LA-13076-MS

Knowledge Fusion: Comparison of Fuzzy Curve Smoothers to Statistically Motivated Curve Smoothers

Los Alamos
NATIONAL LABORATORY

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

Edited by Paul W. Henriksen, Group CIC-1
Prepared by Sharon Hurdle, Group NIS-7

This work was supported by the U.S. Department of Energy, Office of Nonproliferation and National Security.

An Affirmative Action/Equal Opportunity Employer

Footnote, if there is one

LA-13076-MS

UC-700
Issued: March 1996

Knowledge Fusion: Comparison of Fuzzy Curve Smoothers to Statistically Motivated Curve Smoothers

Tom Burr
Richard B. Strittmatter

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

Knowledge Fusion: Comparison of Fuzzy Curve Smoothers to Statistically Motivated Curve Smoothers

by

Tom Burr and Richard Strittmatter

ABSTRACT

This report describes work during FY 95 that was sponsored by the Department of Energy, Office of Nonproliferation and National Security (NN) Knowledge Fusion (KF) Project. The project team selected satellite sensor data to use as the one main example to which its analysis algorithms would be applied. The specific sensor-fusion problem has many generic features, which make it a worthwhile problem to attempt to solve in a general way. The generic problem is to recognize events of interest from multiple time series that define a possibly noisy background. By implementing a suite of time series modeling and forecasting methods and using well-chosen alarm criteria, we reduce the number of false alarms. We then further reduce the number of false alarms by analyzing all suspicious sections of data, as judged by the alarm criteria, with pattern recognition methods. A companion report (Ref. 1) describes the implementation and application of this 2-step process for separating events from unusual background. Reference 1 presents a somewhat detailed application of 12 different time series forecasting methods, followed by 7 different pattern recognition methods. One of the 12 forecasting methods and 1 of the 7 pattern recognition methods was written for the KF project. This report gives a detailed comparison of two of the forecasting methods (fuzzy forecaster and statistically motivated curve smoothers as forecasters). The two methods are compared on five simulated and five real data sets. One of the five real data sets is satellite sensor data. **The conclusion is** the statistically motivated curve smoother is superior on simulated data of the type we studied. The statistically motivated method is also superior on most real data. In defense of the fuzzy-logic motivated methods, we point out that fuzzy-logic methods were never intended to compete with statistical methods on numeric data. Fuzzy logic was developed to handle real-world situations where either real data was not available or was supplemented with either "expert opinion" or some sort of linguistic information. That sort of application is one the KF team began to address during 1995 and is discussed in a separate KF report (Ref. 2).

1. Introduction and Summary

This study supports the DOE/NN-sponsored Knowledge Fusion (KF) project's effort to assemble a suite of time series forecasting and modeling methods. Fuzzy-logic-motivated forecasting methods have recently become popular. Therefore, we implemented a fuzzy forecaster and compared its performance to a statistically motivated forecaster on a variety of data sets. We applied both methods to a randomly selected set of five real data sets and to five simulated data sets. We knew that for the kinds of simulated data we studied, the statistically motivated methods would be superior because they were designed to be superior for that type of data. However, real data never follows any model exactly, so we did not know how the two methods would compare on real data. The conclusion here is that the statistically motivated methods are also superior on most real data. In defense of the fuzzy-logic motivated methods, we point out that fuzzy-logic methods were never intended to compete with statistical methods on numeric data. Fuzzy-logic was developed to handle real-world situations where real data was either not available or supplemented with either "expert opinion" or some sort of linguistic information. The KF team began to address that sort of application during 1995, and the effort is discussed in a separate KF report (Ref. 2).

This report is organized as follows. In section 2 we review analysis methods for scalar or vector time series, with emphasis on a fuzzy forecaster and loess: a statistically motivated forecaster that is reasonable to compare to the fuzzy forecaster for reasons we explain. In section 3 we compare the performances of the fuzzy forecaster and loess on five simulated data sets. In section 4 we compare the performances of the fuzzy forecaster and loess to five real data sets. Section 5 is a summary of the time series models considered, and section 6 is a summary of the performances of the two methods.

