Struct Multidisc Optim (2008) 37:197–202
DOI 10.1007/s00158-007-0224-1

INDUSTRIAL APPLICATION

Model parameter tuning by cross validation and global optimization: application to the wing weight fitting problem

Humberto Rocha

Received: 7 September 2007 / Revised: 11 November 2007 / Accepted: 12 December 2007 / Published online: 23 April 2008

Abstract Model parameter tuning is a fundamental step in any data-fitting problem and of great importance in the final quality of the resulting approximation. Two different sets of model parameters will lead to two different interpolation models that behave very differently between the data points even if both sets of parameters lead to perfect interpolation. The main goal of this paper is to discuss the importance of finding the optimal parameters that will lead to the best prediction model of the given data. This task can be hard, particularly when the number of model parameters is high (usually when the dimension of the problem is high). The wing weight fitting problem is used to illustrate the difficulties in obtaining the best possible approximation in practice.

Keywords Global optimization · Model parameter tuning · Data fitting · Cross validation · Principal component regression

1 Introduction

The least squares quadratic polynomial fitting (also known as the response surface method), Kriging interpolation, and radial basis function (RBF) interpolation are commonly used for approximation of functions $f(\mathbf{x})$ with more than three variables, where $\mathbf{x}$ is the input vector of n variables $x_1, \ldots, x_n$. All these methods find various approximation models of $f(\mathbf{x})$ based on a set of given input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$ and the corresponding function values $f(\mathbf{x}^1), \ldots, f(\mathbf{x}^N)$.

The response surface method is ideal to capture the local quadratic trend of $f(\mathbf{x})$, where the input vectors $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are clustered around a point of interest. Kriging interpolation and RBF interpolation are based on completely different mathematical concepts, but their numerical calculations are almost identical. They are mainly used for accurate global approximations of $f(\mathbf{x})$.

One fundamental difference between the response surface and Kriging/RBF interpolation is that the former is invariant under scaling while the latter is not. That is, if we scale the input variable x_i by θ_i, compute the least squares quadratic polynomial fitting of the scaled data, and rewrite the fitting polynomial in terms of the original variables, then we get the same least squares polynomial fitting of the original data. For Kriging/RBF interpolation, scaling of the input variables will lead to different interpolants. In other words, for the same set of data, one could generate infinitely many Kriging/RBF interpolants by scaling the input variables differently.

In theory, Kriging interpolation method treats each input variable x_i as an independent random variable with a normal distribution and scales x_i by the reciprocal of its standard deviation (see Sacks et al. 1989). For almost all engineering applications, x_i are deterministic variables with physical meanings. However, the Kriging method will generate a good approximation of $f(\mathbf{x})$ if a sufficient amount of data is available, because it is essentially a RBF interpolant. Theory for RBF interpolation is also based on the availability of a sufficient amount of data (see Buhmann 2003). In general, the

H. Rocha (✉)
Universidade Católica Portuguesa, Pólo de Viseu,
3504-505 Viseu, Portugal
e-mail: hrocha@mat.uc.pt

approximation error approaches zero as more and more data points are used to generate the RBF interpolant, no matter what scaling parameters are used.

In practice, due to the curse of dimensionality and the cost of obtaining data points, there is only a limited amount of data for building an approximation of $f(\mathbf{x})$. In such a case, scaling will have a significant effect on the characteristics of the Kriging/RBF interpolants between and beyond the data points. In statistics, the scaling parameters θ_i are called the model parameters, and the cross validation (CV) (Stone 1974; Efron and Tibshirani 1993) is a standard method to find the optimal model parameters that lead to the approximation model with the least prediction error.

In this paper, an aircraft wing weight approximation problem is used to demonstrate the difficulty of finding the optimal model parameters for RBF interpolants during CV error minimization and to show how model parameters affect approximation models. The paper is organized as follows: Section 2 gives a brief discussion of principal component regression with cross validation. The wing weight data fitting problem is formulated in Section 3, while in Section 4, numerical results are included to show advantages of approximations obtained with optimal model parameters. The paper ends with concluding remarks in Section 5.

2 Principal component regression with cross validation

Principal component regression (PCR) with cross validation was proposed in Rocha et al. (2006). For easy reference, a description of its key elements is given in this section.

