The Journal of Machine Learning Research, 6, pp. 1817-1853, 2005.
21. R. Johnson, and T. Zhang, Graph-based semi-supervised learning and spectral kernel design. IEEE Transactions on Information Theory, 54, pp. 275-288, 2008.
22. J. Zhang, Z. Ghahramani, and Y. Yang, Learning multiple related tasks using latent independent component analysis. In Advances in Neural Information Processing Systems 18, pp. 1585-1592, 2006.
23. T. Jebara. Multi-task feature and kernel selection for SVMs. In Proceedings of the 21st International Conference on Machine Learning, 2004.

24. H. Liu, L. Wang, and T. Zhao. Multivariate regression with calibration. In Advances in Neural Information Processing Systems, Vol. 27, pp. 127-135, 2014.
25. A. Izenman, Reduced-rank regression for the multivariate linear model, Journal of multi-variate analysis, 5, 248-264, 1975.
26. A. Izenman, Modern multivariate statistical techniques: regression, classification, and manifold learning, 2008, Springler.
27. G. Reinsel, and R. Velu, Multivariate reduced-rank regression: theory and applications, Springer New York, 1998.
28. M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, Dimension reduction and coefficient estimation in multivariate linear regression, Journal of the Royal Statistical Society: Series B 69, p. 329-346, 2007.
29. Y. Amit, M. Fink, N. Srebro, and S. Ullman, Uncovering shared structures in multiclass classification, In Proceedings of the 24th international conference on Machine Learning, ACM, 2007.
30. S. Negahban, and M. J. Wainwright, Estimation of (near) low-rank matrices with noise and high-dimensional scaling, The Annals of Statistics, 39, pp. 1069-1097, 2011.
31. A. Rohde, and A. B. Tsybakov, Estimation of high-dimensional low-rank matrices. The Annals of Statistics, 39, pp. 887-930, 2011.
32. F. Bunea, Y. She, and M. Wegkamp, Optimal selection of reduced rank estimators of high-dimensional matrices. The Annals of Statistics, 39, pp. 1282–1309, 2011.
33. F. Bunea, Y. She, and M. Wegkamp, Joint variable and rank selection for parsimonious estimation of high dimensional matrices. The Annals of Statistics, 40, pp. 2359–2388, 2012.
34. F. Bunea and A. Barbu, Dimension reduction and variable selection in case control studies via regularized likelihood optimization, Electronic Journal of Statistics, 3, p. 1257-1287, 2009.
35. R. Salakhutdinov, and N. Srebro, Collaborative Filtering in a non-uniform world: Learning with the weighted trace norm. In Advances in Neural Information Processing Systems, V. 23, pp. 2056-2064, 2010.
36. R. Foygel, and N. Srebro, Concentration-based guarantees for low-rank matrix reconstruction, In 24th Annual Conference on Learning Theory, vol. 19, 2011.
37. T. Evgeniou, C. A. Micchelli, and M. Pontil, Learning multiple tasks with kernel methods. Journal of Machine Learning Research, 6 615, 2006.
38. T. Heskes, Empirical Bayes for learning to learn. In Proceedings of the 17th International Conference on Machine Learning, 2000.
39. Y. W. Teh, M. Seeger, and M. I. Jordan, Semiparametric latent factor models. In Proceedings of the International Workshop on Artificial Intelligence and Statistics, vol. 10, 2005.
40. K. Yu, V. Tresp, and A. Schwaighofer, Learning Gaussian processes from multiple tasks. In Proceedings of the 22nd international conference on Machine Learning, 2005.
41. H. Zhu, H. Leung, Z. He, A variational Bayesian approach to robust sensor fusion based on Student-t distribution, Inform. Sci., 221, 201–214, 2013.
42. S. Weisberg, Applied Linear Regression, John Wiley and Sons, p.528, 2005
43. H. Wang, Y. Wang, Q. Hu, Self-adaptive robust nonlinear regression for unknown noise via mixture of Gaussians, Neurocomputing, 235, 274–286, 2017.
44. M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, A multi-step outlier-based anomaly detection approach to network-wide traffic, Inf. Sci., 348, pp. 243–271, 2016.
45. M. Thottan, C. Ji, Anomaly detection in IP networks, IEEE Trans. Signal Process., 51, 8, pp. 2191–2204, 2003.
46. E.M. Knorr, R.T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, VLDB J., 8, 3–4, pp. 237–253, 2000.
47. A. Koufakou, M. Georgiopoulos, A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes, Data Min. Knowl. Discov., 20, 2, pp. 259–289, 2010.
48. F. Shaari, A.A. Bakar, A.R. Hamdan, Outlier detection based on rough sets theory, Intell. Data Anal., 13, 2, pp. 191–206, 2009.
