

## A nearest neighbor bootstrap for resampling hydrologic time series

Upmanu Lall and Ashish Sharma

Utah Water Research Laboratory, Utah State University, Logan

**Abstract.** A nonparametric method for resampling scalar or vector-valued time series is introduced. Multivariate nearest neighbor probability density estimation provides the basis for the resampling scheme developed. The motivation for this work comes from a desire to preserve the dependence structure of the time series while bootstrapping (resampling it with replacement). The method is data driven and is preferred where the investigator is uncomfortable with prior assumptions as to the form (e.g., linear or nonlinear) of dependence and the form of the probability density function (e.g., Gaussian). Such prior assumptions are often made in an ad hoc manner for analyzing hydrologic data. Connections of the nearest neighbor bootstrap to Markov processes as well as its utility in a general Monte Carlo setting are discussed. Applications to resampling monthly streamflow and some synthetic data are presented. The method is shown to be effective with time series generated by linear and nonlinear autoregressive models. The utility of the method for resampling monthly streamflow sequences with asymmetric and bimodal marginal probability densities is also demonstrated.

### Introduction

Autoregressive moving average (ARMA) models for time series analysis are often used by hydrologists to generate synthetic streamflow and weather sequences to aid in the analysis of reservoir and drought management. Hydrologic time series can exhibit the following behaviors, which can be a problem for commonly used linear ARMA models: (1) asymmetric and/or multimodal conditional and marginal probability distributions; (2) persistent large amplitude variations at irregular time intervals; (3) amplitude-frequency dependence (e.g., the amplitude of the oscillations increases as the oscillation period increases); (4) apparent long memory (this could be related to (2) and/or (3)); (5) nonlinear dependence between  $x_t$  versus  $x_{t-\tau}$  for some lag  $\tau$ ; and (6) time irreversibility (i.e., the time series plotted in reverse time is “different” from the time series in forward time). The physics of most geophysical processes is time irreversible. Streamflow hydrographs often rise rapidly and attenuate slowly, leading to time irreversibility.

*Kendall and Dracup* [1991] have argued for simple resampling schemes, such as the index sequential method, for streamflow simulation in place of ARMA models, suggesting that the ARMA streamflow sequences usually do not “look” like real streamflow sequences. An alternative is presented in the work of *Yakowitz* [Yakowitz, 1973, 1979; Schuster and Yakowitz, 1979; Yakowitz, 1985, 1993; Karlsson and Yakowitz, 1987a, b], *Smith et al.* [1992], *Smith* [1991], and *Tarboton et al.* [1993] who consider the time series as the outcome of a Markov process and estimate the requisite probability densities using nonparametric methods. A resampling technique or bootstrap for scalar or vector-valued, stationary, ergodic time series data that recognizes the serial dependence structure of the time series is presented here. The technique is nonpara-

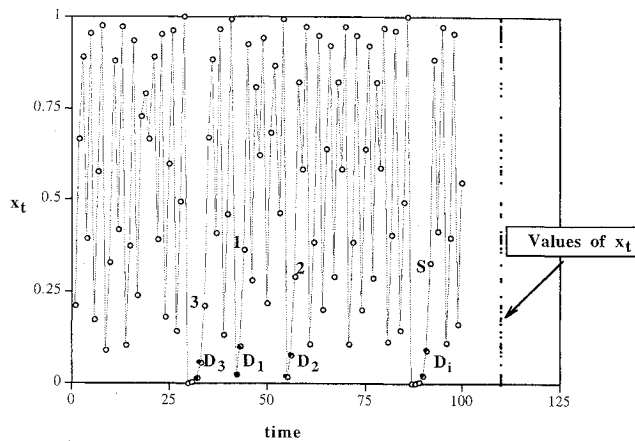
metric; that is, no prior assumptions as to the distributional form of the underlying stochastic process are made.

The bootstrap [Efron, 1979; Efron and Tibishirani, 1993] is a technique that prescribes a data resampling strategy using the random mechanism that generated the data. Its applications for estimating confidence intervals and parameter uncertainty are well known [see Härdle and Bowman, 1988; Tasker, 1987; Woo, 1989; Zucchini and Adamson, 1989]. Usually the bootstrap resamples with replacement from the empirical distribution function of independent, identically distributed data. The contribution of this paper is the development of a bootstrap for dependent data that preserves the dependence in a probabilistic sense. This method should be useful for the Monte Carlo analysis of a variety of hydrologic design and operation problems where time series data on one or more interacting variables are available.

The underlying concept of the methodology is introduced through Figure 1. Consider that the serial dependence is limited to the two previous lags; that is,  $x_t$  depends on the two prior values  $x_{t-1}$  and  $x_{t-2}$ . Denote this ordered pair, or bituple, at a time  $t_i$  by  $\mathbf{D}_i$ . Let the corresponding succeeding value be denoted by  $S$ . Consider the  $k$  nearest neighbors of  $\mathbf{D}_i$  as the  $k$  bituples in the time series that are closest in terms of Euclidean distance to  $\mathbf{D}_i$ . The first three nearest neighbors are marked as  $\mathbf{D}_1$ ,  $\mathbf{D}_2$ , and  $\mathbf{D}_3$ . The expected value of the forecast  $S$  can be estimated as an appropriate weighted average of the successors  $x_t$  (marked as 1, 2, and 3, respectively) to these three nearest neighbors. The weights may depend inversely on the distance between  $\mathbf{D}_i$  and its  $k$  nearest neighbors  $\mathbf{D}_1$ ,  $\mathbf{D}_2$ ,  $\dots$ ,  $\mathbf{D}_k$ . A conditional probability density  $f(x|\mathbf{D}_i)$  may be evaluated empirically using a nearest neighbor density estimator [see Silverman, 1986, p. 96] with the successors  $x_1, \dots, x_k$ . For simulation the  $x_t$  can be drawn randomly from one of the  $k$  successors to the  $\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_k$  using this estimated conditional density. Here, this operation will be done by resampling the original data with replacement. Hence the procedure developed is termed a nearest neighbor time series bootstrap. In

Copyright 1996 by the American Geophysical Union.

Paper number 95WR02966.  
0043-1397/96/95WR-02966\$05.00



**Figure 1.** A time series from the model  $x_{t+1} = (1 - 4(x_t - 0.5)^2)$ . This is a deterministic, nonlinear model with a time series that looks random. A forecast of the successor to the bituple  $D_i$ , marked as  $S$  in the figure, is of interest. The “patterns” or bituples of interest are the filled circles, near the three nearest neighbors  $D_1$ ,  $D_2$ , and  $D_3$  to the pattern  $D_i$ . The successors to these bituples are marked as 1, 2, and 3, respectively. Note how the successor (1) to the closest nearest neighbor ( $D_1$ ) is closest to the successor ( $S$ ) of  $D_i$ . A sense of the marginal probability distribution of  $x_t$  is obtained by looking at the values of  $x_t$  shown on the right side of the figure. As the sample size  $n$  increases, the sample space of  $x$  gets filled in between 0 and 1, such that the sample values are arbitrarily close to each other, but no value is ever repeated exactly.

summary, one finds  $k$  patterns in the data that are “similar” to the current pattern and then operates on their respective successors to define a local regression, conditional density, or resampling.

The nearest neighbor probability density estimator and its use with Markov processes is reviewed in the next section. The resampling algorithm is described after that. Applications to synthetic and streamflow data are then presented.

## Background

It is natural to pursue nonparametric estimation of probability densities and regression functions through weighted local averages of the target function. This is the foundation for nearest neighbor methods. The recognition of the nonlinearity of the underlying dynamics of geophysical processes, gains in computational ability, and the availability of large data sets have spurred the growth of the nonparametric literature. The reader is referred to work by *Silverman* [1986], *Eubank* [1988], *Härdle* [1989, 1990], and *Scott* [1992] for accessible monographs. *Györfi et al.* [1989] provide a theoretical account that is relevant for time series analysis. *Lall* [1995] surveys hydrologic applications. For time series analysis a moving block bootstrap (MBB) was presented by *Kunsch* [1989]. Here a block of  $m$  observations is resampled with replacement, as opposed to a single observation in the bootstrap. Serial dependence is preserved within, but not across, a block. The block length  $m$  determines the order of the serial dependence that can be preserved. Objective procedures for the selection of the block length  $m$  are evolving. Strategies for conditioning the MBB on other processes (e.g., runoff on rainfall) are not obvious. Our investigations indicated that the MBB may not be able to

reproduce the sample statistics as well as nearest neighbor bootstrap presented here.

The  $k$  nearest neighbor ( $k$ -nn) density estimator is defined as [*Silverman*, 1986, p. 96]

$$f_{\text{NN}}(\mathbf{x}) = \frac{k/n}{V_k(\mathbf{x})} = \frac{k/n}{c_d r_k^d(\mathbf{x})} \quad (1)$$

where  $k$  is the number of nearest neighbors considered,  $d$  is the dimension of the space,  $c_d$  is the volume of a unit sphere in  $d$  dimensions ( $c_1 = 2$ ,  $c_2 = \pi$ ,  $c_3 = 4\pi/3$ ,  $\dots$ ,  $c_d = [\pi^{d/2}/\Gamma(d/2 + 1)]$ ),  $r_k(\mathbf{x})$  is the Euclidean distance to the  $k$ th-nearest data point, and  $V_k(\mathbf{x})$  is the volume of a  $d$ -dimensional sphere of radius  $r_k(\mathbf{x})$ .

This estimator is readily understood by observing that for a sample of size  $n$ , we expect approximately  $\{nf(\mathbf{x})V_k(\mathbf{x})\}$  observations to lie in the volume  $V_k(\mathbf{x})$ . Equating this to the number observed, that is,  $k$ , completes the definition.