Let $\mathbf{x}^1, \ldots, \mathbf{x}^N$ be a given set of input vectors in the n-dimensional Euclidean space $\mathbb{R}^n$ (of column vectors); $f(\mathbf{x})$ is the value of an unknown response at $\mathbf{x} \in \mathbb{R}^n$; $f_k = f(\mathbf{x}^k)$ for $k = 1, \ldots, N$; $\mathbf{x}_i^k$ and $\mathbf{x}_i$ are the ith component of $\mathbf{x}^k$ and $\mathbf{x}$, respectively; σ_i is the standard deviation of the ith component of $\mathbf{x}^1, \ldots, \mathbf{x}^N$, i.e.,

$$\sigma_i = \frac{1}{N-1}\left(\sum_{j=1}^{N}\left(\mathbf{x}_i^j - \text{ave}(\mathbf{x}_i)\right)^2\right)^{\frac{1}{2}},$$

where $\text{ave}(\mathbf{x}_i) = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_i^k$; $\hat{\mathbf{x}}$ (or $\hat{\mathbf{x}}^k$) is the scaled version of $\mathbf{x}$ (or $\mathbf{x}^k$) and its ith component is x_i/σ_i (or x_i^k/σ_i); superscript "T" denotes the transpose of a vector or matrix; $\mathbf{C} = \frac{1}{N-1}\sum_{i=1}^{N}(\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))(\hat{\mathbf{x}} - \text{ave}(\hat{\mathbf{x}}))^T$ denotes the covariance matrix of the scaled input vectors $\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^N$; $\text{ave}(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}^k$ (the average of the given input vectors); $\text{ave}(\hat{\mathbf{x}}) = \frac{1}{N}\sum_{k=1}^{N}\hat{\mathbf{x}}^k$ (the average of the given scaled input vectors); $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_n$ are the eigenvalues of $\mathbf{C}$; $\mathbf{u}^j$ is an eigenvector of $\mathbf{C}$ corresponding to γ_j; r is any integer between 1 and n; and $\mathbf{v}$ (or $\mathbf{v}^k$) is a feature vector in $\mathbb{R}^r$ whose jth component is $\hat{\mathbf{x}}^T\mathbf{u}^j$ (or $(\hat{\mathbf{x}}^k)^T\mathbf{u}^j$) for $1 \leq j \leq r$. One requirement for r is that $\mathbf{v}^1, \ldots, \mathbf{v}^N$ are distinct vectors in $\mathbb{R}^r$.

Note that the mapping $\mathbf{x} \to (\mathbf{u}^1, \ldots, \mathbf{u}^r)^T\mathbf{x}$ is really a projection of the input vectors in a space with a reduced dimension r. If all $(\mathbf{x}^j - \text{ave}(\mathbf{x}))(1 \leq \text{j} \leq \text{N})$ are on a r-dimensional subspace in $\mathbb{R}^n$, then the given set of pairs $(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^N, f_N)$ only provides information on how $f(\mathbf{x})$ changes with respect to change of $(\mathbf{x} - \text{ave}(\mathbf{x}))$ on the r-dimensional subspace, and it is appropriate to use the data for an approximation of $f(\mathbf{x})$ with $(\mathbf{x} - \text{ave}(\mathbf{x}))$ restricted on this r-dimensional subspace. This is the basic concept behind the PCR in the context of approximation theory. Transformation of $f(\mathbf{x})$ leads to the following function in $\mathbb{R}^r$:

$$\hat{f}(\mathbf{v}) = \hat{f}\left(\hat{\mathbf{x}}^T\mathbf{u}^1, \ldots, \hat{\mathbf{x}}^T\mathbf{u}^r\right) = f(\mathbf{x}),$$

whenever $\mathbf{x} - \text{ave}(\mathbf{x})$ is a linear combination of $\mathbf{u}^1, \ldots, \mathbf{u}^r$. In particular, $\hat{f}(\mathbf{v}^k) = f_k = f(\mathbf{x}^k)$ $(1 \leq j \leq N)$.

