A new Bayesian recursive technique for parameter estimation

Yasir H. Kaheil,[1] M. Kashif Gill,[1] Mac McKee,[1] and Luis Bastidas[1]

Received 22 August 2005; revised 1 May 2006; accepted 16 May 2006; published 16 August 2006.

[1] The performance of any model depends on how well its associated parameters are estimated. In the current application, a localized Bayesian recursive estimation (LOBARE) approach is devised for parameter estimation. The LOBARE methodology is an extension of the Bayesian recursive estimation (BARE) method. It is applied in this paper on two different types of models: an artificial intelligence (AI) model in the form of a support vector machine (SVM) application for forecasting soil moisture and a conceptual rainfall-runoff (CRR) model represented by the Sacramento soil moisture accounting (SAC-SMA) model. Support vector machines, based on statistical learning theory (SLT), represent the modeling task as a quadratic optimization problem and have already been used in various applications in hydrology. They require estimation of three parameters. SAC-SMA is a very well known model that estimates runoff. It has a 13-dimensional parameter space. In the LOBARE approach presented here, Bayesian inference is used in an iterative fashion to estimate the parameter space that will most likely enclose a best parameter set. This is done by narrowing the sampling space through updating the ''parent'' bounds based on their fitness. These bounds are actually the parameter sets that were selected by BARE runs on subspaces of the initial parameter space. The new approach results in faster convergence toward the optimal parameter set using minimum training/calibration data and fewer sets of parameter values. The efficacy of the localized methodology is also compared with the previously used BARE algorithm.

Citation: Kaheil, Y. H., M. K. Gill, M. McKee, and L. Bastidas (2006), A new Bayesian recursive technique for parameter estimation, *Water Resour. Res.*, *42*, W08423, doi:10.1029/2005WR004529.

1. Introduction

[2] Accurate state estimates in hydrology can be difficult to obtain due to the uncertainties in the parameters, data, and the model structure. One way to improve such estimates is through proper calibration of models by estimating the best parameter set. Another important concept that has been well addressed by *Beven* [1993] and *Savenije* [2001] is the notion of ''equifinality.'' There can be multiple parameter sets that can produce equally good model performance. Hence the aim of calibration is to obtain at least one range of parameter sets within which is included a ''best'' parameter set. Previous attempts to do this have focused on ''batch'' techniques that require a great deal of data and substantial computational effort and computer time to determine optimal parameter ranges. An example of batch calibration procedures is the shuffled complex evolution (SCE-UA) algorithm by *Duan et al.* [1992] that employs downhill simplex and complex evolution strategies. This algorithm was further extended to account for the multiobjective functions in multiobjective complex evolution (MOCOM-UA) by *Yapo et al.* [1998] and *Gupta et al.* [1998] and multiobjective shuffle complex evolution Metropolis (MOSCEM-UA) by *Vrugt et al.* [2003]. The primary approach used by these earlier efforts has been the ''batch'' calibration approach. This approach assumes the model parameter values to be time invariant and typically requires that considerable data be collected before the procedure can be implemented to arrive at ''optimal'' estimates for those parameters. Only after the model has been calibrated and tested can it be employed to make predictions [*Thiemann et al.*, 2001].

[3] *Freer et al.* [1996] used a Bayesian approach for evaluating the predictive uncertainties for TOPMODEL parameters, known as generalized likelihood uncertainty estimation (GLUE). The GLUE approach employs an online technique in contrast to batch techniques. The GLUE approach weighs the combination of models and parameter sets. Such a combination is termed a ''model'' in the GLUE parlance. Weights are used to formulate a cumulative distribution of predictions from which uncertainty quantiles can be calculated [*Freer et al.*, 1996]. (Readers interested in the details of the calculation of weights are referred to *Freer et al.* [1996].) *Thiemann et al.* [2001] introduced an alternative parameter estimation approach known as Bayesian recursive estimation (BARE), which is based on the solid ground of Bayesian inference. Nevertheless, *Beven and Young* [2003] argue that both the GLUE and BARE methodologies are on two extremes of the range of possible Bayesian approaches with respect to their treatment of different types of errors, e.g., model structure error, input error, etc. *Beven and Young* [2003] further emphasize that in the BARE methodology the most critical issue is the implicit assumption

[1]Department of Civil and Environmental Engineering and Utah Water Research Laboratory, Utah State University, Logan, Utah, USA.

0043-1397/06/2005WR004529

that the model structure is correct, which leads BARE to effectively converge to a single parameter set with very little parameter uncertainty. To overcome this problem, we introduce a few modifications to the BARE algorithm.

[4] The idea is to avoid converging to a single parameter set by looking for a plausible parameter space defined by the best parameter sets obtained through various BARE runs, which we call the "initial BARE runs." Further, an evolutionary genetic-algorithm-like approach is used to define the bounds of the parameter space by choosing the best two parameter sets (parents) resulting from the initial BARE runs. In an iterative approach, the algorithm continues to look for the best parents, thus "localizing" the search further; hence the name localized Bayesian recursive estimation (LOBARE.) The two best parameter sets are defined as the ones with the best goodness of fit values. To avoid convergence to a local optimum parameter space, a "mutation" process can be randomly applied. Another very important aspect that has been addressed here is that a small number of sets of parameter values (say 40 versus 10,000) sampled from the parameter space is required to assure the desired convergence. This will minimize the computational cost that has been one of the drawbacks in BARE.

[5] Another very important issue is that, due to the complex nature of large physical or conceptual models and their large set of parameters that must be estimated, acquisition of sufficient data to populate them can be a difficulty. The problem becomes worse when the data are scarce, which is usually the case. For example, according to *Yapo et al.* [1996] the Sacramento soil moisture accounting (SAC-SMA) model requires 8 years of data for the calibration of its 13 parameters before it can be used for predictions. On the other hand, models derived from methods developed in the field of Artificial Intelligence (AI) generally have smaller parameter sets, which results in less difficulty in parameter estimation. Advancements in statistical learning theory (SLT) by *Vapnik* [1995, 1998] have resulted in a new machine-learning tool called support vector machines (SVMs). The SVM was initially developed for solving classification problems but was later extended to regression problems. The idea in using SVMs for regression is to find a function as flat as possible that can fit the data. In order to address nonlinearity of the problem, the data are mapped into a higher-dimensional space, called the "feature space," using a nonlinear mapping function. Dot product operations have to be done in the feature space between the mapped vectors. However, explicit computation of the dot products is prohibitive due to the high dimensionality of the feature space. To address this difficulty, kernel functions are introduced as a replacement for both the mapping into feature space and the dot product calculations. SVMs have recently been applied in hydrology with great success [e.g., *Dibike et al.*, 2001; *Liong and Sivapragasam*, 2002; *Asefa and Kemblowski*, 2002; *Gill et al.*, 2003, 2006; *Asefa et al.*, 2004; *Khalil*, 2005; *Khalil et al.*, 2005a, 2005b, 2005c, 2006]. For detail on the mathematics and development of SVMs, readers are directed to *Vapnik* [1995, 1998], *Asefa et al.* [2004]; *Gill et al.* [2006], and *Khalil* [2005].

[6] The paper is organized as follows. The theoretical background on BARE and LOBARE is provided in section 2. The case studies, results, and discussion are presented in section 3, and conclusions are drawn in section 4.

2. Theoretical Background

2.1. Localized Bayesian Recursive Estimation

[7] Before we present the details of the LOBARE algorithm, we will provide the fundamental equations that explain Bayesian inference. Consider the following model equation:

$$y_{T+1} = \eta(\xi\backslash\theta) + \varepsilon_{T+1} \tag{1}$$

where y_{T+1} is the observed output, $\eta(\xi\backslash\theta)$ represents model prediction denoted by $\hat{y}_{T+1}$, for the time step $T + 1$, η denotes the model structure with $\boldsymbol{\xi}$ being a vector of inputs, and θ is the parameter set. $\varepsilon_{T+1} = y_{T+1} - \hat{y}_{T+1}$ is the residual error.

