Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

Locally weighted polynomial regression: Parameter choice and application to forecasts of the Great Salt Lake

Upmanu Lall,[1] Young-Il Moon,[2] Hyun-Han Kwon,[1] and Ken Bosworth[3]

Received 17 December 2004; revised 8 February 2006; accepted 27 February 2006; published 25 May 2006.

[1] Relationships between hydrologic variables are often nonlinear. Usually, the functional form of such a relationship is not known a priori. A multivariate, nonparametric regression methodology is provided here for approximating the underlying regression function using locally weighted polynomials. Locally weighted polynomials consider the approximation of the target function through a Taylor series expansion of the function in the neighborhood of the point of estimate. Cross-validatory procedures for the selection of the size of the neighborhood over which this approximation should take place and for the order of the local polynomial to use are provided and shown for some simple situations. The utility of this nonparametric regression approach is demonstrated through an application to nonparametric short-term forecasts of the biweekly Great Salt Lake volume. Blind forecasts up to 1 year in the future using the 1847–2004 time series of the Great Salt Lake are presented.

Citation: 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.

1. Introduction

[2] Most hydrologists are familiar with linear regression and its use for developing a relationship between two or more variables. The general linear model (where the variables are presumed to be linearly related after applying some predetermined transform) is used as a building block in many activities ranging from spatial surface estimation, missing value imputation, sediment load estimation to autoregressive time series modeling. Often, such a procedure is not very satisfying. The choice of an appropriate transform to use may not be obvious, and scatterplots of the data may not visually support the assumed model. An alternative to such regression approaches that is capable of representing relatively complex relationships between variables, through ‘‘local’’ or pointwise approximation of the underlying function, is presented here. Procedures that determine model parameters automatically from the data, and also allow the user to choose between a parametric (i.e., linear model) and a nonparametric model (i.e., a local linear model) are provided, for a special case.

[3] There has been a surge in the development and application of nonparametric regression and density estimation methods in the last decade as computational resources have become more accessible. Some monographs that make this literature accessible are by *Silverman* [1986], *Eubank* [1988], and *Härdle* [1989]. Examples of nonparametric estimators include orthogonal series expansions, moving averages, splines, kernel, nearest neighbor, and neural network estimators. The reader is referred to *Lall* [1995] for a review of the application of nonparametric function estimation in hydrology, to *Owosina* [1992] for a comparative performance evaluation of selected methods, and to *Hastie et al.* [2001] for an expository text that compares all these methods and their parameter estimation. Some attributes of these methods are as follows: (1) The estimator of the target function can often be expressed as a weighted moving average of the observations. (2) The estimates are defined locally or using data from a small neighborhood of each point of estimate, i.e., the weights of the moving average are zero or very small beyond some distance from the point of estimate. (3) As a result, a wide class of target, underlying functions can be approximated without knowing their actual functional form. (4) The ‘‘nonparametric’’ estimator has parameters that control the local weights and the size of the neighborhood used for estimation.

[4] The last point deserves further explanation since it often creates confusion. The role played by the parameters in these methods is quite different. The parameters of a nonparametric model have a different role than those of a parametric model. No global functional form is assumed, and the parameters merely control how local averages are to be formed. Unlike linear or parametric regression, where the parameters (e.g., intercept and slope) are sufficient to provide an estimate at any point, the nonparametric estimator needs the observations in the neighborhood to provide an estimate at any point. For example the parameter of a moving average is the number of points (e.g., 3) to average over. One still needs the three points surrounding the point of estimate to report the answer. By contrast, once a linear

[1] Department of Earth and Environmental Engineering, Columbia University, New York, New York, USA.
[2] Department of Civil Engineering, University of Seoul, Seoul, South Korea.
[3] Department of Mathematics, Idaho State University, Pocatello, Idaho, USA.

Copyright 2006 by the American Geophysical Union.
0043-1397/06/2004WR003782$09.00

regression has been evaluated, the parameters are all one needs to provide an estimate at any point since the overall behavior is of an assumed form (e.g., linear or log linear).

[5] Background information on multivariate, locally weighted polynomial regression is provided in the next section. Methods for parameter selection and the estimation of error variances and confidence intervals are presented next. An algorithm for local regression is then summarized. An application of local regression to time series forecasting is presented. A biweekly record of the Great Salt Lake volume from 1847 to 2004 was used for the forecasts. The dynamics of the time series of the Great Salt Lake has been studied by *Sangoyomi* [1993], *Lall et al.* [1996] and *Abarbanel et al.* [1996]. *Sangoyomi* [1993] investigated a spectral analysis that showed 14 peaks that pass the 90% significance test.

[6] This prior work suggests that the dynamics of the Great Salt Lake (GSL) volume is nonlinear, and particularly demonstrated the success of a nonparametric regression scheme (multivariate adaptive regression splines, or MARS [*Friedman*, 1991]) for forecasting the GSL for up to 4 years ahead during extreme hydrologic periods. The local polynomial regression scheme presented here is a computationally faster alternative to MARS, and is easier to explain and analyze. Results from the two methods are comparable. We do not directly and quantitatively compare MARS and the local regression approach. Our intention here is to focus on certain aspects of choosing the parameters of the local regression model and in demonstrating how these work to permit a flexible choice between a full sample parametric estimator and an estimator that uses a fraction of the full sample. The improvement and the conditions under which one can expect improvement using the proposed estimator relative to the usual cross-validation-based criteria for model selection are demonstrated.

2. Background

[7] Consider a general regression model given as

$$y_i = f(\mathbf{x}_i) + e_i \qquad i = 1, \ldots, n \tag{1}$$

where, **x** is a M vector of M explanatory variables, y is the "response" variable, $f(.)$ represents the underlying functional relationship between y and **x**, e_i are noise or measurement errors, that may or may not depend on $\mathbf{x}_i$, and n is the number of observations.

[8] Linear regression considers the case where the function $f(.)$ is linear in **x**, i.e., $f(\mathbf{x}) = \mathbf{x}\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a M vector of coefficients. Theoretical methods for parameter selection, analysis of variance, hypothesis testing, estimating confidence and prediction intervals, outlier identification and other related statistical activities are well developed and understood for linear regression [*Stuart and Ord*, 1991]. Statistical software is readily available to perform these computations. Consequently, these methods see widespread use. Nonlinear relationships between **x** and y are admitted in this framework through prior transforms applied to **x**, and/or y. For instance, the components of **x** may include polynomial terms ($1, v, v^2, \ldots$) in an original variable v. The Box-Cox family of power transformations is often used in hydrology [see *Helsel and Hirsch*, 1992] to select an appropriate transform. There are a number of difficulties with such an approach. These include (1) the inability to find an appropriate transform for a given data set, (2) bias in the estimates upon back transforming, (3) nonunique selection of applicable transforms for a given data set, which can lead to an unquantifiable uncertainty of estimate (particularly while extrapolating the data, which is the usual application!), and (4) reconciliation of the distribution of errors in the transformed and original coordinates relative to the underlying error distribution. An approach for the pointwise estimation of the unknown function $f(.)$ from data, based on a local Taylor series expansion of $f(\mathbf{x})$ at the point of estimate **x**, was proposed by *Macauley* [1931]. *Cleveland* [1979], *Cleveland and Devlin* [1988], and *Cleveland et al.* [1988] pioneered this idea into a statistical methodology for local approximation of functions from data. Recently [e.g., *Loader*, 1999; *Hastie and Loader*, 1993; *Fan and Gijbels*, 1992; *Müller*, 1987; *Lejeune*, 1985], these methods have been recognized as very useful generalizations of kernel regression (weighted moving averages). Applications to a suite of statistical estimation problems are emerging. Our presentation here is specific to the estimator we describe. The material presented here builds directly on the method of *Cleveland and Devlin* [1988]. The reader is referred to a monograph by *Wand and Jones* [1995] for general background on the methods.