2. Analysis of Scalar or Vector Time Series

In this section we review vector time series modeling.

2.1 ARIMA Time Series Models

One widely studied class of time series model is the auto-regressive, integrated moving average (ARIMA) model, which we describe now. Suppose that the time series $\{X_1, X_2, \ldots, X_n\}$ has been de-trended by differencing or, equivalently, by fitting polynomial functions of time. The detrended series is said to follow an auto-regressive moving average (ARMA) model.

We can write the general scalar $ARMA\ (p, q)$ model as

$$X_t = a_0 + \sum_{j=1}^{p} a_j X_{t-j} + \sum_{j=0}^{q} b_j \varepsilon_{t-j} \qquad (1)$$

where the a_j and b_j are real constants, the ε_{t-j} are independently and identically distributed (iid) random variables, and $t \in \{1, 2, \ldots, n\}$. It is common to refer to ε_{t-j} as the shock at time $t-j$. Usually ARIMA models further specify that the shock ε_t follows a Gaussian distribution with mean 0 and variance σ^2, denoted $N(0, \sigma)$. In addition, to ensure that the time series is

stationary (constant mean, variance, and covariances) and invertible [representable as an auto-regressive (AR) model], conditions are imposed on the values of the a_j and b_j. See Ref. 3 for further details.

Note that X_t is a linear combination of the past p values of the series (auto-regressive) and of the past q shocks (moving average). We have not restricted this study to the class of linear ARMA models, but because ARMA models are a convenient and easy-to-discuss class of models, most of our discussion will use the ARMA model for an example. Reference 4 introduced 12 candidate ways to model and forecast a vector-based time series. However, at the time Ref. 4 was written, we had no algorithms for vector ARMA time series. All the commercial software only treated vector AR time series. It is more computationally challenging to handle vector ARMA models. To date, we have heard, but not confirmed, that one lesser-known statistical programming package (JMP) can handle vector ARMA time series. We have added the ability within the statistical and graphical programming language S+ by using a user-contributed library of state-space (in the sense of the Kalman Filter language) functions. Details are part of another 1995 KF report (Ref. 5).

2.2 Vector AR Models

In this section we present a quite general AR time series model. Because of issues discussed in Ref. 4, we do not consider nonlinear moving average (MA) models or ARMA models. An AR time series can be written as follow:

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-l}) + e_t \quad . \tag{2}$$

The error e_t is usually assumed to be from some convenient distribution such as the Gaussian but need not be. Nearly always, the distribution of the e_t is at least assumed to be the same for all t. We will assume that the errors have the same distribution F, which we write as $e_t - F(\ )$. Not all functions f combined with error distribution F lead to a stationary time series. We do not attempt a formal treatment of this issue, but rather accept Equation (2) as a basis for trying our two modeling approaches. We also easily generalize Equation (2) to include other time series, say Y and Z:

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-l_x}, Y_t, \ldots, Y_{t-l_y}, Z_t, \ldots, Z_{t-l_z}) + e_t \quad . \tag{3}$$

Note that present values of both Y and Z are allowed because the goal is to forecast Z. In our setting, X is the gamma series, Y is the electron series, and Z is the proton series. Or, in another setting, we have only two series: gammas and a series proportional to the sum of electrons and protons.

Equation (3) can be treated exactly like an ordinary regression model for which there are many techniques, loosely described by the degree of assumptions placed on the functional form for f. We will discuss the two models shortly, but first we point out two advantages to using restrictive models such as models that assume f is linear:

(1) Often there is not enough data to estimate complicated models or a completely unrestricted f. In fact, the old standby, linear regression, will be with us forever for this reason and represents an extreme example of "combining information" in that data at one end of the range is assumed to have the same functional form as data at another end of the range. By the way, because we are insisting on working with stationary series, we often must restrict the time window so that the series can be considered stationary over that window, thereby reducing the effective size of the data set and essentially forcing us to use only the simplest models. More complicated models suffer from the "curse of dimensionality," which we will explain below.

