AGU PUBLICATIONS

Water Resources Research

RESEARCH ARTICLE

10.1002/2013WR013969

Key Points:

- A multimodel framework for streamflow simulation
- Generates rich streamflow scenarios
- Preserves complex distributional properties

Supporting Information:

- Regression models and Figures S1–S33

Correspondence to:
C. Li,
lichsunny@gmail.com

Citation:
Li, C., and V. P. Singh (2014), A multimodel regression-sampling algorithm for generating rich monthly streamflow scenarios, *Water Resour. Res.*, *50*, 5958–5979, doi:10.1002/2013WR013969.

Received 13 APR 2013
Accepted 23 JUN 2014
Accepted article online 26 JUN 2014
Published online 17 JUL 2014

A multimodel regression-sampling algorithm for generating rich monthly streamflow scenarios

Chao Li[1] and Vijay P. Singh[1,2]

[1]Department of Biological and Agricultural Engineering, Texas A&M University, College Station, Texas, USA, [2]Department of Civil and Environmental Engineering, Texas A&M University, College Station, Texas, USA

Abstract This paper presents a multimodel regression-sampling algorithm (MRS) for monthly streamflow simulation. MRS is motivated from the acknowledgment that typical nonparametric models tend to simulate sequences exhibiting too close a resemblance to historical records and parametric models have limitations in capturing complex distributional and dependence characteristics, such as multimodality and nonlinear autocorrelation. The aim of MRS is to generate streamflow sequences with rich scenarios while properly capturing complex distributional and dependence characteristics. The basic assumptions of MRS include: (1) streamflow of a given month depends on a feature vector consisting of streamflow of the previous month and the dynamic aggregated flow of the past 12 months and (2) streamflow can be multiplicatively decomposed into a deterministic expectation term and a random residual term. Given a current feature vector, MRS first relates the conditional expectation to the feature vector through an ensemble average of multiple regression models. To infer the conditional distribution of the residual, MRS adopts the *k*-nearest neighbor concept. More precisely, the conditional distribution is estimated by a gamma kernel smoothed density of historical residuals inside the *k*-neighborhood of the given feature vector. Rather than obtaining the residuals from the averaged model only, MRS retains all residuals from all the original regression models. In other words, MRS perceives that the original residuals put together would better represent the covariance structure between streamflow and the feature vector. By doing so, the benefit is that a kernel smoothed density of the residual with reliable accuracy can be estimated, which is hardly possible in a single-model framework. It is the smoothed density that ensures the generation of sequences with rich scenarios unseen in historical record. We evaluated MRS at selected stream gauges and compared with several existing models. Results show that (1) compared with typical nonparametric models, MRS is more apt at generating sequences with richer scenarios and (2) in contrast to parametric models, MRS can reproduce complex distributional and dependence characteristics. Since MRS is flexible at incorporating different covariates, it can be tailored for other potential applications, such as hydrologic forecasting, downscaling, as well as postprocessing deterministic forecasts into probabilistic ones.

1. Introduction

A historical streamflow record represents only one of the many possible realizations under the present conditions of land use and land cover, human intervention, climate variability, and other forcing factors [*Nazemi et al.*, 2013]. With one observed sequence, it is impractical to evaluate the effects of alternative policies and plans for catchment water resources management. On the other hand, historical records with sufficient length are needed for the formulation of reservoir operation policies, but in practice they are not always available. To surmount these obstacles, one approach is to simulate synthetic sequences using stochastic models.

Over the past several decades, a number of stochastic models have been developed for streamflow simulation. In general, as noted by *Bras and Rodriguez-Iturbe* [1985], streamflow simulation can be thought of as an exercise in the conditional distribution of $Y \mid \mathbf{X}$, wherein Y stands for streamflow of current month and $\mathbf{X}$ the corresponding feature vector. Simulation can be done through sequential sampling from the conditional distribution. The feature vector $\mathbf{X}$ may consist of single or multiple variables. Typical examples include lagged flow variables, aggregated flow variables, exogenous climate variables, and/or their combinations, to mention a few. No matter how the feature vector $\mathbf{X}$ changes, the fundamental objective remains the same, namely, to derive the conditional distribution of current month flow in a parametric, semiparametric, or nonparametric way.

Commonly used parametric models include the ARMA-type models [*Bras and Rodriguez-Iturbe*, 1985; *Fernandez and Salas*, 1990], the entropy theory-based model [*Hao and Singh*, 2011], and the copula theory-based model [*Lee and Salas*, 2011]. These parametric models assume that the conditional distribution of $Y \mid \mathbf{X}$ is from some well-known family or at least should follow some rigid functional form. Parametric models indeed have advantages as long as their assumptions are correct, or at least are not seriously violated. However, they lack flexibility in modeling complex distributional and dependence characteristics, such as multimodality and nonlinear autocorrelation. In this context, alternative semiparametric and nonparametric models have been put forward [*Lall and Sharma*, 1996; *Sharma et al*., 1997; *Sharma and O'Neill*, 2002; *Srinivas and Srinivasan*, 2001a, 2001b, 2005a, 2005b, 2006; *Srivastav et al*., 2011; *Prairie et al*., 2006; *Salas and Lee*, 2010; *Lee et al*., 2010; *Keylock*, 2012]. These models are typically based on nonparametric techniques like *k*-nearest neighbor resampling (KNN), kernel density estimation (KDE), their variants, and/or combinations. Semiparametric and nonparametric models avoid restrictive assumptions and allow the data to speak for themselves. Simulations can thus relatively closely replicate characteristics of the historical data. Nonparametric models are not without limitations, however. The representative one is that the simulated sequence exhibits too close a resemblance to historical record, as first recognized by *Maheepala and Parera* [1996] and more recently highlighted by *Salas and Lee* [2010] and *Lee et al*. [2010, 2012]. This limitation should be addressed as it is against the primary purpose of stochastic simulation, namely, to examine scenarios that are possible to occur but did not occur in history.

Considering the fact that parametric models have limitations in simulating complex distributional and dependence characteristics and typical nonparametric models tend to simulate sequences exhibiting too close a resemblance to historical record, the objective of this research is to seek an enhanced but straightforward simulation scheme that alleviates or eliminates typical shortcomings of existing parametric and nonparametric models. To this end, we take advantage of several commonly used nonparametric techniques in capturing complex distributional and dependence characteristics and introduce a novel multimodel simulation scheme with an attempt to simulate diverse streamflow realizations. The multimodel simulation scheme represents the most important contribution of this research in a sense that it provides a simple-to-understand and easy-to-implement approach being capable of overcoming the inadequacy of typical nonparametric models in simulating diverse streamflow realizations, which, to our knowledge, has not yet been well addressed in the existing literature.

In the following sections, we shall first explain the development of the enhanced multimodel simulation scheme and present its step-by-step implementation procedure, then evaluate the developed model at selected stream gauges and compare it with three commonly used nonparametric alternatives, and finally summarize the major conclusions of this research.

2. Development of Simulation Scheme

Basically, the enhanced simulation scheme is built upon the regression-resampling framework of the modified KNN model (MKNN) in *Prairie et al*. [2006]. Our innovation lies in that we advanced this framework such that it can generate rich streamflow scenarios, at the same time with other recognized limitations of MKNN being addressed. In order to set the stage for the enhanced scheme, we will first briefly summarize the logic of MKNN. We will then enumerate its limitations. Through proposing solutions to each of the limitations, we will finally unfold the overview of the enhanced scheme.

2.1. Logic of MKNN

For simplicity of explanation, we first clarify some notations. We consider streamflow of month *t* as a random variable and denote it as Y_t. We let Y_{t-1} represent streamflow of the previous month. We use $y_t(i)$ to mean the observation of Y_t at time instance *i*, where $i = 1, 2, \ldots, N$, and herein *N* represents the total number of time instances. Then, for each $y_t(i)$, there is an associated observation $y_{t-1}(i)$.

MKNN first assumes that Y_t can be additively decomposed into an expectation term M_t and a random residual term e_t as follows:

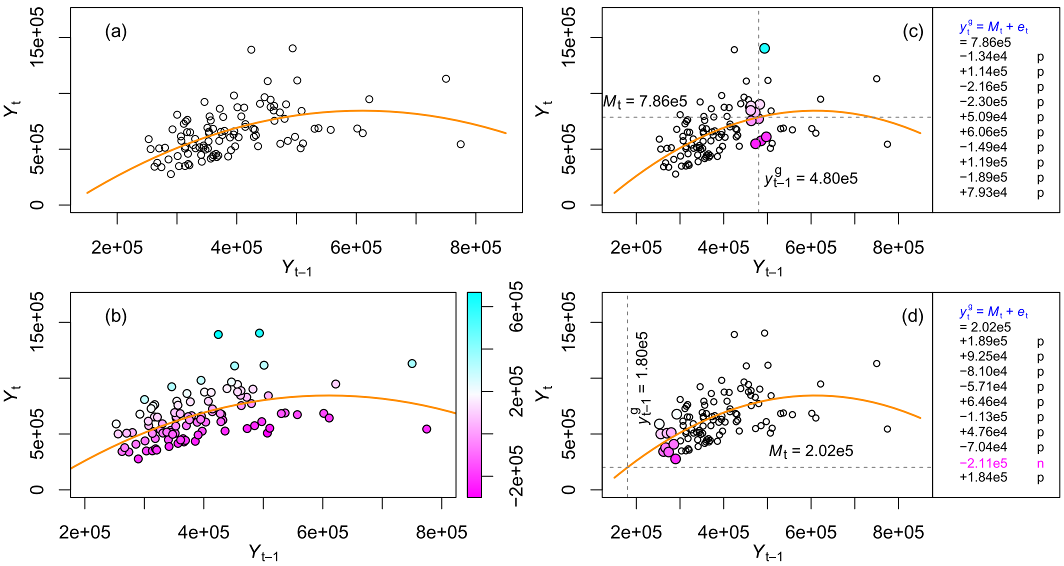

Figure 1. Schematic illustration for the logic of MKNN. (a) An LWPR function (solid orange curve) is fitted to the historical streamflow observations (scatter points) of June and May from the Lees Ferry gauge 09380000. (b) From the fitted regression curve, historical residuals are calculated and represented by the color coding. (c) Given a feature value y^g_{t-1} as shown by the dashed vertical line, i.e., the generated streamflow of month $t-1$, at the regression stage, the expectation M_t is estimated by the fitted LWPR function, i.e., $M_t = f\left(y^g_{t-1}\right)$ as shown by the dashed horizontal line; at the sampling stage, a historical residual is randomly selected from the *k*-neighborhood (here $k = 10$) of the given feature value, as highlighted by the colored points; the simulated streamflow of month *t* can be obtained by adding the selected residual the estimated expectation M_t. (d) The same as Figure 1c illustrating the regression-sampling procedure but for a situation where a negative simulation may be obtained. Note that the unit of the axes is acre-foot.

$$Y_t = M_t + e_t \tag{1}$$

Then it is assumed that streamflow of the current month Y_t depends on that of the previous month, i.e., $\mathbf{X}_t = Y_{t-1}$. To estimate the conditional distribution of $Y_t \mid Y_{t-1}$, MKNN first relates the expectation M_t to the feature variable Y_{t-1} through a locally weighted polynomial regression (LWPR) function $f(\cdot)$ [*Loader*, 1999; *Lall et al*., 2006]

$$M_t = f(Y_{t-1}) \tag{2}$$

which can be estimated from historical observations $[y_{t-1}(i), y_t(i)]$, $i = 1, 2, \ldots, N$.

After knowing the expectation M_t, one further needs to infer the distribution of the residual e_t conditional on Y_{t-1}. To this end, MKNN assumes that the conditional distribution of e_t can be approximated by the empirical distribution of historical residuals inside a *k*-neighborhood of the conditional feature point, i.e., $Y_{t-1} = y_{t-1}$. Hence, the conditional distribution of $Y_t \mid Y_{t-1} = y_{t-1}$ can be perceived as equivalent to the empirical distribution of the data obtained by adding the *k* residuals, respectively, back to the estimated expectation $M_t = f(y_{t-1})$. With this conditional distribution, simulation can be completed by random number generation.

We resummarize the logic of MKNN from the perspective of estimating the conditional distribution of $Y_t \mid Y_{t-1}$. We do that because it is helpful to explain where and how our improvements are made. In real simulations, one does not necessarily need to explicitly estimate the conditional distribution but follow the procedure in *Prairie et al*. [2006], presented in the following. Suppose now that one wants to simulate streamflow of month *t*. Prior to the simulation, an LWPR function $f(\cdot)$ has to be fitted to historical observations of months $t-1$ and *t*; see the solid line in Figure 1a. Then historical residuals are calculated following equations (1) and (2), as illustrated in Figure 1b. Now if it is assumed that the generated streamflow of month $t-1$ is y^g_{t-1}, simulation can be achieved by first estimating the expectation through $M_t = f\left(y^g_{t-1}\right)$ and then adding it to a historical residual, which is randomly selected from the *k*-neighborhood of $Y_{t-1} = y^g_{t-1}$. The random selection is weighted such that the closer neighbors are more likely to be selected. Figures 1c and 1d illustrate the above two-stage regression-sampling procedure.