[8] It is reasonable to impose a uniform distribution as a priori on the parameter space to start with, in case there is no prior information about the distribution. The a priori density function is termed as $p(\theta)$. The probability distributions of the model outputs are conditional distributions. Hence $p(y_{T+1}, \mathbf{y}\backslash\boldsymbol{\xi}; \theta)$ is the conditional probability density of the outputs y_o through y_{T+1}, given the inputs and the parameters. Now, the conditional density of the outputs and the model parameters, given only the inputs, is defined by

$$p(y_{T+1}, \mathbf{y}; \theta\backslash\boldsymbol{\xi}) = p(y_{T+1}, \mathbf{y}\backslash\boldsymbol{\xi}; \theta)p(\theta) \tag{2}$$

The posterior density of the outputs and the model parameters given all the inputs and the observations at T is given by Bayes rule

$$p(y_{T+1}; \theta\backslash\boldsymbol{\xi}, \mathbf{y}) = Cp(y_{T+1}, \mathbf{y}; \theta\backslash\boldsymbol{\xi}) \tag{3}$$

where

$$C^{-1} = \int p(y_{T+1}, \mathbf{y}; \theta\backslash\boldsymbol{\xi})d(\mathbf{y}) \tag{4}$$

The marginal density can be obtained by integrating the joint density of the variables that are not of interest. Using this, we can write prediction and calibration equations as follows. The marginal posterior density of y_{T+1} is

$$p(y_{T+1}\backslash\boldsymbol{\xi}, \mathbf{y}) = \int_{\Theta} p(y_{T+1}; \theta\backslash\boldsymbol{\xi}, \mathbf{y})d\theta \tag{5}$$

Similarly, the calibration equation is

$$p(\theta\backslash\boldsymbol{\xi}, \mathbf{y}) = \int_{\Theta} p(y_{T+1}; \theta\backslash\boldsymbol{\xi}, \mathbf{y})dy_{T+1} \tag{6}$$

A highest posterior density (HPD) region for this density describes a set of plausible models. A subset $\boldsymbol{R}$ of the domain of p is called an **HPD** region of content $1 - \alpha$ if $P(R) = 1 - \alpha$ and $p(y_1) \geq p(y_2)$ for any $y_1 \in R$ and $y_2 \notin R$ [*Thiemann et al.*, 2001]. Next, we will explore how to use Bayesian inference to make use of new observations. So far,

($\boldsymbol{\xi}$, $\mathbf{y}$)are inputs and measurements until time T. The marginal posterior density of θ is given by equation (6),

$$p(\theta\backslash\boldsymbol{\xi},\mathbf{y})p(y) = p(\mathbf{y};\theta\backslash\boldsymbol{\xi}) = p(\mathbf{y}\backslash\boldsymbol{\xi};\theta)p(\theta)p(\theta\backslash\boldsymbol{\xi},\mathbf{y}) \propto p(\mathbf{y}\backslash\boldsymbol{\xi};\theta)p(\theta) \tag{7}$$

New information available about system outputs at each time step is incorporated according to the following:

$$p\big(\theta\backslash\xi_{T+1},\boldsymbol{\xi},y_{T+1},\mathbf{y}\big) \propto p\big(y_{T+1},\mathbf{y}\backslash\xi_{T+1},\boldsymbol{\xi};\theta\big)p(\theta) \tag{8}$$

Now assuming y_{T+1} and y_T are independent

$$\begin{aligned} p\big(y_{T+1},\mathbf{y}\backslash\xi_{T+1},\boldsymbol{\xi};\theta\big) &= p\big(y_{T+1}\backslash\xi_{T+1},\boldsymbol{\xi};\theta\big)p(\mathbf{y}\backslash\boldsymbol{\xi};\theta)p(\theta) \\ &\propto p\big(y_{T+1}\backslash\xi_{T+1},\boldsymbol{\xi};\theta\big)p(\theta\backslash\boldsymbol{\xi},\mathbf{y}) \end{aligned} \tag{9}$$

Therefore equation (8), becomes

$$p\big(\theta\backslash\xi_{T+1},\boldsymbol{\xi},y_{T+1},\mathbf{y}\big) \propto p\big(y_{T+1}\backslash\xi_{T+1},\boldsymbol{\xi};\theta\big)p(\theta\backslash\boldsymbol{\xi},\mathbf{y}) \tag{10}$$

which requires a recursive form to update θ. Next, we will follow the approach of *Thiemann et al.* [2001] for the selection of the error model. Assuming that the error model looks like

$$v = g(y) - g(\hat{y}) \tag{11}$$

where $g(y)$ represents a transformed space, and assuming an exponential power density function, $E(\sigma,\beta)$, as described by *Box and Tiao* [1973],

$$p(v\backslash\sigma,\beta) = \omega(\beta)\sigma^{-1}\exp\Big[-c(\beta)|v/\sigma|^{2/(1+\beta)}\Big] \tag{12}$$

$$c(\beta) = \left\{\frac{\Gamma[3(1+\beta)/2]}{\Gamma[(1+\beta)/2]}\right\}^{1/(1+\beta)}$$

$$\omega(\beta) = \frac{\{\Gamma[3(1+\beta)/2]\}^{1/2}}{(1+\beta)\{\Gamma[(1+\beta)/2]\}^{3/2}}$$

β is a shape factor $(-1,1]$ giving the measure of kurtosis indicating the "nonnormality" of the parent population. $\beta = 0$ gives a normal density, $\beta = 1$ gives a double exponential density, and when β approaches -1 the density becomes uniform. $\sigma > 0$ is the unknown standard deviation, assumed to be constant with respect to time.

[9] The recursive form is derived here as by *Thiemann et al.* [2001] and *Box and Tiao* [1973]. The maximum likelihood estimate of σ_t at the current time step is given as

$$\hat{\sigma}_T(\theta)^{2/(1+\beta)} = 1/T\frac{2c(\beta)}{(1+\beta)}\sum_{t=1}^{T}|v_t(\theta)|^{2/(1+\beta)} \tag{13}$$

The recursive form of the above equation is given as

$$\hat{\sigma}_T(\theta)^{2/(1+\beta)} = \frac{T-1}{T}\hat{\sigma}_{T-1}(\theta)^{2/(1+\beta)} + 1/T\frac{2c(\beta)}{(1+\beta)}\sum_{t=1}^{T}|v_t(\theta)|^{2/(1+\beta)} \tag{14}$$

The next step is to substitute equation (14) into equation (12), giving the posterior density of θ:

$$p(\theta\backslash\boldsymbol{\xi},z_{T+1},\mathbf{z};\beta) \propto N_T(\theta)p(\theta\backslash\boldsymbol{\xi},\mathbf{z};\beta) \tag{15}$$

$$N_T(\theta) = \frac{1}{\hat{\sigma}_T(\theta)}\exp\left[-c(\beta)\left|\frac{v_T(\theta)}{\hat{\sigma}_T(\theta)}\right|^{2/(1+\beta)}\right] \tag{16}$$

The above posterior density will converge to the optimal parameter set as more data are employed.

2.2. LOBARE Algorithm

[10] The BARE algorithm will converge to a single set of parameter values from the sampled sets. The whole idea behind LOBARE is to avoid doing this. LOBARE starts by sampling a number of sets from $\frac{1}{N_1}$th of the initial plausible parameter space. At the end of N_1 separate BARE runs, N_1 sets of parameter values would have been selected by BARE. The best of these N_1 sets are used to provide the upper and lower bounds on a new parameter space that is a proper subset of the initial space.

[11] It is important to note that the number of sets of parameter values selected by BARE that will be used to define the next parameter space (the "parent parameter sets") does not have to be the same as N_1. In fact, in both applications reported in this paper, two (out of N_1) sets were used to redefine the parameter search space. Selection of parents is based on a goodness of fit measure that is preferably different from the error function used in the BARE algorithm.

[12] Reproducing the next population out of the N_1 sets of parameter values does not strictly follow the conventional rules of genetic algorithms. Instead of implementing a crossover process, the new population is sampled from the whole space. Crossover would result in sampling sets on the edges of the newly defined parameter space, yet sampling from a space enclosed by these edges guarantees a better coverage of the parameter space. The newly defined space is then subdivided into N_2 parts, and BARE is run again on each of them. This produces N_2 new sets of parameter values. For a high-dimensional parameter space, the initial number of subspaces, N_1, used for the first LOBARE run might be larger than the number of subdivisions used in the next (N_2) and subsequent iterations in order to provide better coverage of the entire plausible parameter space. In addition, narrowing the parameter space was performed in this way to help assure convergence.