Next, one finds an approximation $g(\mathbf{v})$ of $\hat{f}(\mathbf{v})$ by using the given set of data $(\mathbf{v}^1, f_1), \ldots, (\mathbf{v}^N, f_N)$. Then, the following approximation relationship

$$f(\mathbf{x}) \approx g\left(\hat{\mathbf{x}}^T\mathbf{u}^1, \ldots, \hat{\mathbf{x}}^T\mathbf{u}^r\right)$$

usually yields a more meaningful approximation of $f(\mathbf{x})$ than constructing an approximation of $f(\mathbf{x})$ from the original data $(\mathbf{x}^1, f_1), \ldots, (\mathbf{x}^N, f_N)$, especially when $\mathbf{x}^1, \ldots, \mathbf{x}^N$ are clustered around a r-dimensional affine space. See Subsection 3.6 of Li and Padula (2005) for details on reformulating the data-fitting problem in the r-dimensional feature space.

For the wing weight data-fitting problem, multiquadric RBF interpolation will be used to construct the approximation function $\hat{g}(\mathbf{v})$ (see Rocha et al. 2006). Let $\varphi(t) = \sqrt{1+t^2}$ denote the standard multiquadric RBF and let $\|\mathbf{v} - \mathbf{v}^j\|$ be the parameterized distance between $\mathbf{v}$ and $\mathbf{v}^j$ defined as

$$||\mathbf{v} - \mathbf{v}^j|| = \left(\sum_{i=1}^{r}|\theta_i|\left(\mathbf{v}_i - \mathbf{v}_i^j\right)^2\right)^{\frac{1}{2}},$$

where θ_i are positive numbers. Then, the RBF interpolant $g(\mathbf{v}) = \sum_{j=1}^{N}\alpha_j\varphi(\|\mathbf{v} - \mathbf{v}^j\|)$ is the unique linear combination of $\varphi(\|\mathbf{v} - \mathbf{v}^j\|)$ $(1 \leq j \leq N)$ that satisfies the following interpolation conditions:

$$\sum_{j=1}^{N}\alpha_j\,\varphi\left(||\mathbf{v}^k - \mathbf{v}^j||\right) = f_k, \quad \text{for } k = 1, \ldots, N.$$

To determine the optimal model parameters $\theta_1, \ldots, \theta_r$ for the RBF interpolant $g(\mathbf{v})$, we use the following

leave-one-out cross validation error minimization procedure:

- Fix a set of parameters $\theta_1, \ldots, \theta_r$.
- For $j = 1, \ldots, N$, construct the RBF interpolant $g_{-j}(\mathbf{v})$ of the $(N-1)$ data points $(\mathbf{v}^k, f_k)$ for $1 \le k \le N, k \ne j$.
- Use the following CV root mean square error as the prediction error:

$$E^{CV}(\theta_1, \ldots, \theta_r) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(g_{-j}(\mathbf{v}^j) - f_j\right)^2}. \qquad (1)$$

- Find the model parameters $\theta_1^*, \ldots, \theta_r^*$ that minimize $E^{CV}(\theta_1, \ldots, \theta_r)$, i.e.,

$$E^{CV}(\theta_1^*, \ldots, \theta_r^*) = \min_{\theta_1, \ldots, \theta_r} E^{CV}(\theta_1, \ldots, \theta_r).$$

Note that the interpolation function $g_{-j}(\mathbf{v}) = \sum_{i=1, i\ne j}^{N} \alpha_i \varphi(\|\mathbf{v} - \mathbf{v}^i\|)$ in the CV procedure can be obtained by solving the following interpolation equations:

$$\sum_{i=1, i\ne j}^{N} \alpha_i \varphi\left(\|\mathbf{v}^k - \mathbf{v}^i\|\right) = f_k \quad \text{for } 1 \le k \le N, k \ne j.$$

It is worth pointing out that it is difficult to minimize $E^{CV}(\theta_1, \ldots, \theta_r)$ numerically because $E^{CV}(\theta_1, \ldots, \theta_r)$ is a highly nonlinear and nonconvex function. One could make the model parameter tuning much easier by assuming $\theta_1 = \cdots = \theta_r$, which reduces the problem to unconstrained minimization of a univariate function (see Tu 2003). This approach has the obvious benefit of dealing with a much easier optimization problem but the disadvantage of not using all different θ_i. Optimization with respect to $\theta_1, \ldots, \theta_r$ allows the model parameter tuning process to scale each component of $\mathbf{v}$ based on its significance in modeling the variance in the response; thus, it facilitates implicit variable screening.