2.2. Rationale of the Enhanced Scheme

Despite the advantages, literature review revealed three major limitations [*Prairie et al*., 2006; *Salas and Lee*, 2010; *Lee et al*., 2010]. First, like most of the current nonparametric models, MKNN-simulated sequence exhibits too close a resemblance to historical record. Second, it may simulate negative values. Third, it is less apt at preserving the interannual variability. We will now reason how to address these limitations.

From section 2.1, one may have realized that the resemblance of historical record in simulated data is mainly due to the limited choices available at the residual sampling stage. To circumvent this problem, it seems intuitive to expand the sampling space by increasing the neighborhood volume *k*. However, in order to obtain a model with balanced performance with respect to a suite of characteristics, *k* should be neither too small nor too large. A too large *k* tends to underestimate some of the characteristics, like variance and autocorrelation [*Buishand and Brandsma*, 2001]. A second remedy assumes that the residual e_t in equation (1) is normally distributed with mean 0 and variance approximated by the sample variance of historical residuals inside the *k*-neighborhood. Considering the relatively small sample size (usually *k* is less than 10 for seasonal streamflow simulation), the accuracy of the estimated variance is doubtable, let alone the questionable normality assumption. In addition, *Lee et al*. [2010] suggested a mixing procedure based on the "crossover" machinery of the genetic algorithm. This procedure was then used for stochastic weather simulation [*Lee et al*., 2012]. Because crossover may affect the serial correlation, a trial-and-error procedure is required to determine a suitable mixing parameter such that the autocorrelation is not distorted too much.

To generate streamflow with rich scenarios, here we neither want to impose the subjective distributional assumption nor to adopt the relatively complex crossover mixing, but to introduce an application-oriented multimodel simulation scheme. Note that at the regression stage MKNN makes little use of statistical ideas, if any. It follows that in theory any surrogate function is feasible as long as it properly generalizes the underlying M_t–$\mathbf{X}_t$ relationship. We thus hypothesize that improvement can be achieved if several regression models are used. The motivating idea behind this is that with various models at hand, various regression results can be obtained, which will in turn benefit the subsequent residual sampling by offering more choices. Arguably, this should improve the simulation scheme in generating diverse streamflow scenarios.

Pertaining to the generation of negative values, it mainly stems from the additive decomposition assumption. Equation (1) indicates that if the selected residual e_t happens to be negative while with a magnitude greater than the estimated expectation M_t, then a negative simulation will result, as the case shown in Figure 1d. To circumvent this risk, we hypothesize that instead of the additive decomposition, Y_t follows a multiplicative decomposition:

$$Y_t = M_t \times e_t \tag{3}$$

The multiplicative decomposition assures that no negative value will be generated. It should be noted that for stream gauges in arid regions where a substantial amount of zero values may exist, in this case it would be better to simulate the wet-dry sequence first using, for example, a Markov process and then simulate streamflow of wet months using the hypothesis in equation (3).

The inadequacy of preserving the interannual variability is recognized as a consequence of the first-order assumption of streamflow autocorrelation. To reproduce the interannual variability, additional feature variables retaining the respective signals are required. *Prairie et al*. [2006] discussed that it is expected to improve MKNN in characterizing the interannual variability by incorporating the dynamic aggregated flow variable of the past 12 months into the feature vector. To our knowledge, the mentioned work has not yet been reported by the authors themselves or others. The idea of including a dynamic aggregated variable was initially discussed by *Lall and Sharma* [1996] and later applied by *Sharma and O'Neill* [2002] for monthly streamflow simulation. A second approach suggested by *Salas and Lee* [2010] is to include the static aggregated flow of a water year as an extra feature. *Sharma and O'Neill* [2002] argued that to maintain the dependence between annual and monthly time scales, a more realistic way should be to use a moving aggregated flow variable as compared to the static water year aggregated flow. We therefore adopt the first logic. Then, the feature vector $\mathbf{X}_t$ can be represented as

$$\mathbf{X}_t=[Y_{t-1}, Y_a] \qquad (4)$$

where Y_a denotes the dynamic aggregated flow of the past 12 months. In all analyses that follow the feature vector $\mathbf{X}_t$ should be understood as the one given in equation (4) unless otherwise stated.

To establish the multimodel simulation model one needs to select a number of suitable regression models. Since different models may have different performance, the next question to be addressed is one of how to combine them such that put together they can approximate the M_t–$\mathbf{X}_t$ relationship better or at least no worse than any individual model. Following this, how to properly estimate the conditional distribution of $e_t \mid \mathbf{X}_t$ plays a crucial role in generating rich streamflow scenarios. These three questions are discussed in order in the ensuing subsections 2.2.1–2.2.3.

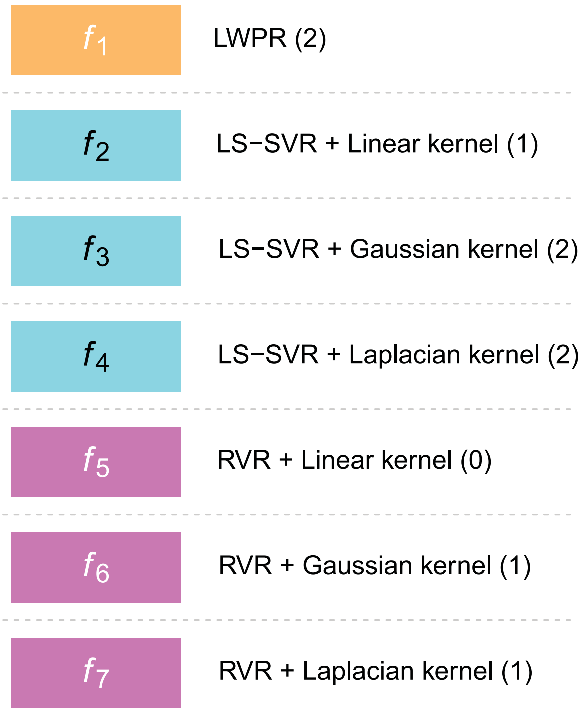
f_1 LWPR (2)
f_2 LS−SVR + Linear kernel (1)
f_3 LS−SVR + Gaussian kernel (2)
f_4 LS−SVR + Laplacian kernel (2)
f_5 RVR + Linear kernel (0)
f_6 RVR + Gaussian kernel (1)
f_7 RVR + Laplacian kernel (1)

Figure 2. Reference card for the regression models used in this research (LWPR: locally weighted polynomial regression; LS-SVR: least squares support vector regression; and RVR: relevance vector regression). Values in the parentheses are the numbers of hyperparameters of the respective regression models.

2.2.1. Regression Models

As was noted, any regression model can be used as long as it approximates the M_t–$\mathbf{X}_t$ relationship reasonably well. Our application-oriented simulation scheme requires models that must be simple to use, but must also be sophisticated enough to capture the dependence structure between M_t and $\mathbf{X}_t$. Keeping this point in mind, three types of regression methods were selected, including LWPR, least squares support vector regression (LS-SVR), and relevance vector regression (RVR). All of them are capable of both linear and nonlinear regression analyses. Moreover, there exist free software packages available for their implementation, making them easily accessible to practitioners. In order to streamline the presentation in the main text, we provide their background information in the supporting information. Combining them with different kernel functions, seven models can be created, as listed in the reference card in Figure 2.

Among the seven models, except for f_5 (RVR with linear kernel), the others contain at least one hyperparameter that has to be determined beforehand, as shown by the numbers in the parentheses in Figure 2. To identify the optimal hyperparameters, we apply a leave-one-out cross-validation (LOOCV) procedure with the mean squared error of the regression estimates at the sample points being the objective function; see details in the supporting information. It is argued that LOOCV requires a fair amount of computation. Yet this should not be a serious limitation here because in general the sample size of available observed data for monthly streamflow simulation is not large (around 100). In addition, the availability of ever-fast computers significantly accelerates the computation speed.

Different models will produce different regression results. Even when assuming the underlying M_t–$\mathbf{X}_t$ relationship to be linear, the regression plane fitted by f_2 may differ from the one by f_5. One may argue that only one of the seven models reflects the best knowledge about the underlying M_t–$\mathbf{X}_t$ relationship and should therefore be used as its surrogate function. However, it is emphasized that our intention is not to identify the one with the best performance. Instead, we deem that each model provides a reasonable approximation. Then it follows that for a given neighborhood size k, the number of historical residuals e_t obtained from the seven regression models will be 6k times more than that obtained from a single model.

As will be seen later, the enriched residuals will in turn benefit the enhanced simulation scheme in generating rich streamflow scenarios.

2.2.2. Multimodel Weighting Scheme

Within the multimodel simulation scheme, for each calendar month t there are seven admissible models approximating the M_t–$\mathbf{X}_t$ relationship. Properly weighting competing models at hand is necessary. Commonly used model weighting techniques includes naively equal weighting, inverse-variance weighting, least squares weighting, and Bayesian weighting. Considering the simplicity of the naively equal weighting and its decent performance [*Hansen*, 2007], it is adopted for this research. Note that the equal weights, all positive and summing up to 1, can be explained as the likelihood measures of individual models being the best. Then, given a feature value $\mathbf{X}_t = \mathbf{x}_t$, there are two options to estimate the expectation M_t. One can first select one of the seven models at random and then make estimation on $\mathbf{x}_t$. One can also average the seven models and then make estimation with the averaged model. Here we choose the second way. In other words, we consider the averaged model as the unique surrogate function of the M_t–$\mathbf{X}_t$ relationship.

2.2.3. Conditional Residual Sampling

To approximate the conditional distribution of $e_t \mid \mathbf{X}_t$, we retain all the original residuals of the seven models rather than those associated with the averaged surrogate function only. We believe that the original residuals put together reflect a more comprehensive exploration of the covariance structure between Y_t and $\mathbf{X}_t$. Consequently, options available at the residual sampling stage are increased 6 times more. Attributing to the increased residual numbers, one can further smooth these discrete residuals to obtain a continuous distribution function and then do sampling from the smoothed distribution.

We apply the nonparametric KDE to smooth the discrete residuals. Commonly used KDEs typically employ the Gaussian kernel. However, negative residuals may be generated because the Gaussian kernel has an unbounded support. Our simulation scheme requires the multiplicative residual that must be positive. To avoid negatives, we adopt the gamma KDE introduced by *Salas and Lee* [2010]

$$f(e_t|\mathbf{X}_t=\mathbf{x}_t)=\frac{1}{n}\sum_{j=1}^{n}\text{gamma}\left(e_t; e_t(j)^2/h^2, h^2/e_t(j)\right) \tag{5}$$

where n is the total number of historical residuals inside the k-neighborhood of the conditional feature point $\mathbf{X}_t = \mathbf{x}_t$, herein $n = 7k$ with k being a heuristic value of $\sqrt{N}$ with N holding the same meaning as defined in section 2.1; $e_t(j)$ is the jth residual; gamma($\cdot$; a, b) is the probability density function (PDF) of gamma distribution with shape parameter a and scale parameter b, respectively; and h is the kernel smoothing parameter and can be determined by

$$h_1=\sigma(e_t)/0.5n^{1/2} \tag{6}$$

where $\sigma(e_t)$ is the standard deviation of e_t and can be estimated by the sample standard deviation of historical residuals $e_t(i)$, $i = 1, 2, \ldots, n$. It is experienced that equation (6) tends to yield an overall small value. We therefore modify it as

$$h_2=\sigma(e_t)/n^{1/4} \tag{7}$$

Comparing these two estimators, one can find their relationship: (1) $h_1 > h_2$, when $n < 16$; (2) $h_1 = h_2$, when $n = 16$; and (3) $h_1 < h_2$, when $n > 16$. Generally, when the sample size is small, they have approximately the same performance, whereas when the sample size is large, h_1 tends to result in an undersmoothed density.

To illustrate this effect, we carried out a simple Monte Carlo simulation experiment. First, random sample sets with different sizes (10, 15, 500, and 1000) were simulated from a gamma distribution with shape parameter 5.0 and scale parameter 1.0. Equation (5) with the smoothing parameter estimated respectively by h_1 and h_2 was used to smooth the sample data. For each sample size, random sampling and kernel smoothing were repeated 4 times. Results were summarized in Figure 3. When the sample size is small (10), although $h_2 < h_1$, their difference is very slight. Thus, it is not surprising to observe almost the same density estimates. No evidence for over or undersmooth was found with either estimator. As the sample size increases, the denominator of h_1 increases much faster than that of h_2. Consequently, undersmoothed curves with irrational bumps were observed along with h_1. Although the suggested estimator h_2 may not be the theoretically optimal option, the simple simulation experiment shows that at a minimum it is a practically safe one.