(2) For years the regression literature has been filled with informal confirmation of the "parsimony principle," which states that simplest models are preferred when possible because no data follows any model exactly, and model departures can be more severe when the training data is overfitted by using an overly complex model. Fortunately, we have a straightforward way to guide us toward the proper degree of model complexity: use the model to forecast a held-out (not used for training) testing set and accept the model that performs best on the testing set. A complete treatment of this issue uses what is known as cross-validation to repeatedly divide the data into training and testing sets. Cross-validation is very important when the data sets are small. In our case, we simply did a one-time division into training and testing sets because the data sets were reasonably large.

For notational convenience we will drop the distinction between, for example, the time series X, Y, Z, and will lump all candidate predictors together to form p predictors as in Equation (4):

$$x_t = f(x_{1t}, x_{2t}, \ldots, x_{tp}) + e_t \ . \tag{4}$$

We write Equation (4) without the explicit reference to lagged predictors primarily to not restrict our data sets to AR time series. Any regression setting will be appropriate, as we will see. However, in the context of time series, an important issue is how to choose the lag for each of the time series. We always use a "trial-and-error" approach starting with lag 1 only for the X series (the series to be predicted) and starting with lag 0 for the other series. The winning method is the one that minimizes the sum of squared errors on a held-out testing set.

In our view, using a "curve smoother" algorithm is usually feasible in one dimension or perhaps in a few dimensions. Occasionally, if there is enough data following Equation (2), then a multi-dimensional curve smoother could be feasible. But, because many real data sets exhibit only quasi-stationarity, there is often not as much data following a model like Equation (2) as one might first think. This is an important practical issue that we will treat in a later report. In this report, we restrict attention to data sets where is it feasible to try some kind of curve smoother method (fuzzy-logic motivated or statistically motivated) because there is enough data for the problem's dimensionality.

In Fig. 1 we present one simple idea that pervades much of this report. Consider the model $y = f(x) + e$ in one dimension. In Fig. 1, we have $fx = 4x(1-x)$, which is the well-studied logistic map. The logistic map has properties that are interesting in the study of chaotic time series, but our concern is simply to estimate f as well as possible so that future values of x can be used to forecast future values of y. Now that we know the true functional form, we know that the estimate $\hat{f}$ shown is close to the true f but exhibits a slight overfit of the data because it is more bumpy than the true f. Our estimate $\hat{f}$ was based on 200 observations of (x,y) pairs.