A generalized nearest neighbor density estimator [*Silverman*, 1986, p. 97], defined in (2), can improve the tail behavior of the nearest neighbor density estimator by using a monotonically and possibly rapidly decreasing smooth kernel function.

$$f_{\text{GNN}}(\mathbf{x}) = \frac{1}{r_k^d(\mathbf{x})n} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{r_k(\mathbf{x})}\right) \quad (2)$$

The “smoothing” parameter is the number of neighbors used,  $k$ , and the tail behavior is determined by the kernel  $K(\mathbf{t})$ . The kernel has the role of a weight function (data vectors  $\mathbf{x}_i$  closer to the point of estimate  $\mathbf{x}$  are weighted more), and can be chosen to be any valid probability density function. Asymptotically, under optimal mean square error (MSE) arguments,  $k$  should be chosen proportional to  $n^{4/(d+4)}$  for a probability density that is twice differentiable. However, given a single sample from an unknown density, such a rule is of little practical utility. The sensitivity to the choice of  $k$  is somewhat lower as a kernel that is monotonically decreasing with  $r_k(\mathbf{x})$  is used. A new kernel function that weights the  $j$ th neighbor of  $\mathbf{x}_i$  using a kernel that depends on the distance between  $\mathbf{x}_i$  and its  $j$ th neighbor is developed in the resampling methodology section.

*Yakovitz* (references cited earlier) developed a theoretical basis for using nearest neighbor and kernel methods for time series forecasting and applied them in a hydrologic context. In those papers *Yakovitz* considers a finite order, continuous parameter Markov chain as an appropriate model for hydrologic time series. He observes that discretization of the state space can quickly lead to either an unmanageable number of parameters (the curse of dimensionality) or poor approximation of the transition functions, while the ARMA approximations to such a process call for restrictive distributional and structural assumptions. Strategies for the simulation of daily flow sequences, one-step-ahead prediction, and the conditional probability of flooding (flow crossing a threshold) are exemplified with river flows and shown to be superior to ARMA models. Seasonality is accommodated by including the calendar date as one of the predictors. *Yakovitz* claims that this continuous parameter Markov chain approach is capable of reproducing any possible Hurst coefficient. Classical ARMA models are optimal only under squared error loss and only for linear operations on the observables. The loss/risk functions associated with hydrologic decisions (e.g., whether to declare a flood warning or not) are usually asymmetric. The nonparametric framework allows attention to be focused directly on

calculating these loss functions and evaluating the consequences.

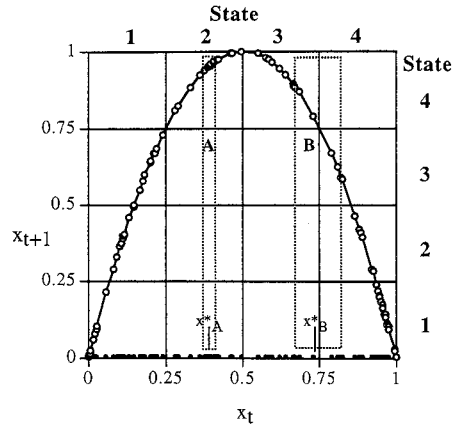
The example of Figure 1 is now extended to show how the nearest neighbor method is used in the Markov framework. One step Markov transition functions are considered. The relationship between  $x_{t+1}$  and  $x_t$  is shown in Figure 2. The correlation between  $x_t$  and  $x_{t+1}$  is 0, even though there is clear-cut dependence between the two variables.

Consider an approximation of the model in Figure 1 by a multistate, first-order Markov chain, where transitions from, say, state 1 for  $x_t$  (0 to 0.25 in Figure 2) to states 1, 2, 3, or 4 for  $x_{t+1}$  are of interest. The state  $i$  to state  $j$  transition probability  $p_{ij}$  is evaluated by counting the relative fraction of transitions from state  $i$  to state  $j$ . The estimated transition probabilities depend on the number of states chosen as well as their actual demarcation (e.g., one may need a nonuniform grid that recognizes variations in data density). For the nonlinear model used in our example, a fine discretization would be needed. Given a finite data set, estimates of the multistate transition probabilities may be unreliable. Clearly, this situation is exacerbated if one considers higher dimensions for the predictor space. Further, a reviewer has observed that a discretization of a continuous space Markov process is not necessarily Markov.

Now consider the nearest neighbor approach. Consider two conditioning points  $x_A^*$  and  $x_B^*$ . The  $k$  nearest neighbors of these points are in the dashed windows  $A$  and  $B$ , respectively. The neighborhoods are seen to adapt to variations in the sampling density of  $x_t$ . Since such neighborhoods represent moving windows (as opposed to fixed windows for the multistate Markov chain) at each point of estimate, we can expect reduced bias in the recovery of the target transition functions. The one step transition probabilities at  $x_t^*$  can be obtained through an application of the nearest neighbor density estimator to the  $x_{t+1}$  values that fall in windows like  $A$  and  $B$ . A conditional bootstrap of the data can be obtained by resampling from this set of  $x_{t+1}$  values. Since each transition probability estimate is based on  $k$  points, the problem faced in a multistate Markov chain model of sometimes not having an adequate number of events or state transitions to develop an estimate is circumvented.

### The Nearest Neighbor Resampling Algorithm

In this section a new algorithm for generating synthetic time series samples by bootstrapping (i.e., resampling the original time series with replacement) is presented. Denote the time series by  $x_t, t = 1, \dots, n$ , and assume a known dependence structure, that is, which and how many lags the future flow will depend on. This conditioning set is termed a “feature vector,” and the simulated or forecasted value, the “successor.” The strategy is to find the historical nearest neighbors of the current feature vector and resample from their successors. Rather than resampling uniformly from the  $k$  successors, a discrete resampling kernel is introduced to weight the resamples to reflect the similarity of the neighbor to the conditioning point. This kernel decreases monotonically with distance and adapts to the local sampling density, to the dimension of the feature vector, and to boundaries of the sample space. An attractive probabilistic interpretation of this kernel consistent with the nearest neighbor density estimator is also offered. The resampling strategy is presented through the following flowchart:



**Figure 2.** A plot of  $x_{t+1}$  versus  $x_t$  for the time series generated from the model  $x_{t+1} = (1 - 4(x_t - 0.5)^2)$ . The state space for  $x$  is discretized into four states, as shown. Also shown are windows  $A$  and  $B$  with “whiskers” located over two points  $x_A^*$  and  $x_B^*$ . These windows represent a  $k$  nearest neighborhood of the corresponding  $x_t$ . In general, these windows will not be symmetric about the  $x_t$  of interest. One can think of state transition probabilities using these windows in much the same way as with the multistate Markov chain. A value of  $x_{t+1}$  conditional to point  $A$  or  $B$  can be bootstrapped by appropriately sampling with replacement one of the values of  $x_{t+1}$  that fall in the corresponding window.

1. Define the composition of the “feature vector”  $\mathbf{D}_t$  of dimension  $d$ , e.g.,

Case 1  $\mathbf{D}_t: (x_{t-1}, x_{t-2}); \quad d = 2$

Case 2  $\mathbf{D}_t: (x_{t-\tau_1}, x_{t-2\tau_1}, \dots, x_{t-M_1\tau_1}; x_{t-\tau_2}, x_{t-2\tau_2}, \dots, x_{t-M_2\tau_2});$   
 $d = M_1 + M_2$

Case 3  $\mathbf{D}_t: (x_{1-\tau_1}, \dots, x_{1-M_1\tau_1}; x_{2-\tau_2}, x_{2-\tau_2}, \dots, x_{2-M_2\tau_2});$   
 $d = M_1 + M_2 + 1$

where  $\tau_1$  (e.g., 1 month) and  $\tau_2$  (e.g., 12 months) are lag intervals, and  $M_1, M_2 \geq 0$  are the number of such lags considered in the model.

Case 1 represents dependence on two prior values. Case 2 permits direct dependence on multiple time scales, allowing one to incorporate monthly and interannual dependence. For case 3,  $x_1$  and  $x_2$  may refer to rainfall and runoff or to two different streamflow stations.

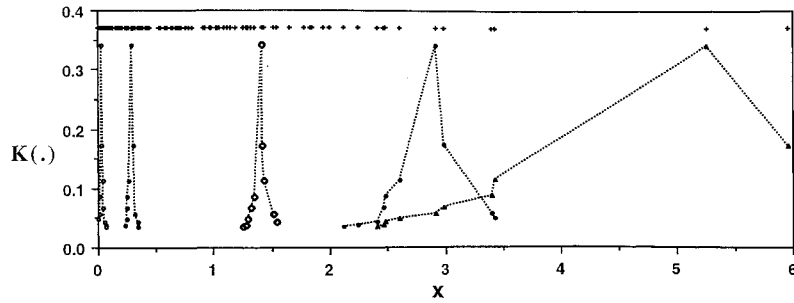
2. Denote the current feature vector as  $\mathbf{D}_t$  and determine its  $k$  nearest neighbors among the  $\mathbf{D}_t$ , using the weighted Euclidean distance

$$r_{it} = \left( \sum_{j=1}^d w_j (v_{ij} - v_{it})^2 \right)^{1/2} \tag{3}$$

where  $v_{ij}$  is the  $j$ th component of  $\mathbf{D}_i$ , and  $w_j$  are scaling weights (e.g., 1 or  $1/s_j$ , where  $s_j$  is some measure of scale such as the standard deviation or range of  $v_j$ ).

The weights  $w_j$  may be specified a priori, as indicated above, or may be chosen to provide the best forecast for a particular successor in a least squares sense [see *Yakowitz and Karlsson, 1987*].

Denote the ordered set of nearest neighbor indices by  $J_{i,k}$ . An element  $j(i)$  of this set records the time  $t$  associated with the  $j$ th closest  $\mathbf{D}_t$  to  $\mathbf{D}_i$ . Denote  $x_{j(i)}$  as the successor to  $\mathbf{D}_{j(i)}$ .



**Figure 3.** Illustration of resampling weights,  $K(j(i))$ , at selected conditioning points  $x_i$ , using  $k = 10$  with a sample of size 100 from an exponential distribution with parameter of 1. The original sampled values are shown at the top of the figure. Note how the bandwidth and kernel shape vary with sampling density.

If the data are highly quantized, it is possible that a number of observations may be the same distance from the conditioning point. The resampling kernel defined in step 3 (below) is based on the order of elements in  $J_{i,k}$ . Where a number of observations are the same distance away, the original ordering of the data can impact the ordering in  $J_{i,k}$ . To avoid such artifacts, the time indices  $t$  are copied into a temporary array that is randomly permuted prior to distance calculations and creation of the list  $J_{i,k}$ .