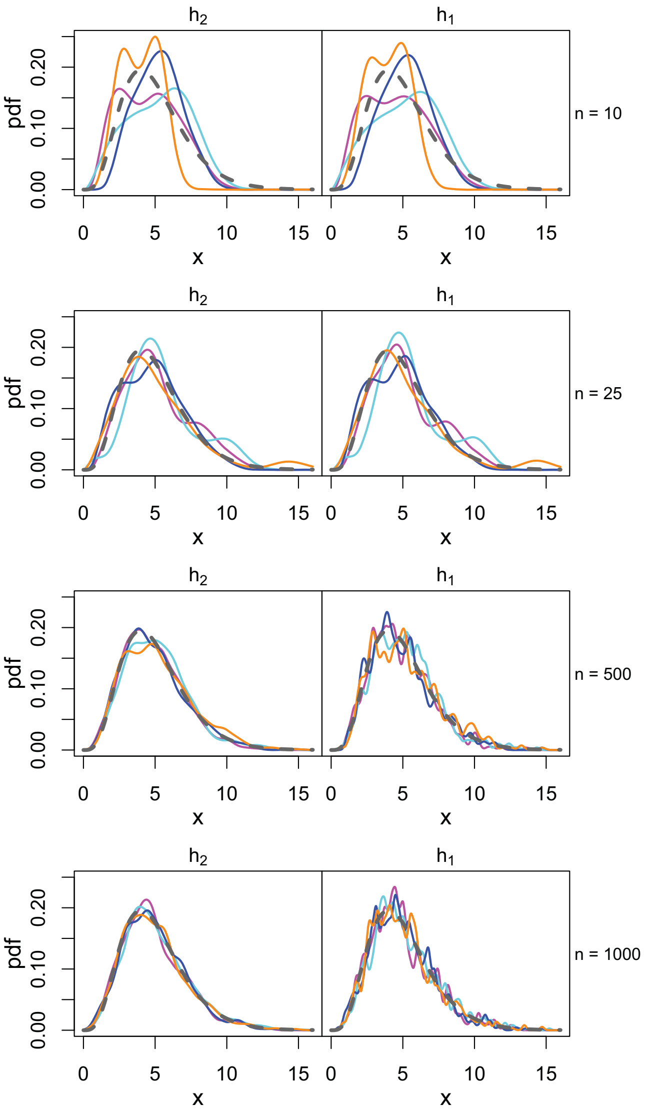

Figure 3. Behavior of different smoothing parameter estimators (h_1 and h_2) for gamma KDE as the sampling size increases from 10 to 1000. In each subplot, the gray dashed curve is for the true density and the solid curves for the kernel smoothed densities with different colors representing different Monte Carlo simulation trials.

3. Model Implementation

To make the multimodel simulation scheme easy to reproduce, we summarize here its step-by-step implementation procedure. We divide the whole procedure into two phases. The first phase is for preprocessing the observed data and estimating the model hyperparameters, and the second for generating synthetic streamflow sequences.

3.1. Data Preprocessing and Hyperparameters Estimating

Formulating historical streamflow matrix: Suppose there are N years of monthly streamflow observations at the gauge of interest. These observations are first stored in an N-by-12 matrix with one row for each year, as illustrated in Figure 4a. For simplicity, we use $\mathbf{Y}$ to denote the matrix thus obtained. Thereby, the tth column of $\mathbf{Y}$ stores the historical observations of Y_t, i.e., streamflow of the calendar month t.

Creating conditional feature matrices: From the historical flow matrix $\mathbf{Y}$, another matrix $\mathbf{Y}_{-1}$, referred to as the matrix for streamflow of the previous month, can be obtained through a three-step procedure that follows. Step 1: create an N-by-12 matrix of zeros, denoted by $\mathbf{Y}_{-1}$, as illustrated by Figure 4b. Step 2: assign the submatrix of $\mathbf{Y}$ formed by all rows and columns 1 through 11 to the submatrix of $\mathbf{Y}_{-1}$ formed by all rows and columns 2 through 12, as illustrated by Figure 4c. Step 3: assign the submatrix of $\mathbf{Y}$ formed by rows 1 through $N-1$ and column 12 to the submatrix of $\mathbf{Y}_{-1}$ formed by rows 2 through N and column 1, as illustrated by Figure 4d. Upon recursively repeating the above three steps, other matrices for streamflow of l prior months are obtained and designated by $\mathbf{Y}_{-l}$, $l = 1, 2, \ldots, 12$. One may have realized that the above operations artificially introduced varying numbers of zeroes in the first rows of these matrices. We remove the first rows of **Y** and $\mathbf{Y}_{-l}$, $l = 1, 2, \ldots, 12$, leading to matrices with $N-1$ rows, as demonstrated by Figures 4e and 4f. In order not to complicate the notation, in subsequent descriptions we shall use N rather than $N-1$ to refer to the number of rows of these matrices. From $\mathbf{Y}_{-l}$, $l = 1, 2, \ldots, 12$, the matrix for the dynamic aggregated flow of past 12 months can be obtained as

$$\mathbf{Y}_a = \sum_{l=1}^{12} \mathbf{Y}_{-l} \tag{8}$$

The foregoing matrix operations assure that the tth columns of $\mathbf{Y}_{-1}$ and $\mathbf{Y}_a$ store observations of the first and second variables in the feature vector $\mathbf{X}_t$ in equation (4), respectively.

Standardizing conditional feature matrices: Let $\mathbf{m}_{-1}$ and $\mathbf{s}_{-1}$ denote, respectively, the vectors of sample mean and standard deviation of $\mathbf{Y}_{-1}$, each with a size of 1-by-12. Note that the tth element of $\mathbf{m}_{-1}$ is computed from the data in the tth column of $\mathbf{Y}_{-1}$. The same holds true for $\mathbf{s}_{-1}$. Similarly, denote the corresponding vectors of $\mathbf{Y}_a$ as $\mathbf{m}_a$ and $\mathbf{s}_a$, respectively. Standardization of $\mathbf{Y}_{-1}$ is achieved by subtracting the tth element of $\mathbf{m}_{-1}$ from data in the tth column of $\mathbf{Y}_{-1}$ and then dividing by the tth element of $\mathbf{s}_{-1}$. Likewise, standardization of $\mathbf{Y}_a$ can be accomplished. For simplicity, let $\mathbf{Z}_{-1}$ and $\mathbf{Z}_a$ denote the standardized matrices of $\mathbf{Y}_{-1}$ and $\mathbf{Y}_a$, respectively.

Identifying hyperparameters of regression models: For a given month t, $t = 1, 2, \ldots, 12$, data in the tth columns of **Y**, $\mathbf{Z}_{-1}$, and $\mathbf{Z}_a$ are fed to identify the hyperparameters of the regression models. The identified hyperparameters are then stored in different arrays $\mathbf{P}^m$, $m = 1, 2, \ldots, 7$, one for each regression model. Since different models may have different numbers of hyperparameters (see the numbers in Figure 2), the size of $\mathbf{P}^m$ may vary from one model to another. For instance, for model f_1, the size of $\mathbf{P}^1$ is 2-by-12 with one column for each calendar month; whereas for f_2, the size of $\mathbf{P}^2$ is 1-by-12. Note that since f_5 has no hyperparameters, $\mathbf{P}^5$ is an empty array.

Calculating historical residuals: To store historical residuals, we first create seven matrices $\mathbf{e}^m$, $m = 1, 2, \ldots, 7$, each with a size of N-by-12 (the same as that of **Y**, $\mathbf{Y}_{-1}$, and $\mathbf{Y}_a$). We take $\mathbf{e}^1$ as an example to explain how these residual matrices are filled. For a given month t, $t = 1, 2, \ldots, 12$, first access the tth columns of $\mathbf{P}^1$, **Y**, $\mathbf{Z}_{-1}$, $\mathbf{Z}_a$, denoted by $\mathbf{P}^1(:, t)$, $\mathbf{Y}(:, t)$, $\mathbf{Z}_{-1}(:, t)$, $\mathbf{Z}_a(:, t)$, respectively. With the hyperparameters in $\mathbf{P}^1(:, t)$, model f_1 is calibrated with data in $\mathbf{Y}(:, t)$ and $[\mathbf{Z}_{-1}(:, t), \mathbf{Z}_a(:, t)]$. The calibrated model is then used to make regression on $[\mathbf{Z}_{-1}(:, t), \mathbf{Z}_a(:, t)]$. Upon dividing each element of $\mathbf{Y}(:, t)$ by the corresponding regression estimate thus obtained, finally the multiplicative residuals can be yielded and stored in the tth

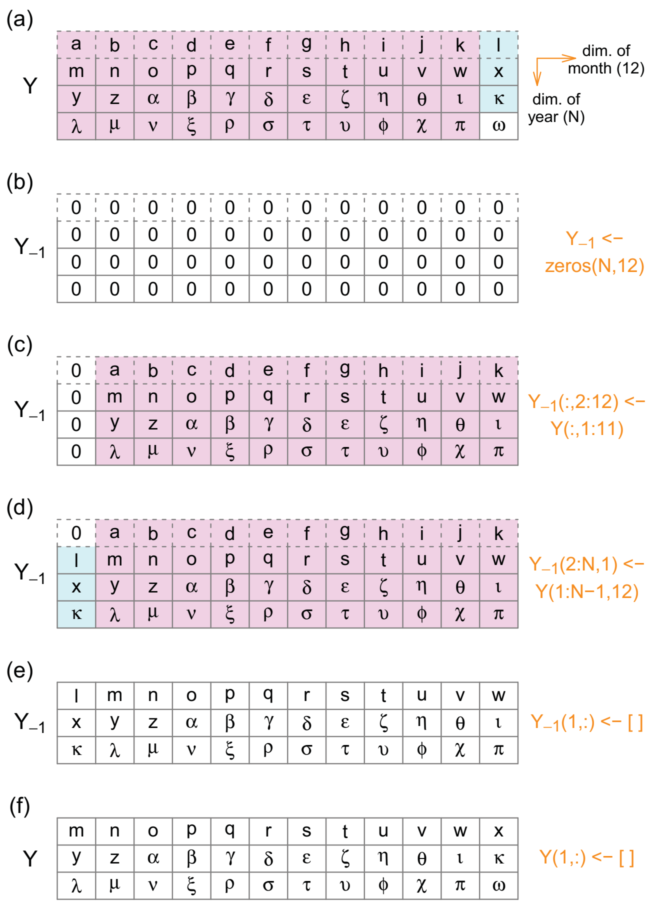

Figure 4. Schematic illustration for (a) formulating the historical streamflow matrix and for (b–f) creating the conditional feature matrices.

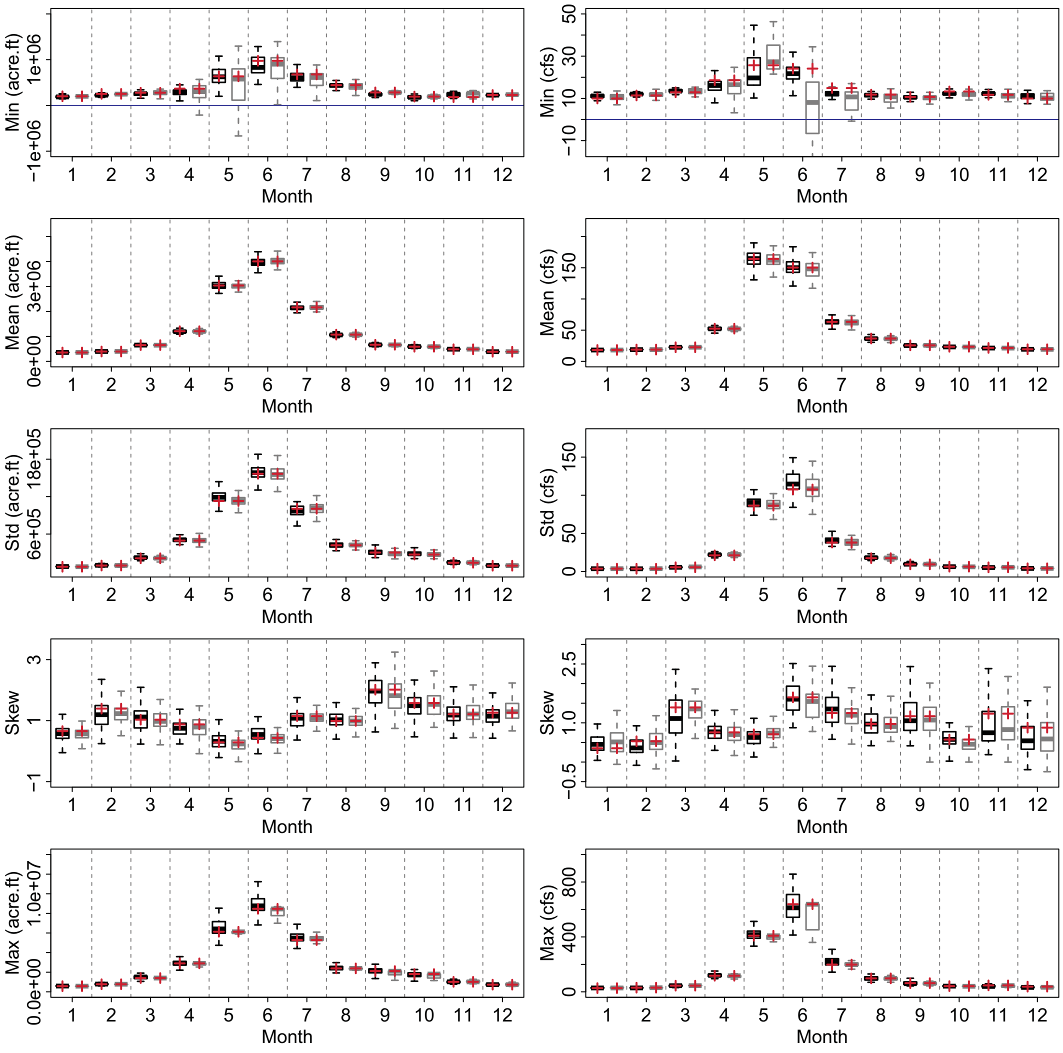

Figure 5. Box plots of basic summary statistics (minimum, mean, standard deviation, skewness, and maximum) of MRS-simulated (black) and MKNN-simulated (gray) streamflow sequences, calculated at the monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