[9] Generally, the strategy for local polynomial regression is to choose a certain number, k, of nearest neighbors (in terms of Euclidean distance) of the estimation point **x**, and to form the estimate $f(\mathbf{x})$ through a locally weighted, polynomial regression over the $(\mathbf{x}, y)$ data that lie in the neighborhood. Consider the general regression model described in (1). The sampling locations $\mathbf{x}_i$ are usually not regularly spaced. We assume the e_i are uncorrelated, mean zero, random variables, assumed to be approximately identically distributed in the k nearest neighborhood of the point of estimate.

[10] To motivate the local polynomial regression technique for approximating a wide class of functions $f(.)$, first suppose that the errors e_i are identically zero. Then locally, about some $\mathbf{x}^*$, we can estimate $f(.)$ using a (multivariate) polynomial in **x** which is chosen to interpolate $y_i = f(\mathbf{x}_i{}^*)$ at $\mathbf{x}_i{}^*$, $i = 1, \ldots k$, in the k nearest neighborhood of $\mathbf{x}^*$. The data indices $i = 1, \ldots k$ are arranged so that the data $\mathbf{x}_i{}^*$ are arranged in order of increasing distance from $\mathbf{x}^*$. Here, we assume the underlying f is locally a smooth (i.e., continuous and differentiable to some order) function. In the univariate setting we can write the following Newton-Taylor expansion

$$\begin{aligned} f(x) &= f[x_i^*] + f[x_1^*, x_2^*](x - x_1^*) + \ldots + f[x_1^*, x_2^*, \ldots x_k^*](x - x_1^*) \\ &\quad \cdot (x - x_2^*) \ldots (x - x_{k-1}^*) + f^{(k)}(\zeta)(x - x_1^*) \ldots (x - x_{k-1}^*) \\ &\quad \cdot (x - x_k^*)/k! \\ &= p_k(x) + f^{(k)}(\zeta)(x - x_1^*) \ldots (x - x_k^*)/k! \end{aligned} \tag{2}$$

where the points x and ζ are in the interior (convex hull) of the set of points $\{x_1^*, \ldots, x_k^*\}$, and $f[x_1^*, x_2^*, \ldots x_j^*]$ is the $(\mathrm{j} - 1)^{th}$ divided difference (defined as the coefficient of the term $x^{j\text{-}1}$ of the polynomial of degree j - 1 that agrees with $f(x_1), f(x_2), \ldots f(x_j)$ of $f(.)$ at the sample points $\{x_1^*, x_2^*, \ldots x_{\mathrm{j}}^*\}$). The point ζ is generally unknown, but is guaranteed to exist by the mean value theorem provided $f(.)$ is C^k

smooth (i.e., continuous and differentiable to order k). We notice that if the sample points are closely grouped (i.e., $|x - x_k^*|$ is small) and $f(.)$ is locally C^k smooth (i.e., the kth derivative is finite), then the last term in the above equality can be neglected, and one obtains the local kth-order polynomial approximation $p_k(x)$ to $f(x)$.

[11] The choice of the x_i^* values for any given x can be problematic in several ways. First, x^* should be "centered" within the set $\{x_1^*,\ldots x_k^*\}$. This is not a problem if the sample points x_i^* are uniformly distributed, and x is not near the boundary of the region. However, in situations where one has nonuniform sampling, and when one estimates $f(.)$ near the boundary, we expect a deterioration in the estimate of $f(.)$ provided by $p_k(.)$. Secondly, there is a choice of how many points, k, to use in building the estimator $p_k(.)$. Choosing too few neighbors about x results in loss of information contained in higher derivatives; choosing too many neighbors (resulting in a higher-degree polynomial) yields an estimator with too many degrees of freedom, allowing too much variation between the interpolation points. That is, the true, but unknown local behavior of f near x^* may not be well represented by bringing in interpolation points "far away" from x^*. Thus, even working with error free, perfect data, the idea of representing f by a local polynomial requires a means of balancing the trade-off occurring between choosing small local neighborhoods (i.e., small k), resulting in a possibly oversmoothed or biased estimate, and choosing large neighborhoods, resulting in superfluous degrees of freedom. Asymptotically, (i.e., as the sample size $n \rightarrow \infty$, one would decrease k, since the distance $|\mathrm{x} - x_k^*|$ will approach zero.

[12] With the introduction of noise, the above procedure can be readily adapted by requiring that an estimate of $f(.)$ near a given $\mathbf{x}^*$ be obtained by performing a local least squares polynomial regression on the k nearest neighborhood, and using this regression polynomial's value at $\mathbf{x}^*$ to estimate $f(\mathbf{x}^*)$. It is desirable to prescale each data vector or coordinate in $\mathbf{x}$ to have similar scale, prior to computing nearest neighbor distances. In our implementation, we consider scaling each component to have a mean 0, and variance 1, or to lie between 0 and 1. Practically, linear, quadratic and cubic polynomials are useful. Using a polynomial of degree p, we may have n_p parameters that need to be estimated locally. Note that $n_p < k$, else we can interpolate, with potentially disastrous results. In practice, we enforce $k > 2n_p$. It is desirable to form estimates that are "local", since the "end points" in a linear regression can have a large influence on the resulting estimate. Given these observations, and the observation that the point of estimate should be approximately centered in the locale of estimation, it is desirable to weight the local polynomial fit, such that observations close to the point of estimate are accorded a higher importance in the least squares fit, than those that lie further away in the neighborhood.

[13] Then, the locally weighted polynomial regression at each point of estimate $\mathbf{x}_l^*$, $\boldsymbol{l} = 1\ldots m$, given a $(n \times M)$ data matrix $\mathbf{X}$ and a $(n \times 1)$ response vector $\mathbf{y}$, is obtained through the solution of the weighted least squares problem:

$$\min_{\beta_l} (\mathbf{y}_l - \mathbf{Z}_l\beta_l)^T \mathbf{W}_l (\mathbf{y}_l - \mathbf{Z}_l\beta_l) \qquad (3)$$

where the subscript $\boldsymbol{l}$ recognizes that the associated element is connected with the point of estimate $\boldsymbol{x}_l^*$; β_l are estimates of the coefficients of the terms in the basis defined by $\mathbf{Z}_l$; $\mathbf{Z}_l$ is a matrix formed by augmenting $\mathbf{X}$, with columns that represent the polynomial expansion of $\mathbf{X}$ to degree p (including cross-product terms if desired); $\mathbf{W}_l$ is a $k \times k$ diagonal weight matrix with elements $w_{ii,l} = K(u_{i,l})/\sum_{j=1}^{k} K(u_{j,l})$, where $u_{i,l} = d_{i,l}/d_{k,l}$; $d_{i,l}$ is the distance from $\mathbf{x}_i^*$ to $\mathbf{x}_j$ using an appropriate metric, and $K(.)$ is a weight function.

[14] We have implemented a bisquare kernel ($K(u) = 15/16(1 - \mathrm{u}^2)^2$). The latter is recommended by *Scott* [1992] because of its smoothness properties. The matrices $\mathbf{y}_l$ and $\mathbf{Z}_l$ are defined over the k nearest neighborhood of $\mathbf{x}_l^*$. Singular value decomposition (SVD) using algorithms from *Press et al.* [1989] is used to solve the linear estimation problem resulting from (3).

[15] The reader may note that we are in the familiar territory of linear regression, and will hence have the usual statistical tools available to us. The coefficients β_l are obtained as:

$$\beta_l = \left(\mathbf{Z}_l^T \mathbf{W}_l \mathbf{Z}_l\right)^{-1} \mathbf{Z}_l^T \mathbf{W}_l \mathbf{y}_l \qquad (4)$$

The resulting estimate of $\hat{f}(\mathbf{x}_l)$ is then

$$\hat{f}(\mathbf{x}_l) = \mathbf{z}_l \beta_l \qquad (5)$$

where $\mathbf{z}_l$ is the $n_p \times 1$ vector formed by augmenting $\mathbf{x}_l$ with polynomial terms to order p.