3. Define a discrete kernel  $K(j(i))$  for resampling one of the  $x_{j(i)}$  as follows:

$$K(j(i)) = \frac{1/j}{\sum_{j=1}^k 1/j} \tag{4}$$

where  $K(j(i))$  is the probability with which  $x_{j(i)}$  is resampled.

This resampling kernel is the same for any  $i$  and can be computed and stored prior to the start of the simulation.

4. Using the discrete probability mass function (p.m.f.)  $K(j(i))$ , resample an  $x_{j(i)}$ , update the current feature vector, and return to step 2 if additional simulated values are needed.

A similar strategy for time series forecasting is possible. An  $m$ -step-ahead forecast is obtained by using the corresponding generalized nearest neighbor regression estimator:

$$g_{\text{GNN}}(x_{i,m}) = \sum_{j=1}^k K(j(i))x_{j(i),m} \tag{5}$$

where  $x_{i,m}$  and  $x_{j(i),m}$  denote the  $m$ th successor to  $i$  and  $j(i)$ , respectively.

### Parameters of the Nearest Neighbor Resampling Method

#### Choosing the Weight Function $K(v)$

The goals for designing a resampling kernel are to (1) reduce the sensitivity of the procedure to the actual choice of  $k$ , (2) keep the estimator local, (3) have  $k$  sufficiently large to avoid simulating nearly identical traces, and (4) develop a weight function that adapts automatically to boundaries of the domain and to the dimension  $d$  of the feature vector  $\mathbf{D}_i$ . These criteria suggest a resampling kernel that decreases monotonically as  $r_{ij}$  increases.

Consider a  $d$ -dimensional ball of volume  $V(r)$  centered at  $\mathbf{D}_i$ . The observation  $\mathbf{D}_{j(i)}$  falls in this ball when the ball is

exactly of volume  $V(r_{i,j(i)})$ . Assuming that the observations are independent (which they may not be), the likelihood with which the  $j(i)$ th observation should be resampled as representative of  $\mathbf{D}_i$  is proportional to  $1/V(r_{ij(i)})$ .

Now, consider that in a small locale of  $\mathbf{D}_i$ , the local density can be approximated as a Poisson process, with constant rate  $\lambda$ . Under this assumption the expected value of  $1/V(r_{ij(i)})$  is

$$E(1/V(r_{ij(i)})) = \lambda/j \tag{6}$$

The kernel  $K(j(i))$  is obtained by normalizing these weights over the  $k$  nearest neighborhood.

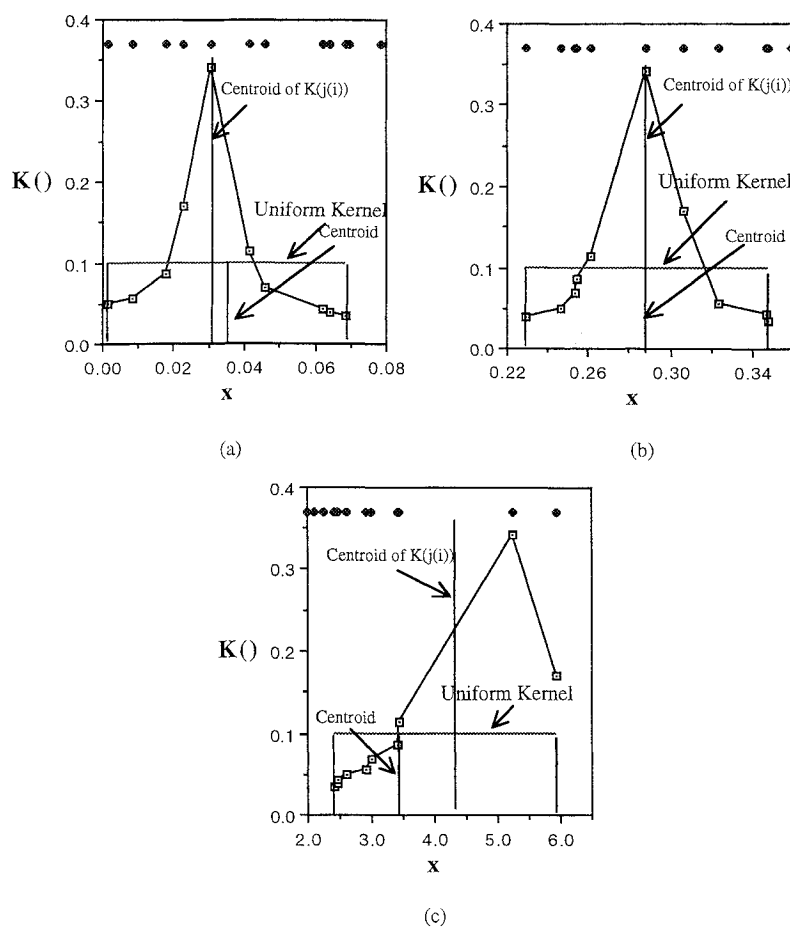
$$K(j(i)) = \frac{c\lambda/j}{\sum_{j=1}^k c\lambda/j} = \frac{1/j}{\sum_{j=1}^k 1/j} \tag{7}$$

where  $c$  is a constant of proportionality.

These weights do not explicitly depend on the dimension  $d$  of the feature vector  $\mathbf{D}_i$ . The dependence of the resampling scheme on  $d$  is implicit through the behavior of the distance calculations used to find nearest neighbors as  $d$  varies. Initially, we avoided making the assumption of a local Poisson distribution and defined  $K(j(i))$  through a normalization of  $1/V(r_{ij(i)})$ . This approach gave satisfactory results as well but was computationally more demanding. The results obtained using (7) were comparable for a given  $k$ .

The behavior of this kernel in the boundary region, the interior, and the tails is seen in Figures 3 and 4. From Figure 3, observe that the nearest neighbor method allows considerable variation in the “bandwidth” (in terms of a range of values of  $x$ ) as a function of position and underlying density. The bandwidth becomes automatically larger as the density becomes sparser and flatter. In regions of high data density (left tail or interior) the kernel is nearly symmetric (the slight asymmetry follows the asymmetry in the underlying distribution). Along the sparse right tail the kernels are quite asymmetric, as expected. Some attributes of these kernels relative to a uniform kernel (with the same  $k$ ), used by the ordinary nearest neighbor method, are shown in Figure 4.

For bounded data (e.g., streamflow that is constrained to be greater than 0), simulation of values across the boundary is often a concern. This problem is avoided in the method presented since the resampling weights are defined only for the sample points. A second problem with bounded data is that bias in estimating the target function using local averages increases near the boundaries. This bias can be recognized by



**Figure 4.** Illustration of weights  $K(j(i))$  versus weights from the uniform kernel applied at three points selected from Figure 3. The uniform kernel weights are  $1/k$  for each  $j \in J(i, k)$ . The effective centroid to each kernel for each conditioning point  $i$  (in each case with the highest value of  $K(j(i))$ ) is shown. For  $i$  in the interior of the data (Figure 4b) the centroids of both the uniform kernel and  $K(j(i))$  coincide with  $i$ . Toward the edges of the data (Figures 4a and 4c) the centroid corresponding to  $K(j(i))$  is closer to  $i$  than that for the uniform kernel. The  $K(j(i))$  are thus less biased than the uniform kernel for a given  $k$ . The kernel  $K(j(i))$  also has a lower variance than the uniform kernel for a given value of  $k$ .

observing that the centroid of the data in the window does not lie at the point of estimate. From Figure 4 note that while the kernel is biased toward the edges of the data, that is, the centroid of the kernel does not match the conditioning point, the bias is much smaller than for the uniform kernel. If one insists on kernels that are strictly positive with monotonically decreasing weights with distance, it may not be possible to devise kernels that are substantially better in terms of bias in the boundary region.

The primary advantages of the kernel introduced here are that (1) it adapts automatically to the dimension of the data, (2) the resampling weights need to be computed just once for a given  $k$ , (there is no need to recompute the weights or their normalization to 1), and (3) bad effects of data quantization or clustering (this could lead to near zero values of  $r_{i,j(i)}$  at some points) on the resampling strategy, that arise if one were to resample using a kernel that depends directly on distance (e.g., proportional to  $1/V(r_{i,j(i)})$ ), are avoided. These factors translate into considerable savings in computational time that can be important for large data sets and high dimensional settings, and into improved stability of the resampling algorithm.

#### Choosing the Number of Neighbors $k$ and Model Order $d$

The order of ARMA models is often picked [Loucks *et al.*, 1981] using the Akaike information criteria (AIC). Such criteria estimate the variance of the residual to time series forecasts from a model, appropriately penalized for the effective degrees of freedom in the model. A similar perspective based on cross validation is advocated here. Cross validation involves “fitting” the model by leaving out one value at a time from the data and forecasting it using the remainder. The model that yields the least-predictive sum of squares of errors across all such forecasts is picked. One can approximate the average effect of such an exercise on the sum of squares of errors without going through the process.