[13] The way we divide a parameter space is by bisecting the dimensions of the most sensitive parameters. To determine the level of sensitivity of a parameter, a generalized sensitivity analysis (GSA) [*Spear and Hornberger*, 1980; *Bastidas et al.*, 1999] could be done before running LOBARE; however for the second application in this paper these parameters were chosen arbitrarily. The number of subspaces will be 2^D where D is the number of parameter dimensions that were chosen to be bisected. We recommend that D does not exceed 4 to reduce the computational effort. A typical value for N_1 for the first run is 8 (or 2^3) for parameter spaces with three or more dimensions. For subsequent iterations, $N_k = 2$ ($k = 2, \ldots,$). For easier implementation of LOBARE, we believe, increasing the

number of parameter sets in the first run could serve just as well as increasing the initial number of partitions N_I. However, one has to be sure of the efficiency of the random sampler used. A recommended sampler that was used for both of the applications reported in this paper is the Latin hypercube.

[14] Mutation on the boundaries of the current parameter space is triggered when a random number is less than a preset mutation probability. The mutation amount is expressed in terms of a fraction (of the new range) by which the edges of the newly defined parameter space should be offset outwardly. A typical value for the mutation probability parameter, *MP*, is 0.2, and a typical mutation fraction amount σ is 0.1. With the use of these typical numbers, a parameter space will be 1.2^M bigger than a parameter space that was never mutated. *M* is the dimension of the parameter space.

[15] The LOBARE algorithm has three main modules. The first module is the preparation and initialization of different parameters of LOBARE. The second module runs BARE and gives the initial guess about the parameter density. The last module is about updating and refinement.

2.2.1. Preparation

[16] 1. Pick the system model, transformation model $z = g(y)$, and error model $v \sim E(\sigma, \beta)$.

[17] 2. Select a goodness of fit measure, *F*, that will be used to validate the parameter set obtained after each BARE run. *F* might be the same as the error model, *v*, but in order to eliminate biasness, it should not be.

[18] 3. Define the number of initial BARE runs, *N*, which are required to define the parameter space in the next LOBARE iteration. A minimum yet sufficient value of *N* is 2.

[19] 4. Define the mutation probability, *MP*, according to which we will impose uncertainty on bounds of the parameter space. As stated earlier, a typical value of *MP* is 0.2.

[20] 5. Define the LOBARE iteration, *IT*, and the maximum number of LOBARE iterations, *MaxIT*, required to halt the algorithm.

2.2.2. Initialization

[21] 1. Set $N = 1$, $IT = 1$, $MaxIT > N$;

[22] 2. Define $\Theta \subset \Re^M$ according to initial lower and upper bounds on the parameters.

[23] 3. Set *MP* between [0, 1].

2.2.3. BARE

[24] The BARE algorithm has three main steps: initialization, prediction, and updating.

2.2.3.1. Initialization

[25] 1. Set T = 0, Initialize the prior $p(\theta^i \setminus \boldsymbol{\xi}, \mathbf{z}; \beta) = p_o(\theta)$ and $\hat{\sigma}_o(\theta^i) = \hat{\sigma}_o$ for each parameter set ($i = 1, \ldots, n$).

[26] 2. Select the kurtosis parameter, β, and the initial estimate of $\hat{\sigma}_o$.

[27] 3. Divide the plausible parameter space $\Theta \subset \Re^M$ into *N* divisions by bisecting sensitive parameter dimensions. Notice that $N \in \{2, 2^2, 2^3, \ldots, 2^D\}$ where *D* is the number of selected dimensions to be bisected. Also notice that *N* for the first run could be different from the next runs.

[28] 4. Define the prior probability density $p_o(\theta)$ for θ (uniform) and sample $n \times N$ parameter sets from this distribution. Notice that the sampled parameter sets will be sufficient to run the BARE algorithm *N* times. $\Theta_1, \Theta_2, \ldots, \Theta_N \subset \Theta$ and ranges of $any(\Theta_1, \Theta_2, \ldots, \Theta_N) \cap any(\Theta_1, \Theta_2, \ldots, \Theta_N) = \Phi$. In other words the ranges of these groups do not overlap.

2.2.3.2. Prediction

[29] 1. Compute the transformed output $\hat{z}_{T+1}(\theta^i) = g(\eta(\theta^i) \setminus \xi)$ for each parameter set ($i = 1, \ldots, n$). Sort the outputs, i.e., compute $j = 1, \ldots, n = \text{sort}\{i = 1, \ldots, n\}$, such that $\hat{z}_{T+1}(\theta^j) \geq \hat{z}_{T+1}(\theta^{j-1})$

[30] 2. Next, compute the cumulative distribution function (CDF) of the predicted output in the transformed space $p(z_{T+1} \leq a \setminus \boldsymbol{\xi}, \mathbf{z}, \beta) = \sum_{j=1}^{k} p(\theta^j \} \boldsymbol{\xi}, \mathbf{z}, \beta)$, where $\hat{z}_{T+1}(\theta^k) \leq a \leq \hat{z}_{T+1}(\theta^{k+1})$

[31] 3. Define highest posterior density (HPD) region by computing the percentiles for $\hat{z}_{T+1}$ and transform them back to the original output.

[32] 4. Increase the prediction uncertainty to account for the structural error and as yet unobserved output measurement error as follows: (1) Define output region of interest; find indices $[l, u]$ such that $\hat{z}_{T+1}(\theta^l) = \min\{\hat{z}_{T+1}(\theta^i), i = 1, \ldots, n\}$ and $\hat{z}_{T+1}(\theta^u) = \max\{\hat{z}_{T+1}(\theta^i), i = 1, \ldots, n\}$. (2) Define upper and lower output values $[a_{T+1}^{\min}, a_{T+1}^{\max}]$ such that $a_{T+1}^{\min} = \hat{z}_{T+1}(\theta^l) - 2\hat{\sigma}_T(\theta^l)$ and $a_{T+1}^{\max} = \hat{z}_{T+1}(\theta^u) + 2\hat{\sigma}_T(\theta^u)$. (3) Next, define a discrete set of n_a uniformly spaced sampling points *b* on the output range (e.g., $n_a = 100$) such that $b = \{a_{T+1}^{\min} + \frac{a_{T+1}^{\max} - a_{T+1}^{\min}}{n_a - 1}(m - 1), m = 1, \ldots, n_a\}$.

[33] 5. Compute the probability density of the as yet unobserved output measurement in the transformed space. $p(\tilde{z}_{T+1} = b_k \setminus \boldsymbol{\xi}, \mathbf{y}) = C\sum_{i=1}^{n} N_{T+1}(\theta \setminus \boldsymbol{\xi}, \hat{\sigma}_T, \beta)|_{z_{T+1} = bk}$. C is a normalizing constant.

[34] 6. Compute the CDF of the as yet unobserved output measurement in the transformed space. $p(\tilde{z}_{T+1} \leq b_k \setminus \boldsymbol{\xi}, \mathbf{y}) = \sum_{j=1}^{k} p(z_{T+1} = b_k \setminus \boldsymbol{\xi}, y)$

[35] 7. Define the highest posterior density (HPD) region by computing the percentiles for $\tilde{z}_{T+1}$ and transform them back to the original output.

2.2.3.3. Updating

[36] 1. Once the observation of y_{T+1} becomes available, find the transformed value as $z_{T+1} = g(y_{T+1})$. Update the estimates of the error model variance $\hat{\sigma}_{T+1}(\theta^i)$ for $i = 1, \ldots, n$ according to equation (14).

[37] 2. Compute the posterior parameter density $p(\theta^i \} \boldsymbol{\xi}, \mathbf{z}, z_{T+1}) = CN_{T+1}(\theta \setminus \boldsymbol{\xi}, \hat{\sigma}_{T+1}, \beta)|_{zT+1}\, p(\theta^i \setminus \boldsymbol{\xi}, \mathbf{y})$ for $i = 1, \ldots, n$ and use it as the prior distribution for the next time step.

[38] 3. Set T = T+1 and resume with the prediction step, until all the available measurements are consumed.

2.2.4. Update and Refinement

[39] 1. Set $IT = IT + 1$.

[40] 2. Once all the measurements are consumed, the output is the posterior parameter density, which will give the best parameter set. Compute *F* for the best parameter set against the calibration set.

[41] 3. Repeat BARE step (section 2.2.3) until $IT = N$.

[42] 4. Sort parameter sets according to corresponding *F* values, such that $F|\theta'_1 \leq F|\theta'_2 \leq \ldots \leq F|\theta'_N$, where lower values of *F* indicate better goodness of fit against the calibration set. Also notice that θ'_i, $i = 1, 2, \ldots, N$ are the best parameter sets to which BARE converged in each BARE run on Θ_i, $i = 1, 2, \ldots, N$ respectively.