[16] Expressions for the asymptotic bias and variance of an estimate equivalent to (5) are presented by *Wand and Jones* [1995, p. 140], and are not reproduced here. Instead, we shall develop some data driven estimates of variance and mean square error of estimate, under the assumption that the number of nearest neighbors, k, and the order of the local polynomial, p has been chosen appropriately. We shall consider k and p the "smoothing parameters" of the estimation scheme and consider the slopes β_l to be "structural parameters".

[17] The mean square error of estimate at $\mathbf{x}_l$ is estimated from the local regression as

$$MSE\left(\hat{f}(\mathbf{x}_l)\right) = \mathbf{e}_l^T \mathbf{W}_l \mathbf{e}_l \qquad (6)$$

[18] Note that this is a weighted average of the squared errors from the local regression, with the same weights as those used for estimating the regression. Essentially, the mean square error of fit is localized to the point of estimate.

[19] An estimate of the variance of the local errors $e_{i,l}$, $i = 1\ldots k$, is available as

$$s_{e_l}^2 = \mathbf{e}_l^T \mathbf{W}_l \mathbf{e}_l / \left((k - n_p)/k\right) \qquad (7)$$

where n_p is the number of parameters fit.

[20] One can thus allow an error structure that is localized to the point of estimate as well. Hypothesis tests can then be designed to see if the error variance at two points is comparable or not. Such tests can either be implemented nonparametrically (e.g., through a bootstrap [*Efron and*

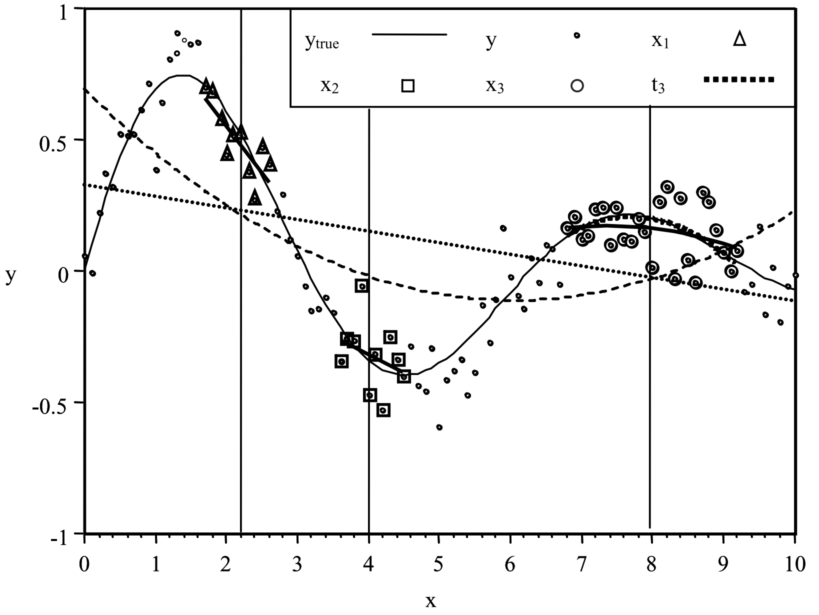

Figure 1. Illustration of local linear and local quadratic regression.

Tibishirani, 1993] of the data), or parametrically with assumptions as to the probability distribution of errors.

[21] Under the assumption that the local errors $e_{i,l}$ are approximately normally distributed, one can construct a t test for a hypothesis that individual coefficients β_j are significantly different from zero, at a desired significance level α. A judicious application of this t test with a significance level α, allows one to screen spurious coefficients and improve the degrees of freedom of the local fit, retaining n'_p of the n_p original terms in the regression. The t statistic of interest [*Stuart and Ord*, 1991, p. 1042], with $(k - n_p)$ degrees of freedom is

$$t_j = \beta_j / \left(s_{e_l}^2 a_{jj}\right)^{0.5} \tag{8}$$

where a_{jj} is the (j,j) element of the matrix $(\mathbf{Z}_l^T \mathbf{W}_l \mathbf{Z}_l)^{-1}$.

[22] Approximate confidence and prediction intervals for the estimate in (5) can also be obtained as for the regular linear regression model. We shall defer the presentation of these estimates until the end of the next section which focuses on the selection of the number of nearest neighbors and the order of the polynomial.

3. Smoothing Parameter Selection and Confidence Limits

[23] Some of the issues that are important for picking the order of the local polynomial and for selecting the number of nearest neighbors to use were touched on in the last section. Here, we shall present the selection process we have adopted for smoothing parameter selection. It is helpful to begin with a simple univariate example. Consider the estimation of the function $f(x) = \sin(x) \exp(-0.2x)$, from the data (x_i, y_i), generated such that the x_i are equally spaced values from 0 to 10, and the y_i are then generated as $f(x_i) + e_i$, $i = 1 \ldots 100$, and $e_i \sim N(0, 0.1)$. This data set, the true, underlying function, and 3 local regressions are shown in Figure 1.

[24] Figure 1 provides an illustration of local linear and local quadratic regression, with uniform weights $w_{ii} = 1/k$. The data (small circles) are 100 points generated from $y = \sin(x)e^{-0.2x} + N(0, 0.1)$. The true function is shown as the solid line. The thin dotted line is a linear regression through the full data. It shows the bias incurred if a neighborhood of size 100 is used at any point. The thin dashed line is a quadratic fit through the data. It shows that the bias may be reduced by going to a higher-order polynomial. Estimates are considered at 3 points, $x = 2.2$, 4 and 8. Local linear fits with 10 neighbors are used at the first two points, and a locally quadratic fit with 25 neighbors is used at $x = 8$. The data in each neighborhood are shown with large triangles, rectangles and circles, respectively. The local fits are shown as thick solid lines. The approximation at the first two points is quite good. The estimate at $x = 8$ is worse. Since we know the true function, we can use those values with the local quadratic fit. The resulting fit at $x = 8$ is shown as a thick dashed line. It is seen to coincide with the target function. Consequently, the approximation error at $x = 8$ is a consequence of the local noise realization.

[25] We observe that the quality of the local regressions is quite good. The higher-order (quadratic) fit is less biased than the linear, and the bias decreases substantially as the size of the neighborhood is reduced from 100 to 10 or 25 points. However, the problem with local regressions of increased variability of estimate due to the reduced sample size is also shown. This is exacerbated as one moves to a higher-order polynomial fit with the same number of data points.

[26] Thus there is a trade-off between bias and variance as one changes the order of the local polynomial and the number of points used to fit it. Another potential problem arises if the x_i values are randomly sampled or are clustered in certain locations. Consider for instance the following values of x_i (3.1, 3.2, 3.5, 4.5, 5, 20, 22, 25, 30, 31). Now consider a local regression at x =10. In this case, the first 5 neighbors of the point of estimate are all to its left, and no

information is used from the data on the right. An estimate with $k \leq 5$ will then certainly be an extrapolation of the data used. Recall from section 2, that it is desirable that the point of estimate be centered in its estimation neighborhood. Recall also that the variance of estimate of linear regression increases dramatically as one moves toward the edges of, or out of the range of data. Consequently, in the multivariate setting, we may find it desirable to devise a strategy by which we symmetrize the neighborhood of the point of estimate, and to explicitly consider the degree of extrapolation involved in a candidate local regression.

[27] Parameter selection approaches are usually based on an estimate of the Mean Square Error of the estimation scheme. In the current setting, we have to be careful, since one can easily choose k and p to drive the mean square error of fit to zero, and the associated R^2 will be 1! Of course, this may not say much about performance in a predictive as opposed to fitting setting. Consequently, we shall investigate some cross-validatory choices of the predictive mean square error for parameter selection.