Here, the forecast is formed (equation (5)) as a weighted average of the successors. When using the full sample, define the weight used with the successor to the current point as  $w_{jj}$ . This weight recognizes the influence of that point on the estimate at the same location. Hence the influence of the rest of the points on the fit at that point is  $(1 - w_{jj})$ . This suggests that if estimated full sample forecast error  $e_j$  is divided by  $(1 -$

**Table 1.** Statistical Comparison of  $k$ -nn and AR1 Model Simulations Applied to an AR1 Sample

	AR1 Sample	Simulations					
		5% Quantile		Median		95% Quantile	
		$k$ -nn	AR1	$k$ -nn	AR1	$k$ -nn	AR1
Mean	0.04	-0.14	-0.12	0.02	0.04	0.24	0.20
Standard deviation	1.11	1.02	1.03	1.10	1.11	1.18	1.20
Skew	-0.17	-0.32	-0.25	-0.18	0.00	-0.03	0.21
Lag 1 correlation	0.63	0.56	0.57	0.62	0.63	0.68	0.69

$w_{jj}$ ), a measure of what the error may be if the data point ( $\mathbf{D}_j, x_j$ ) was not used in developing the estimate is provided. Note that the degrees of freedom (e.g., 0 for a  $k = 1$  and  $4/3$  for a  $k = 3$  using a uniform kernel) of estimate are implicit in this idea. Craven and Wahba [1979] present a generalized cross validation (GCV) score function that considers the average influence of excluded observations for estimation at each sample point and approximates the predictive squared error of estimate. The GCV score is given as

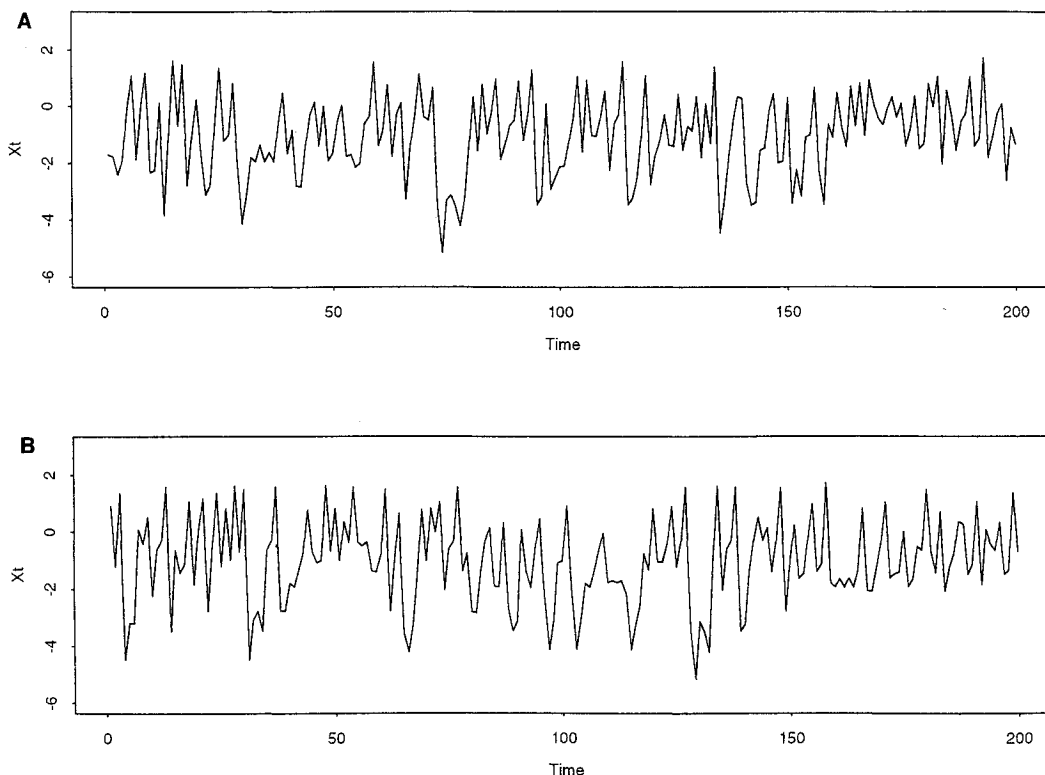
$$GCV = \frac{\sum_{i=1}^n e_i^2/n}{\left(\sum_{j=1}^n (1 - w_{jj})/n\right)^2} \tag{8}$$

The GCV score function can be used to choose both  $k$  and  $d$ . For the kernel suggested in this paper,  $w_{jj}$  is a constant for a given  $k$ , and the GCV can be written as

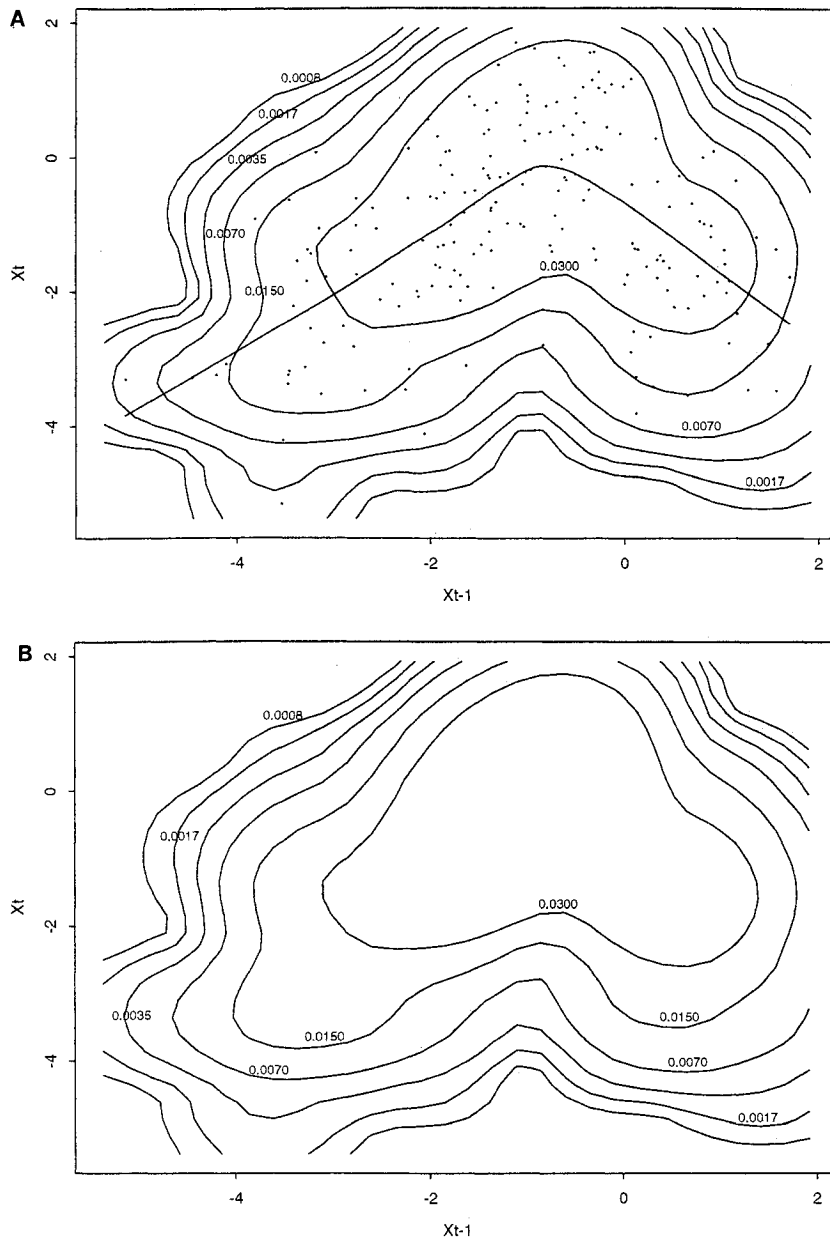
$$GCV = \frac{\sum_{i=1}^n e_i^2/n}{\left(1 - 1/\sum_{j=1}^k 1/j\right)^2} \tag{9}$$

A prescriptive choice of  $k = n^{1/2}$  from experience is also suggested. This is a good choice for  $1 \leq d \leq 6$ , and  $n \geq 100$ . Sensitivity to the choice of  $k$  in this neighborhood is small, and where computational resources are limited this choice can be recommended. Typically, with a sample size  $n$  of 50 to 200, this corresponds to a choice of  $k$  ranging from 7 to 14. When using the GCV criteria with the same sample size, it is our experience that varying  $k$  within 5 to 10 units of the optimal selected value does not appreciably change the GCV score.

Criteria such as the GCV and the AIC are known to overfit or over parameterize time series relationships. With the nearest neighbor resampler, a model with order higher than necessary will have increased variability for a given  $k$ . The extra or superfluous coordinates serve to degrade rather than enhance identification of the patterns that describe the system. Likewise, a smaller-than-optimal choice of  $d$  would lead to traces that lack the appropriate memory. Comparison of the attributes of the series generated by models with different values of  $k$  and  $d$  is consequently desirable. These comparisons can be based on how well attributes of direct interest to the investigator such as run lengths or the frequencies of threshold cross-



**Figure 5.** (a) A time series trace from the SETAR model described by (11) and (b) a time series trace from a  $k$ -nn resample of the original SETAR sample.



**Figure 6.** (a) An ASH estimate of the bivariate probability density  $f(x_t, x_{t-1})$  for the SETAR sample, the thick curve denoting a LOWESS smooth, and (b) an average of the ASH estimates of the bivariate probability density  $f(x_t, x_{t-1})$  across 100  $k$ -nn resamples from the SETAR sample.

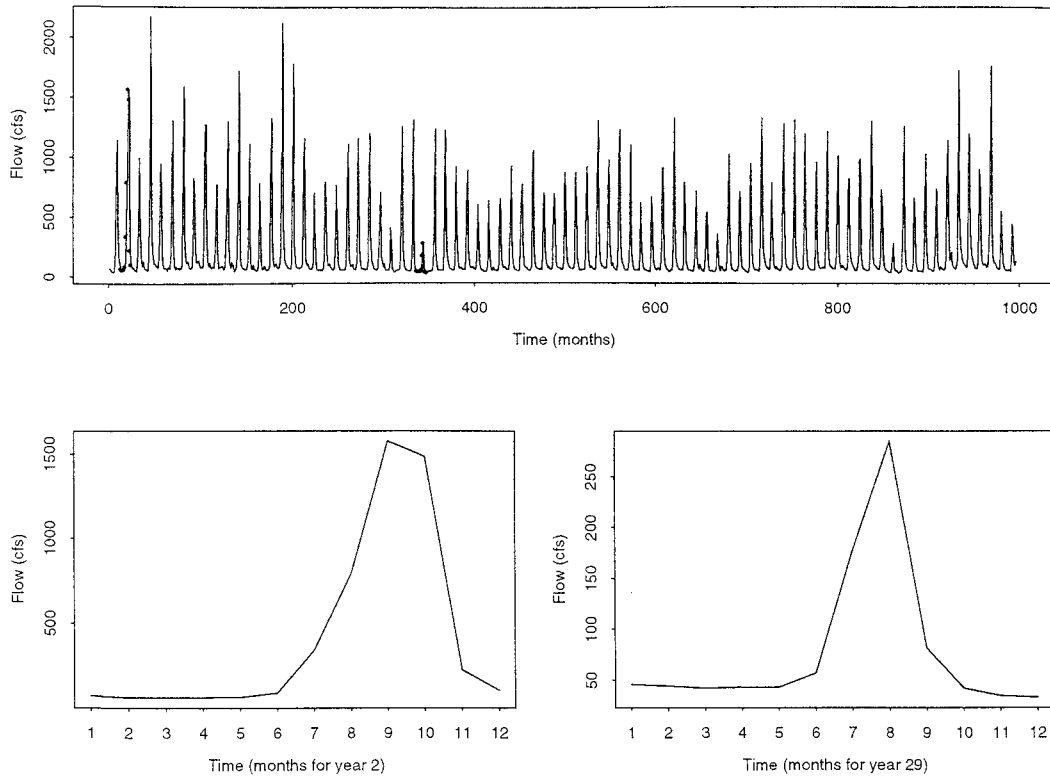
ings are reproduced. One can try various combinations of  $k$  and  $d$  and visually compare resampling attributes with historical sample attributes (sample moments and marginal or joint probability densities).