column of $\mathbf{e}^1$. Following the same vein, the other residual matrices can be filled. Note that there exists an one-to-one corresponding relationship between the elements of $\mathbf{Y}$ and $\mathbf{e}^m$, $m = 1, 2, \ldots, 7$.

Thus far, it seems somewhat tedious due to a number of matrix operations and notations. It must, however, be noted that first the above data manipulations are computationally very efficient and can be easily accomplished with only a few commands in the environment of MATLAB; second, the data organization is clear and convenient for the following simulation experiments, as will become obvious in section 3.2. Note that not all the mentioned arrays need to be passed to the simulation phase. We end this section by enumerating those which will be used afterward: (1) $\mathbf{Y}$, (2) $\mathbf{m}_{-1}$, (3) $\mathbf{s}_{-1}$, (4) $\mathbf{m}_a$, (5) $\mathbf{s}_a$, (6) $\mathbf{Z}_{-1}$, (7) $\mathbf{Z}_a$, (8) $\mathbf{P}^m$, and (9) $\mathbf{e}^m$, $m = 1, 2, \ldots, 7$. All other intermediate terms can be omitted hereinafter.

3.2. Synthetic Streamflow Generating

Suppose now that we want to generate a synthetic sequence of N^s years. We first create a row vector $\mathbf{Y}^s$ to store streamflows to be simulated. The length of $\mathbf{Y}^s$ is $12N^s$. Next, replicate N^s copies of each array retained in the first phase, i.e., $\mathbf{Y}$, $\mathbf{m}_{-1}$, $\mathbf{s}_{-1}$, $\mathbf{m}_a$, $\mathbf{s}_a$, $\mathbf{Z}_{-1}$, $\mathbf{Z}_a$, $\mathbf{P}^m$, and $\mathbf{e}^m$, $m = 1, 2, \ldots, 7$. Subsequently, these copies are concatenated one after another to form an expanded array. For instance, for $\mathbf{Y}$, the expanded array will

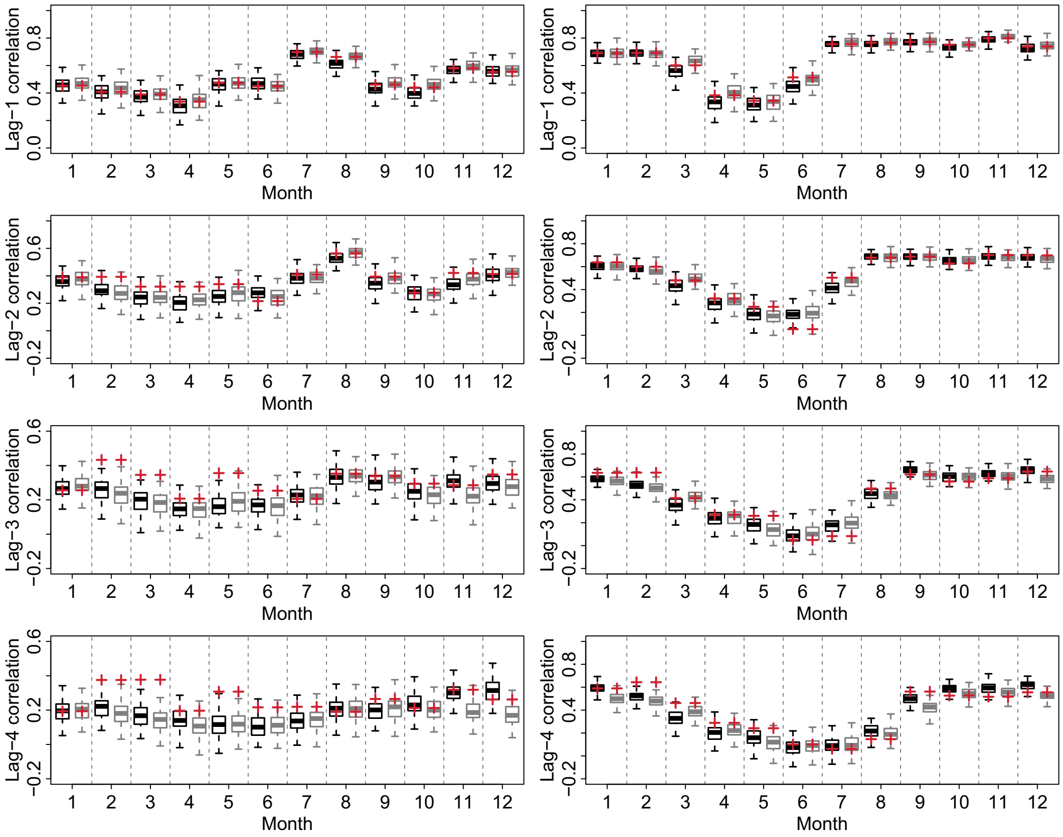

Figure 6. Box plots of lag-1 to lag-4 autocorrelations of MRS-simulated (black) and MKNN-simulated (gray) streamflow sequences, calculated at the monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations.

be [**Y**, **Y**, . . . , **Y**] with a size of N-by-12N^s. For simplicity without introducing confusion, we will use the same notations for the expanded arrays. Streamflow simulation proceeds as follows:

1. Randomly select any integer r from 1, 2, . . . , N.

2. Access the data in row r of **Y** and store these data in order in the first 12 entries of $\mathbf{Y}^s$.

3. Now start with $t = 13$.

4. Using the simulated data, compute values of the conditional feature variables at the current step as $\mathbf{x}_t^s = \left[y_{t-1}^s, y_a^s\right]$, in which

$$y_{t-1}^s = \mathbf{Y}^s(t-1)$$

$$y_a^s = \sum_{i=t-12}^{t-1} \mathbf{Y}^s(i)$$

5. Standardize the conditional feature point $\mathbf{x}_t^s$ as

$$\mathbf{z}_t^s = \frac{\mathbf{x}_t^s - [\mathbf{m}_{-1}(t), \mathbf{m}_a(t)]}{[\mathbf{s}_{-1}(t), \mathbf{s}_a(t)]}$$

Of particular note is that the above division is performed on an element-by-element basis.

6. Access data in the tth column of **Y** and store them in an column vector $\mathbf{y}_t$, i.e., $\mathbf{y}_t = \mathbf{Y}(:, t)$.

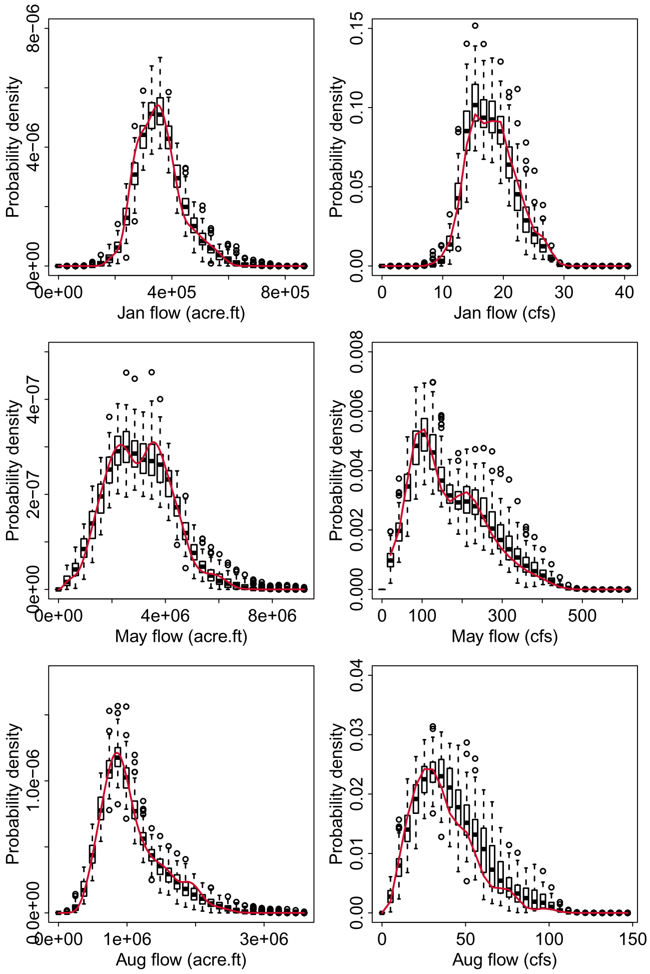

Figure 7. Selected comparison examples of gamma kernel density estimates of observed and MRS-simulated streamflow sequences, for (left) gauge 09380000 and (right) gauge 10234500. The solid red curves are for the density estimates of historical observations and the boxes represent the range of density estimates of the 100 simulated sequences. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

7. Access data in the *t*th columns of $\mathbf{Z}_{-1}$ and $\mathbf{Z}_a$ and store them in an *N*-by-2 matrix $\mathbf{z}_t$, i.e., $\mathbf{z}_t = [\mathbf{Z}_{-1}(:, t), \mathbf{Z}_a(:, t)]$.

8. Access data in the *t*th columns of $\mathbf{P}^m$ and store them separately in different arrays $\mathbf{p}_t^m$, i.e., $\mathbf{p}_t^m \ = \mathbf{P}^m(:, t)$, $m = 1, 2, \ldots, 7$.

9. Feed $\mathbf{y}_t$ and $\mathbf{z}_t$ to calibrate each regression model f_m with hyperparameters $\mathbf{p}_t^m$, $m = 1, 2, \ldots, 7$. Then given $\mathbf{z}_t^s$, estimate M_t^m with the calibrated model f_m, $m = 1, 2, \ldots, 7$.

10. Naively average M_t^m, $m = 1, 2, \ldots, 7$, and denote the averaged value as M_t^s.

11. Calculate the Euclidean distance between $\mathbf{z}_t^s$ and each point in $\mathbf{z}_t$ as

$$d_i = \|\mathbf{z}_t^s - \mathbf{z}_t(i, :)\|, \quad i = 1, 2, \ldots, N$$

Store the resulting values in order in an *N*-by-1 column vector $\mathbf{d}$.

12. Find the location indices of the first *k* ($k = \sqrt{N}$) smallest values in $\mathbf{d}$. Store the indices in a column vector **idx**.

13. Access the historical residuals stored in the submatrix of $\mathbf{e}^m$ formed by rows specified by **idx** and column *t* and store them separately in different column vectors $\mathbf{e}_t^m$, i.e., $\mathbf{e}_t^m = \mathbf{e}^m(\mathbf{idx}, t)$, $m = 1, 2, \ldots, 7$.

14. Smooth the discrete residuals $\mathbf{e}_t^m$, $m = 1, 2, \ldots, 7$, using gamma KDE with the smoothing parameter estimated by h_2.

15. Generate a random number e_t^s from the smoothed density.

16. Assign $M_t^s \times e_t^s$ as the simulated streamflow at the current step and store this value in the *t*th entry of $\mathbf{Y}^s$.

