Many models of the dynamics of nonlinear time series have large numbers of parameters and tend to overfit. This paper discusses algorithms for selecting the best basis functions from a dictionary for a model of a time series. Selecting the optimal subset of basis functions is typically an NP-hard problem which usually has to be solved by heuristic methods. In this paper, we propose a new heuristic that is a refinement of a previous one. We demonstrate with applications to artificial and real data. The results indicate that the method proposed in this paper is able to obtain better models in most cases.

Keywords: Model selection; nonlinear modeling; description length.
building larger models using a basis selection procedure [Judd & Mees, 1995]; described in the next section. Because the method starts with a small model and builds large models, we will refer to it as the "bottom-up" method.

The bottom-up method has proven to be effective [Judd & Mees, 1995, 1998; Small & Judd, 1998] in modeling nonlinear dynamics, so in this paper we will first compare this bottom-up approach to a "top-down" approach, which starts by taking a linear combination of all basis functions in the dictionary and selectively removes basis functions one at a time to obtain a good model. We also propose a hybrid "up-and-down" approach that iterates between the bottom-up and top-down approach using the best model of the previous iteration as the starting point of the next iteration.

We find that the up-and-down method combines the complementary strengths of the bottom-up and top-down approaches and obtains models that are smaller than, but just as accurate as, those obtained by using just the bottom-up or top-down methods.

2. Basis Selection and Model Improvement

The principal idea behind the basis selection procedure used by Judd and Mees is the following; for a detailed description see [Judd & Mees, 1995]. Given a model which is a selection of basis functions from the dictionary, one first tries to obtain a better model with the same number of basis functions, by first enlarging the model by one basis function. The basis function added is that in the dictionary that best fits the error vector of the current model. The enlarged model is then returned to the size of the original model by removing the basis function that makes the least contribution to the enlarged model. The "bringing-in" and "throwing-out" procedure is repeated until the model does not change in this process. This model is taken to be the best model of its size. The bottom-up method then attempts to find the next biggest best model by taking the last enlarged model as the starting point and applies the above bringing-in and throwing-out procedure to this model.

The top-down method uses the same bringing-in and throwing-out procedure as the bottom-up method. The difference is that when the best model of a given size is formed the top-down method proceeds to throw out another basis function to obtain smaller sized model, whereas the bottom-up method brings in an extra basis function to obtain a large sized model.

There are alternatives to the basis selection procedure used by Judd and Mees. For example, for a small dictionary one could perform an exhaustive search of the entire dictionary to obtain the truly best model. In a subsequent paper we will make a comparative study of basis selection procedures.

3. The Minimum Description Length Principle

The length and numerical accuracy of a time series are limited even for simulations. Worse still, real time series are contaminated by noise to some extent. So the model should not be fitted to the data too exactly, this is overfitting. If the model is too poorly fitted to the data, the model is underfitting. The aim is to model complicated phenomena as well as take into account finite length, numerical accuracy and noise of the data.

From the viewpoint of description length, one considers the number of bits of information required to reconstruct the original time series. A convenient, though suboptimal, way to reconstruct the data would be to use the model and its parameters, including initial data, and to then correct the model's output by adding in the prediction errors. This gives a so-called two-part code: the data is described by the model and its error vector. An important point is that the original data has finite accuracy and the parameters and errors need only be specified to finite accuracy. If the model is good the total number of bits required to transmit parameters, initialization data and errors will be less than the number of bits required to send the raw data and the two-part code will have compressed the data.

The minimum description length (MDL) principle states that the best model is the one that attains the greatest compression. Roughly speaking, the two-part code's description length is given by an expression of the form

\[
\text{Description Length} \approx (\text{Number of data}) \times \log (\text{Mean square prediction error}) + (\text{Penalty for number and accuracy of parameters}).
\]
The prediction error must decrease when the number of parameters in a model increases, however, the penalty for introducing another parameter eventually outweighs the benefit to the description of the data (see Fig 1). The model whose description length is the smallest in the class of models is considered the optimal model. Note that this may be the null model, where the raw data is sent; in such a case, the data is indistinguishable from a stream of random numbers to within the powers of the modeling method being used.