### Applications

Two synthetic examples, one from a linear and one from a nonlinear model, are presented first. These are followed by an application to monthly flows from Weber River near Oakley, Utah. In all cases a lag 1 model with  $k$  chosen as  $n^{1/2}$  was used.

Comparative performance of the simulations is judged using sample moments and sample p.d.f.'s estimated using adaptive shifted histograms (ASH; Scott [1992, chap. 5]). In all applica-

tions using the univariate ASH, a bin width of  $(x_{\max} - x_{\min})/9.1$ , where  $x_{\max}$  and  $x_{\min}$  are the respective maximum and minimum values of the data, and five shifted histograms were used. For bivariate densities a bin width of  $(x_{\max} - x_{\min})/3.6$  and five shifted histograms in each coordinate direction were used. These are the default settings for the computer code distributed by D. Scott. Conditional expectations,  $E(x_t|x_{t-1})$ , are estimated using LOWESS [Cleveland, 1979]. LOWESS is a popular robust, locally weighted linear regression technique that allows a flexible curve to be fit between two variables. We used default parameter choices (three iterations for computing the robust estimates based on two thirds of the data) with the "lowess" function available on S-Plus [Statistical Sciences, Inc., 1991].



**Figure 7.** The Weber River monthly streamflow time series. Flows for 2 years (1906 (year 2) and 1933 (year 29)) are shown in the lower panels. Note the asymmetry about the peak monthly flow in 1906 clearly shows that the time series is irreversible; that is, its properties will be quite different in reverse time.

### Example E1

The first data set considered is a sample of size 500 from a linear autoregressive model of order 1, AR1, defined as

$$x_t = 0.6x_{t-1} + 0.8e_t \quad (10)$$

where  $e_t$  is a mean zero, variance 1, Gaussian random variable.

One hundred realizations, each of length 500, were then generated from an AR1 model fitted to the sample and from the nearest neighbor ( $k$ -nn) bootstrap. Selected statistics from these simulations are compared in Table 1. The sample statistics considered are reproduced by the  $k$ -nn method, while only the sample statistics used in fitting the parametric AR1 model are reproduced. The AR1 simulations instead reproduce population or model statistics (e.g., skew = 0) for parameters that are not explicitly fitted to the sample. With repeated applications to a number of samples from the same distribution, the  $k$ -nn procedure reproduces the population statistics as well. On the other hand, a parametric model only reproduces fitted statistics. The ASH estimated median p.d.f. of  $x_t$ , from the  $k$ -nn resamples, matches the sample p.d.f., and the scatter of the estimated p.d.f.'s across resamples is comparable to the scatter from the AR1 samples. In order to save space, these results are not reproduced here.

### Example E2

A sample of size 200 was generated from a self-exciting threshold autoregressive (SETAR) model described by Tong [1990, pp. 99–101]. The general structure of such models is similar to that of a linear autoregressive model, with the dif-

ference being that the parameters of the model change upon crossing one or more thresholds. Such a model may be appropriate for daily streamflow, since crossing a flow threshold (defined on a single past flow or collectively on a set of past flows) with flow increasing may signal runoff response to rainfall or snow melt, and crossing the threshold with flow decreasing may signal return to base flow or recession behavior. Here a lag 1 SETAR model was used:

$$\begin{aligned} x_t &= 0.4 + 0.8x_{t-1} + e_t & \text{if } x_t \leq 0.0 \\ x_t &= -1.5 - 0.5x_{t-1} + e_t & \text{otherwise} \end{aligned} \quad (11)$$

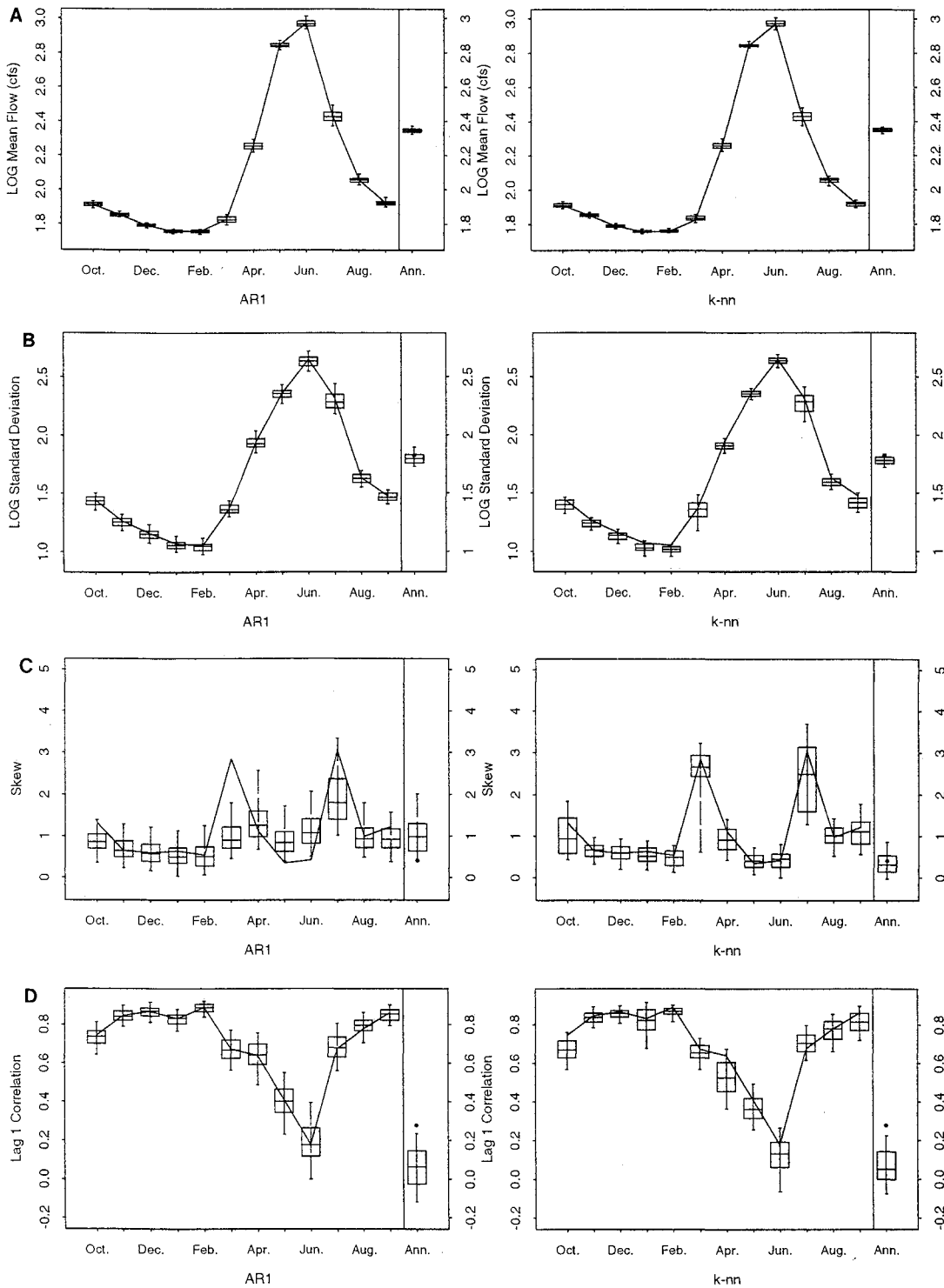
where  $e_t$  is a Gaussian random variable with mean 0 and variance 1.

The time series generated from the SETAR model and a time series simulated by the nearest neighbor method are shown in Figure 5. The bivariate probability densities  $f(x_t, x_{t-1})$  for the original SETAR sample and for 100 nearest-neighbor samples, each of length 200, were computed using ASH. The estimated  $f(x_t, x_{t-1})$  from the original sample along with a LOWESS fit of  $E(x_t|x_{t-1})$  and an average of the  $f(x_t, x_{t-1})$  estimates taken across the nearest neighbor realizations are illustrated in Figure 6. We see that the bivariate density of the data is reproduced quite well by the simulations.

### Example E3

The 1905–1988 monthly flow record from U.S. Geological Survey (USGS) station 10128500, Weber River, near Oakley, Utah, located at 40°44'10"N and 111°14'45"W, at an elevation