Even knowing that PCR and CV are well known topics and widely used, the combination of both in an automatic procedure proved to be more effective (see Rocha et al. 2006). However, the effectiveness of this procedure depends on the quality of the model parameter obtained on the CV error minimization. Therefore, obtaining better solutions for this global optimization problem is an important aspect to improve.

3 Wing weight data fitting problem

The following wing weight fitting problem is used to demonstrate the difficulties and benefits of finding optimal model parameters. For system analysis of conceptual design of aircraft, an important task is to resize a conceptual aircraft for a mission analysis. To conduct a mission analysis of a resized aircraft, system analysts need to estimate the gross takeoff weight w_{to} of the aircraft. Specifically, one commonly resized component of an aircraft is the wing. As a result, system analysts need a relationship between the wing weight w and sizing parameters of the wing (such as span (b),plan area of the wing (s), taper ratio (λ), and sweep angle (Λ)). The goal is the construction of an approximation $\bar{w} \approx w$ of the relationship between the actual wing weight and various key configuration parameters of the wing by using actual wing weight data of 41 subsonic transports. Such a procedure is called an empirical approach by Ardema et al. (1996). However, system analysts tend to reject the idea of using a general regression or approximation model for wing weight estimation because a general model lacks any engineering insight and usually leads to a nonphysical weight estimation formula that gives negative wing weight or exhibits nonmonotonicity of wing weight vs some key configuration parameters, such as s.

There are many studies (Rocha et al. 2006; Ardema et al. 1996; Keane 2003) on building approximation models for weight estimation. So far, useful weight models are mainly derived from knowledge and insight of experienced engineers instead of rigorous principles of physics. In some cases, useful weight estimation models are considered proprietary information not supposed to be shared with the public. Weight information of existing aircraft is not necessarily available to the public. System analysts at NASA Ames Research Center were able to collect weight information of 41 subsonic transports including Boeing 747, Douglas DC-7C, Fokker F-28 twin-engine jet liner, and Lockheed C-130B cargo aircraft (see Fig. 1).

Each wing weight data point consists of the actual wing weight w and relevant configuration parameters,

Fig. 1 A variety of subsonic transports in the data set

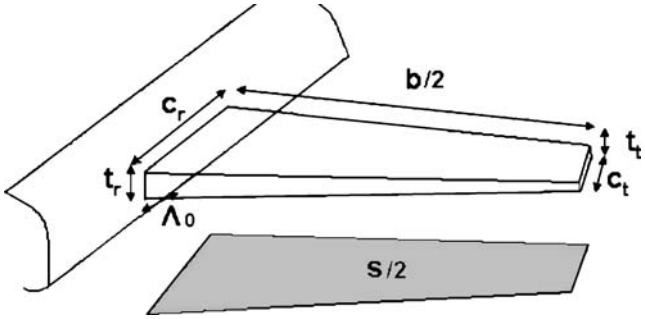

Fig. 2 Airplane wing geometry parameters

including wingspan (b), root chord of wing at fuselage intersection (c_r), tip chord of wing (c_t), plan area of wing (s), thickness of airfoil at fuselage intersection (t_r), gross takeoff weight of aircraft (w_{to}), wing sweep angle in radian (Λ), and ultimate load (μ). Figure 2 shows the wing geometry parameters for a trapezoidal approximation of the actual wing. A detailed explanation of these wing parameters can be found in Raymer (1999).

System analysts are mostly interested in whether an estimation model captures the weight growth trends "correctly" between and beyond the known data points. There is no physics-based criterion for verification of a correct solution; instead, expert opinions determine whether a mathematical solution is useful in practice.

For wing weight estimation, a desirable approximation should have the following properties: w is an increasing function with respect to each of b, s, λ, and Λ, and w is a decreasing function with respect to t_r/c_r. These properties are derived from simple engineering rules on the relationships between the wing weight and each of the five key configuration parameters. There are infinitely many 2D plots to inspect the 2D relationships between an input variable and the wing weight. In practice, we only check the 2D plot between the wing weight and each of the input variables b, s, t_r/c_r, λ, and Λ at each of the input vectors $\mathbf{x}^j$ $(1 \le j \le N = 41)$. That is, each plot is generated by setting all the input variables (except one specified variable among $b, s, t_r/c_r, \lambda$, and Λ) to the corresponding values of $\mathbf{x}^j$.