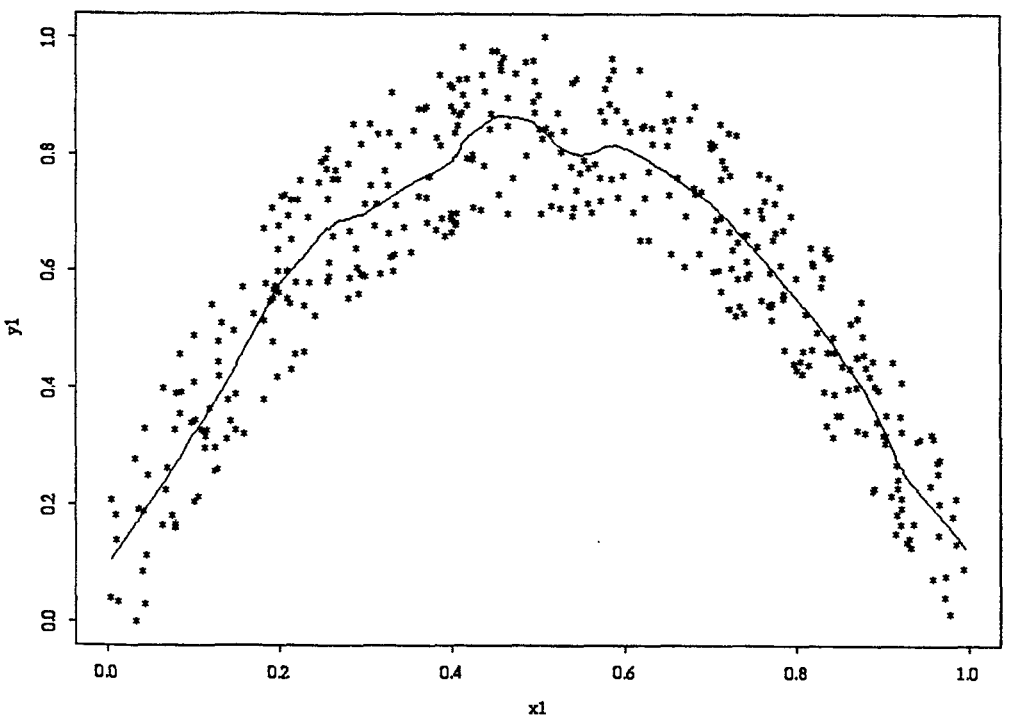

FIGURE 1. Scatterplot illustration of a one-dimensional curve smoother.

The estimate $\hat{f}$ in Fig. 1 was obtained by a local regression (statistically motivated) algorithm available in the statistical programming language called S+. The local regression algorithm is called loess and the following is a brief description of loess in one dimension.

2.2.1 Local regression (loess)

Assume there are n data pairs (x_i,y_i). In the context of a time series, replace (x_i,y_i) with (x_{t-1},x_t) because the previous value could be used in a lag-one model to predict the present value. The loess estimate at any point x uses the q of n data pairs that are closest to the value x. The closest q data pairs are used in a weighted linear or quadratic regression with weights according to the distance from x. Because loess must perform the regression at each point x where the estimate is desired, it is important to use efficient algorithms. See Ref. 6 for computational details. The same idea carries over to multiple dimensions, but the current implementation of loess is limited to up to 15 predictors for local linear fits or up to 4 predictors for local quadratic fits. To complete the discussion of Fig. 1, we mention that we also use a nonparametric smoother (no local linear or quadratic assumption) that we will describe here because it is closest in spirit to the fuzzy-logic motivated curve smoother.

2.2.2 Nonparametric curve smoother

We will present the equations for our one-dimensional curve smoother. The higher dimensional smoother is a natural extension. Our smoother is similar conceptually to other statistically motivated curve smoothers.

$$\hat{f}(x) = \left[(1/(n-1)) \sum_{j=1}^{n-1} X_{j+1} w\left(\frac{x-X_j}{h}\right)\right] / \left[(1/n) \sum_{j=1}^{n} w\left(\frac{x-X_j}{h}\right)\right] \tag{5}$$

Equation (5) gives an intuitive way to estimate f at the point x as follows. All data contributes to the estimated value via a weighted average, with weight given by the distance of the data points from x. A data point, say X_k, that is far from x simply won't contribute much to the estimate at x, provided we choose the smoothing parameter h and the weight function w so that $w((x-X_j)/h)$, which is the weighting term, is small when $x-X_j$ is large. Consider Fig. 1 again, in the light of this description: for a given value $X = x$ the estimate for Y is primarily determined by those Y values that correspond to X values near x. This is a simple idea, but selecting the smoothing degree remains somewhat of an art despite attempts to automate the choice of bandwidth h. However, by using held-out testing sets, it is possible to do a reasonable job of automating the choice of h. Experience and theory suggest that the choice of h is more critical than the choice of w. Typically simple smooth functions such as a Gaussian-shaped function e^{-x^2} are a good choice. Theoretically optimal weight functions such as the Epanechnikov kernel are sometimes suitable, but the best theoretical shape for the weight function depends on the true function f, so we are not fond of using specially motivated weight functions. In fact, we nearly always use simple Gaussian weight functions and concern ourselves with searching for good h. The basic idea is to balance the tradeoff between bias and variance in that too little smoothing overfits the data, reducing bias but increasing variance, and the reverse occurs for too

much smoothing. Trial-and-error is the best way to select bandwidth h. We also have implemented an adaptive kernel method, which allows the bandwidth h to be a function of the local density of data. That is, for x values where the data is sparse, h will be larger so that more neighbors are included in the effective neighborhood size. Currently we favor the non-adaptive kernel method because it is easier and it usually performs just as well.