Usually it is not easy to calculate the exact description length of a model, even with the two-part code simplification, but explicit approximations of the description length can be calculated in special model classes, such as pseudo-linear models [Judd & Mees, 1995]. These have the general form

\[ x(t+1) = \sum_{i=1}^{m} \lambda_i f_i(v(t)) + \varepsilon(t), \]

where \( f_i \) are some selection of nonlinear functions (that is, these are basis functions), \( \lambda_i \) are unknown parameters, \( d \) is the embedding dimension, and \( \varepsilon(t) \) are independent identically distributed errors. Define

\[ V_i = (f_i(v(t)), f_i(v(t+1)), f_i(v(t+2)), \ldots, f_i(v(t+n)))^T, \quad i = 1, 2, \ldots, m \]

where \( T \) indicates the transpose, \( n \) is the number of data, and \( m \) is the size of the dictionary of basis functions. Let \( V \) be the matrix whose columns are \( V_i, i = 1, 2, \ldots, m \). When \( \lambda \) is chosen to minimize the sum of squares of the prediction errors \( \mathbf{e} = \mathbf{y} - V\lambda \), then the description length \( S_k \) is approximated by [Judd & Mees, 1995]

\[ S_k = \left( \frac{n}{2} - 1 \right) \ln \frac{e^T e}{n} + (k + 1) \left( \frac{1}{2} + \ln \gamma \right) \]

\[ - \sum_{i=1}^{k} \ln \delta_i \]  

(2)

where \( k \) is the number of nonzero components of \( \lambda \), \( \gamma \) is related to the scale of the data (see below), and \( \delta \) solves \([Q\delta]_j = 1/\delta_j\) where \( Q = n\hat{V}^T \hat{V}/e^T e \) and \( \hat{V} \) is composed of just those columns of \( V \) that correspond to nonzero parameter \( \lambda \). The variables \( \delta \) can be interpreted as the relative precision to which the parameters \( \lambda \) are specified.

The attraction of pseudo-linear models is that the parameters \( \lambda \) are easily calculated, because the sum of squares of the prediction errors \( \mathbf{e} = \mathbf{y} - V\lambda \) can be minimized efficiently using singular value decomposition or any of its many equivalents.

A more precise interpretation of \( \gamma \) is that it is the exponent in a floating point representation of the parameters scaled relative to some fixed amount. Typically \( 1 \leq \gamma \leq 32 \). A value of \( \gamma = 1 \) gives larger models and a value of \( \gamma = 32 \) gives smaller models. When the basis functions involve hidden parameters (such as, radial basis functions, where the position and scale of basis functions constitute additional parameters), then \( \gamma = 32 \) is better because it implicitly penalizes for the hidden parameters. However, when there are hidden parameters it would be better to explicitly penalize for the extra parameters, but this requires more complex calculation than we describe above. In this paper to avoid the additional technical details we use \( \gamma = 32 \) throughout. The purpose of this paper is to compare selection algorithms, and the conclusions are the same when we use \( \gamma = 1 \). (The value of \( \gamma = 32 \) is, however, probably not the correct thing to do, and we will address the technical details in a subsequent paper.)

4. Selection Algorithms

We now review the bottom-up method of Judd and Mees [1995], and consider a situation where it does
not produce the correct model. We then consider alternative selection algorithms, namely, the top-down and up-and-down methods.

Usually, it is impossible to know beforehand which of the basis functions in the dictionary are the best for the model, so several techniques for selecting basis functions have been proposed, for example, [Chen et al., 1991]. The algorithm for selecting basis functions proposed by Judd and Mees [1995] has been shown to obtain good results for short and long term predictions for many nonlinear systems [Judd & Mees, 1995, 1998; Small & Judd, 1998]. As mentioned before, we will refer to this method as the bottom-up method, because the method increases the model size from one parameter.

Although the algorithm for selecting basis functions often works well, there are situations where it does not. Details of such situations are described elsewhere [Judd & Mees, 1995]. One concrete example of the deficiencies is the following Example 1.