4 Numerical results

The multiquadric RBF approximation is the most desirable wing weight approximation among several generated in Rocha et al. (2006). The multiquadric RBF approximation has the following form

$$g(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \varphi_j(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \sqrt{1 + \sum_{i=1}^{n} \theta_i \left(x_i - x_i^j\right)^2},$$

Table 1 Minimized CV errors for the data set with eight input variables

	CV error minimization using **fminsearch**	CV error minimization using **PSOt**
$n = 8$	2,697	3,726
$r = 8$	4,321	2,647
$r = 7$	3,065	2,643
$r = 6$	3,724	4,961

where α_j are determined by the interpolation conditions $g(\mathbf{x}^k) = f_k$ $(k = 1, \ldots, N)$. For the wing weight data-fitting problem, we used $n = 8$ (the set of variables used for the best empirical model by Ardema et al. 1996) and $n = 14$ (the set of all variables) as the dimensions of the $\mathbf{x}$ vector (see Rocha et al. 2006). The CV error $E^{CV}(\theta_1, \ldots, \theta_n)$ in (1) was minimized using the MATLAB code **fminsearch** to find the best model parameters $\theta_1^*, \ldots, \theta_n^*$.

The multiquadric PCR with $r = 7$ for the data set with eight variables was the most desirable wing weight approximation among all multiquadric RBF approximations generated and correspond to the smallest final CV error obtained for multiquadric PCR approximations [even though the CV error for the multiquadric fitting without principle component analysis (PCA) is smaller than the multiquadric PCR with $r = 7$, its 2D trends are almost always less desirable than the latter].

MATLAB code **fminsearch**, an implementation of the Nelder–Mead (Nelder and Mead 1965) multidimensional search algorithm, is very reliable for finding local optimal solutions. The local optimal solution generated by MATLAB code **fminsearch** for minimization of the CV error is very sensitive to the initial guess. Multiple initial guesses were used for searching a global minimizer of the CV error by **fminsearch**. However, there was no guarantee that the best solution among the calculated local optimal solutions was good enough

Table 2 Minimized CV errors for the data set with fourteen input variables

	CV error minimization using **fminsearch**	CV error minimization using **PSOt**
$n = 14$	18,585	4,155
$r = 14$	12,755	6,759
$r = 13$	10,711	7,414
$r = 12$	12,540	7,342
$r = 11$	34,515	6,067
$r = 10$	14,235	6,128
$r = 9$	4,653	7,973
$r = 8$	9,497	6,056
$r = 7$	12,425	6,012
$r = 6$	7,242	12,558

(the final CV error value obtained is close enough to the CV error global minimum).

In general, it is difficult to find global minimizers of nonconvex objective functions. Heuristic search methods like simulated annealing, tabu search, and genetic algorithms can be applied to find approximate solutions of global minimizers of the CV error. Lately, one of the most used heuristics and with more success is particle swarm optimization (PSO). PSO is a subset of evolutionary computation such as genetic algorithms (see Kolda et al. 2003). PSO is a heuristic that was developed out of attempts to model different animal behaviors such as bird flocks or fish schools. MATLAB code **PSOt** (MATLAB toolbox) (see Birge 2003) was used to minimize the cross validation error for the set of $n = 8$ variables and for the set of all variables $n = 14$ as an attempt for improving the parameter tuning process.

Table 1 shows the minimized CV errors for multiquadric RBF interpolation models by using **fminsearch** with multiple initial guesses and by using **PSOt** for $n = 8$ and Table 2 shows the results for $n = 14$. The first row of each table shows the fitting result without PCA. That is, RBF interpolation and cross validation error minimization are done with respect to the original input vector **x**.