[28] A variety of estimators of predictive mean square error $P(\hat{f})$ have been proposed in the literature. *Cleveland and Devlin* [1988] considered Mallows C_p in their work on local polynomial regression. *Li* [1985] discusses the theoretical foundations of this and other measures such as ordinary and generalized cross validation, the finite prediction error, the Akaike information criteria (AIC) and the Bayesian information criteria (BIC). Of these the generalized cross-validation (GCV) statistic proposed by *Craven and Wahba* [1979] is of particular interest as an estimate of $P(\hat{f})$since it has performed well [*Härdle*, 1984, 1989] in practical applications. It is defined as

$$GCV\left(\hat{f}\right) = MSE\left(\hat{f}\right) / \left(n^{-1} tr[\mathbf{I} - \mathbf{H}]\right)^2 \tag{9}$$

where the global mean square error of fit averaged over all the original sample values is given by

$$MSE\left(\hat{f}\right) = n^{-1} \sum_{i=1}^{n} \left(y_i - \hat{f}_i\right)^2 \tag{10}$$

H is the influence matrix defined through

$$\hat{\mathbf{f}} = \mathbf{H}\mathbf{y}, \tag{11}$$

I is the identity matrix, and $tr[.]$ represents the trace of the matrix.

[29] Note that (11) represents a linear estimator, and the ith diagonal element of **H** can be thought of as the "weight" of that data point on the estimate at that point. *Eubank* [1988, p. 406] states that for linear regression (local or global, and on the raw variable or a polynomial in it) it is easy to show that $0 \leq h_{ii} \leq 1$. Thus, if each h_{ii} is 1, and the other h_{ij} are 0, we see that we have 0 degrees of freedom, and the estimate at each point is simply the original data, i.e., the model completely overfits or undersmooths. The corresponding MSE is zero, and the GCV is infinity. On the other hand if all the h_{ij} are equal, the estimate at every point is the sample average of the y_i, the degrees of freedom are $(n - 1)$, since we fit one parameter, and the MSE may be large if $f(.)$ is not a constant, and for n large, MSE and GCV will approach each other in magnitude. Consider also the case where the h_{ij} are equal for the k nearest neighbors of a point and 0 elsewhere. In this case we may approximate $f(.)$ better since we form a moving average of y values, and hence have a lower MSE. However, the degrees of freedom will only be $(k - 1)$, and the GCV may be larger. The denominator in (4) consequently has the role of a penalty for the effective number of parameters used in fitting the model. The effective number of parameters is determined by the number of neighbors and the number of terms in the local polynomial.

[30] The motivation for using $GCV(\hat{f})$ comes from a theorem proven by *Craven and Wahba* [1979]. They showed that $GCV(\hat{f})$ is a nearly unbiased estimator of $P(\hat{f})$. *Eubank* [1988] also shows that the GCV is closely related to the ordinary cross-validation (OCV) estimate of $P(\hat{f})$ (p. 30), and to the Akaike information criteria (p. 40). OCV considers the Mean Square Error in dropping one observation at a time from the fitting set and then predicting it using the remaining data. It also estimates $P(\hat{f})$, but at a higher computational cost.

[31] We shall consider global (over the whole data set) and local estimates of $GCV(\hat{f})$to aid parameter selection. The global GCV can be estimated after performing n local regressions, one at each data point x_i, as

$$GGCV\left(\hat{f}\right) = \left(\sum_{i=1}^{n} e_i^2 / n\right) / \left(1 - \sum_{i=1}^{n} h_{ii}/n\right)^2 \tag{12}$$

where $h_{ii} = \mathbf{z}_i^T \left(\mathbf{Z}_i^T \mathbf{W}_i \mathbf{Z}_i\right)^{-1} \mathbf{z}_i w_{ii,i}$, z_i is the augmented n'_p vector corresponding to the k nearest neighbors of $\mathbf{x}_i$, $\mathbf{W}_i$ is the weight matrix for estimation at $\mathbf{z}_i$, and $w_{ii,i}$ is the weight applied to $\mathbf{z}_i$, and where $e_i = y_i - \hat{f}(x_i)$. One can select appropriate values of k and p, as the minimizers of the GGCV value computed in equation (12) for each combination of k and p. These would be the values of k and p that would do well on the average. However, in certain situations (e.g., where the curvature of the target function varies over the data, and where the variance of the noise varies over the range of the data), one may wish to make such choices locally at the point of estimate.

[32] A local GCV score that uses data directly from the local regression at the point of estimate can also be constructed. In this case the errors $e_{i,l}$ are the residues of the model fitted over the k nearest neighbors of the point $\mathbf{x}_l$*, and $\mathbf{W}_i$ is the corresponding weight matrix. The trace of the matrix **H** in this case is simply n'_p, the number of coefficients fit. The local GCV score is then given as

$$LGCV_l\left(\hat{f}\right) = \mathbf{e}_l^T \mathbf{W}_l \mathbf{e}_l / \left(\left(k - n'_p\right)/k\right)^2 \tag{13}$$

[33] The appropriate values of k and p can then be obtained as the ones that minimize the local GCV score for the local regression. The $LGCV_l$ value also provides insight into the local predictive error variance. This approach to parameter selection is particularly useful when making a few estimates from a large data set. This can be the case when local regression is used for forecasting a nonlinear time series model, as is done later in this paper. Note that an improved estimate of the LGCV at a point of estimate could

be obtained as a weighted average of the LGCV values at a suitable number of neighbors of that point. This adds to the computational burden but can reduce the variance of LGCV estimates.

[34] A problem with the two selectors introduced thus far is that they presume a regular or well behaved sampling of the data x_i, and do not address the issue raised earlier of local data clustering and its effects on the regression. Since, both the global and the local GCV statistic are based on estimation at observed points (which may not coincide very well on average with the points at which we need estimates), they provide only limited insight into the best parameters to use locally, particularly in areas that are sparsely sampled. A modification of the local GCV score that attempts to address this problem is now introduced.

[35] Comparing equations (7) and (13) we see that they differ by a factor $(k - n'_p)/k$. In this context, the predictive error variance is merely a penalized version of the error variance for fitting, with a penalty that comes directly from the average weight (n'_p/k) ascribed to each data point during the fitting process. Now, following *Stuart and Ord* [1991 p. 1080], the estimation error variance for an estimate $\hat{y}_l$ at a data point z_l is obtained in terms of the "leverage" h_l exerted by the point of estimate on the observational data set as:

$$\mathrm{var}(\hat{y}_l) = s^2_{e_l} h_l \tag{14}$$

where $h_l = \{\mathbf{z}_l\ (\mathbf{Z}_l^T\ \mathbf{W}_l\ \mathbf{Z}_l)^{-1}\ \mathbf{z}_l^T/k\}$.

[36] The leverage h_l for a point of estimate located at the mean of the k nearest neighborhood is n'_p/k. As one moves away from the center, the leverage value increases. The reader may recall that the confidence limits for linear regression expand as one moves away from the center. This corresponds to the increasing pointwise error variance due to increasing leverage, as in equation (14).

[37] We can then define a corresponding measure of local predictive mean square error with consideration of the "leverage" the point of estimate exerts on the data set as

$$LGCVLEV(z_n) = LGCV_l h_l \tag{15}$$

[38] The LGCVLEV score now has a penalty to account for the degree of extrapolation at the point of estimate relative to the k nearest neighborhood. If the data are clustered and the point of estimate lies in the middle of the cluster, h_l and hence the penalty are the lowest. On the other hand, if the point of estimate is outside the cluster, and one is extrapolating, h_l and the penalty will be higher. In this case, if LGCVLEV is used to choose the local k, the value of k should increase until one or more points outside the cluster enter the local data set. Of course, the inclusion of these points could increase the mean square error. A trade-off between symmetrization of the neighborhood and centering it about the point of estimate, and between increasing bias resulting from this growth in the neighborhood size is thus recognized. If symmetrization is not possible (e.g., if the point of estimate lies outside the original data set), the leverage penalty will try to shrink the size of the neighborhood, provided that the LGCV score is not significantly increased. The absolute and relative values of leverage increase as the order p of the polynomial is increased. Thus this statistic will emphasize lower-order fits. Recall from (2) that the approximation error of the scheme depends on both the order of the polynomial and on the size of the neighborhood.