2.2.3 Fuzzy forecaster (also called fuzzy controller)

We wrote a fuzzy forecaster for the KF project in 1995. The fuzzy forecaster has a few forms, and the form we use here is based on that in Ref. 7. It is also known as a fuzzy controller, but we find that name can cause confusion because this algorithm is for forecasting only, not for feedback and control. In our view, the fuzzy logic approaches to modeling and forecasting time series could benefit from a more statistical approach, especially in the area of choosing the number of fuzzy regions, which is comparable to our choice of bandwidth in Equation (5).

The fuzzy forecaster works as follows:

1) scale all predictors and the response to lie in (0,1);

2) for each predictor and for the response, select some number of fuzzy regions and a shape for each region (typically, use triangular fuzzy regions);

3) compute candidate fuzzy rule base with rules such as the following: RULE 1: if x_1 is large and x_2 is small, then y is medium;

4) assign a degree to each rule according to the degree of memberships by each predictor. For example, with RULE 1 above, x_1 was declared to be large because it had the maximum degree of membership in the fuzzy set called "large." Suppose the degree of membership of x_1 in "large" was 0.8, the degree of membership of x_2 in "small" was 0.9, and the degree of membership of y in "medium" was 0.7. Then the degree of RULE 1 is typically defined as $0.8 \times 0.9 \times 0.7 = 0.504504$; and

5) compute final fuzzy rule base by accepting only the rules having maximum degree of membership. For example, if one data pair leads to the candidate rule "if x is small then y is medium," and another data pair leads to the candidate rule "if x is small then y is small," then because these are conflicting rules, accept the rule having maximum degree.

From our discussions of loess and our curve smoother, we can see that it is wasteful to follow step (5) because many data points will never contribute to the rule base. It would be better to at least compute an average rule to resolve conflicting rules such as the two given in step (5). That average rule could be a member of the final fuzzy rule base, so that for a given rule describing the values of the predictors, there would be only one rule for the response. However, that one rule would

be an average of all the rules for relevant data pairs. Also, even for cases where there are no conflicting rules, so no averaging is needed, the rule base contains the center value of the response's fuzzy region rather than the actual value of the response. That is also a waste of information. Also, the statistically motivated curve smoother uses some type of trial-and-error to select a good bandwidth. Therefore, our implementation of the fuzzy forecaster tries a number of fuzzy regions ranging from 2 to 6. Typically, we see improved performance (lower average squared error) as the number of regions (for each predictor and for the response) increases, up to a point, and then worsening results as additional regions are added. In all cases, we report the best performance we could find. We also apply both triangular and Gaussian-shaped regions and report the best results.

In short, the fuzzy forecaster is failing to use all of the data and is incompletely using the data that it does use because it disregards the actual value of the response. Therefore, we implemented both the standard (wasteful) fuzzy forecaster and an improved version. Results presented are the best results (over the range of candidate number of regions and trying both triangular and Gaussian-shaped regions) for the improved version. In all cases considered, the improved version outperformed the standard version but was still inferior to the statistically motivated version.

The conclusion in Ref. 4 is that the fuzzy forecaster offers no advantage over either statistically motivated curve smoothers. In fact, our view is that fuzzy logic offers no advantage over statistically motivated approaches in **any** area in which the data is numeric. Fuzzy logic was developed for situations in which numeric data was not available. It is probably best to apply it only in those unfortunate situations.

In the next two sections, we present results of the fuzzy forecaster and for loess because loess was available in S+ and because it was simple to program the fuzzy forecaster in S+. It is also simple to program the nonparametric curve smoother in S+, but to date we only have a Fortran version.

3. Comparison of Fuzzy Forecaster to Loess for Five Simulated Data Sets

The protocol for the simulated data experiments was as follows.