[43] 5. Select the best $NP \leq N$ parameter sets according to their corresponding *F* values. *NP* denotes the number of parents used to generate the next population. A typical *NP* value is 2.

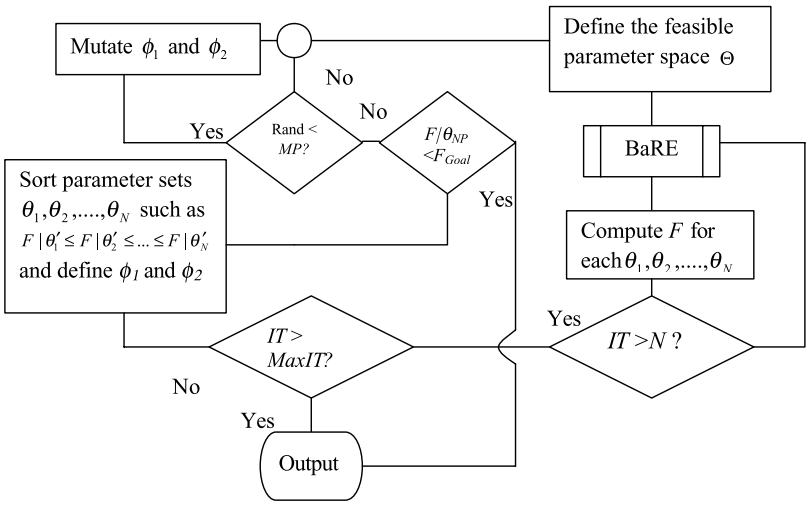

Figure 1. Flowchart for LOBARE.

[44] 6. Mutate $\theta'_1, \theta'_2, \ldots, \theta'_N$ values if a uniform random number is less than *MP*, such as $\phi_1 = \min(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi}) = [\min(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi}) - \sigma.range(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi})] \forall i = \{1, \ldots, q\}$, $\phi_2 = \max(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi}) = [\max(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi}) + \sigma.range(\theta_{1i}, \theta_{2i}, \ldots, \theta_{NPi})] \forall i = \{1, \ldots, q\}$; where q is the length of the parameter set θ and σ is a fraction to define the extent to which the new range will be stretched. A typical σ value used in the applications reported here is 0.1. In case the uniform random number was greater than *MP*, $\sigma = 0$. Higher values of σ and *MP* result in slow but effective convergence, while lower values may result in converging to local optima in high-dimensional parameter spaces.

[45] 7. Check if $F|\theta'_{NP} \le F_{GOAL}$ is true to halt the algorithm and output the new parameter space calculated in the previous step. Mutation in this case should not occur.

[46] 8. Define Θ using ϕ_1 and ϕ_2.

[47] 9. Repeat BARE step (section 2.2.3) until *IT* = *MaxIT*. A flowchart diagram of the LOBARE algorithm is provided in Figure 1.

3. Results and Discussion

3.1. Application on Soil Moisture

[48] As the first illustration of the use of the LOBARE approach, a support vector machine model for the prediction of soil moisture was tested. There are three parameters associated with the SVM model that need to be calibrated (namely, the trade-off, C, between model complexity and goodness of fit, the error tolerance ε, and the kernel parameter γ). SVMs have recently been applied in hydrology with great success [e.g., *Dibike et al.*, 2001; *Liong and Sivapragasam*, 2002; *Asefa and Kemblowski*, 2002; *Gill et al.*, 2003, 2006; *Asefa et al.*, 2004; *Khalil*, 2005; *Khalil et al.*, 2005a, 2005b, 2005c, 2006]. For detail on the mathematics and development of SVMs, readers are directed to *Vapnik* [1995, 1998], *Asefa et al.* [2004]; *Gill et al.* [2006], and *Khalil* [2005]. In the following section, the LOBARE approach is used to obtain the best parameter set.

3.2. Description of the Study Area and Data

[49] The plausibility of the proposed models is evaluated using data from the Soil Climate Analysis Network (SCAN). The Soil Climate Analysis Network was created by the U.S. Department of Agriculture for the purpose of providing a soil and climate data bank. There are more than 90 SCAN stations all over the U.S. at which daily and hourly measurements for meteorological and soil moisture data are obtained using various sensors and instruments. The data in this study were taken from the SCAN site located at the Little Washita River Experimental Watershed (LWREW) in Southwestern Oklahoma in the Southern Great Plains region of the United States (see Figure 2). The Little Washita watershed has an area of 611 km^2. It has been a center of soil-water-related research activities for many decades, including the 1936 soil erosion control project, the USDA Agricultural Research Service (ARS) data collection effort in 1961, and the remote sensing experiments of NASA-USDA (T. J. Jackson, Southern Great Plains 1997 (SGP97) Hydrology Experiment plan, available at http://hydrolab.arsusda.gov/~tjackson). The area of the LWREW is about 60% rangeland, 20% cropland, and 20% miscellaneous (riparian, forest, urban, etc.). LWREW has a wide range of soil types with fine sand, loamy fine sand, fine sandy loam, loam, and silty loams being the predominant soil surface textures [*Allen and Naney*, 1991].

[50] The SVM model is trained to predict soil moisture at time step t+5 (t is in days), an approach similar to that used by *Gill et al.* [2003] and *Khalil et al.* [2005b]. The inputs to the models are soil moisture and meteorological data (relative humidity, average solar radiation, soil temperature at depths of 5 cm and 10 cm, air temperature, and wind speed) at time steps t-1 and t. The output is the soil moisture value at t+5. The training set in the previous study [*Khalil et al.*, 2005b] was 600 daily inputs, whereas in the case discussed here, only about 150 daily inputs were sufficient to determine the correct parameter range. The data window from 1 January 1999 to 31 July 2002 was selected. The results that follow in the next section are shown for the 352 days starting 14 May 2001 (test sample size), when the

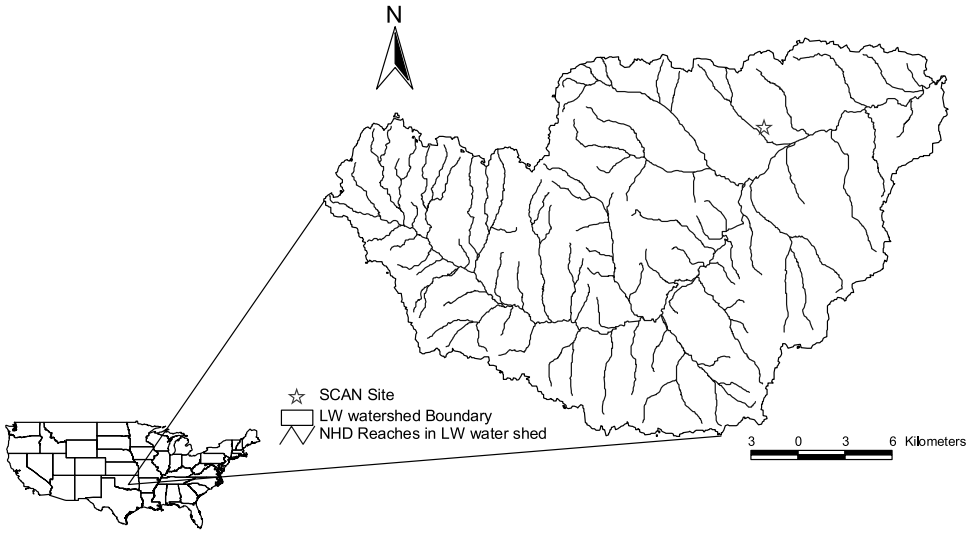

Figure 2. Little Washita River Experimental Watershed (LWREW) along with one Soil Climate Analysis Network station.

LOBARE was used. It is also important to mention that the results that follow in the next section are for soil moisture values that have been normalized between −1 and 1.

3.3. Soil Moisture Results

[51] The performance of the localized strategy is shown in Figure 3 where soil moisture prediction results are plotted along with the observed soil moisture data. The prediction curves follow the observations closely. The algorithm also gives uncertainty bounds as shown in Figure 3. The soil moisture time series that is shown exhibits variability that is not easy to capture. Still, LOBARE performed very well as compared against the traditional BARE algorithm [*Thiemann et al.*, 2001], as shown in Figure 4. The results in Figure 4 are the best obtained after 100 BARE runs, though it can be seen that the BARE algorithm still needs improvement. Because of faster convergence, the number of support vectors when LOBARE was used was 118, whereas it is 450 with BARE. A smaller number of support vectors helps maintain the generalizing capability of the SVM machine or, in other words, fewer support vectors in the model reduce the chance of overfitting. The goodness of fit measure values are summarized in Table 1, which also shows the comparison with the *Thiemann et al.* [2001] algorithm.