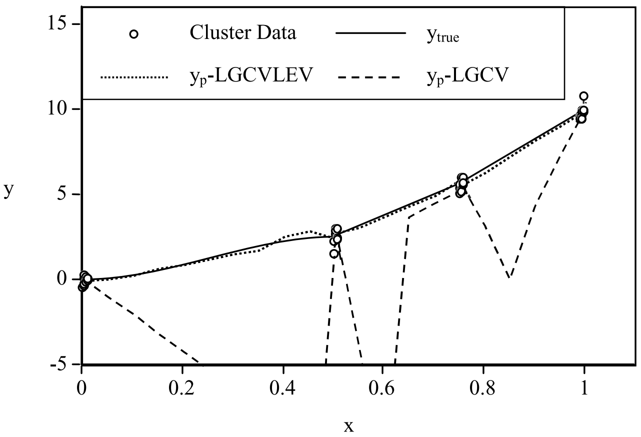

Figure 2. Univariate regression example of the utility of LGCVLEV.

[39] A simple univariate regression example that shows the performance of LGCV and LGCVLEV for selecting the number of neighbors and polynomial order to use is presented in Figure 2. A data set of length 40 is generated from the model $y = 10x^2 + N(0, 0.1)$. The sampling locations are organized into four clusters centered at (x_c = 0.005, 0.505, 0.755, 0.995), with each cluster spread between $x_c \pm 0.005$. Twenty-five points (4 at the cluster centers, and 21 equally spaced from 0 and 1) were considered for estimation. Locally linear fits were considered with k ranging from 4 to 40. LGCV almost always (23 out of 25 cases) picks k = 5. GGCV also picks k = 5, and hence gives the same estimates as LGCV. Since each cluster has 10 points in it, in most cases the resulting estimates will be based only on data from one cluster. The resulting fit is seen to be quite poor except at the estimation points that lie in or very close to the clusters. LGCV estimates that are really poor are off the graph (worst fit is at x =0.3 where the value is −70.2). LGCVLEV picks k = 8 to 17 for the estimates within clusters, and k =17 to 29 for the estimation points that are not in the clusters. In this case, the data from each cluster is used for estimates within clusters, and data from more than one cluster is used when the point of estimate is in between clusters. The resulting estimates are seen to be quite good over the range of the data. The "bias" in the estimate between x = 0.4 and x = 0.5, corresponds to a LGCVLEV choice of k = 29. Such increased variability is a consequence of the increased number of parameter choices when choosing parameters locally.

[40] We see that when the point of estimate lies in a local cluster of data, both measures give the same optimal choice. When the point of estimate, lies in between clusters, LGCV will pick a value of k that leads to estimates from one of the clusters only, whereas LGCVLEV picks a value that forms the k nearest neighborhood using the 2 adjoining clusters, in a proportion that depends on the relative distance of x_l from the 2 clusters. The behavior of LGCV and LGCVLEV near the boundary of the data set is also different as was indicated earlier. For this example the global GCV choices are the same as those for the local GCV.

[41] Local parameter selection can lead to increased variability in the overall estimation scheme. In practice, we recommend that a number of local estimates at the *m* desired locations be made. Now one can form the average LGCV or LGCVLEV score as a function of *k* and *p* across this set of *m* prediction points. Determine the *k* and *p* values that minimize these average scores. If these values are not significantly different from the values obtained locally, it is expedient to use the same *k* and *p* across all prediction points. Variants of this idea across subregions of data are also useful.

[42] Approximate confidence intervals assuming normally distributed local errors with local error variance $s_{e_1}^2$ can be obtained for each estimate $\hat{f}(x_l)$ following *Stuart and Ord* [1991, p. 1043] as

$$\hat{f}(x_l)_{\alpha_P} = \hat{f}(x_l) \pm t_{k-n'_p,1-\alpha/2} \times s_{e_l} \qquad (16)$$

where $t_{k-d',1-\alpha/2}$ is a Student's *t* variate with $(k - n'_d)$ degrees of freedom, and significance level α.

[43] Note that the above confidence interval is likely to be narrower than it should be, because it does not recognize the variability one should associate with the choice of *k* and *p*. Some consideration of these factors is possible through the use of LGCVLEV$^{0.5}$ in place of s_{e_l} in equation 16, since LGCVLEV corresponds to an estimate of error variance that recognizes the price to be paid for estimating the smoothing parameters. Prediction intervals for regression are usually given as

$$\hat{f}(x_l)_{\alpha_P} = \hat{f}(x_l) \pm t_{k-n'_p,1-\alpha/2} \times s_{e_l}(1 + h_l) \qquad (17)$$

Here they are analogously, specifed by using LGCVLEV$^{0.5}$ in place of s_{e_l}.

[44] A better, but computationally more intensive approach to prediction intervals is provided by *Yao and Tong* [1994], who directly solve for the α conditional percentiles of *y* given x_l, by considering an appropriate asymmetric quadratic loss function that depends on the percentile of interest α.

4. Algorithm

[45] The local polynomial regression algorithm is now summarized.

1. Given data (x_i, y_i), $i = 1 \ldots n$
   1a. Plot Scatterplots of *y* vs. each component of *x*.
   1b. Decide on a range $\{k_1, k_2\}$ of nearest neighbors, and $\{p_1, p_2\}$ of polynomial order. Typically $p_1 = 0$ or 1, and p_2 is 1 or 2; $k_1 \geq 2 \times$ n$_p$, where n_p is the total number of coefficients to solve for, and $k_2 = n$. The case $p_1 = 0$ takes care of locally constant fitting, i.e., kernel regression.
2. For each point of estimate $\mathbf{x}_l$, $l = 1 \ldots L$, estimate a local regression (equations (4) and (5)) for each value of *k* and *p* considered.
   2a. Form the augmented $k \times n_p$ matrix $\mathbf{Z}_l$ for the polynomial basis indicated by *p* with *k* nearest neighbors.
   2b. Form the $k \times k$ matrix of weights $\mathbf{W}_l$ indicated by the number of neighbors *k*.
   2c. Identify for each such regression, the coefficients β_j, $j = 1 \ldots n'_p$, and associated columns of the matrix $\mathbf{Z}_l$ to retain a subset of n'_p predictors using equation (8). Re-solve the model with the n'_p predictors. Retain the optimal number of predictors for this *k* and *p* choice based on LGCV or LGCVLEV.
   2d. Select optimal $(k^*, p^*)_l$ as the minimizers of LGCVLEV (or LGCV) over *k* and *p*.
   2e. Retain for each such regression, the following statistics: (1) the estimate $\hat{f}(x_l)$, (2) the residual $e_l(k, p)$, (3) $LGCV_l$ (k,p), (4) $LGCVLEV_l$ (k,p), (5) $s_{e_l}{}^2$, (6) h_{ii} (equation 12), and h_l (equation 14), (7) the number of coefficients, n'_p at were selected, and (8) $(k^*, p^*)_l$
   2f. Provide confidence and prediction intervals for $\hat{f}(x_l)$ corresponding to n'_p, k^*, p^* if needed.
3. (OPTIONAL) Compute the following statistics from the information retained in 2.
   3a. GGCV(k, p) (equation 12), Average ($LGCVLEV_l$ (k,p)), Average ($LGCV_l$ (k,p)). The averages are taken over $l = 1 \ldots L$.
   3b. Determine the k^*, p^* that minimize (1) GGCV, (2) Average (LGCVLEV) and (3) Average (LGCV). If these are all similar and the spread of $(k^*, p^*)_l$ is relatively small, use these values at all *l*, else use the $(k^*, p^*)_l$ determined in 2.

5. Application

[46] The primary application we consider in this paper is the forecast of the volume of the Great Salt Lake at key points in time from its 1847–2004, biweekly time series. However, we shall begin by forecasts of data from two known models to assure ourselves that the forecasting scheme can work. The generic forecasting model is described first.