Generate 400 observations of the response and predictors. Randomly select 200 to train and 200 to test. Repeat 10 times and report the average and standard deviation of the average squared forecast error. By repeating 10 times we can compute approximate confidence intervals for the true mean squared forecast error on each data set for each method.

Data set 1: $y = x(1-x) + e$, where $e - U(0, 1) \times 0.1$, where $U(0, 1)$ denotes a uniform distribution over (0,1).

Data set 2: $x - U(0, 1)$ and $y = \sin\left(2\pi (1-x)^2\right) + x \times e$, where $e - N(0, 1)$ and $N(0, 1)$ denotes the normal distribution with mean 0 and variance 1.

Data set 3: $x_1 - U(0, 1)$, $x_2 \sim U(0, 1)$, and $y = 2 \times x_1 - 3 \times x_2 + 0.1 \times e$, where $e - N(0, 1)$.

Data set 4: $x_1 - U(0, 1)$, $x_2 - U(0, 1)$, and $y = (2 \times x_1 - 3 \times x_2 + x_1 \times x_2) + 0.1 \times e$, where $e - N(0, 1)$.

Data set 5: $x_1 - U(0, 1)$, $x_2 - U(0, 1)$, $x_3 - U(0, 1)$, and $y = (2 \times x_1 - 3 \times x_2 + x_1 \times x_2 + x_3 + x_2 \times x_3 + x_1 \times x_3) + 0.1 \times e$, where $e - N(0, 1)$.

A rescaled version of data set 1 was shown in Fig. 1. Still working with data set 1, in Fig. 2 we show the loess fit and the fuzzy forecaster fit using triangular and Gaussian regions. Clearly, the loess fit is superior to both of the best fitting fuzzy forecaster methods.

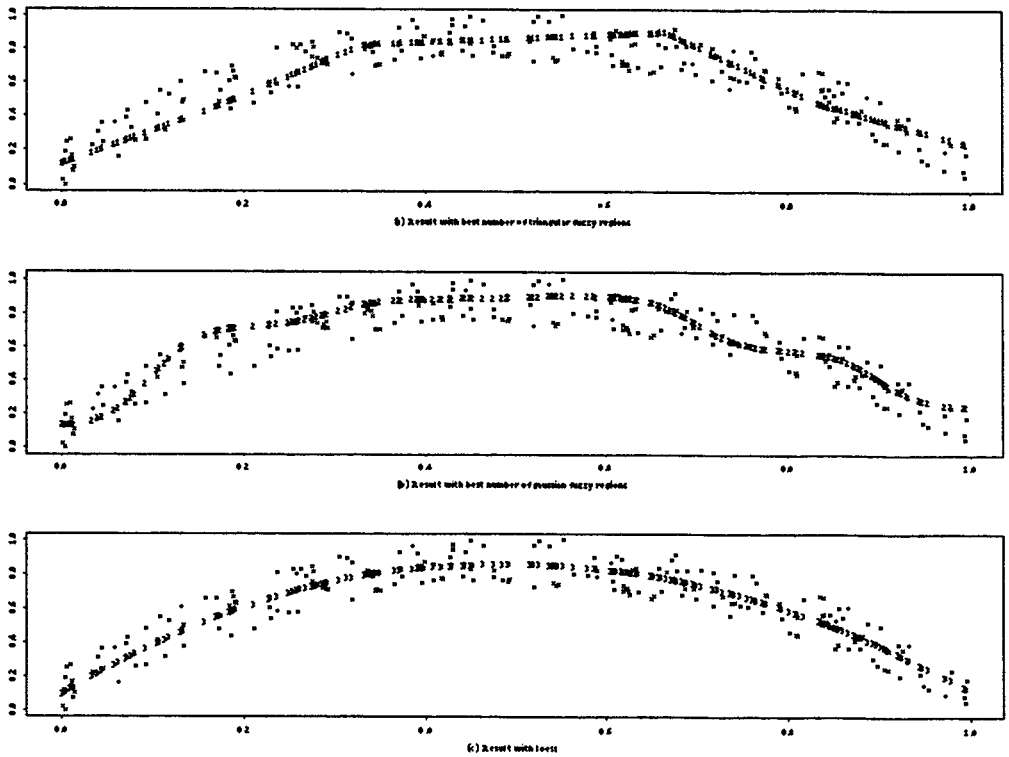

FIGURE 2. Visual comparison of result of best fitting fuzzy forecaster to result of loess.