[52] The evolution of the three SVM parameters using LOBARE is shown in Figure 5, where the horizontal axis is the parameter value and the vertical axis is the time step in

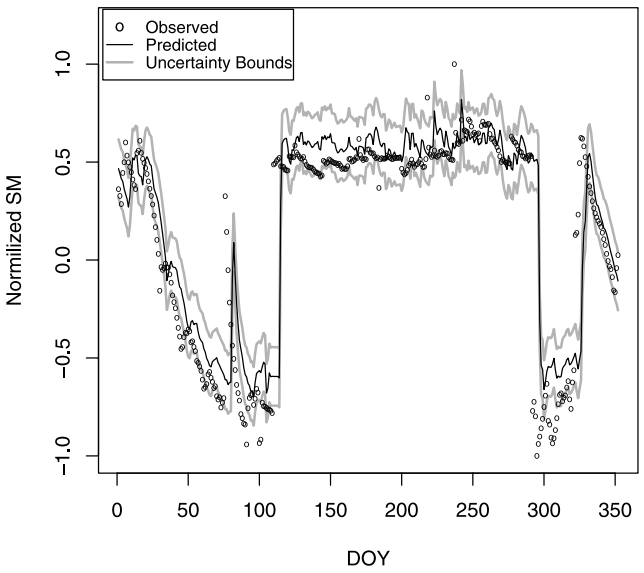

Figure 3. Soil moisture prediction results from the testing phase, using LOBARE along with uncertainty bounds.

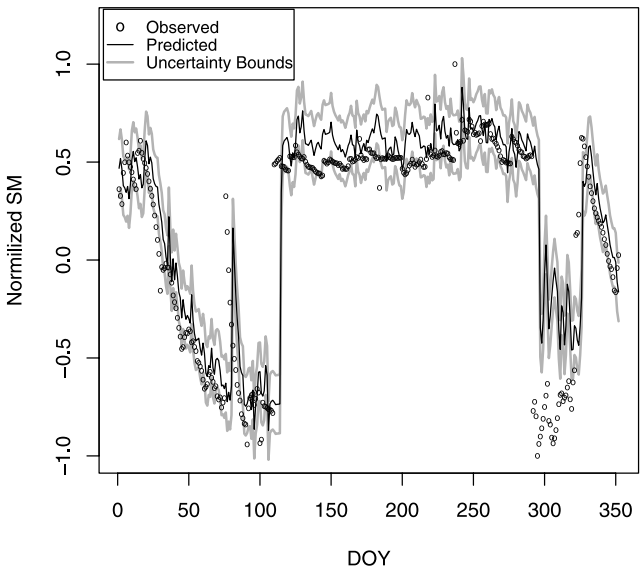

Figure 4. Soil moisture prediction results using BARE along with uncertainty bounds.

Table 1. SVM Parameter Results

Statistic	Observed Data	Predicted LOBARE	Predicted BARE
Mean	0.1480	0.2308	0.2420
Maximum	1.0000	0.8074	0.8808
Variance	0.2816	0.2201	0.2376
Skewness	−0.8370	−0.7460	−0.8395
Kurtosis	2.0606	1.8424	2.1529
SVM Parameters	LOBARE	BARE	
Cost C	11.456	7.832	
γ	0.0023	0.0126	
Tolerance ε	0.166	0.1108	
Normalized RMSE	0.288	0.3155	
RMSE	3.302	3.598	
E	0.6943	0.6466	
MAE	0.1818	0.1925	
R^2	0.7199	0.6862	
BIAS	−0.0828	−0.0941	

days. The plots are obtained by taking the projection of the parameter set on each parameter axis; the darker the bars, the more likely the chance of parameter selection. Since we start with a uniform distribution, the lower portions of the plot look alike until a few of the best parameter values take the lead as the update continues. As more data come in, only a few parameter values are updated leaving behind the sets that do not perform as well. An additional thing to notice is that we start with just 40 sets of parameter values, which reduces the computational cost but does not compromise the

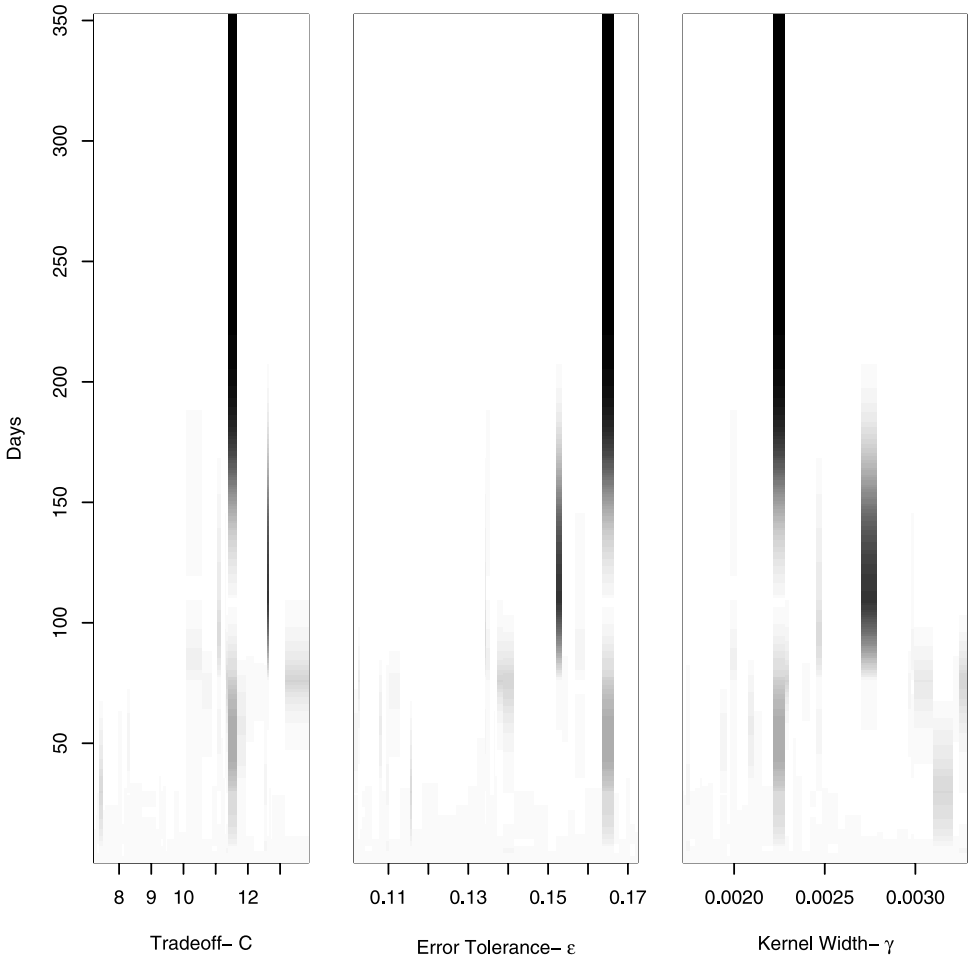

Figure 5. SVM parameter evolution. The dark bar indicates the likelihood of the parameter.

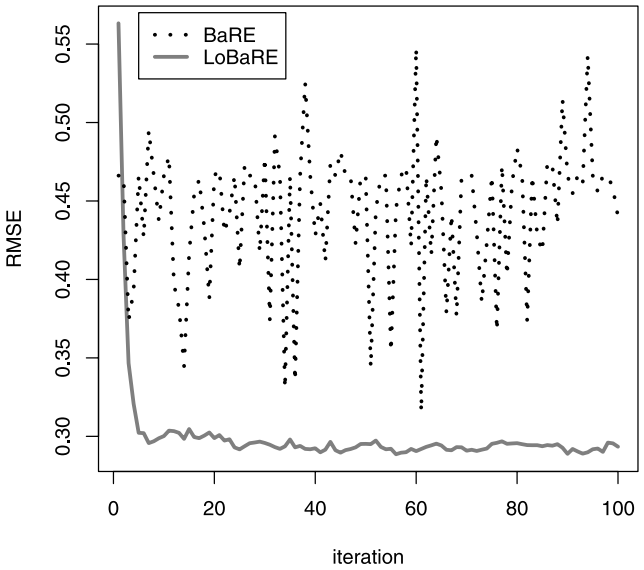

Figure 6. SVM convergence plot for LOBARE and BARE.