[47] Let us say that we have a time series, G_t, $t = 1 \ldots nt$. We shall presume that G_t is the outcome of a Markovian (i.e., future values depend only on a finite set of past values) process, and consider a nonlinear, autoregressive model as appropriate for forecasts of such a system. We refer the reader to *Tong* [1990] for theoretical background and attributes of such an approach. The model of interest is stated as:

$$G_{t+m} = f_m(\mathbf{V}_t) + e_t \qquad (18)$$

where G_{t+m} is an m step ahead forecast, $f_m(\mathbf{V}_t)$ is a forecast or recursion function (the conditional expectation of G_{t+m} given $\mathbf{V}_t$), that is presumed to be continuous and twice differentiable, e_t is a independent and locally (in the *k* nearest neighborhood of $\mathbf{V}_t$) identically distributed noise process with zero mean and finite variance, $\mathbf{V}_t$ is a state vector of length m_1 with components $(G_t, G_{t-\tau}, \ldots G_{t-(m-1)\tau})$, where τ is a delay or lag between coordinates.

[48] Now, one can consider two approaches for forecasting the *m* step ahead value for the time series. One could forecast 1 step ahead *m* times, updating $\mathbf{V}_t$ to $\mathbf{V}_{t+1}, \ldots \mathbf{V}_{t+j}$ etc., with the *j* forecasted values G_{t+j} that are available at that point. This is called an iterated forecast. Alternately, one could directly forecast m steps ahead using the current

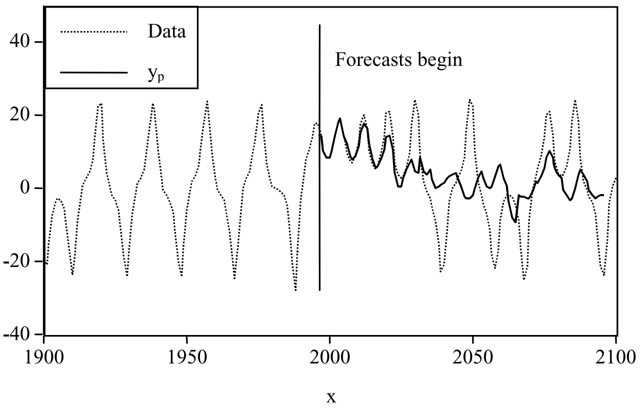

Figure 3. Direct m-step forecasts of Lorenz data starting from index 1997. The dotted line represents the actual values, while the solid lines are the forecasted values ($m_1 = 5$, $\tau = 3$, $k = 50$ at the first 21 points and 90 or 150 at the rest of the points, $p = 1$ at the first 9 points, and $p = 2$ at the remaining points). The divergence of the forecasted and the observed trajectories near index 2030 is characteristic of the loss of predictability in the Lorenz system as trajectories pass near the unstable point ($x = y = z = 0$). The increase in k and p after the first 20 points may reflect increasing derivatives of $f(V_t)$ as one approaches the origin.

$\mathbf{V}_t$. In either case, m successive estimates of $f_m(\mathbf{V}_t)$ are needed. We have explored both strategies in our work, and found them to give comparable results for the data sets tested. In what follows, the direct prediction method is described. Iterated forecasts follow a very similar strategy.

5.1. Direct Forecasts of a Time Series for m Steps Forward From Time Index *t* = *nt*

[49] 1. Select a lag τ and an embedding dimension m_1. These are parameters that can be varied and picked as the ones that minimize the GGCV for the m step forecasts G_{t+m}, or they can be chosen based on other prescriptive criteria (e.g., τ as the delay that corresponds to the first minimum of the average mutual information and m_1 as the value that minimizes the percentage of false nearest neighbors; see *Abarbanel et al.* [1993] for details of both prescriptions). Usually, these will not be changed during the m step prediction, but may change if the forecasts are started under rather different conditions.

[50] 2. Form a data matrix $\mathbf{X}$ with m_1 columns corresponding to $\mathbf{V}_t$, and $(nt - (m_1 - 1)^* \tau - m)$ rows from the time series G_t using the appropriate lags of the time series. For example say $\tau = 2$, $m_1 = 3$. Then the first row of $\mathbf{X}$ will have G_5, G_3, G_1; and the last row will have $G_{nt\text{-}m}$, $G_{nt\text{-}m\text{-}2}$, $G_{nt\text{-}m\text{-}4}$.

[51] 3. Form a data vector $\mathbf{y}$ corresponding to $\mathbf{X}$, as the value G_{t+m} corresponding to each row of the first column of $\mathbf{X}$. For example, for m =5 in the previous example, the first entry of $\mathbf{y}$ is G_{10}, and the last entry is G_{nt}.

[52] 4. Solve the local regression problem at $\mathbf{x}_l = G_{nt}$, $G_{nt\text{-}\tau}$, $G_{nt\text{-}(m\text{-}1)\tau}$ as in section 4.

[53] 5. Evaluate the forecast and provide confidence/prediction intervals if needed.

[54] Note that since $\mathbf{V}_t$ does not change for each step in the direct forecasting method, if k and p do not change one can retain the matrix $(\mathbf{Z}_1\mathbf{W}_1\mathbf{Z}_1)^{-1}$ between forecasts, and speed up the computations.

5.2. Synthetic Series

[55] The first scenario is a time series of length 200 from an AR(2), or autoregressive model of lag 2. The model is defined by:

$$y_t = y_{t-1} - 0.5y_{t-2} + \varepsilon_t \tag{19}$$

where ε_t is a Normally distributed random variable with mean 0 and variance 1.

[56] In this case, we selected $\tau = 1$, and varied m_1 from 1 to 5. The number of nearest neighbors to use was varied from 50 to 195 (200 embedding dimension), and linear and quadratic (without cross-product terms) fits were considered from 1 to 20 various points in the series. In all cases LGCVLEV was used to select the parameters of interest. The number of nearest neighbors was consistently picked as the full sample size, m_1 was typically picked to be 2, and linear fits were always selected. Since the models selected were essentially global, linear, autoregressive models each time, the resulting statistical properties were satisfactory.

5.3. Lorenz Equations

[57] The Lorenz equations [*Lorenz*, 1963] are given as

$$\begin{aligned} \dot{x} &= -\sigma(x + y) \\ \dot{y} &= -xz + rx - y \\ \dot{z} &= xy - bz \end{aligned} \tag{20}$$

[58] Here we took $\sigma = 16$, $r = 45.92$, $b = 4$, and a numerical integration time step $\partial t = 0.05$, and sampled the x state variable. This is a chaotic system that has been well studied by many investigators. From prior work [*Lall et al.*, 1996; *Moon et al.*, 1995] we knew that one should expect $\tau = 2$ to 4, and $m_1 = 4$ to 6. Consequently we investigated these values and $k_1 = 50$ to $k_2 = 150$, and $p_1 = 1$, $p_2 = 2$. Forecasts from index 1996 of the x time series are presented in Figure 3. No data after index 1996 were used for the forecasts.

[59] The Lorenz system has an instability near $x = y = z = 0$. Trajectories that approach this state tend to diverge rapidly. We notice from Figure 3 that the forecasts of the Lorenz x variable are quite good until the trajectory passes near the unstable point. A small uncertainty in the values of the state vector at index 1996 leads to a similar divergence in the trajectories from the numerical simulation of the Lorenz equations. Thus this divergence is intrinsic to this model. Subsequent similarity in the forecast and actual trajectories is coincidental. Similar results were obtained by *Lall et al.* [1996] using MARS.