By a simple inspection of Tables 1 and 2, one can immediately verify that a significant improvement is obtained in the CV error minimization results when **PSOt** is used instead of **fminsearch** for most of the cases. In particular, an improvement in the final CV error was obtained for the previous best approximation (multiquadric PCR with $r = 7$ for the data set with eight variables). That improvement has a correspondence on the behavior of the approximations with respect to the

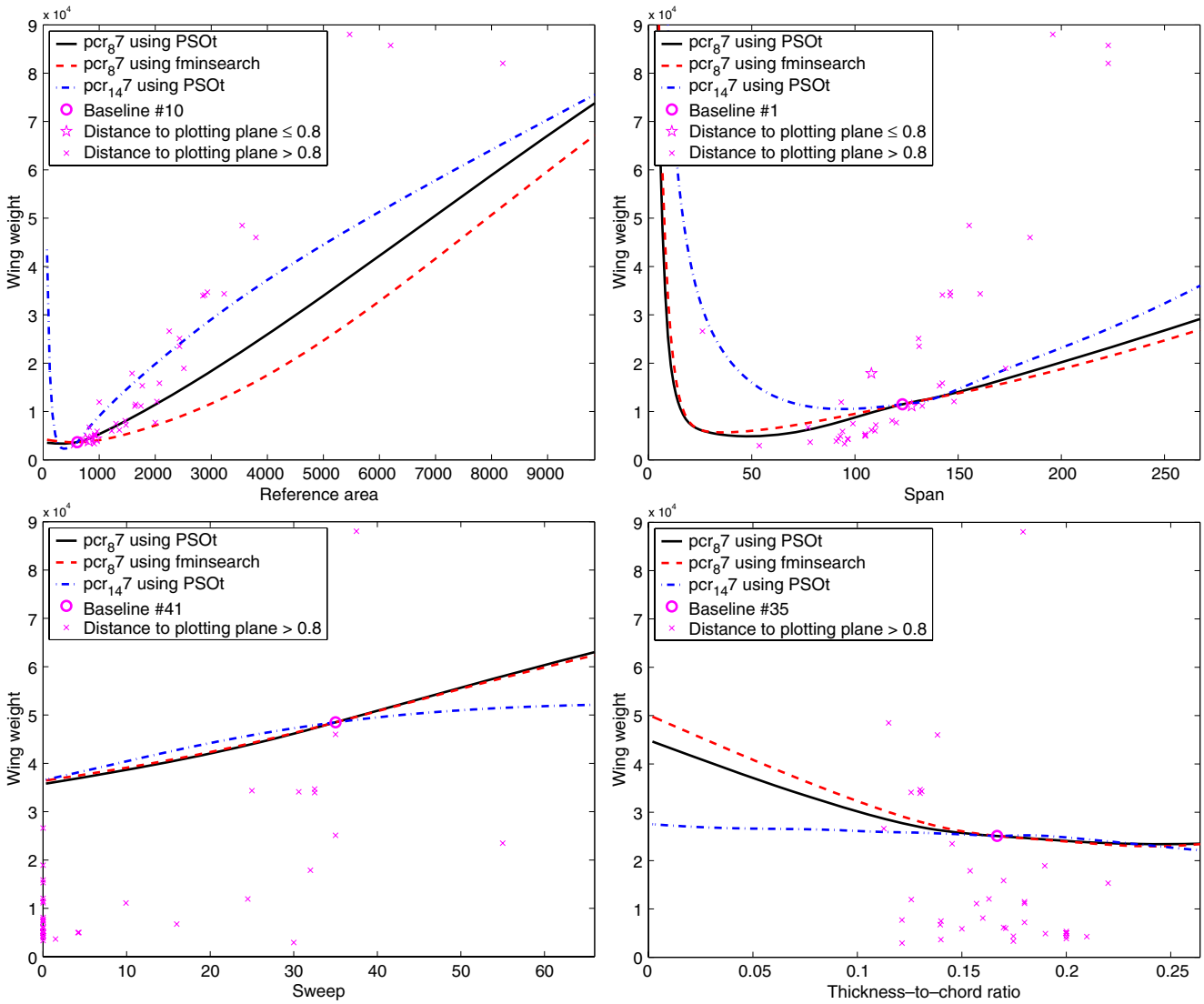

Fig. 3 Two-dimensional plots of best multiquadric PCR approximations

2D plot trends. In fact, the multiquadric PCR with $r = 7$ for the data set with eight variables (using **PSOt** in the minimization process) is now the most desirable wing weight approximation among all the generated approximations. Figure 3 shows that $\text{pcr}_8 7$ ($\text{pcr}_n r$ denotes the PCR approximation in a space with a reduced dimension r using a data set with n variables; thus, $\text{pcr}_8 7$ is the PCR approximation with $r = 7$ for the data set of eight variables) using **PSOt** has better trends than $\text{pcr}_8 7$ using **fminsearch** in the CV error minimization process. The approximation that has better behavior for the 14 variables set ($\text{pcr}_{14} 7$) was also introduced in Fig. 3. It is important to note that if the plan area of the wing is fixed, then the wing configuration becomes unrealistic when the span b approaches zero. As a result, the sharp increase of wing weight as b approaches zero (see Fig. 3) is a very useful weight penalty that automatically prevents a conceptual designer from generating unrealistic wings if the multiquadric PCR approximation is used for wing weight estimation. The baseline data point $\mathbf{x}^j$ is also plotted for perspective, as well as the rest of the points in two groups: (1) distance to the plotting plane ≤ 0.8 and (2) distance to the plotting plane > 0.8.

Without inspecting the 2D plot trends of the different approximations for the $n = 14$ variables set, we suspected already that $\text{pcr}_{14} 7$ was the best approximation for that data set due to the fact of it being one of the approximations with the smallest CV error (see Table 2). This result is another indication of an intrinsic seven degrees of freedom in the data.

5 Concluding remarks

Model parameter tuning is a fundamental step in any data-fitting problem and of great importance in the final quality of the resulting approximation. Two different sets of model parameters will lead to two different interpolation models that behave very differently between the data points even if both sets of parameters lead to perfect interpolation. Global optimization is a very important tool when CV is used for parameter tuning. The performance of the approximation between data points depends on the quality of the output parameters of the CV error minimization procedure.