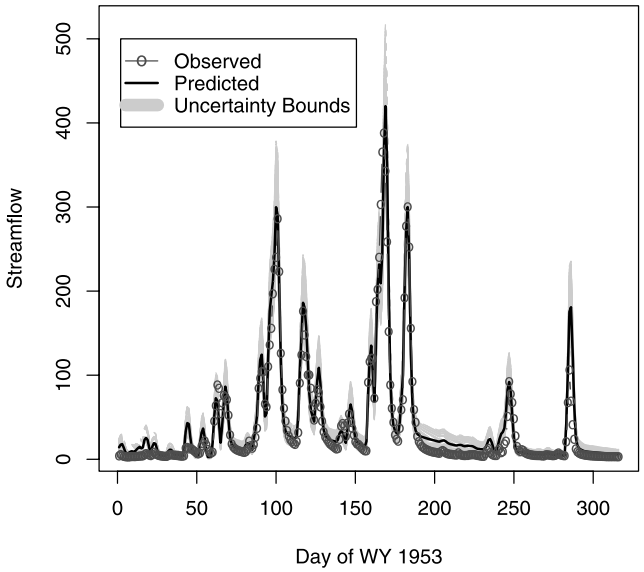

Figure 8. Prediction results for SAC-SMA using LOBARE for the water year 1953 along with the uncertainty bounds.

convergence. It is also seen that the best set of parameter values was picked after about 150 days of updating. This is illustrated in Figure 5 by the bars that grow dark after about 150 days. The LOBARE convergence results are shown in Figure 6. The results are compared with the traditional BARE algorithm of *Thiemann et al.* [2001]. It is noted that the LOBARE shows much faster convergence when compared to BARE. The root mean square error (RMSE) value decreased rapidly in the iterative method of LOBARE, whereas for this problem the RMSE of BARE oscillated without convergence.

3.4. Application to Streamflow Modeling

[53] As the second case study, the LOBARE algorithm is tested on a very well known conceptual rainfall-runoff (CRR) model, the Sacramento soil moisture accounting (SAC-SMA) model for streamflow estimation. The SAC-SMA (Figure 7) model is a component of the National

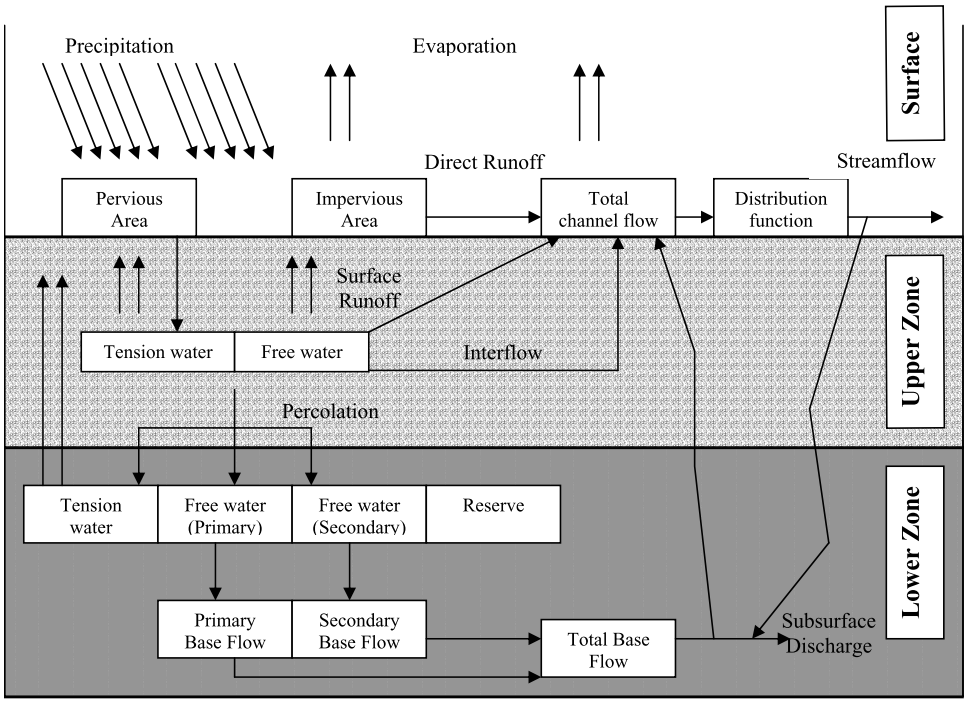

Figure 7. Conceptual flow diagram of the SAC-SMA model (adapted from *Thiemann et al.* [2001]).

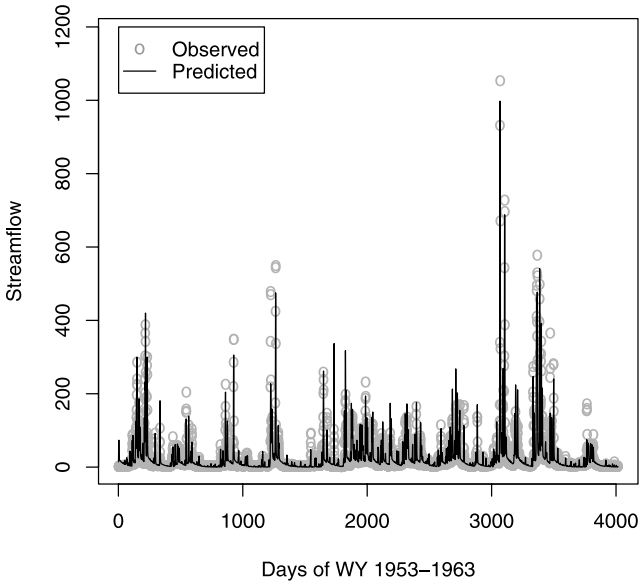

Figure 9. Prediction results for SAC-SMA using LOBARE for the 11 years (WY 1953–1963).

Weather Service River Forecast model and is well known for its complex of 13 parameters. Further details about the SAC-SMA model are given by *Burnash et al.* [1973].

3.5. Description of the Study Area and Data

[54] The SAC-SMA model has been widely used for streamflow predictions on the Leaf River in recent optimization applications [*Sorooshian and Gupta*, 1983; *Brazil*, 1988; *Duan et al.*, 1994; *Gupta et al.*, 1998; *Thiemann et al.*, 2001; *Vrugt et al.*, 2003]. The Leaf River watershed is located in Southern Mississippi and has an area of 1950 km^2 [*Vrugt et al.*, 2003]. In order to have a comparison with the previous study [*Thiemann et al.*, 2001], we are working with the same data. The inputs to the SAC-SMA model are precipitation and evapotranspiration data taken at 6-hour intervals. The output is streamflow, forecast at 6-hour intervals at the same time step as the inputs. The output was averaged over the full day to obtain daily predictions. It is assumed that the streamflow observations became available at the beginning of the 1953 water year (WY). That is when the LOBARE update procedure was started. The training set for LOBARE was 548 data points, only 302 of which were required for the estimated parameter range to stabilize. The three parameter dimensions that were bisected in the first