Here are the approximate 95% confidence levels for the mean squared forecast errors over the testing set for the five data sets:

Fuzzy Forecaster	Loess	Theoretical Best
1. (.002, .004)	(.0013, .0014)	.00083
2. (.42, .69)	(.40,.49)	difficult to calculate
3. (.029, .035)	(.0096, .01)	.01
4. (.025,.033)	(.010, .011)	.01
5. (.057, .065)	(.011, .013)	.01

The loess method outperformed the best fuzzy-forecaster method in all five data sets, and achieved quite close to the theoretical best in the four cases for which we have calculated the theoretical best.

4. Comparison of Fuzzy Forecaster to Loess for Five Real Data Sets

The protocol for the real data sets was to use 50% to train and 50% to test, and then to repeat the process 10 times. As in section 3, repeating the experiment 10 times allows us to report approximate confidence intervals for the true mean squared error of the forecast error.

Data set 1: This is the same data set that was analyzed in section 5.1.2 of the 1994 KF report (Ref. 8). The response is the gamma count at time t. Predictors are gamma count at time t-1, gamma count at time t-2, electron count at time t, and proton count at time t. We also tried using higher lags for all predictor series, but the average squared forecast error was not reduced. (4 predictors, n=878)

Data set 2: Use NOx (nitric oxide and nitrogen dioxide combined) to predict the equivalence ratio at which the car engine was run (a measure of the richness of the air/ethanol mix) (1 predictor, n=88). Source: "Ethanol Fuel—A Single-Cylinder Engine Study of Efficiency and Exhaust Emissions," *SAE Transactions* **90**(810345), pp. 1410-1424.

Data set 3: Use radiation, temperature, and wind to predict atmospheric ozone (3 predictors, n=111). Source: New York environmental study.

Data set 4: Use previous day's bond yield of type A and present day's bond yield of type B to predict present day's bond yield of type A (2 predictors, n=192). Source: daily yields of ATT bonds from April 1975 to December 1975.

Data set 5: Use a car's weight and engine displacement to predict gallons/mile (2 predictors, n=60). Source: April 1990 issue of *Consumer Reports*.

Here are the approximate 95% confidence levels for the mean-squared forecast errors over the testing set for the five data sets:

	Fuzzy Forecaster	**Loess**
1.	(55798, 60652)	(40588,42810)
2.	(.24, .36)	(.23, .32)
3.	(.24, .29)	(.23, .27)
4.	$(711,4921)\times 10^{-7}$	$(1.6,2.0)\times 10^{-7}$
5.	(.13, .21)	(.14, .21)

So, loess is better on data sets 1-4, while the fuzzy forecaster is very slightly better on data set 5. Actually, there is little difference between the two on data sets 2, 3, and 4, but with data sets 1 and 4, loess does significantly better. Recall that we are presenting the best results we found for the fuzzy forecaster (an idea we have not found in any of the fuzzy-logic literature), searching for the best number of regions and whether to use triangular or Gaussian regions.

5. Summary of the Vector AR Models Considered

Recall that we are considering vector AR models in the context of time series so that forecasting time series can be viewed as an ordinary regression problem. Although it may be possible (see Ref. 4 for an ad hoc "two-step" procedure that is under investigation) to treat nonlinear MA models, we have not attempted that here. Once we restrict attention to AR models, we get access to a host of regression techniques that can be applied as if the data were in the usual regression setting: observe independent cases of data "pairs," $(\tilde{x},y)$ where the $\tilde{x}$ vector is a p-component predictor vector. The only difference in our setting is that successive cases are not independent because of the serial correlation. However, asymptotically (as the number of cases increases) this serial correlation can be ignored for estimating functions (Ref. 9).