Finding the optimal parameters that will lead to the best prediction model of the given data can be hard, particularly when the number of model parameters is high (usually when the dimension of the problem is high). The wing weight fitting problem was used to illustrate the difficulties on obtaining the best possible approximation in practice. Among all the heuristic search methods used, particle swarm optimization proved to be the more effective to obtain the best parameter values during the parameter tuning process. Even knowing that this result concerns only the wing weight data-fitting problem, it confirms the growing reputation of particle swarm optimization in finding global minimizers of nonconvex objective functions in high dimensional spaces.

The wing weight fitting problem is a real-world problem that illustrate the need of global optimization for obtaining the best possible approximation when using CV. For our data set, the numerical results show that an improvement of the minimization process leads to an improvement of the best approximation, as well as the confirmation of the indication of an intrinsic seven degrees of freedom in the data. Even though the correspondence between CV minimization improvement and best approximation improvement are only demonstrated by the wing weight data-fitting problem, the result can lead to significant advantages in fitting other historical or sparse data when approximation effectiveness is harder to verify.

References

Ardema M, Chambers M, Patron A, Hahn A, Miura H, Moore M (1996) Analytical fuselage and wing weight estimation of transport aircraft. NASA Technical Memorandum 110392

Birge B (2003) PSOt, a particle swarm optimization toolbox for matlab. In: IEEE swarm intelligence symposium proceedings, Indianapolis, 24–26 April 2003

Buhmann M (2003) Radial basis functions: theory and implementations. Cambridge University Press, Cambridge

Efron B, Tibshirani RJ (1993) An introduction to the Bootstrap. Chapman & Hall, London

Keane A (2003) Wing optimization using design of experiments, response surface, and data fusion methods. J Aircr 40: 741–750

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new prespectives on some classical and modern methods. SIAM Rev 45:385–482

Li W, Padula S (2005) Approximation methods for conceptual design of complex systems. In: Chui C, Neamtu M, Schumaker L (eds) Approximation XI. Nashboro, Brentwood, pp 241–278

Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7:308–313

Raymer D (1999) Aircraft design: a conceptual approach, 3rd edn. AIAA, Reston

Rocha H, Li W, Hahn A (2006) Principal component regression for fitting wing weight data of subsonic transports. J Aircr 43:1925–1936

Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments. Stat Sci 4:409–423

Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J R Stat Soc 36:111–147

Tu J (2003) Cross-validated multivariate metamodeling methods for physics-based computer simulations. In: Proceedings of the IMAC-XXI, Kissimmee, 3–6 February 2003