49. J.A.K. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing, 48, pp. 85–105, 2002.
50. W. Wen, Z. Hao, X. Yang, A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression, Neurocomputing, 71, pp. 3096–3103, 2008.
51. X. Chen, J. Yang, J. Liang, Q. Ye, Recursive robust least squares support vector regression based on maximum correntropy criterion, Neurocomputing, 97, pp. 63–73, 2012.
52. X. Yang, L. Tan, L. He, A robust least squares support vector machine for regression and classification with noise, Neurocomputing, 140, pp. 41–52, 2014.
53. G.J. McLachlan, K.E. Basford, Mixture Models: inference and Applications to Clustering, Marcel Dekker, 1988.
54. V. Mazya, G. Schmidt, On approximate approximations using Gaussian kernels, IMA J. Numer. Anal., 16 (1), pp. 13–29, 1996.
55. Y.X. Zhao, X.H. Zhuang, S.J. Ting, Gaussian mixture density modeling of nongaussian source for autoregressive process, IEEE Trans. Signal Process. 43, pp. 894–903, 1995.

56. G. Galimberti, G. Soffritti, A multivariate linear regression analysis using finite mixtures of t distributions, Comput. Stat. Data Anal., 71, pp. 138–150, 2014.
57. C.B., Zeller, C.R.B. Cabral, V.H. Lachos, Robust mixture regression modeling based on scale mixtures of skew-normal distributions, TEST-Springer, 2015.
58. V.G. Cancho, D.K. Dey, V.H. Lachos, M.G. Andrade, Bayesian nonlinear regression models with scale mixtures of skew-normal distributions: estimation and case influence diagnostics, Comput. Stat. Data Anal., 55, pp. 588–602, 2011.
59. A.M. Garay, V.H. Lachos, C.A.A. Valle, Nonlinear regression models based on scale mixtures of skew-normal distributions, J. Korean Stat. Soc., 40, pp. 115–124, 2011.
60. V.H. Lachos, D. Bandyopadhyay, A.M. Garay, Heteroscedastic nonlinear regression models based on scale mixtures of skew-normal distributions, Stat. Probab. Lett., 81, pp. 1208–1217, 2011.
61. Elkan, C. The foundations of cost-sensitive learning. *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 973-978, 2001.
62. R. Goetschalckx, K. Driessens, S. Sanner, Cost-sensitive parsimonious linear regression, Proc. 8th IEEE International Conference on Data Mining (ICDM'08), pp. 809-814, 2008.
63. G. Bansal, A.P. Sinha, H. Zhao, Tuning data mining methods for cost-sensitive regression: a study in loan charge-off forecasting Journal of Management Information Systems, 25 (3), pp. 317-338, 2008.
64. B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, FL (November 2003), pp. 435-442.
65. D. J. C. MacKay, Bayesian Interpolation, Neural Computation, Vol. 4, pp. 415-447, 1992.
66. M. T. Hagan, M.B. Menhaj Training feedforward networks with Marquardt algorithm, IEEE transactions on neural networks, Vol. 5, No. 6, pp. 989 - 993, 1994.
67. F. D. Foresee, and M. T. Hagan, Gauss-Newton approximation to Bayesian regularization, Proceedings of the International Joint Conference on Neural Networks, pp. 1930-1935, 1997.
68. D. Nguyen and B. Widrow, Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, *Proceedings of the IJCNN*, vol. 3, pp. 21–26, 1990.
69. Y. Nesterov. Introductory lectures on convex optimization: a basic course, volume 87 of Applied optimization. Kluwer Academic Publishers, 2004.
70. A. Khrabrov, K. Kolinko, Y. Vinogradov, A. Zhuk, I. Grishin, I. Ignatyev, Experimental investigation and mathematical simulation of unsteady aerodynamic characteristics of a transonic cruiser model at small velocities in a wide range of angles of attack, Visualization of Mechanical Processes: An International Online Journal, 1 (2), 2011.
71. K. S . Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Transactlons on Neural Networks, Vol. I . No. I, 1990.
72. D. I. Ignatyev, A. N. Khrabrov, Neural network modeling of unsteady aerodynamic characteristics at high angles of attack, Aerospace Science and Technology, 41, pp. 106–115, 2015
73. M. T. Hagan, H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996.
74. D. I. Ignatyev, M. E. Sidoryuk, K. A. Kolinko, and A. N. Khrabrov "Dynamic Rig for Validation of Control Algorithms at High Angles of Attack", Journal of Aircraft, Vol. 54, No. 5, pp. 1760-1771, 2017.
75. D. Ignatyev, A. Khrabrov "Experimental study and neural network modeling of aerodynamic characteristics of canard aircraft at high angles of attack", Aerospace, vol. 5, No. 1, 2018.