17. Update *t* to $t + 1$ and repeat steps 4–16.

18. Repeat step 17 until *t* is equal to $12N^s$.

4. Model Evaluation and Comparison

Among many streamflow data sets (from 17 gauges) with which we have tested the developed model, all with satisfactory performance, in the following we will only report the results for two representative gauges and reserve those of the other gauges in the supporting information. It is noted that almost all the observations and remarks reported in the following are in general applicable to other gauges (Figures S1–S33 in

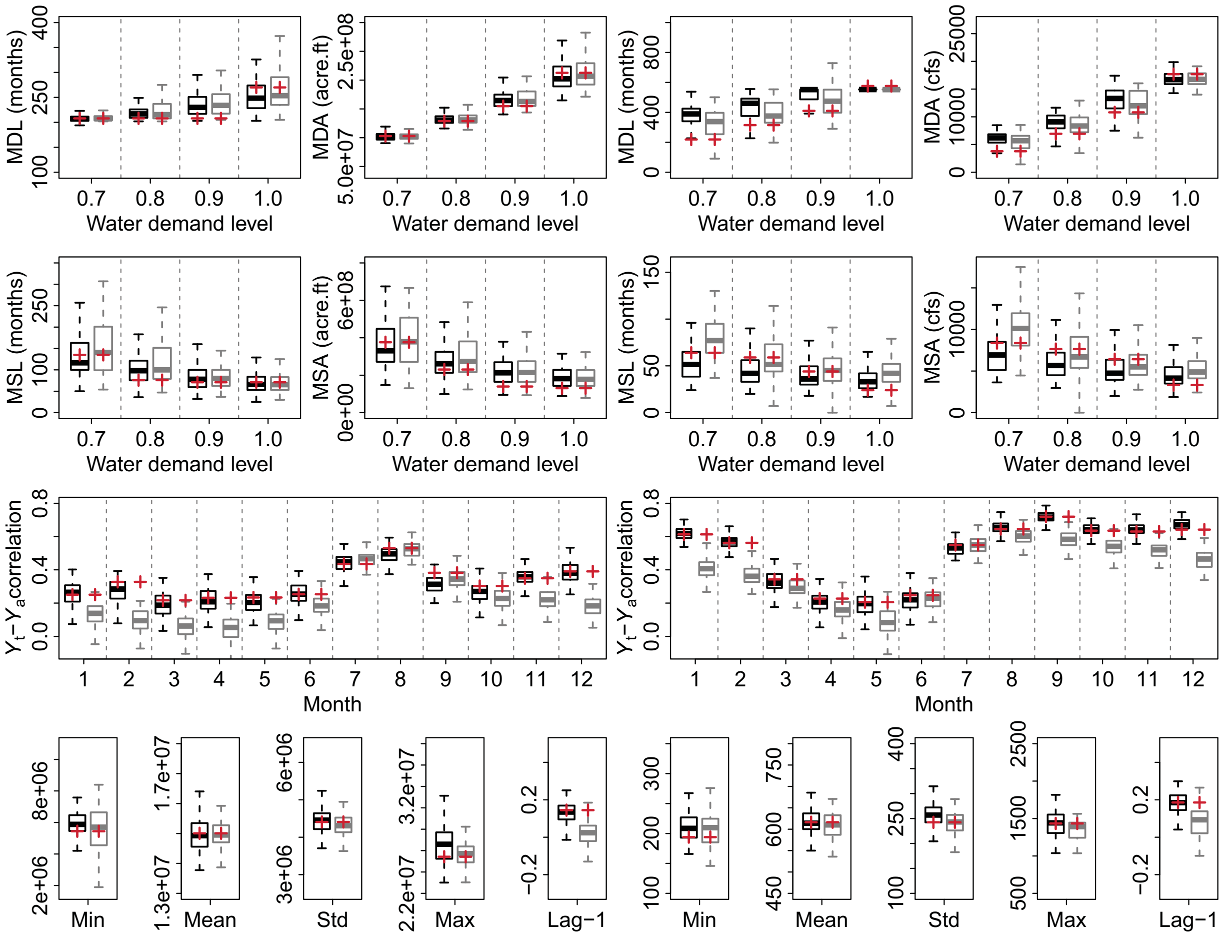

Figure 8. Box plots of maximum drought length and amount (first row), maximum surplus length and amount (second row), Y_t–Y_a correlation (third row), and annual scale basic summary statistics and lag-1 autocorrelation of MRS-simulated (black) and MKNN-simulated (gray) streamflow sequences, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

the supporting information). The two representative gauges are gauge 09380000 (U.S. Geological Survey gauge number) on the Colorado River at Lees Ferry, Arizona, with monthly records from 1906 to 2007 downloaded from http://www.usbr.gov/lc/region/g4000/NaturalFlow/index.html; gauge 10234500 on the Beaver River near Beaver, Utah, with records from 1915 to 2010 downloaded from http://waterdata.usgs.gov/usa/nwis/nwisman/?site_no=10234500&agency_cd=USGS. We focused on these two gauges because they have been extensively studied to test different streamflow simulation approaches and also because they have quite different characteristics. Detailed description of these data sets can be found elsewhere in *Prairie and Russell* [2005], *Lee et al.* [2010], *Sharma et al.* [1997], and *Sharma and O'Neill* [2002], and will not be repeated here.

For simplicity in description, we shall refer to the enhanced multimodel regression-sampling algorithm as MRS. Table S1 in the supporting information presents the identified hyperparameters of the regression models for the two representative gauges. We applied MRS independently at each gauge. One hundred sequences, each with length equal to the corresponding effective historical record (101 years for gauge 09380000 and 95 years for gauge 10234500), were generated for each gauge. The model performance was evaluated from the following three aspects: (1) reproduce basic summary and autocorrelation statistics; (2) reproduce the overall historical distribution; and (3) reproduce interannual variability.

Basic summary and autocorrelation statistics used here include (1) mean, (2) standard deviation, (3) skewness, (4) minimum, (5) maximum, and (6) lag-1 to lag-4 autocorrelations at monthly scale. Nonparametric Kendall's τ was used as the autocorrelation measure. Reproduction of the overall historical distribution was

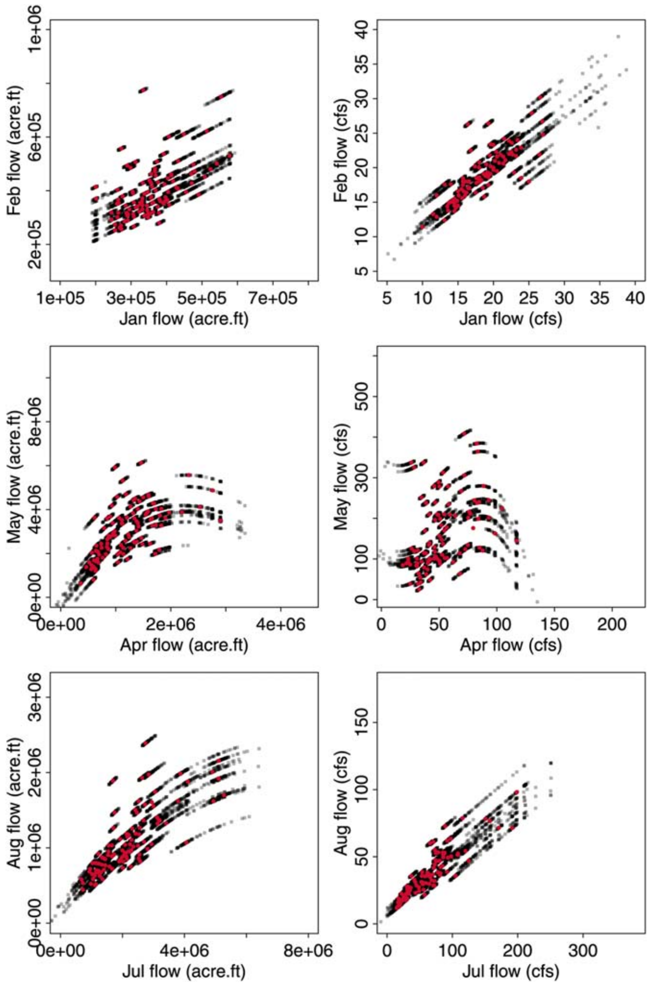

Figure 9. Scatter plots of observed (red) and MKNN-simulated (gray) streamflow sequences of selected adjacent months (January–February: top row; April–May: middle row; and July–August: bottom row), for (left) gauge 09380000 and (right) gauge 10234500. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

checked by comparing the gamma kernel smoothed density of the simulated data against that of the observed data. Drought and surplus statistics, Y_t–Y_a correlation, as well as the above-enumerated basic summary statistics but calculated at annual scale were adopted as interannual variability measures. Drought and surplus statistics include maximum drought length (MDL) and amount (MDA), maximum surplus length (MSL), and amount (MSA). MDL (MSL) is defined as the longest period of deficit (excess) relative to the water demand level in a given period of years. Correspondingly, MDA (MSA) is the maximum deficit (excess) obtained from all the drought (surplus) events that occurred in that period [*Salas and Lee*, 2010]. Static water demand levels, expressed as fractions (0.7, 0.8, 0.9, and 1.0) of the historical mean, were used in this research.

4.1. Preliminary Evaluation

Figures 5 and 6 present basic summary and autocorrelation statistics of observed and simulated sequences for the representative gauges. Apparently, MRS effectively reproduced all the summary statistics for both gauges (Figure 5). One point worth noting is with respect to the minimum and maximum statistics, which have been recognized as being relatively hard to simulate. For each gauge and each month, the historical minimum and maximum fell inside the respective box of simulations. More importantly, the simulated data were nearly symmetrically distributed. This is an interesting feature of MRS compared with most other nonparametric models, in which the simulated minimum and maximum values present, respectively, L-shaped and J-shaped distributions, as will be discussed later. This suggests MRS' decent extrapolation capability. Figure 6 shows that MRS, by construction, can reproduce lag-1 autocorrelation; for high-order autocorrelations, MRS' performance is, to varying degrees, tied to the order of the autocorrelation, gauges and months, despite all examined autocorrelations being reasonably reproduced for most months.

To check how well MRS reproduces the historical distribution, we compared the gamma KDE of simulated data against that of historical data. Figure 7 presents representative results for January, May, and August for both gauges. In general, MRS can capture complex distributional properties, such as strong asymmetry and bimodality. Reproducing bimodality is valuable for river basins affected by seasonal precipitation concentrations or those jointly fed by both rainfall and snowmelt.

Figure 8 compares observed and simulated drought and surplus statistics, Y_t–Y_a correlation, and summary statistics and lag-1 autocorrelation calculated at the annual scale. In general, MRS did quite a good job of

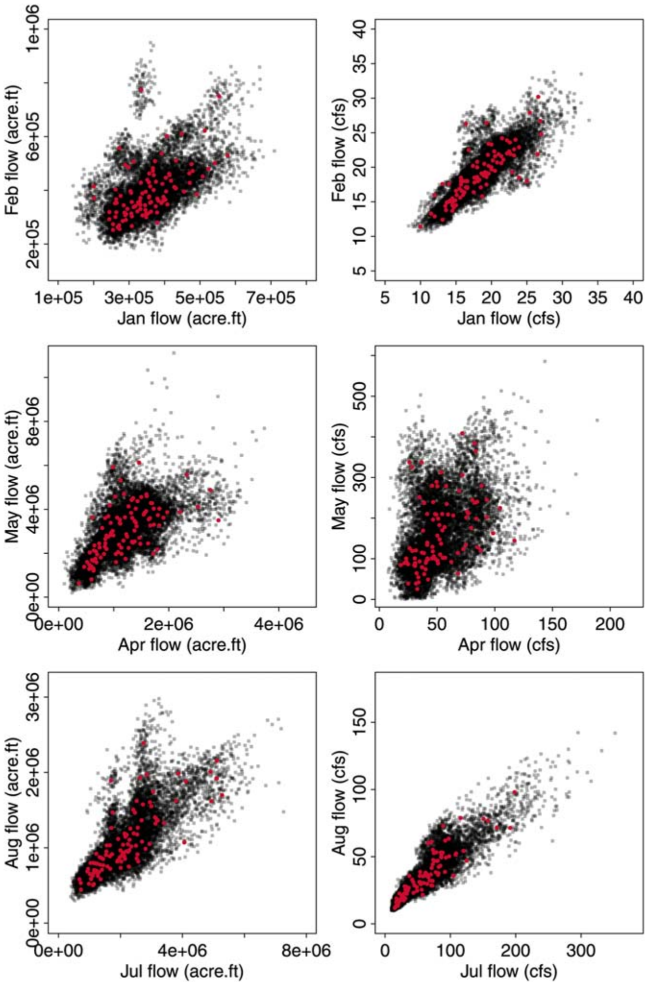

Figure 10. Scatter plots of observed (red) and MRS-simulated (gray) streamflow sequences of selected adjacent months (January–February: top row; April–May: middle row; and July–August: bottom row), for (left) gauge 09380000 and (right) gauge 10234500. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

preserving these interannual statistics at both gauges. Due to the inclusion of Y_a as an additional feature variable, it is not surprising to observe MRS' skill in reproducing the Y_t–Y_a correlation. Although MRS has not been explicitly structured for reproducing annual summary statistics and lag-1 autocorrelation, all of them were resolved fairly well. Since interannual variability is often related to sustained droughts or periods of high flows, the remarkable performance of MRS in this respect implies that it could serve as a reliable aid for catchment water resources management and drought analysis. As was noted, the most important contribution of MRS, making it distinctive from most existing nonparametric models, is that it can simulate diverse streamflow realizations unseen in historical records. This feature is more readily appreciated through comparisons with other nonparametric models, presented in the following subsections.