5.4. Great Salt Lake Forecasts

[60] The Great Salt Lake (GSL) of Utah is a closed lake in the lowest part (elevation 1280 m above mean sea level) of the Great Basin (latitudes 40°20′ and 41°40′N, and longitudes 111°52′ and 113°06′W), in the arid western United States. The GSL is approximately 113 km long and 48 km wide, with a maximum depth of 13.1 m and an average depth of 5.0 m. The large surface area and

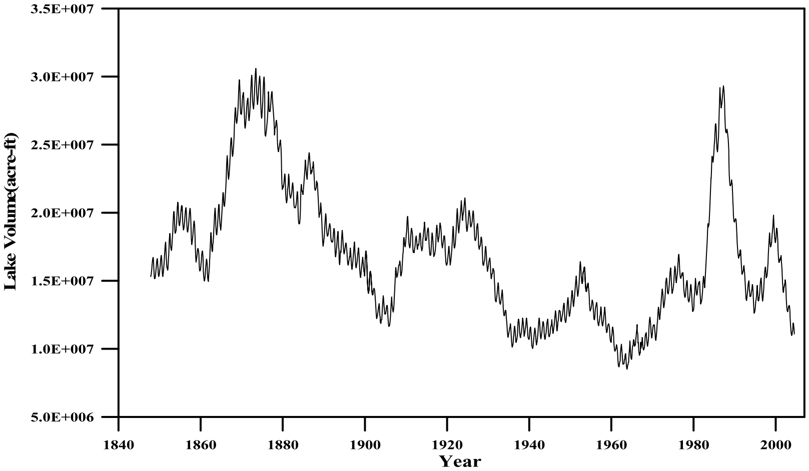

Figure 4. Biweekly time series of the Great Salt Lake, 1847–2004.

shallow depth make the lake very sensitive to fluctuations in long-term climatic variability. As shown in Figure 4, the lake volume has varied considerably over decadal timescales during the last 160 years. The low-frequency character makes this an interesting time series to forecast.

[61] We considered blind forecasts of the GSL volume from different states for 1 to 2 years into the future from the date of forecast. The forecasted values are then compared with the volumes that were actually recorded subsequently. They are presented in Figures 5 and 6. The lag τ was selected as 10 as in the range of the first minimum of the average mutual information [*Moon et al.*, 1995] limited experimentation with other lags suggested that this was a good choice. An embedding of $m_1 = 5$ was selected after experimentation with various values in the range 1 to 9. Usually, this value corresponded to the one that minimized GGCV, LGCV or LGCVLEV. We searched over $k_1 = 50$ to $k_2 = 150$ nearest neighbors and typically selected 120 to 150. Locally linear and quadratic (without x products) fits were considered. Typically linear was selected.

[62] Figure 5 provides a sequence of 1 year blind forecasts of the GSL using the direct m step method, from August 1977 to July 1987. The dots represent the observed GSL time series. The solid lines represent 12 forecasts, one for each month of the next year. Only data available up to the beginning of the forecast period is used for fitting and forecasting. Given the extreme nature of the 1983–1987 period the predictions appear to be quite good. Of particular interest is the forecast starting in August 1983. The predictability is quite poor for this forecast.

[63] Figure 6 shows a sequence of 2 year blind forecasts of the GSL using the direct m step method, from August 1977 to July 1988. The dots represent the observed GSL time series. The solid lines represent 24 forecasts, one for each month of the next year. Only data available up to the beginning of the forecast period is used for fitting and forecasting. Comparing with Figure 5, we see that some trajectories (e.g., August 1980 start, August 1983 start) that appeared to be departing from the observed trajectory at the end of 1 year have followed the observed trajectory during the next year. However, others (e.g., August 1982 start) have continued to drift away. The forecast started in August 1981 does quite well till it reaches August 1982 when it behaves much like the first year of the forecast started in August 1982. The second year of the forecast started in August 1985 diverges from the observed trajectory, while

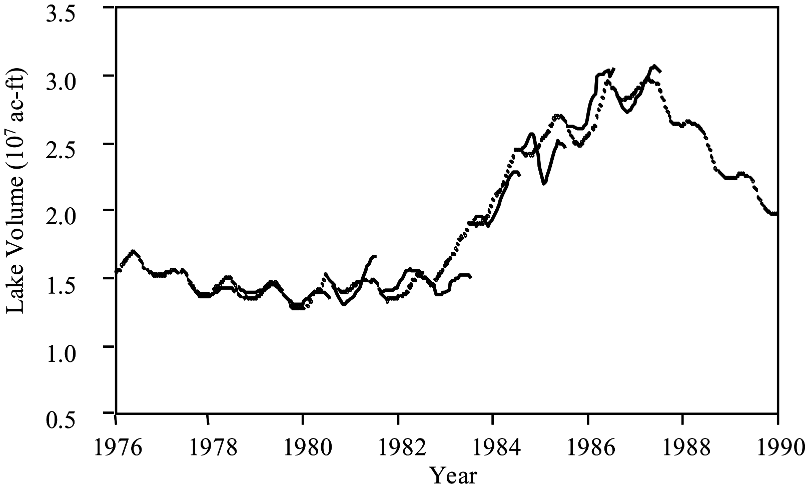

Figure 5. Sequence of 1 year blind forecasts of the GSL.

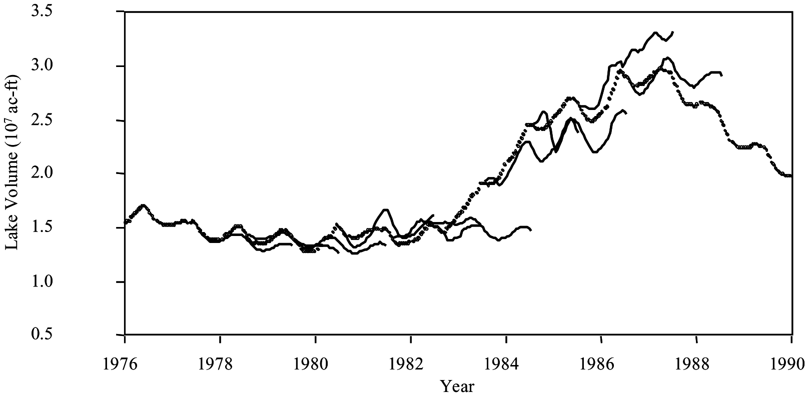

Figure 6. Sequence of 2 year blind forecasts of the GSL.

the forecast with information up to August 1986 does much better. Clearly, predictability is quite high for the conditions in the 1970s.

[64] Finally, a forecast of the Great Salt Lake volume for 1 year beginning July 2004 is presented in Figure 7.

6. Summary

[65] A locally weighted polynomial regression methodology for approximating nonlinear regressions was introduced in this paper. The basic algorithm presented is derived from the work by *Cleveland and Devlin* [1988]. A new, local cross-validatory criteria was introduced for selecting the smoothing parameters of the method while considering a bias-variance trade-off in estimation and symmetrization of the *k* nearest neighborhood of the point of estimate. The utility of this selector for this purpose was demonstrated with a simple example.

[66] From Taylor series, the approximated error of a local linear model with small *k* is comparable to a local quadratic model with large *k*. So, if *n* is much greater than *d* and f'' is high then a local quadratic model will be picked. If *n* is not much greater than *d* then a local linear model is picked.

[68] The methodology presented was then applied to the forecast of selected time series with encouraging results. These methods are still evolving. We can expect improvements in procedures for estimating prediction intervals and for selecting parameters of the method. Algorithmic improvements for more efficiently exploiting multivariate data structures are also to be expected.

[69] Generally, if a linear, stochastic time series model is appropriate, the LGCVLEV based approach leads to a selection of *k*, the number of nearest neighbors that is close to the full sample size. Conversely, where the target function is highly nonlinear, and the noise in the data is low, a small value of *k* is selected. Thus the model building process also helps us get some insight as to the degree of linearity/nonlinearity and noise associated with the time series. Since all real world data will have varying degrees of measurement or representation error associated with them, it is difficult to uniquely ascribe the dynamics to stochastic or chaotic origins. Linear versus nonlinear dynamics can be diagnosed subject to the degree of noise in the data.