Example 1. Consider the Henon map [Henon, 1976]

\[
\begin{align*}
\{ x(t) &= 1.0 - 1.4x(t - 1)^2 + y(t - 1) \\
y(t) &= 0.3x(t - 1).
\end{align*}
\]

This model can be expressed as \( x(t) = 1.0 + 0.3x(t - 2) - 1.4x(t - 1)^2 \) and we consider using multivariate polynomial models. The data used here is 1000 points with Gaussian observational noise with standard deviation 0.01. We will use time delay polynomial models. Choosing \( \text{lag} = 2 \) and degree = 3 gives 10 candidate basis functions in the dictionary, that is, the constant function, \( x(t - 1), x(t - 2), x(t - 1)^2, x(t - 2)^2, x(t - 1)x(t - 2), x(t - 1)^3, x(t - 2)^3, x(t - 1)^2x(t - 2) \) and \( x(t - 1)x(t - 2)^2 \). Applying the bottom-up method, we obtain the minimum description length for increasing model size as shown in Fig 2. The overall minimum description length is obtained when the number of parameters is 5. The basis functions in this MDL model are constant, \( x(t - 2), x(t - 1)^2, x(t - 1)^3 \) and \( x(t - 1)^2x(t - 2) \). Three of these are the correct choices but other two are not. This is one example where the bottom-up method does not work well.

4.1. The top-down method

An alternative to the bottom-up method is the top-down method in which instead of being increased, the model size is decreased from the model that uses all candidate basis functions.

In the following algorithm, \( B \) is the set of indices of the current basis set for model. We write \( V_B \) for the \( n \times k \) matrix formed from the columns of \( V \) with indices in \( B \), where \( n \) is the length of vector and \( k \) is the number of parameters. Let \( \lambda_B \) be the least squares solution to \( y = V_B\lambda_B \), and let \( e_B = y - V_B\lambda_B \).

\[ \text{It is a separate problem to determine what are appropriate basis functions to put in the dictionary. This problem is not addressed in this paper; for an example of how it might be solved see [Judd & Mees, 1998].} \]
(1) Normalize the columns of $V$ to have unit length, and initialize $H$ to the empty set. Set $B = \{1, \ldots, m\}$ where $m$ is the size of the complete dictionary. Calculate $\lambda_B$ and calculate the description length $S_k$, where $k = m = |B|$. Let $i$ be the index in $B$ corresponding to the component of $\lambda_B$ with the smallest absolute value. Let $B = B \setminus \{i\}$ and $H = H \cup \{i\}$. (Throw out the basis function that contributes the least to the model.\footnote{Note that step 2 repeats the throwing out. To get the algorithm started, we have to perform this throwing out step twice.})

(2) Calculate $\lambda_B$. Let $i$ be the index in $B$ corresponding to the component of $\lambda_B$ with the smallest absolute value. Let $B' = B \setminus \{i\}$ and $H' = H \cup \{i\}$, then calculate $\lambda_{B'}$. (Throw out the basis function that contributes the least to the model.)

(3) Calculate $e_{B'} = y - V_{B'} \lambda_{B'}$ and $\mu_{H'} = V_{H'}^T e_{B'}$, and let $j$ be the index of the component of $\mu$ with maximum absolute value. (The component of the vector $\mu$ measures how closely each of the basis functions not currently in use will match the error of the current model. Extend the current model with basis function that best matches the current error.)

(4) If $i \neq j$, then $B = B' \cup \{j\}$ and $H = H' \setminus \{j\}$, then go to step 2. Otherwise, that is, if $i = j$, let $H = H'$ and $B$ is left as it is (that is, without modifying), then go to step 5. (Step 2 threw out a basis function, and step 3 brought one in.

If $i \neq j$, then the basis was changed, so go back and see if these can be further changes or if necessary. If $i = j$, then the basis was not changed, so the best basis at the current size $k = |B|$ has been found.)

(5) The current values of $\lambda_B$ (not $\lambda_{B'}$) are the best parameter values for the model of size $k = |B|$ and let $B_k = B$. Calculate the description length $S_k$ as in Eq. (2) with these $B_k$ and $\lambda_B$.

(6) Let be $B = B'$. If $k > 1$, then go to step 2.

(7) When $k = 1$, then let $B = \{j\}$ where $V_j$ is the column of $V$ such that $|V_j^T y|$ is maximum. (This selects as the one basis function that most closely matches the data $y$.)

(8) Let $S_0 = ((n/2) - 1) \ln(y^T y/n) + 1/2 + \ln \gamma$. (The description length of the raw data.)