4.2. Comparison With MKNN

MRS represents an enhanced version of MKNN. It will be interesting to see their comparison. Figures 5 and 6 also include the corresponding results for MKNN. MRS exhibited more or less the same skills as MKNN in reproducing mean, standard deviation, skewness, and lag-1 to lag-4 autocorrelations. Comparing with MRS, MKNN is less apt at capturing extreme statistics. In particular, a number of negatives were simulated, especially in wet seasons, such as April, May, and Jun for gauge 09380000 and June and July for gauge 10234500, consistent with the previous findings of *Salas and Lee* [2010]. In MRS, the multiplicative decomposition mechanism ensures the avoidance of negative simulations unless, as was mentioned, the simulated expectation M_t happens to be negative. It is experienced that for almost all the gauges only few instances (less than 3) where the simulated expectation is negative might occur (due to overextrapolation) in one or two of the regression models. It never happens that all seven models simultaneously yield negative M_t. In practice, it is, therefore, safe to simply neglect the negative estimates and take average over the others, attributing to the sevenfold insurance of MRS. One can also see that MKNN-simulated maximum values presented a J-shaped distribution in most of the months at both gauges, implying its limited extrapolation capacity.

Pertaining to reproduction of historical distribution, MRS and MKNN performed comparably well. We therefore omit their comparison here. The observed and MKNN-simulated interannual variability statistics were included in Figure 8 as well. MRS showed similar performance to MKNN in reproducing drought and surplus statistics and the summary statistics at the annual scale. MRS outperformed MKNN in preserving the Y_t–Y_a

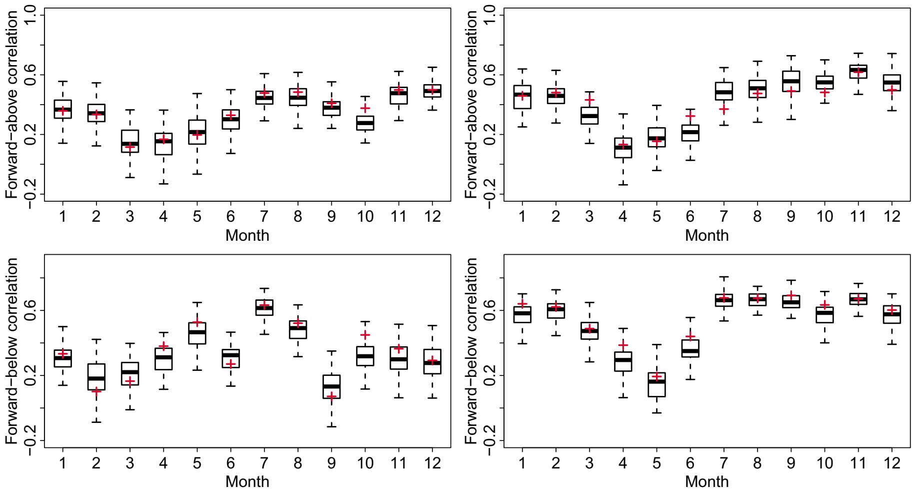

Figure 11. Box plots of forward-above (top row) and forward-below (bottom row) state-dependent autocorrelations of MRS-simulated streamflow sequences, calculated at monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations.

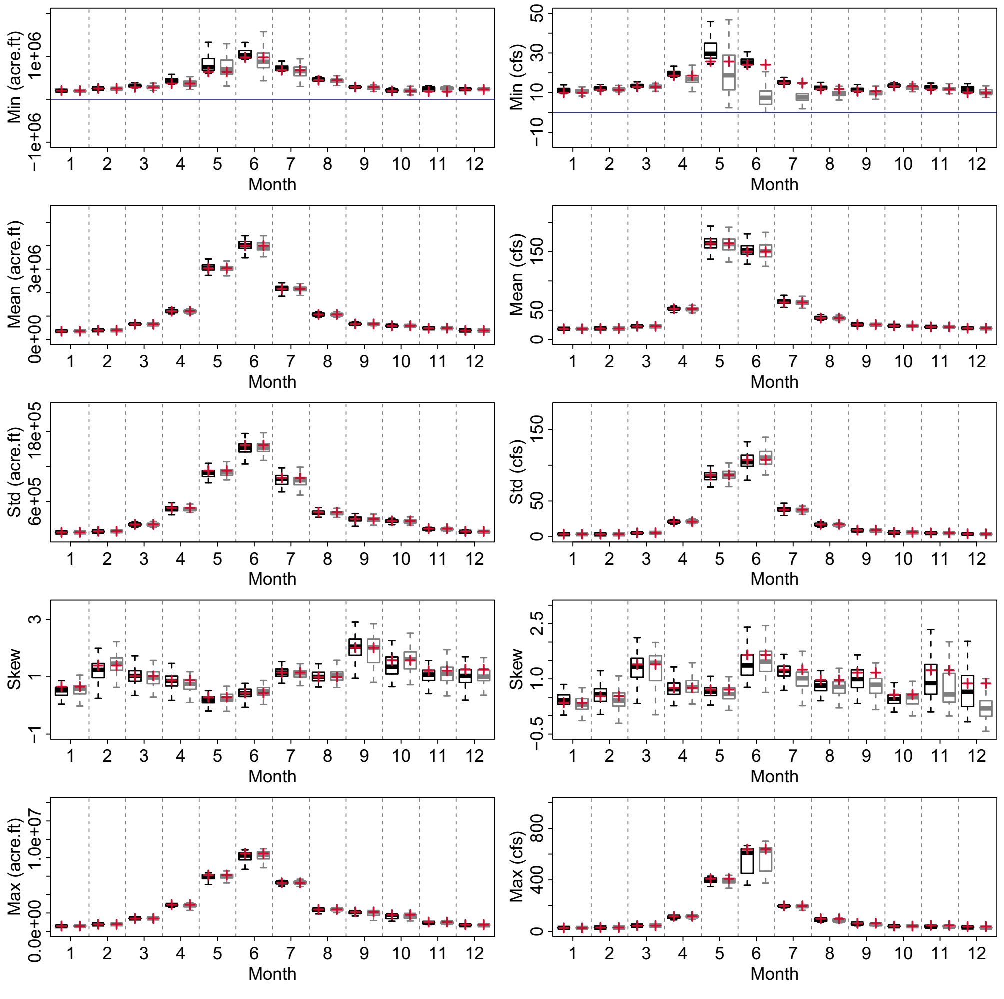

Figure 12. Box plots of basic summary statistics (minimum, mean, standard deviation, skewness, and maximum) of NPL-simulated (black) and KGKA-simulated (gray) streamflow sequences, calculated at monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

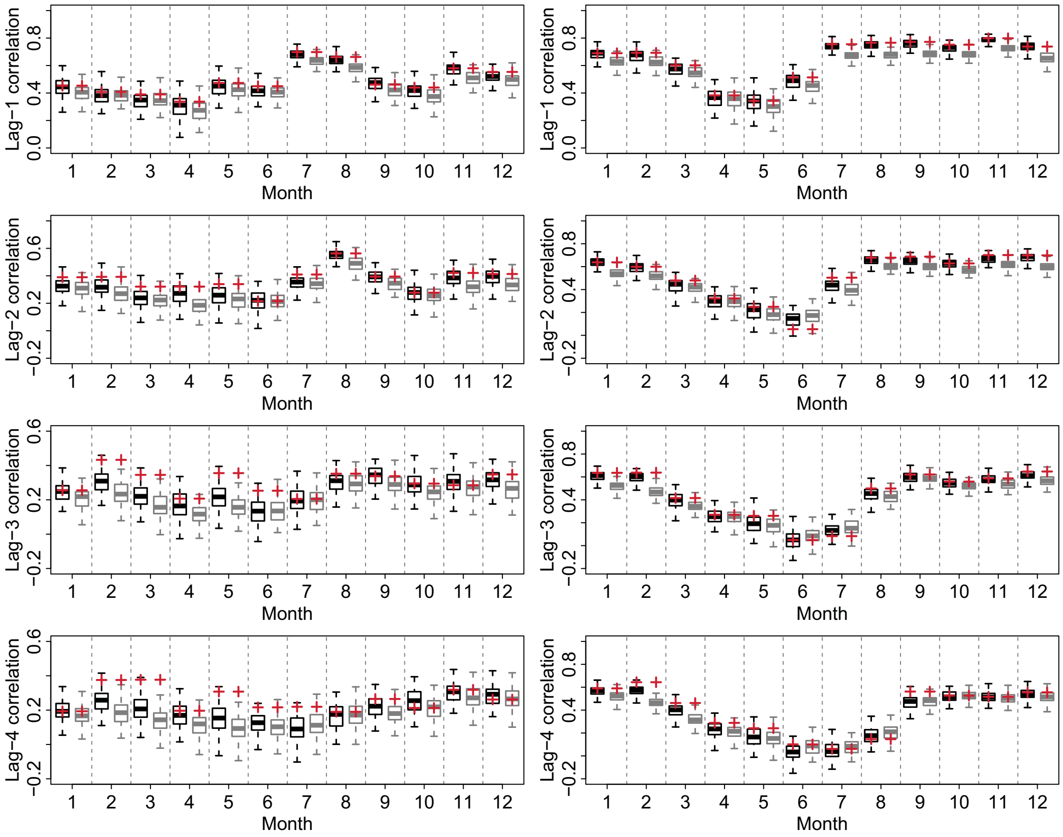

Figure 13. Box plots of lag-1 to lag-4 autocorrelations of NPL-simulated (black) and KGKA-simulated (gray) streamflow sequences, calculated at monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations.

correlation and the lag-1 autocorrelation of annual streamflow. In detail, both of the two statistics was consistently underestimated throughout the year by MKNN. The improved performance of MRS stems from the inclusion of Y_a as an additional feature variable.

In contrast to the original KNN model of *Lall and Sharma* [1996], MKNN enables simulating values unseen in historical records [*Prairie et al.*, 2006]. However, its extrapolation ability is rather limited, as can be seen from the scatter plots in Figure 9. Simulations from MKNN tend to distribute around the observations following a linear or nonlinear pattern, depending on the underlying relationship of adjacent months' streamflows. This implies that the simulated sequences represent too close a resemblance to historical records. With such simulations, it is hardly possible to obtain a comprehensive picture about alternative management policies and plans for a water resources system.

Comparing Figures 9 and 10, one can easily appreciate how good MRS is at generating rich streamflow scenarios unseen in historical records, underlining the added value of the multimodel simulation scheme. Careful inspection of Figure 10 reveals MRS' skill in capturing nonlinear autocorrelation, like most nonparametric models; see the two bottom left plots. This feature is further distilled in Figure 11, which presents the observed and simulated state-dependent correlations (above-and-forward and below-and-forward correlations) for both gauges. The definitions for these state-dependent correlations can be found in *Sharma et al.* [1997]. Obviously, MRS, by construction, is able to reproduce these state-dependent correlations, implying its ability in capturing nonlinear autocorrelation. The Monte Carlo simulation experiment in Appendix A also confirmed this point.