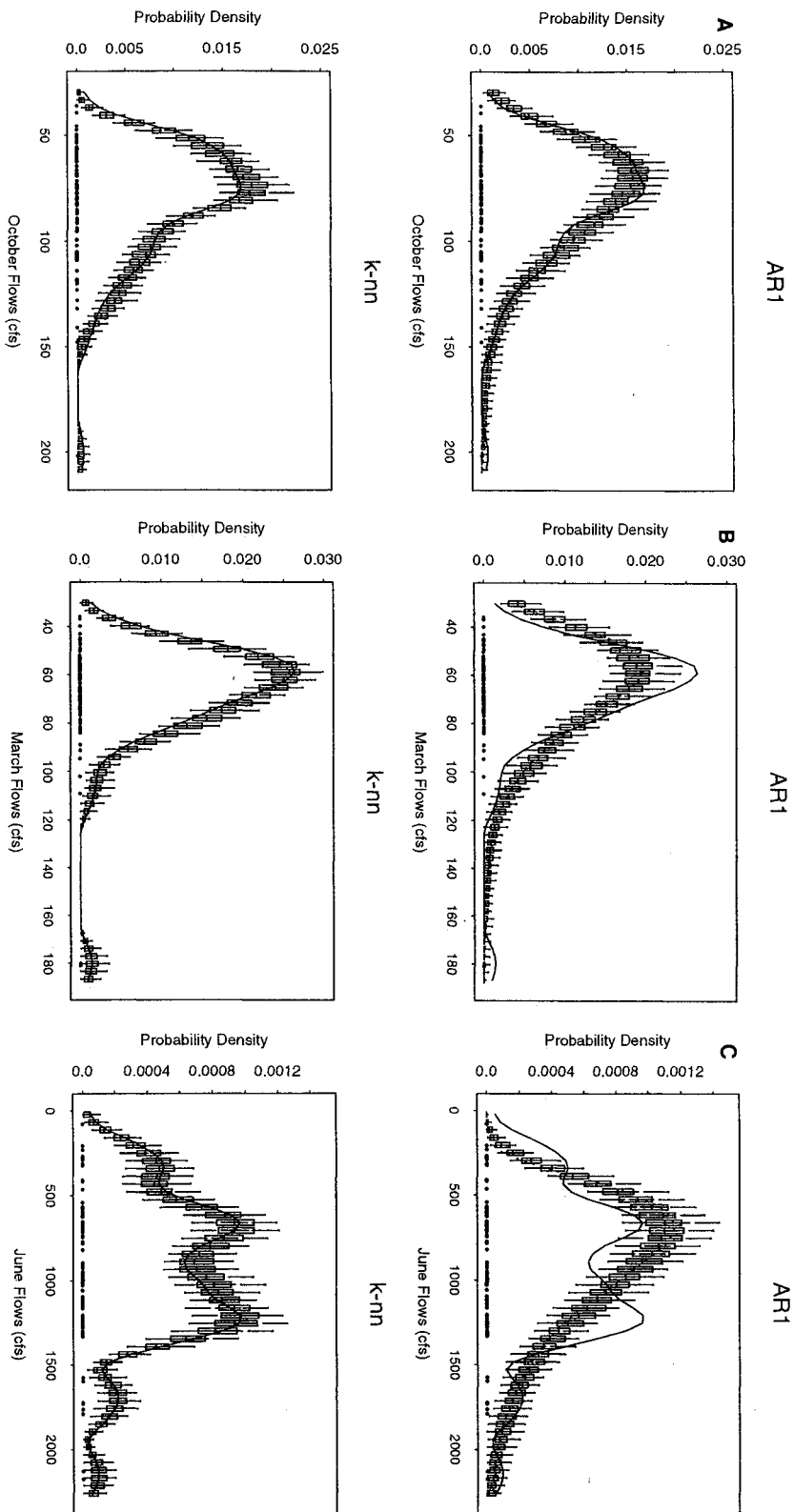




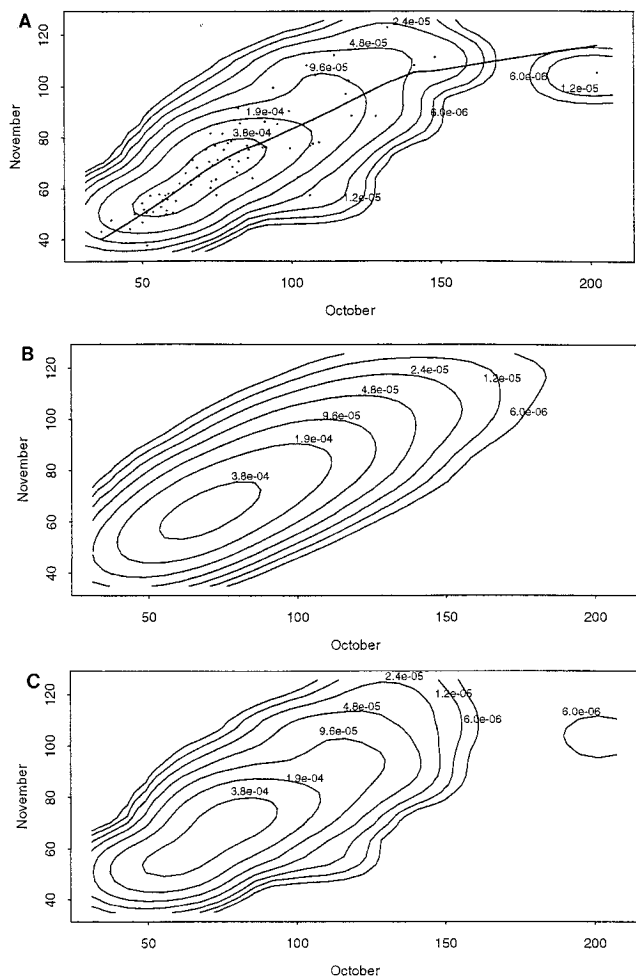
**Figure 8.** Monthly and annual statistics for simulated traces using AR1 and *k*-nn models for the Weber River data: (a) log (mean flow), (b) log (standard deviation), (c) skew, and (d) lag 1 correlation. The solid line in each figure represents the statistic for the historical sample. Box plots compare the statistic over simulations. Box plots are composed of a box being placed on the interquartile range from the multiple realizations of the statistic being compared, the line in the center of this box being the median. The “whiskers” extend to the 5% and 95% quantiles of the compared statistic. The dot above “Ann” in each figure gives the historical annual statistic. Annual flows are not modeled explicitly by either simulator used.

of 6600 feet (2012 m) above mean sea level was extracted from the USGS Hydro Climate Data Network (HCDN) CD-ROM [Slack *et al.*, 1992]. This data set is presumed to be free of the effects of regulation, diversion, and similar factors. The Weber

River at this location is a snow melt-fed, perennial, mountain stream, with a drainage area of 162 square miles (420 km<sup>2</sup>). The mean annual flow is 223 cubic feet per second (cfs) (6.24 m<sup>3</sup>/s). June is the month with the highest flow, subsequent to



**Figure 9.** Marginal probability densities estimated by ASH from AR1 and  $k$ -nn samples for the Weber River data. In each case the solid line is the ASH estimate for the historical record; the dashed line in the AR1 figure is the fitted AR1 model; the historical sample points are shown, and the box plots (see description in caption to Figure 8) depict the ASH estimates for all simulations. Results are presented for (a) October, (b) March, and (c) June.



**Figure 10.** (a) An ASH estimate of the bivariate probability density  $f(x_t, x_{t-1})$  for the October/November historical flows, the thick curve denoting a LOWESS smooth; (b) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100 simulations from an AR1 model; and (c) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100  $k$ -nn resamples. Values are given in cubic feet per second (1 cfs equals  $0.028 \text{ m}^3/\text{s}$ ).

snow melt. January is typically the month with the lowest flow. The 1905–1988 monthly time series and flows for two specific years are presented in Figure 7.

A monthly AR1 model was fitted to logarithmically transformed monthly flows, with monthly varying parameters estimated as described by Loucks *et al.* [1981, p. 285] to preserve moments in real space. This entails sequentially simulating monthly flow moving through the calendar year, using (for example) only the 83 monthly values for January and February over the 83-year record to simulate February flows given January flows. One hundred simulations of 83 years were generated in each case. The  $k$ -nn bootstrap is applied in a similar manner using (for example) January flows to find neighbors for a given January flow and the corresponding February successor. The flow data are not logarithmically transformed for the  $k$ -nn bootstrap.

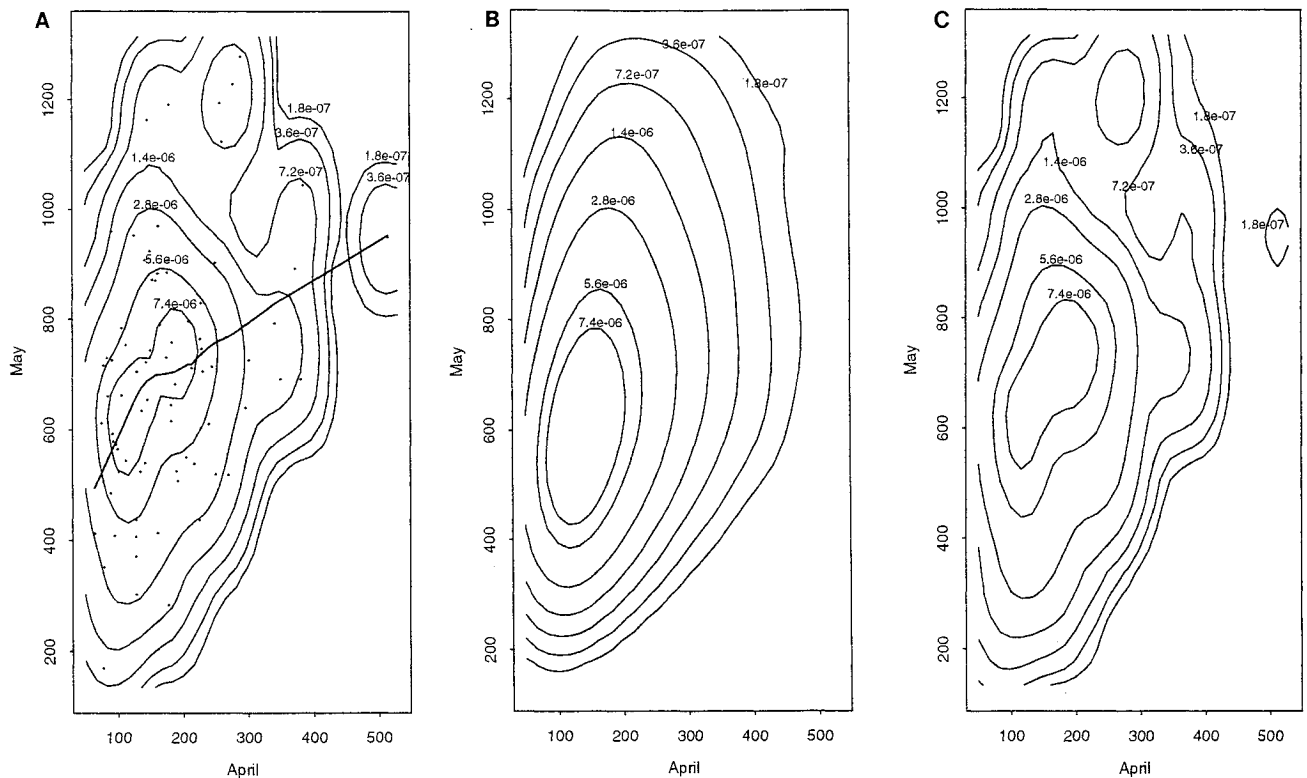
Selected results are presented. A comparison of means, standard deviations, skew, and lag 1 correlations are presented for the 12 months in Figure 8. Both the  $k$ -nn and the AR1 model seem to be comparable in reproducing the mean

monthly flow and its variation across simulations. The standard deviations of monthly flows are somewhat more variable in  $k$ -nn simulations than those from the AR1 model. Some months (low flows) have a slight downward bias in the simulated standard deviation, while others (March and July, the months following a minimum and maximum in monthly flow, respectively) show a larger spread in standard deviation than in the AR1 case. While both models seem to do well in reproducing the historical lag 1 correlation, the  $k$ -nn statistics appear to be more variable across realizations. A difference between the two simulators is apparent in Figure 8c, where the AR1 model fails to reproduce the monthly and the annual skews as well as the  $k$ -nn model does. Recall that the ad hoc prescriptive choice of  $k = n^{1/2}$  was used here, with no attempt at fine tuning the  $k$ -nn simulator.