[70] Nonparametric regression algorithms include kernel regression (locally constant polynomial), spline smoothing, spline regression, orthogonal series estimators, and artifi-

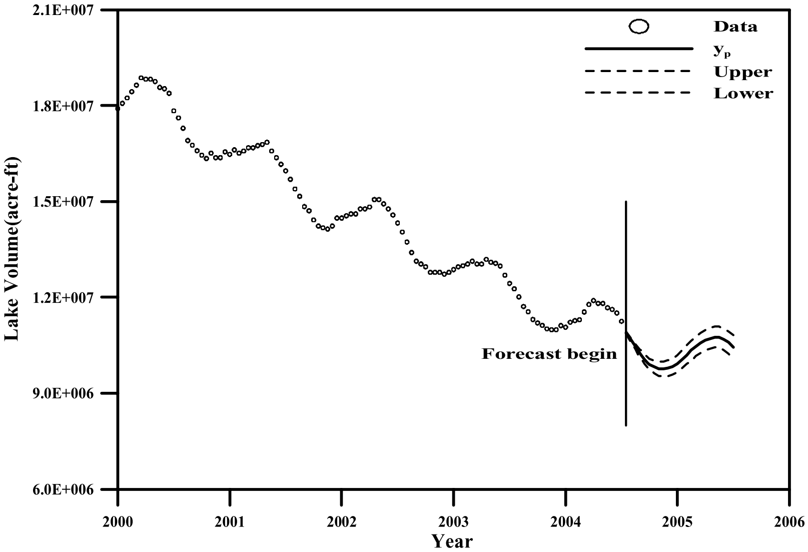

Figure 7. Forecasts for 1 year starting July 2004. The solid line is the forecasted sequence, while circles are the actual series. Approximate 90% prediction intervals using LGCVLEV for the forecast are also shown.

cial neural networks. Such algorithms are attractive because they can approximate a wide class of functions without knowledge of the function. Consequently, as sample sizes of available data have grown, and computational factors are no longer limiting, the approximation of arbitrary nonlinear structures in relationships has become feasible. The real world performance of these algorithms with proper parameter selection criteria is usually similar. Typically differences across these algorithms arise largely due to stability and overparameterization during predictor and parameter identification. The advantage of the local polynomial regression is that it is easy to understand and implement and provides access to well developed results from linear model theory. The LGCVLEV criteria introduced here addresses an important aspect of predictor sampling design through symmetrization of the neighborhoods selected and through a reduction in the potential for overparameterization. This criteria is uniquely suited to the local polynomial setting since it exploits the local structure of the regression/function estimation problem.

[71] **Acknowledgments.** Partial support of this work by the USGS grant 1434-92-G-226 and NSF grant EAR-9205727 is acknowledged. Comments from three anonymous reviewers contributed to a significant improvement of the manuscript.

References

Abarbanel, H. D. I., R. Brown, J. J. Sidorowich, and L. S. Tsimring (1993), The analysis of observed chaotic data in physical systems, *Rev. Modern Phys.*, *65*(N4), 1331–1392.

Abarbanel, H. D. I., U. Lall, Y.-I. Moon, M. Mann, and T. Sangoyomi (1996), Nonlinear dynamics of the Great Salt Lake: A predictable indicator of regional climate, *Energy*, *21*(7/8), 655–665.

Cleveland, W. S. (1979), Robust locally weighted regression and smoothing scatterplots, *J. Am. Stat. Assoc.*, *74*(368), 829–836.

Cleveland, W. S., and S. J. Devlin (1988), Locally weighted regression: An approach to regression analysis by local fitting, *J. Am. Stat. Assoc.*, *83*(403), 596–610.

Cleveland, W. S., S. J. Devlin, and E. Grosse (1988), Regression by local fitting, *J. Econometrics*, *37*, 87–114.

Craven, P., and G. Wahba (1979), Smoothing noise data with spline functions, *Numer. Math.*, *31*, 377–403.

Efron, B., and R. J. Tibishirani (1993), *An Introduction to the Bootstrap*, CRC Press, Boca Raton, Fla.

Eubank, R. (1988), *Spline Smoothing and Nonparametric Regression*, Marcel Dekker, New York.

Fan, J., and I. Gijbels (1992), Variable bandwidth and local linear regression smoothers, *Ann. Stat.*, *20*, 196–216.

Friedman, J. (1991), Multivariate adaptive regression splines (with discussion), *Ann. Stat.*, *19*, 1–141.

Härdle, W. (1984), How to determine the bandwidth of some nonlinear smoothers in practice, *Lect. Notes Stat.*, *26*, 163–184.

Härdle, W. (1989), *Applied Nonparametric Regression*, *Econometric Soc. Monogr.*, vol. 19, 333 pp., Cambridge Univ. Press, New York.

Hastie, T. J., and C. Loader (1993), Local regression: Automatic kernel carpentry (with discussion), *Stat. Sci.*, *8*, 120-43.

Hastie, T. J., R. Tibshirani, and J. Friedman (2001), *The Elements of Statistical Learning: Data Mining*, *Inference, and Prediction*, 533 pp., Springer, New York.

Helsel, D. R., and R. M. Hirsch (1992), *Statistical Methods in Water Resources*, 524 pp., Elsevier, New York.

Lall, U. (1995), Nonparametric function estimation: Recent hydrologic contributions, U.S. Natl. Rep. Int. Union Geod. Geophys. 1991–1994, *Rev. Geophys.*, *33*, 1093–1102.

Lall, U., T. Sangoyomi, and H. D. I. Abarbanel (1996), Nonlinear dynamics of the Great Salt Lake: Nonparametric short term forecasting, *Water Resour. Res.*, *32*, 975–985.

Lejeune, M. (1985), Estimation non-parametrique par noyaux: Regression polynomial mobile, *Rev. Stat. Appl.*, *33*, 43–67.

Li, K.-C. (1985), From Stein's unbiased risk estimates to the method of generalized cross validation, *Ann. Stat.*, *13*(4), 1352–1377.

Loader, C. (1999), *Local Regression and Likelihood*, Springer, New York.

Lorenz, E. N. (1963), Deterministic nonperiodic flow, *J. Atmos. Sci.*, *20*, 130–141.

Macauley, F. R. (1931), *The Smoothing of Time Series*, Natl. Bur. of Econ. Res., New York.

Moon, Y.-I., B. Rajagopalan, and U. Lall (1995), Estimation of mutual information using kernel density estimators, *Phys. Rev. E*, *52*(3), 2318–2321.

Müller, H.-G. (1987), Weighted local regression and kernel methods for nonparametric curve fitting, *J. Am. Stat. Assoc.*, *82*, 231–238.

Owosina, A. (1992), Estimation of the space and time variability of nonpoint source groundwater contamination, M.S. thesis, 191 pp., Utah State Univ., Logan.

Press, W. H., B. P. Flannery, S. Teukolsky, and W. T. Vetterling (1989), *Numerical Recipes: The Art of Scientific Computing*, Cambridge Univ. Press, New York.

Sangoyomi, T. B. (1993), Climatic variability and dynamics of Great Salt Lake hydrology, Ph.D. dissertation, 247 pp., Utah State Univ., Logan.

Scott, D. W. (1992), *Multivariate Density Estimation*, 127 pp., John Wiley, Hoboken, N. J.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, 175 pp., CRC Press, Boca Raton, Fla.

Stuart, A., and J. K. Ord (1991), *Kendall's Advanced Theory of Statistics*, vol. II, 1323 pp., Oxford Univ. Press, New York.

Tong, H. (1990), *Non-linear Time Series: A Dynamical System Approach*, 564 pp., Clarendon, Oxford, U. K.

Wand, M. P., and M. C. Jones (1995), *Kernel Smoothing*, 232 pp., CRC Press, Boca Raton, Fla.

Yao, Q., and H. Tong (1994), On prediction and chaos in stochastic systems, *Philos. Trans. R. Soc. London, Ser. A.*, *348*, 357–369.

K. Bosworth, Department of Mathematics, Idaho State University, Campus Box 8060, Pocatello, ID 83209, USA.

H.-H. Kwon and U. Lall, Department of Earth and Environmental Engineering, Columbia University, 918 Mudd, 500 West 120th Street, New York, NY 10027, USA. (ula2@columbia.edu)

Y.-I. Moon, Department of Civil Engineering, University of Seoul, Seoul, South Korea.