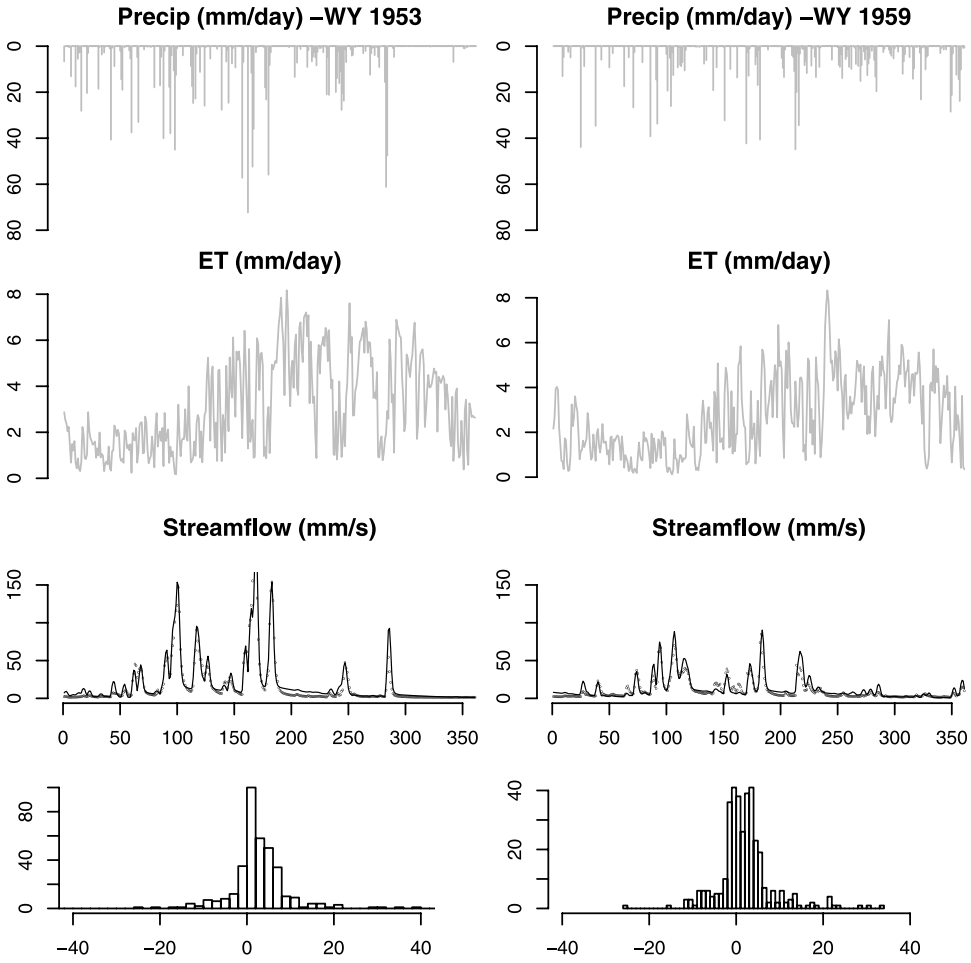

Figure 10. LOBARE prediction results for years 1953 and 1959 along with inputs (precipitation and ET) and residual error histograms.

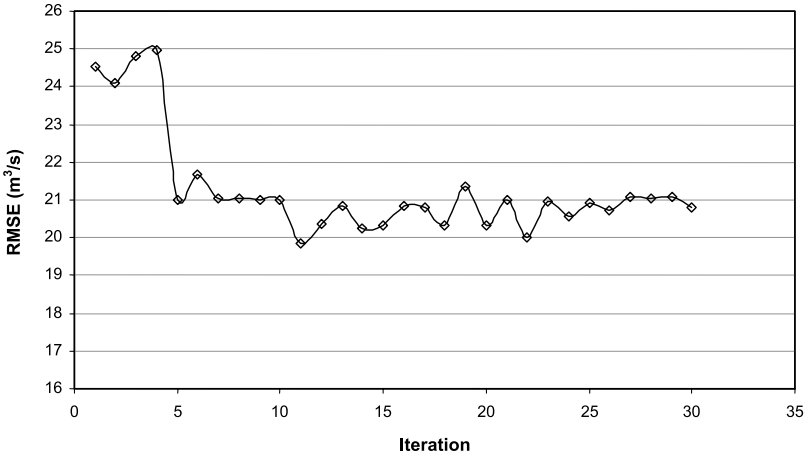

Figure 11. SAC-SMA convergence plot for LOBARE.

LOBARE run were: pctim, lzsk, and lzpk, which have ranges of 0–0.1, 0.01–0.25, and 0.0001–0.025, respectively. The results that follow in the next section span an 11-year period from WY 1953 through 1963.

3.6. Streamflow Results

[55] The streamflow predictions for WY 1953 are shown in Figure 8. A significant strength of the Bayesian-based method is that it also gives the confidence interval for the predictions. The shaded region in Figure 8 shows the confidence interval, whereas the continuous line is the prediction; the observations are the line with diamonds. Figure 8 shows that the predictions are bracketed very well within the bounds, and the observation trend is well captured. The full 11-year modeling results produced by the best parameter set are shown in Figure 9 where the gray diamonds are the observed flows and the solid line is the prediction. Figure 10 shows predictions for two separate years (1953 and 1959) that are different in terms of hydrologic inputs and outputs; one wet represented by 1953 and one dry represented by 1959. Streamflow per unit catchment area is shown in Figure 10 along with residual error histograms. It can be noticed from residual error histograms that the model performed reasonably well for both the wet and the dry year.

[56] Figure 11, which shows the evolution of LOBARE with the RMSE plotted on the vertical axis, depicts the efficiency of the LOBARE algorithm in converging to the 13 parameter values needed for the SAC-SMA model. The LOBARE algorithm shows convergence after 12 iterations. Table 2 gives a comparison between statistics of the observed data vis à vis modeled data with LOBARE and BARE. Values of goodness of fit measures are shown in Table 3. We also compared our results with those of selected previous studies [*Brazil*, 1988; *Thiemann et al.*, 2001]. In the previous method 10,000 parameter sets were used over a calibration period of a year. LOBARE uses only 1200, such as 2^3 divisions with 40 parameter sets each at the first run, then 2 divisions with 40 parameter sets each for the next 11 runs. Convergence is achieved at the end of the 12th run. Fixing these parameters the LOBARE algorithm was run 15 times and a test of stability was performed on the final results. A maximum variation of only ±6.5% of the presented parameter ranges was detected with the lzfpm parameter, which has a range of 1–1000. The LOBARE shows considerable improvement over the studies of *Brazil* [1988] and *Thiemann et al.* [2001] in terms of convergence rate, computational effort, and stability of solution.

4. Conclusions

[57] In the present application, a localized Bayesian recursive estimation algorithm is devised to perform parameter estimation for two very different models, namely, a support vector machine model for soil moisture estimation, and a well-known CRR model, the Sacramento soil moisture accounting (SAC-SMA) model. The SVM has three parameters that must be determined through a calibration process. The SAC-SMA model is very complex with 13 undetermined parameters.

[58] The results for the LOBARE parameter estimation efforts for both models are very encouraging. The LOBARE method converged to parameter values for both models using significantly fewer evaluations of points in the parameter space. Further, the LOBARE algorithm proved to be very stable in its convergence as compared to BARE, especially for the parameter estimation runs for the SVM model (Figure 6).

[59] Because of its design, LOBARE is superior to BARE with respect to the question of equifinality. BARE converges to a single set of parameter values, but there is nothing to say that another set would not be equally good. LOBARE converges to a range of parameter values that (approximately) represent the points in parameter space for which the model performance is indistinguishable. The LOBARE range does not necessarily include parameter sets obtained by BARE or by manual calibration [*Brazil*, 1988].

Table 2. SAC-SMA Observed and Modeled Data Statistics

Statistic	Observed Data	Modeled LOBARE	Modeled BARE
Mean, mm/s	27.4486	28.9439	29.1086
Maximum	1,313.9146	997.6590	921.3714
Variance	3,939.9217	3,600.2428	3,609.3087
Skewness	7.7921	6.3986	5.6337
Kurtosis	100.5736	65.5960	52.6570

Table 3. SAC-SMA Parameter Values and Comparison With Previous Studies

Parameters	Range	LOBARE	*Brazil* [1988]	*Thiemann et al.* [2001]
uztwm	1–150	26.22–37.12: 32.62	9	33.61
uzfwm	1–150	38.57–44.10: 42.75	39.8	76.12
uzk	0.1–0.5	0.20–0.22: 0.22	0.2	0.332
pctim	0–0.1	0.017–0.025: 0.024	0.003	0.016
adimp	0–0.4	0.28–0.35: 0.32	0.25	0.266
zperc	1–250	178.66–244.13: 230.75	250	117.3
rexp	0–5	3.036–4.02: 3.033	4.27	4.948
lztwm	1–500	255.07–272.19: 263.53	240	235.6
lzfsm	1–1000	13.43–89.03: 29.12	40	131.9
lzfpm	1–1000	112.94–145.08: 134.36	120	123.5
lzsk	0.01–0.25	0.081–0.21: 0.21	0.2	0.089
lzpk	0.0001–0.025	0.0051–0.007: 0.0056	0.006	0.015
pfree	0–0.1	0.010–0.015: 0.0126	0.024	0.146