The marginal probability density functions were estimated by ASH for flow in each month. Selected results for simulations from the  $k$ -nn and for the AR1 model applied to logarithmically transformed flows for 3 months are illustrated in Figure 9. We see from Figure 9 that the  $k$ -nn samples are indeed a bootstrap; that is, the simulated marginal probability densities behave much like the empirical sample probability density. The usual shortcoming of the bootstrap in reproducing only historical sample values is also apparent. We see that while the lognormal density used by the AR1 model is plausible in a number of months (e.g., October), the lognormal model seems to be inappropriate in other months, for example, March, where the skew is too extreme for the AR1, and June, which exhibits a distinct bimodality that may be related to the timing or amount of snow melt. The latter is interesting, since the 100 simulations from the AR1 model fail to bracket the two prominent modes of the ASH density estimate, lending support to the idea of bimodality under a pseudohypothesis test obtained from this Monte Carlo experiment.

The bivariate probability density functions for flows in each consecutive pair of months (e.g., May and June) were also computed by ASH. Results for selected months are presented in Figures 10 through 12. Other month pairs were found to exhibit features similar to those in Figures 10 through 12. In each case we present a scatterplot of the flows (in cfs) in the 2 months, with a LOWESS [Cleveland, 1979] smooth of the conditional expectation  $E(x_t|x_{t-1})$ . An examination of the October/November density in Figure 10 reveals that the AR1 model may be quite appropriate for this pair of months. The ASH estimated density from the sample and the averages of the ASH estimated densities from the AR1 and the  $k$ -nn samples are all very similar. The LOWESS estimate of the conditional expectation of the November flow, given the October flow, is very nearly a straight line.

Figures 11 and 12 refer to the months of April/May and May/June, where runoff from snow melt becomes important. The timing of the start and of the peak rate of snow melt vary over this period. Consequently, one can expect some heterogeneity in the sampling distributions of flows in these months. From Figure 11a we see that the LOWESS estimate of  $E(x_t|x_{t-1})$ , exhibits some degree of nonlinearity for April/May. The slope of  $E(x_t|x_{t-1})$  for  $x_{t-1} < 150$  cfs ( $4.2 \text{ m}^3/\text{s}$ ) is quite different from the slope for  $x_{t-1} > 150$  cfs. This is reminiscent of the SETAR model examined earlier. We could belabor this point through formal tests of significance for difference in slope. Our purpose here is to show that the  $k$ -nn approach can adapt to such sample features, while the AR1 model may not. The average bivariate densities of the simula-



**Figure 11.** (a) An ASH estimate of the bivariate probability density  $f(x_t, x_{t-1})$  for the April/May historical flows, the thick curve denoting a LOWESS smooth, (b) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100 simulations from an AR1 model, and (c) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100  $k$ -nn resamples. Values are given in cubic feet per second (1 cfs equals  $0.028 \text{ m}^3/\text{s}$ ).

tions based on ASH once again reinforce this difference between the  $k$ -nn and the AR1 models and their attributes. Note also that the AR1 simulations do not reproduce the sample skews for this period either.

The May/June analysis, shown in Figure 12, is marked by considerably increased variability in stream flow as the snow melt runoff develops. Once again, LOWESS shows some degree of nonlinearity in  $E(x_t|x_{t-1})$ , with the slope of the relationship smaller for high flows than for low flows. A comparison of the ASH bivariate density contours in Figures 12a and 12c reveals that the AR1 density is oriented quite differently from the ASH estimate for the raw sample and is unable to reproduce the degree of heterogeneity in the sample density. Recall that the marginal density of June flows was bimodal, with an antimode around 1000 cfs ( $28 \text{ m}^3/\text{s}$ ). The antimode suggests that the data is clustered into two classes of events: those with flow below 1000 cfs (mode at 700 cfs ( $19.6 \text{ m}^3/\text{s}$ )) and those with flow above 1000 cfs (mode at 1300 cfs ( $36.4 \text{ m}^3/\text{s}$ )). The LOWESS fit suggests that the June flows have an expectation close to 1000 cfs for May flows greater than about 700 cfs. It appears that the conditional expectation averages across the two modes for June flows and that the conditional density (Figure 12b) of June flows, given May flow, may be bimodal, as seen in the marginal density plot for June flows.

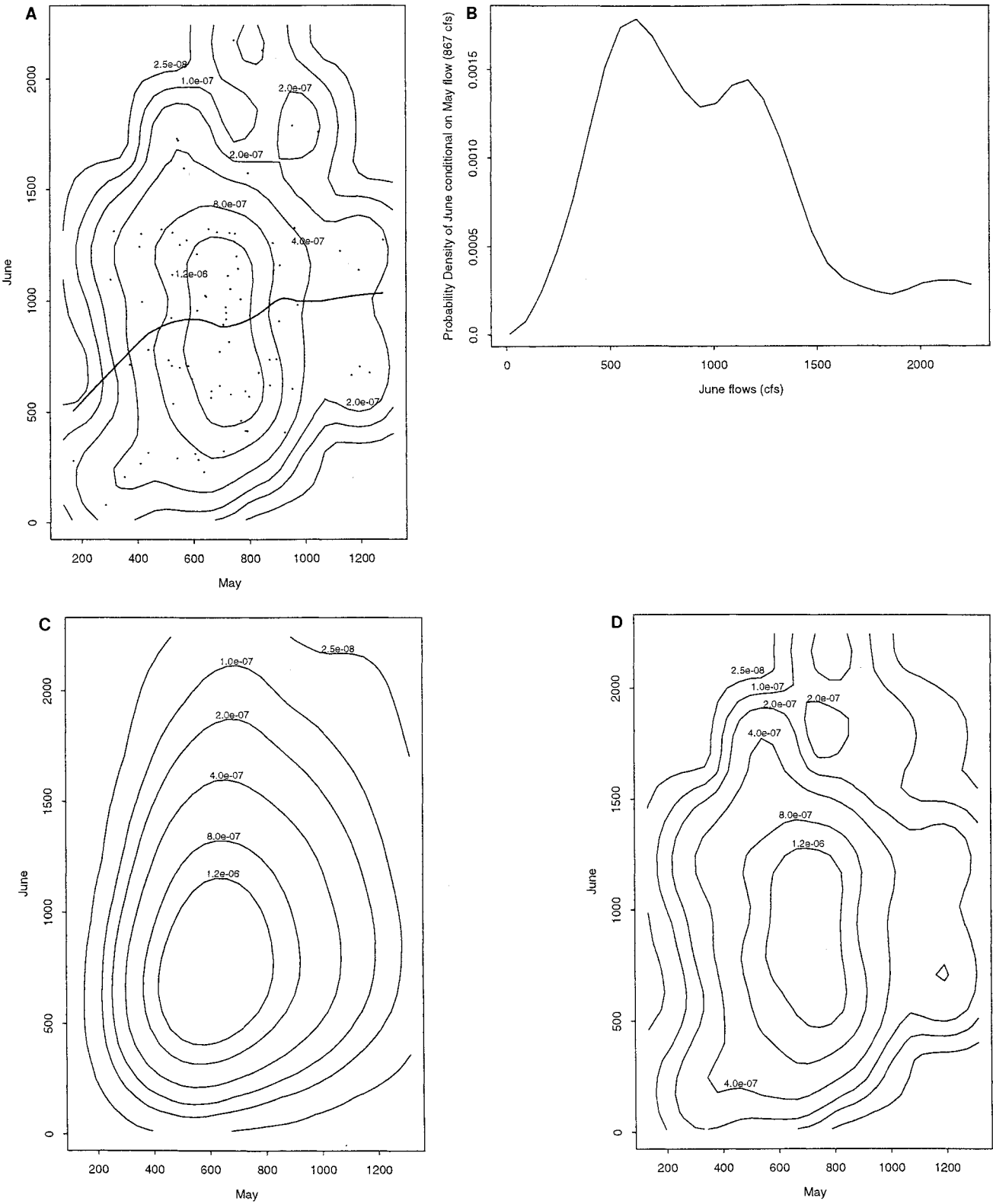
The significance of the findings reported above is that the nearest neighbor bootstrap provides a rather flexible and adaptive method for reproducing the historical frequency distribution of streamflow. The possibly tenuous issue of choosing between a variety of candidate parametric models month by month is avoided. Matching the historical frequency distribu-

tion of flows properly is important for properly estimating storage requirements for a reservoir and analyzing reservoir release options. For snowmelt-driven streams in arid regions the timing and amount of melt is important in determining reservoir operation. The bimodality in the probability density of monthly streamflow during the melt months may be connected to structured low-frequency (interannual and interdecadal) climatic fluctuations in this area [see Lall and Mann, 1995]. This would be a significant factor for reservoir operation, since the timing and amount of snowmelt may correspond to a circulation pattern that corresponds to specific flow patterns in subsequent months as well. The nearest neighbor bootstrap would be an appropriate technique for simulating sequences conditioned on such factors. Work in this direction is in progress.

## Summary and Discussion

A nearest neighbor method for a conditional bootstrap of time series was presented and exemplified. A corresponding forecasting strategy was indicated. Our contributions here lie primarily in the development of a new kernel, suggestion of a parameter selection strategy, application to a conditional bootstrap, and demonstration of the methodology. It was shown that sample attributes are reproduced quite well by this nonparametric method for both synthetic and real data sets. Given the flexibility of these techniques, we consider them to have tremendous practical potential.