6. Summary of the Performances of Fuzzy Forecaster and Loess

First, recall that we have improved the fuzzy forecaster in three ways:

1) The fuzzy-rule base uses all of the training data.

2) The fuzzy rule is calculated by averaging the actual value of the response rather than the center value of the fuzzy region to which the response has maximum membership. By only recording the center value, information is needlessly being thrown away with the fuzzy forecaster in Ref. 7.

3) We use trial and error to choose the number and type of fuzzy regions so we can explicitly optimize a particular criterion, which in this paper has been the average squared forecast error.

When we undertook this study, we knew that the statistically motivated methods, such as our curve smoother or loess, would outperform the fuzzy forecaster on simulated data. That is quite obvious if for no other reason than the fuzzy forecaster does not use all of the training data. Among other reasons, the fuzzy forecaster does not use all of the data because it was developed for models such as the one in Equation (4) but without the stochastic component. That is, it was developed for deterministic models. We find that ironic because fuzzy-logic proponents claim that fuzzy logic offers a new and complementary way to model uncertainty. See Ref. 10 for more of the ongoing debate on the topic.

In the case of the fuzzy forecaster, we conclude that fuzzy logic offers an inferior way to model uncertainty. Namely, it simply ignores uncertainty. Perhaps sometimes there is so much training data that it is acceptable to ignore part of it, but that is almost never our experience. Nevertheless, we did not know how the fuzzy forecaster would compare to, for example, loess, on real data. The message here is that loess or similar methods are superior to the fuzzy forecaster in most situations: with both real and simulated data. Fuzzy-logic-based methods should be reserved for situations with non-numeric data, and even in some of those cases, statistical methods are likely to be more mature and therefore, at least for now, superior. We have not addressed that issue in this paper. Also, we remind the reader that the simpler methods such as global linear or global polynomial should always be applied first. If the data is nearly linear or polynomial in form, then no other method will be as good as the linear or polynomial method.

References

1. T. Burr, J. Doak, J. Howell, D. Martinez, R.Strittmatter, "Knowledge Fusion: Time Series Modeling Followed by Pattern Recognition Applied to Unusual Sections of Background Data," Los Alamos National Laboratory report LA-13075-MS (March 1996).

2. C. Scovel, "Knowledge Fusion: Combining Information," Los Alamos National Laboratory document LA-UR-95-4408 (1995).

3. C. Chatfield, *The Analysis of Time Series, An Introduction*, Fourth Edition (Chapman and Hall, London, 1989).

4. T. Burr, "Prediction of Linear and Nonlinear Time Series With an Application in Nuclear Safeguards and Nonproliferation," Los Alamos National Laboratory report LA-12766-MS (April 1994).

5. Shirley Bleasdale, Tom Burr, Clint Scovel, and Richard Strittmatter, "Knowledge Fusion: An Approach to Time Series Model Selection Followed by Pattern Recognition," Los Alamos National Laboratory report LA-13045-MS (February 1996).

6. W. Cleveland, S. Devlin, and E. Grosse, "Regression by Local Fitting: Methods, Properties, and Computational Algorithms," *Journal of Econometrics* **37**, 87-114 (1988).

7. L. Wang and J. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Transactions on Systems, Man, and Cybernetics* **22**(6), 1414-1427 (Nov/Dec 1992).

8. S. Bleasdale, T. Burr, A. Coulter, J. Doak, B. Hoffbauer, D. Martinez, J. Prommel, C. Scovel, R. Strittmatter, T. Thomas, and A. Zardecki, "Knowledge Fusion: Analysis of Vector-Based Time Series with an Example from the SABRS Project," Los Alamos National Laboratory report LA-12931-MS (April 1995).

9. P. Robinson, "Nonparametric Estimators for Time Series," *Journal of Time Series Analysis* **4**(3), 185-207 (1983).

10. M. Lavioette, J. Seaman, J. Barrett, and W. Woodal, "A Probabilistic and Statistical View of Fuzzy Methods," *Technometrics* **37**(3), 249-292 (1995).

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545