4.3. Comparison With Other Long-Term Nonparametric Alternatives

We also compared MRS with another two nonparametric models reported in the literature. One is the Gaussian kernel-based nonparametric model (NPL) of *Sharma and O'Neill* [2002]. The other is the KNN resampling

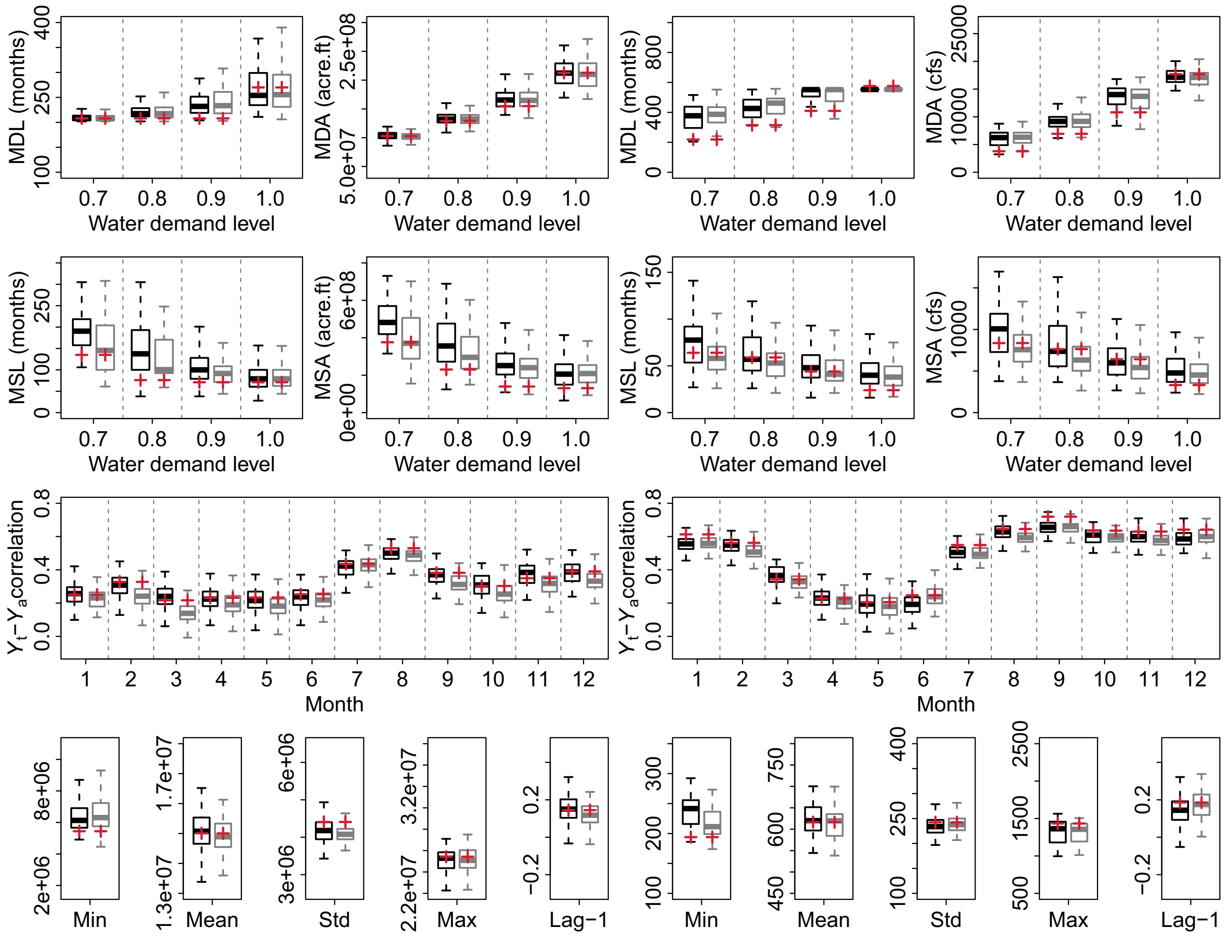

Figure 14. Box plots of maximum drought length and amount (first row), maximum surplus length and amount (second row), Y_t–Y_a correlation (third row), and annual scale basic summary statistics and lag-1 autocorrelation of NPL-simulated (black) and KGKA-simulated (gray) streamflow sequences, for (left) gauge 09380000 and (right) gauge 10234500. Red pluses represent the corresponding statistics of historical observations. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

with gamma kernel perturbation with aggregated variable (KGKA) developed in *Salas and Lee* [2010]. Both are especially structured for preserving the interannual variability. It is noted that NPL involves inverting the covariance matrix of Y_t, Y_{t-1}, and Y_a. Since the magnitude of monthly flow (Y_t and Y_{t-1}) may be greatly smaller than that of Y_a, the covariance matrix sometimes is ill conditioned, which will in turn bring troubles to simulation. To address this issue, we first standardized Y_t, Y_{t-1}, and Y_a, and then run NPL with the standardized data, finally back transformed the simulations into the original scale. As such, reported results about NPL might slightly differ from those in *Sharma and O'Neill* [2002] and in *Salas and Lee* [2010]. This should not be confusing.

Basic summary statistics of observed and simulated sequences from NPL and KGKA are summarized in Figure 12. In general, the three models (MRS, NPL, and KGKA) performed comparably well in reproducing monthly mean, standard deviation, and skewness for both gauges. This is slightly different from the findings in *Sharma and O'Neill* [2002] and in *Salas and Lee* [2010] that NPL inflates the standard deviation. However, such a feature was not seen here, partly because of the standardization carried out in our simulation experiment. Although the three models reasonably reproduced the minimum and maximum for both gauges, the advantage of MRS over the other two is obvious, confirming again the promising extrapolation capability of MRS. Both MRS and NPL reproduced lag-1 autocorrelation reasonably well, whereas KGKA demonstrated underestimation throughout the year (Figure 13). This is likely due to the fact that KGKA estimates the gamma kernel smoothing parameter using all historical data rather than those inside the *k*-neighborhood. As to the higher-order autocorrelations, all models did a reasonably good job, with MRS and NPL slightly better than KGKA.

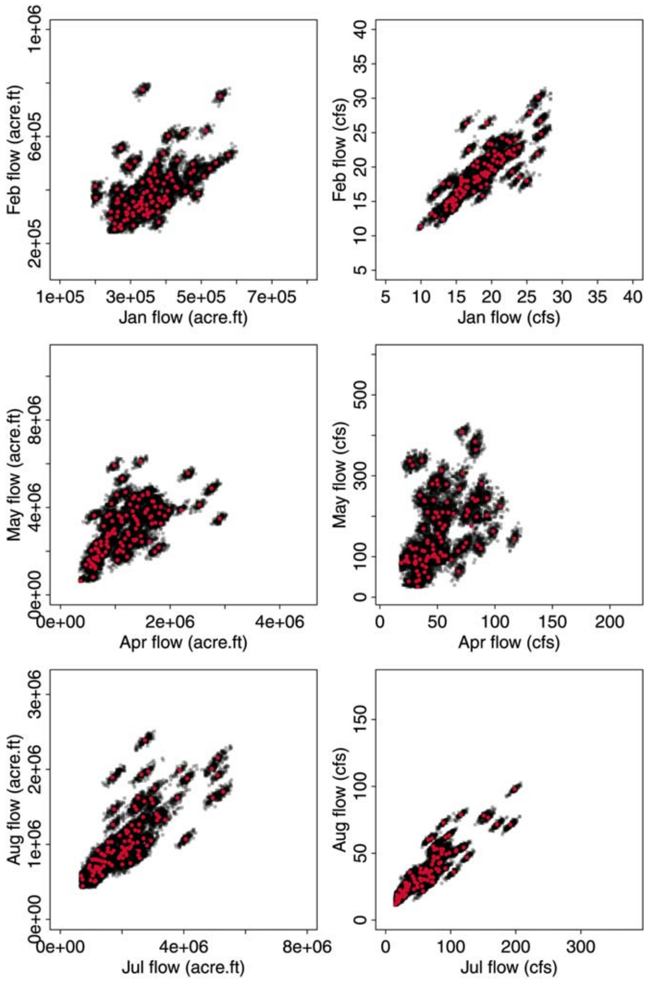

Figure 15. Scatter plots of observed (red) and NPL-simulated (gray) streamflow sequences of selected adjacent months (January–February: top row; April–May: middle row; and July–August: bottom row), for (left) gauge 09380000 and (right) gauge 10234500. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

Careful inspection of Figures 8 and 14 reveals that MRS performed somewhat superior to NPL and KGKA in reproducing drought and surplus statistics for both gauges, despite all of them being able to reasonably capture these statistics. Although KGKA uses the same feature variables as those in MRS and NPL, it underestimated Y_t–Y_a correlation across most months for both gauges due to perhaps the same reason for the underestimation of monthly lag-1 autocorrelation. Compared with MRS, NPL and KGKA overestimated the annual minimum for both gauges.

Figures 15 and 16 are the same as Figures 9 and 10, but for NPL and KGKA models, respectively. Compared with MKNN, NPL and KGKA did improve significantly in generating rich streamflow scenarios. However, the improvement was still not adequate, as signified by the isolated blocks concentrated around extreme observations. KGKA performed better than NPL. Neither KGKA nor NPL could simulate streamflow scenarios as rich as those by MRS, highlighting the most important advantage of MRS over NPL and KGKA. Simulating streamflow sequences with diverse scenarios is of great importance when using the simulations to formulate optimal reservoir operation policies.

5. Concluding Remarks

We present a multimodel regression-sampling algorithm (MRS) for single-site monthly streamflow simulation. MRS is specifically designed for alleviating the issue that typical nonparametric models simulate streamflow sequences always exhibiting too close a resemblance to historical record. In order to retain the merits of nonparametric model in capturing complex distributional and dependence characteristics, we build the enhanced multimodel simulation scheme upon the simple nonparametric regression-sampling framework introduced by *Prairie et al*. [2006] but with several innovative adaptations in an attempt to correct the recognized shortcomings of the original model.

To appreciate the improvements, we applied the enhanced model to 16 stream gauges in the Colorado River basin and compared with other existing nonparametric alternatives such as the NPL model [*Sharma and O'Neill*, 2002] and the KGKA model [*Salas and Lee*, 2010]. We draw two major conclusions from this research:

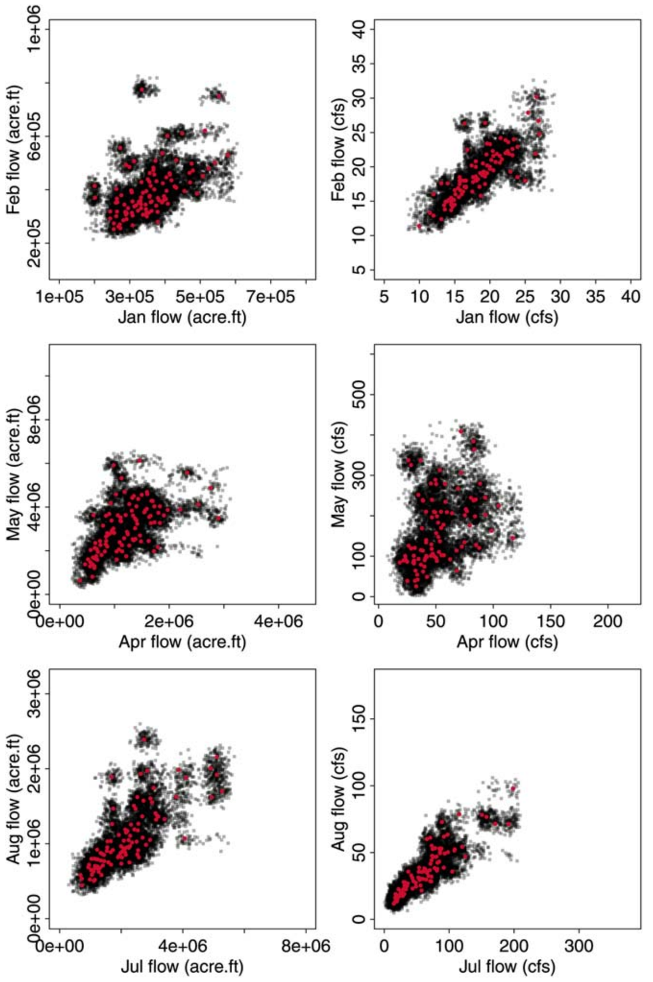

Figure 16. Scatter plots of observed (red) and KGKA-simulated (gray) streamflow sequences of selected adjacent months (January–February: top row; April–May: middle row; and July–August: bottom row), for (left) gauge 09380000 and (right) gauge 10234500. Different units were used for the two gauges simply because the original data acquired from the corresponding websites were used without unit transformation.

1. Similar to other nonparametric models, MRS is capable of capturing complex distributional and dependence characteristics of monthly streamflow, which generally cannot be expected from parametric models.

2. Comparing with the examined nonparametric models, the most pleasing point is that MRS is more apt at generating streamflow sequences with more diverse scenarios in addition to its better performance in several other aspects (e.g., reproducing the minimum and maximum statistics).

Compared to MKNN, NPL, and KGKA, MRS involves much more parameters (9 × 12 hyperparameters in total), making its application relatively cumbersome. It should be realized that the increased number of hyperparameters should not be understood as equivalent to increasing the risk of overfitting because each of the seven regression models works independently in simulating the M_t–$\mathbf{X}_t$ relationship.

MRS cannot simulate nonstationary streamflow, as suggested by the results of a split sample simulation experiment following *Srinivas and Srinivasan* [2001a, 2001b] (Figure 17). To address this issue, external forcing covariates should be included. Exploring which covariate should be used is beyond the scope of this research. The technique discussed in *Sharma and Mehrotra* [2014] provides useful hints to this problem.