Goodness of Fit Measure[a]	LOBARE	*Brazil* [1988]	*Thiemann et al.* [2001]
RMSE	19.8	20.3	24.059
MAE	8.8		10.3796
R^2	0.9007		0.8700
BIAS	−1.4953		−1.6601
E	0.9001		0.8687
RAE	0.2939		0.3468
RSE	0.0999		0.1313
RVE	−0.0545		−0.0605

[a]RMSE (root-mean-square error) = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2}$; MAE (mean absolute error) = $\frac{1}{n}\sum_{i=1}^{n}|X_i - Y_i|$); $R^2 = \left[\frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2\sum_{i=1}^{n}(Y_i-\bar{Y})^2}}\right]^2$; BIAS = $\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)$; E (efficiency coefficient) = $1 - \frac{\sum_{i=1}^{n}(Y_i-X_i)^2}{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}$; RAE (relative absolute error) = $\frac{\sum_{i=1}^{n}|X_i-Y_i|}{\sum_{i=1}^{n}|Y_i-\bar{Y}|}$; RSE (relative square error) = $\frac{\sum_{i=1}^{n}(X_i-Y_i)^2}{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}$; RVE (relative volume error) = $\frac{\sum_{i=1}^{n}(X_i-Y_i)}{\sum_{i=1}^{n}Y_i}$; where Y is the observed value, X is the modeled value, $\bar{Y}$ is the mean of the observed data, $\bar{X}$ is the mean of the modeled data, and n is the number of data points.

This is due to the response surface complexity/roughness which is caused partially by parameter interaction.

[60] In our application, the LOBARE methodology helped to reduce the number of support vectors the SVM machine uses in comparison to the results of the BARE runs. This is potentially significant in SVM applications because this produces a model with better generalization capability and it reduces the computational burden required for the training of the learning machine. This feature opens potential new horizons for modeling hydrologic systems using learning machines that might be built from more limited amounts of data.

[61] **Acknowledgments.** We would like to thank Abedalrazq Khalil at Columbia University for his valuable comments that greatly improved this paper. We would also like to thank the Utah Water Research Laboratory (UWRL) and the Utah Center for Water Resources Research (UCWRR) for the support provided to this research. We are also grateful to the anonymous reviewers of our original manuscript for their extremely valuable comments.

References

Allen, P. B., and J. W. Naney (1991), Hydrology of the Little Washita River Watershed, Oklahoma: Data and analyses, *Rep. ARS-90*, Agric. Res. Serv., U.S. Dep. of Agric., Durant, Okla.

Asefa, T., and M. W. Kemblowski (2002), Support vector machines approximation of flow and transport models in initial groundwater contamination network design, *Eos Trans. AGU*, *83*(47), Fall Meet. Suppl., Abstract H72D-0882.

Asefa, T., M. W. Kemblowski, G. Urroz, M. McKee, and A. Khalil (2004), Support vectors-based groundwater head observation networks design, *Water Resour. Res.*, *40*(11), W11509, doi:10.1029/2004WR003304.

Bastidas, L. A., H. V. Gupta, S. Sorooshian, W. J. Shuttleworth, and Z. L. Yang (1999), Sensitivity analysis of a land surface scheme using multicriteria methods, *J. Geophys. Res.*, *104*(D16), 19,481–19,490.

Beven, K. (1993), Prophesy, reality and uncertainty in distributed hydrological modelling, *Adv. Water Resour.*, *16*, 41–51.

Beven, K., and P. Young (2003), Comment on "Bayesian recursive parameter estimation for hydrologic models" by M. Thiemann, M. Trosset, H. Gupta, and S. Sorooshian, *Water Resour. Res.*, *39*(5), 1116, doi:10.1029/2001WR001183.

Box, G. E. P., and G. C. Tiao (1973), *Bayesian Inference in Statistical Analysis*, Addison-Wesley, Reading, Mass.

Brazil, L. E. (1988), Multilevel calibration strategy for complex hydrologic simulation models, Ph.D. dissertation, Colo. State Univ., Fort Collins.

Burnash, R. J. E., R. L. Ferral, and R. A. McGuire (1973), A generalized streamflow simulation system, report, Joint Fed.-State River Forecast Cent., Sacramento, Calif.

Dibike, B. Y., S. Velickov, D. Solomatine, and B. M. Abbot (2001), Model induction with support vector machines: Introduction and applications, *J. Comput. Civ. Eng.*, *15*(3), 208–216.

Duan, Q., S. Sorooshian, and V. K. Gupta (1992), Effective and efficient global optimization for conceptual rainfall-runoff models, *Water Resour. Res.*, *28*(4), 1015–1031.

Duan, Q., S. Sorooshian, and V. K. Gupta (1994), Optimal use of the SCE-UA global optimization models, *J. Hydrol.*, *158*, 265–284.

Freer, J., K. J. Beven, and B. Ambroise (1996), Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach, *Water Resour. Res.*, *32*(7), 2161–2173.

Gill, M. K., T. Asefa, Q. Shu, and M. W. Kemblowski (2003), Soil moisture prediction using support vector machines, paper presented at INRA 2003 Subsurface Science Symposium, Inst. Natl. de Rech. Agron., Salt Lake City, Utah, 6–8 Oct.

Gill, M. K., T. Asefa, M. McKee, and M. W. Kemblowski (2006), Soil moisture prediction using support vector machines, *J. Am. Water Resour. Assoc*, in press.

Gupta, H. V., S. Sorooshian, and P. O. Yapo (1998), Toward improved calibration of hydrologic models: Multiple and noncommensurate measures of information, *Water Resour. Res.*, *34*(4), 751–763.

Khalil, A. (2005), Computational learning theory in water resources management and hydrology, Ph.D. dissertation, 159 pp., Dep. of Civ. and Environ. Eng., Utah State Univ., Logan.

Khalil, A., M. N. Almasri, M. McKee, and J. J. Kaluarachchi (2005a), Applicability of statistical learning algorithms in ground water quality modeling, *Water Resour. Res.*, *41*, W05010, doi:10.1029/2004WR003608.

Khalil, A., M. K. Gill, and M. McKee (2005b), New applications for information fusion and soil moisture forecasting, paper presented at 8th Conference on Information Fusion, Int. Soc. of Inf. Fusion, Philadelphia, Pa, 25–29 July.

Khalil, A., M. McKee, M. W. Kemblowski, and T. Asefa (2005c), Basin-scale water management and forecasting using multisensor data and neural networks, *J. Am. Water Resour. Assoc.*, *41*(1), 195–208.

Khalil, A., M. McKee, M. Kemblowski, T. Asefa, and L. A. Bastidas (2006), Multiobjective analysis of chaotic dynamic systems with sparse learning machines, *Adv. Water Resour.*, *29*(1), 72–88.

Liong, S., and C. Sivapragasam (2002), Flood stage forecasting with support vector machines, *J. Am. Water Resour. Assoc.*, *38*(1), 173–186.

Savenije, H. H. G. (2001), Equifinality, a blessing in disguise?, *Hydrol. Processes*, *15*, 2835–2838.

Sorooshian, S., and V. K. Gupta (1983), Automatic calibration of conceptual rainfall-runoff models: The question of parameter observability and uniqueness, *Water Resour. Res.*, *19*(1), 260–268.

Spear, R. C., and G. M. Hornberger (1980), Eutrophication in Peel Inlet: II. Identification of critical uncertainties via generalized sensitivity analysis, *Water Res.*, *14*, 43–49.

Thiemann, M., M. Trosset, H. Gupta, and S. Sorooshian (2001), Bayesian recursive parameter estimation for hydrologic models, *Water Resour. Res.*, *37*(10), 2521–2535.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer, New York.

Vapnik, V. (1998), *Statistical Learning Theory*, John Wiley, Hoboken, N. J.

Vrugt, J. A., H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian (2003), Effective and efficient algorithm for multiobjective optimization of hydrologic models, *Water Resour. Res.*, *39*(8), 1214, doi:10.1029/2002WR001746.

Yapo, P., H.V. Gupta, and S. Sorooshian (1996), Calibration of conceptual rainfall-runoff models: Sensitivity to calibration data, *J. Hydrol.*, *181*, 23–48.

Yapo, P. O., H. V. Gupta, and S. Sorooshian (1998), Multi-objective global optimization for hydrologic models, *J. Hydrol.*, *204*, 83–97.

L. Bastidas, M. K. Gill, Y. H. Kaheil, and M. McKee, Department of Civil and Environmental Engineering and Utah Water Research Laboratory, Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200, USA. (yasir@cc.usu.edu)