The parametric versus nonparametric statistical method debate often veers toward sample size requirements and statisti-



**Figure 12.** (a) An ASH estimate of the bivariate probability density  $f(x_t, x_{t-1})$  for the May/June historical flows, the thick curve denoting a LOWESS smooth; (b) ASH estimate of the probability density of June flows conditional to a May flow of 867 cfs ( $24.3 \text{ m}^3/\text{s}$ ); (c) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100 simulations from an AR1 model; and (d) an average of the ASH estimates of  $f(x_t, x_{t-1})$  across 100  $k$ -nn resamples. Values are given in cubic feet per second (1 cfs equals  $0.028 \text{ m}^3/\text{s}$ ).

cal efficiency arguments. In the context of a resampling strategy, as espoused here, these arguments take a somewhat different complexion. For some processes, such as daily streamflow, identification or even definition of an appropriate parametric model is problematic. In these cases, data are relatively plentiful. For such cases, methods such as those presented here are enticing. For monthly and annual flows, there is progressively less structure, and sample sizes are smaller. In these situations, parametric methods may indeed be statistically more efficient provided the correct model is identifiable and parsimonious. In our view, particularly for the snow-fed rivers of the western United States, this may not always be the case. Indeed, for the application presented here at the monthly timescale it is hard to justify choosing the parametric approach over the nearest neighbor method. The consideration of parameter uncertainty is justifiably considered a good idea in parametric time series resampling of streamflow [Grygier and Stedinger, 1990]. Likewise it may be useful to think about model uncertainty when developing parametric models. The latter consideration is implicit in the nonparametric approach, since a rather broad class of models is approximated. The impact of varying the "parameters"  $k$  and the model order on specific attributes of the resamples bears further investigation. Our preliminary analyses suggest that the sensitivity of the scheme is limited over a range of  $k$  values near the "optimal" with the kernel used here. Formal investigations of this issue are being pursued.

One can devise a strategy that allows nearest neighbor resampling with perturbation of the historical data in the spirit of traditional autoregressive models, that is, conditional expectation with an added random innovation. First, one evaluates the conditional expectation using the generalized nearest neighbor regression estimator for each vector  $\mathbf{D}_i$  in the historical record. A residual  $e_i$  can be computed as the difference between the successor  $x_j$  of  $\mathbf{D}_i$  and the nearest neighbor regression forecast. The simulation proceeds by estimating the nearest neighbor regression forecast relative to a conditioning vector  $\mathbf{D}_i$  and then adding to this one of the  $e_j$  corresponding to a data point  $j$  that lie in the  $k$  nearest neighborhood  $J_{i,k}$ . The innovation  $e_j$  is chosen using the resampling kernel  $K(j(i))$ . This scheme will perturb the historical data points in the series, with innovations that are representative of the neighborhood, and will thus "fill in" between the historical data values as well as extrapolating beyond the sample. The computational burden is increased and there is a possibility that the bounds on the variables will be violated during simulation. However, there may be situations where the investigator may wish to adopt this strategy. Further exploration of this strategy is planned.

Issues such as disaggregation of streamflows bear further investigation. One strategy is trivial: resample the flow vector that aggregates to the aggregate flow simulated. A question that arises is whether there is even any need to work with models that disaggregate (especially in time) using these methods. One may wish to work directly with, say, the daily flows, conditioned on a sequence of past daily flows and weekly or monthly flows.

The real utility of the method presented here may lie in exploiting a dependence structure (e.g., in daily flows) that is difficult to treat by traditional methods, as well as complex relationships between variables, and in estimating confidence limits or risk in problems that have a time series structure. The traditional time series analysis framework directs the researcher's attention toward an efficient estimation of model param-

eters under some metric (e.g., least squares or maximum likelihood). The performance metric of interest to the hydrologist may not be the one optimal for the estimation of a certain set of parameters and selected model form. There is reason to directly explore other aspects of the problem that may be of direct interest for reservoir operation and flood control, using flexible, adaptive, data exploratory methods. Such investigations using the  $k$ -nn bootstrap are in progress.

**Acknowledgments.** The work reported here was supported in part by the USGS through grant number 1434-92-G-2265. Discussions with and review of this work by David Tarboton are acknowledged. The comments of the anonymous reviewers resulted in a significant improvement of the manuscript.

## References

- Cleveland, W. S., Robust locally weighted regression and smoothing scatterplots, *J. Am. Stat. Assoc.*, 74, 829–836, 1979.
- Craven, P., and G. Wahba, Smoothing noisy data with spline functions, *Numer. Math.*, 31, 377–403, 1979.
- Efron, B., Bootstrap methods: Another look at the Jackknife, *Ann. Stat.*, 7, 1–26, 1979.
- Efron, B., and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, New York, 1993.
- Eubank, R. L., *Spline Smoothing and Nonparametric Regression, Statistics: Textbooks and Monographs*, Marcel Dekker, New York, 1988.
- Grygier, J. C., and J. R. Stedinger, Spigot, a synthetic streamflow generation package, technical description, version 2.5, School of Civ. and Environ. Eng., Cornell Univ., Ithaca, N. Y., 1990.
- Györfi, L., W. Härdle, P. Sarda, and P. Vieu, *Nonparametric Curve Estimation from Time Series*, vol. 60, *Lecture Notes in Statistics*, Springer-Verlag, New York, 1989.
- Härdle, W., *Applied Nonparametric Regression, Econometric Soc. Monogr.*, Cambridge Univ. Press, New York, 1989.
- Härdle, W., *Smoothing Techniques With Implementation in S*, Springer-Verlag, New York, 1990.
- Härdle, W., and A. W. Bowman, Bootstrapping in nonparametric regression: Local adaptive smoothing and confidence bands, *J. Am. Stat. Assoc.*, 83, 102–110, 1988.
- Karlsson, M., and S. Yakowitz, Nearest-Neighbor methods for nonparametric rainfall-runoff forecasting, *Water Resour. Res.*, 23, 1300–1308, 1987a.
- Karlsson, M., and S. Yakowitz, Rainfall-runoff forecasting methods, old and new, *Stochastic Hydrol. Hydraul.*, 1, 303–318, 1987b.
- Kendall, D. R., and J. A. Dracup, A comparison of index-sequential and AR(1) generated hydrologic sequences, *J. Hydrol.*, 122, 335–352, 1991.
- Kunsch, H. R., The Jackknife and the Bootstrap for General Stationary Observations, *Ann. Stat.*, 17, 1217–1241, 1989.
- Lall, U., Nonparametric function estimation: Recent hydrologic applications, *U.S. Natl. Rep. Int. Union Geod. Geophys. 1991–1994, Rev. Geophys.*, 33, 1093, 1995.
- Lall, U., and M. Mann, The Great Salt Lake: A barometer of low-frequency climate variability, *Water Resour. Res.*, 31(10), 2503–2515, 1995.
- Loucks, D. P., J. R. Stedinger, and D. A. Haith, *Water Resource Systems Planning and Analysis*, 559 pp., Prentice-Hall, Englewood Cliffs, N. J., 1981.
- Schuster, E., and S. Yakowitz, Contributions to the theory of nonparametric regression, with application to system identification, *Ann. Stat.*, 7, 139–149, 1979.
- Scott, D. W., *Multivariate Density Estimation: Theory, Practice and Visualization*, John Wiley, New York, 1992.
- Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, New York, 1986.
- Slack, J. R., J. M. Landwehr, and A. Lumb, A U.S. Geological Survey streamflow data set for the United States for the study of climate variations, 1874–1988, *Rep. 92-129*, U.S. Geol. Surv., Oakley, Utah, 1992.
- Smith, J. A., Long-range streamflow forecasting using nonparametric regression, *Water Resour. Bull.*, 27, 39–46, 1991.
- Smith, J., G. N. Day, and M. D. Kane, Nonparametric framework for

- long range streamflow forecasting, *J. Water Resour. Plann. Manage.*, *118*, 82-92, 1992.
- Statistical Sciences, Inc., *S-Plus Reference Manual, Version 3.0*, Seattle, Wash., 1991.
- Tarboton, D. G., A. Sharma, and U. Lall, The use of non-parametric probability distributions in streamflow modeling, in *Proceedings of the Sixth South African National Hydrological Symposium*, ed. S. A. Lorentz et al., University of Natal, Pietermaritzburg, South Africa, September 8-10, 315-327, 1993.
- Tasker, G. D., Comparison of methods for estimating low flow characteristics of streams, *Water Resour. Bull.*, *23*, 1077-1083, 1987.
- Tong, H., *Nonlinear Time Series Analysis: A Dynamical Systems Perspective*, Academic, San Diego, Calif., 1990.
- Woo, M. K., Confidence intervals of optimal risk-based hydraulic design parameters, *Can. Water Resour. J.*, *14*, 10-16, 1989.
- Yakowitz, S., A stochastic model for daily river flows in an arid region, *Water Resour. Res.*, *9*, 1271-1285, 1973.
- Yakowitz, S., Nonparametric estimation of markov transition functions, *Ann. Stat.*, *7*, 671-679, 1979.
- Yakowitz, S. J., Nonparametric density estimation, prediction, and regression for markov sequences, *J. Am. Stat. Assoc.*, *80*, 215-221, 1985.
- Yakowitz, S., Nearest-neighbor regression estimation for null-recurrent Markov time series, *Stochastic Processes Their Appl.*, *48*, 311-318, 1993.
- Yakowitz, S., and M. Karlsson, Nearest-neighbor methods with application to rainfall/runoff prediction, in *Stochastic Hydrology*, edited by J. B. Macneil and G. J. Humphries, pp. 149-160, D. Reidel, Norwell, Mass., 1987.
- Zucchini, W., and P. T. Adamson, Bootstrap confidence intervals for design storms from exceedance series, *Hydrol. Sci. J.*, *34*, 41-48, 1989.

---

U. Lall and A. Sharma, Utah Water Research Laboratory, Utah State University, UMC82, Logan, UT 84322-8200. (e-mail: ulall@kernel.uwrl.usu.edu; ashar@kernel.uwrl.usu.edu)

(Received February 23, 1995; revised September 20, 1995; accepted September 22, 1995.)