At the current stage, MRS is only suitable for single-site simulation. How to extend it for multisite simulation deserves further research. In addition, it would be interesting to study the performance of MRS at more gauges from different regions and make a more comprehensive comparison with other semiparametric models such as those introduced in *Srinivas and Srinivasan* [2001a, 2001b, 2005a, 2005b, 2006], *Srivastav et al.* [2011], and *Keylock* [2012].

To capture the interannual variability of monthly streamflow, MRS uses dynamic aggregated streamflow of past 12 months as a conditional variable. Another possible way is to apply the multimodel regression-sampling framework directly in the Fourier domain following *Keylock* [2012]. Further effort is required to investigate such a possibility.

Moreover, it should be noted that we have used statistical regression models to approximate the M_t–$\mathbf{X}_t$ relationship. Actually, this is not required. Any models are feasible as long as they can generalize the underlying relationship, not matter they are statistical models, or simple conceptual hydrologic models, or complex physically based models. By changing different surrogate models and/or different feature variables, MRS is

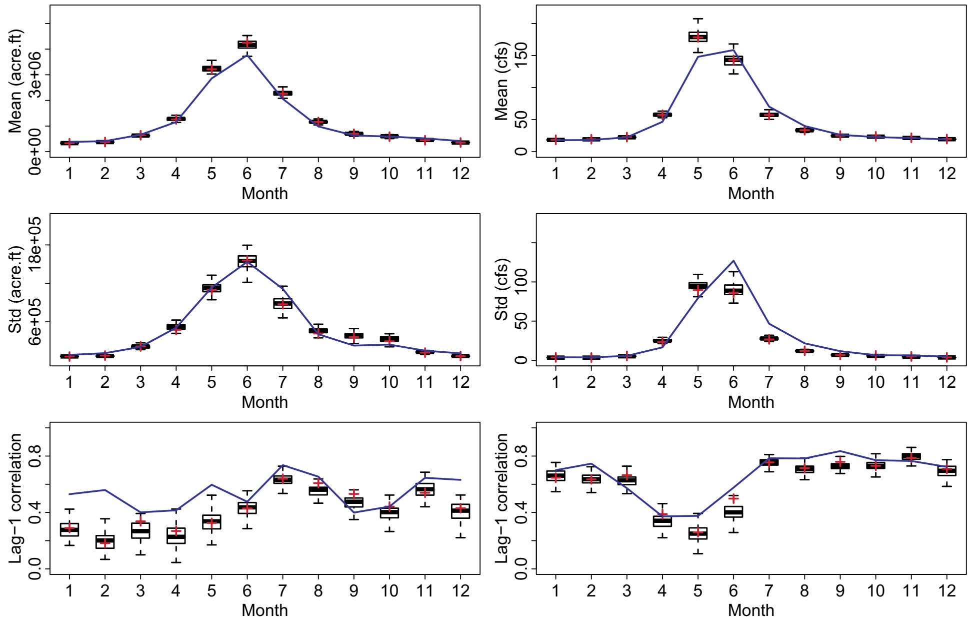

Figure 17. Comparison of mean, standard deviation, and lag-1 autocorrelation of MRS-simulated streamflow sequences during the first-half time period with those of the observed sequences during the first-half period (red pluses) and the second-half period (blue solid lines), calculated at monthly scale, for (left) gauge 09380000 and (right) gauge 10234500. The split-sample experiment was carried out as follows: the whole data set for each gauge was split into two halves; the first half was used to run MRS and the second half was used to test its out-of-sample performance. The distinctive within-sample and out-of-sample performance of MRS implies that the application of MRS to nonstationary streamflow time series should be cautioned.

applicable to other purposes, such as hydrologic simulation, forecasting, downscaling, and postprocessing deterministic forecasts into probabilistic forecasts.

Appendix A: MRS Testing With Synthetic Data

The following Monte Carlo simulation experiment was designed for testing MRS' ability in capturing nonlinear autocorrelations. The experiment was carried out as follows. First, 200 random data pairs were generated from a bivariate distribution whose PDF is given by $f(y_t, y_{t-1}) = N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ if $y_{t-1} \leq 0$; and $f(y_t, y_{t-1}) = N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ otherwise, with $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ being the PDF of a bivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. To mimic a state-dependent nonlinear correlation between Y_t and Y_{t-1}, the bivariate distribution is parameterized as follows:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 6.5 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.69 & 1.01 \\ 1.01 & 1.69 \end{bmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 5.5 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} 1.69 & -0.85 \\ -0.85 & 1.69 \end{bmatrix}$$

The contour plots in Figure A1 show the PDF of the bivariate distribution. It can be clearly seen that Y_t and Y_{t-1} are positively correlated with a relatively strong correlation coefficient (0.6) when $Y_{t-1} \leq 0$, whereas

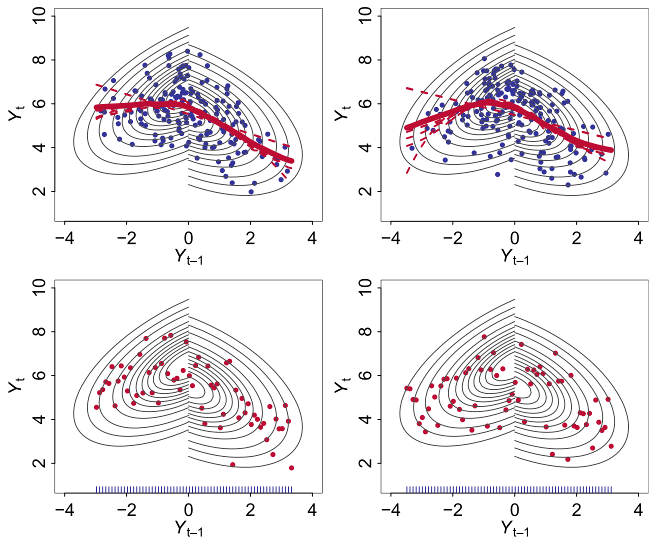

Figure A1. The left and right columns are corresponding to results of two Monte Carlo simulation trials. The contour plots are the probability density function of the assumed bivariate distribution. Blue points are the 200 random data pairs generated from the underlying distribution. Dashed red lines are the estimated regression curves by different models, with one for each regression model. Solid red curves are the ensemble averages of the seven individual curves. Red points are MRS simulations given the conditional feature points shown by the rug plots.

they are negatively correlated with a relatively weak correlation coefficient (-0.5) when $Y_{t-1} > 0$. The scatter points in the top left subplot of Figure A1 represent the generated 200 random data pairs.

With these data pairs, the second step is to estimate the regression curves between Y_t and Y_{t-1} with the seven regression models. The estimated curves are shown by dashed red lines in the top left subplot, one for each regression model. It is seen that models f_2 and f_5 resulted in the same linear curve, whereas the other models, by construction, resulted in somewhat different nonlinear curves. The ensemble average of the seven models is shown by the bold solid curve. Apparently, the averaged curve in general captured the underlying nonlinear relationship, even though models f_2 and f_5 did not.

To see if MRS-simulated random numbers are indeed consistent with the underlying distribution, in the third step we partitioned the interval between the minimum and maximum of the random values of Y_{t-1} that were obtained in the first step into equal-sized subintervals, each with a size of 0.1; we used the endpoints of these subintervals as conditional feature variables (blue rug in the bottom left subplot) to run MRS. The red scatter points in the bottom left subplot represent MRS-simulated values. One can easily see that MRS-simulated values were in good agreement with the underlying distribution. We repeated the above three steps once and obtained similar results, as shown in Figure A1 (right). Overall, this relatively simple Monte Carlo experiment provides a proof for the promising capability of MRS in simulating nonlinear autocorrelations.

Acknowledgments
Editors and three anonymous referees raised constructive comments. Their help in improving this paper is gratefully acknowledged.

References

Bras, R. L., and I. Rodriguez-Iturbe (1985), *Random Functions and Hydrology*, Dover, N. Y.

Buishand, T. A., and T. Brandsma (2001), Multisite simulation of daily precipitation and temperature in the Rhine Basin by nearest neighbor resampling, *Water Resour. Res.*, *37*(11), 2761–2776, doi:10.1029/2001WR000291.

Fernandez, B., and J. D. Salas (1990), Gamma autoregressive models for stream flow simulation, *J. Hydrol. Eng.*, *116*(11), 1403–1414.

Hansen, B. E. (2007), Least-squares model averaging, *Econometrica*, *75*, 1175–1189.

Hao, Z., and V. P. Singh (2011), Single-site monthly streamflow simulation using entropy theory, *Water Resour. Res.*, *47*, W09528, doi:10.1029/2010WR010208.

Keylock, C. J. (2012), A resampling method for generating synthetic hydrological time series with preservation of cross-correlative structure and higher-order properties, *Water Resour. Res.*, *48*, W12521, doi:10.1029/2013WR011923.

Lall, U., and A. Sharma (1996), A nearest neighbor bootstrap for resampling hydrologic time series, *Water Resour. Res.*, *32*(3), 679–693, doi:10.1029/95WR02966.

Lall, U., Y.-I. Moon, H.-H. Kwon, and K. Bosworth (2006), Locally weighted polynomial regression: Parameter choice and application to forecasts of the Great Salt Lake, *Water Resour. Res.*, *42*, W05422, doi:10.1029/2004WR003782.

Lee, T., and J. D. Salas (2011), Copula-based stochastic simulation of hydrological data applied to Nile River flows, *Hydrol. Res.*, *42*(4), 318–330.

Lee, T., J. D. Salas, and J. Prairie (2010), An enhanced nonparametric streamflow disaggregation model with generic algorithm, *Water Resour. Res.*, *46*, W08545, doi:10.1029/2009WR007761.

Lee, T., T. B. M. J. Ouarda, and C. Jeong (2012), Nonparametric multivariate weather generator and an extreme value theory for bandwidth selection, *J. Hydrol.*, *452–453*, 161–171.

Loader, C. (1999), *Local Regression and Likelihood*, Springer, N. Y.

Maheepala, S., and B. J. C. Perera (1996), Monthly hydrologic data generation by disaggregation, *J. Hydrol.*, *178*, 277–291.

Nazemi, A., H. S. Wheater, K. P. Chun, and A. Elshorbagy (2013), A stochastic reconstruction framework for analysis water resources system vulnerability to climate-induced changes in river flow regime, *Water Resour. Res.*, *49*, 291–305, doi:10.1029/2012WR012755.

Prairie, J., and C. Russell (2005), Natural flow and salt computation methods, Bur. of Reclam., U.S. Dep. of the Inter., Salt Lake City, Utah. [Available at http://www.usbr.gov/lc/region/g4000/NaturalFlow/Final-MethodsCmptgNatFlow.pdf.]

Prairie, J., B. Rajagopalan, T. Fulp, and E. Zagona (2006), Modified K-NN model for stochastic streamflow simulation, *J. Hydrol. Eng.*, *11*(4), 371–378.

Salas, J., and T. Lee (2010), Nonparametric simulation of single-site seasonal streamflows, *J. Hydrol. Eng.*, *15*(4), 284–296.

Sharma, A., and R. Mehrotra (2014), An information theoretic alternative to model a natural system using observational information alone, *Water Resour. Res.*, *49*, doi:10.1002/2013WR013845.

Sharma, A., and R. O'Neill (2002), A nonparametric approach for representing interannual dependence in monthly streamflow sequences, *Water Resour. Res.*, *38*(7), 1100, doi:10.1029/2011WR000953.

Sharma, A., D. Tarboton, and U. Lall (1997), Streamflow simulation: A nonparametric approach, *Water Resour. Res.*, *33*(2), 291–308, doi:10.1029/96WR02839.

Srinivas, V. V., and K. Srinivasan (2001a), Post-blackening approach for modeling periodic streamflows, *J. Hydrol.*, *241*(3–4), 221–269.

Srinivas, V. V., and K. Srinivasan (2001b), A hybrid stochastic model for multiseason streamflow simulation, *Water Resour. Res.*, *37*(10), 2537–2549.

Srinivas, V. V., and K. Srinivasan (2005a), Hybrid moving block bootstrap for stochastic simulation of multi-site multi-season streamflow, *J. Hydrol.*, *302*(1–4), 307–330.

Srinivas, V. V., and K. Srinivasan (2005b), Matched block bootstrap for resampling multiseason hydrologic time series, *Hydrol. Processes*, *19*(18), 3659–3682.

Srinivas, V. V., and K. Srinivasan (2006), Hybrid matched-block bootstrap for stochastic simulation of multi-season streamflow, *J. Hydrol.*, *329*(1–2), 1–15.

Srivastav, R. K., K. Srinivasan, and K. P. Sudheer (2011), Simulation-optimization framework for multi-season hybrid stochastic models, *J. Hydrol.*, *404*, 209–225.