(9) Finally, take the basis $B_k$ such that $S_k$ is the minimum as the optimal model.

**Example 2.** Consider the Henon system \cite{Henon} again in Example 1. Figure 3 shows the minimum description length for each model size using the top-down method. The description length is the smallest when the number of parameters selected is 3. Only the correct basis functions (that is, constant, $x(t-2)$ and $x(t-1)^2$) were selected. Also, the smallest description length using the top-down method is smaller than that using the bottom-up method. As Example 2 shows, the top-down method is more

---

Fig. 3. A plot of description length against size of model using the top-down method proposed in this paper. (b) is an enlargement of (a).
effective for the Henon data at selecting the correct basis functions. However, it should be noted that there is a weak point in the top-down method. If there is strong collinearity among the basis vectors $V_i$, that is the basis function $f_i$ evaluated at each embedded point $y$ [see Eq. (1)], then the top-down method may not be effective at selecting the correct basis functions. Example 3 demonstrates this.

**Example 3.** Consider the following nonlinear MA system

$$y(t) = 2.0u(t - 1) + 3.0u(t - 1)^2 + 4.0u(t - 1)^2u(t - 2) + \varepsilon(t),$$

where $\varepsilon(t)$ is Gaussian with standard deviation 0.01.

The input $u(t)$ is the $x$ component of Lorenz equation [Lorenz, 1963]

$$\begin{align*}
\frac{dx}{d\tau} &= -\sigma(x - y), \\
\frac{dy}{d\tau} &= -y - xz + \gamma x, \\
\frac{dz}{d\tau} &= xy - Bz,
\end{align*}$$

where $\sigma = 10.0$, $\gamma = 24.73$, $B = 8/3$, when calculated by the fourth order Runge–Kutta method with sampling interval 0.005, with the addition of the observational noise that is Gaussian with standard deviation 0.01. Again, using time delay polynomial models, choosing lag = 2 and degree = 3.

Fig. 4. A plot of description length against size of model. (a) and (b) are the results of using the bottom-up method, where (b) is an enlargement of (a). (c) and (d) are the results of using the top-down method, where (d) is an enlargement of (c).
determines 10 candidate basis functions in the dictionary. We use 1000 data points to build the model. Because the sampling interval of Runge–Kutta is very small, there is strong collinearity among the basis vectors in the dictionary.

Figure 4 shows that the description length of the model using the bottom-up method is the smallest when the number of parameters selected is 3. Only the correct basis functions (i.e. \(u(t - 1), u(t - 1)^2\) and \(u(t - 1)^2u(t - 2)\)) are selected; however, the description length of the model obtained using the top-down method is the smallest when the number of parameters selected is 6. The basis functions selected are \(u(t - 1), u(t - 2), u(t - 1)u(t - 2), u(t - 1)^3, u(t - 1)^2u(t - 2)\) and \(u(t - 1)u(t - 2)^2\). Note that this model does not involve the basis functions \(u(t - 1)^2\), as it should do. The reason for this appears to be that step 1 in the top-down algorithm (when all basis functions are selected) gives the model

\[
y(t) = 0.140 - 6.497u(t - 1) + 8.419u(t - 2) \\
+ 16.698u(t - 1)^2 + 13.314u(t - 2)^2 \\
- 27.012u(t - 1)u(t - 2) + 48.269u(t - 1)^3 \\
- 44.620u(t - 2)^3 - 137.0678u(t - 1)^2u(t - 2) \\
+ 137.418u(t - 1)u(t - 2)^2.
\]

The last pair of basis functions have large almost equal coefficients of opposite sign, and there are also groups of three basis functions (e.g. \(u(t - 1)^2, u(t - 2)^2\) and \(u(t - 1)u(t - 2)\)) with moderate sized coefficients that almost sum to zero. There is clearly some kind of cancellation going on. It is not clear why this phenomenon occurs, it seems to be connected with the collinearity of the basis vectors, but it might also be that the algorithm is inventing new “basis functions” that are the small difference between cancellations; whatever the reason, this is not desirable.

So, to avoid this problem, the top-down method should not be used when there is significant collinearity among the basis vectors in the dictionary. As we discussed earlier, the bottom-up method is effective even when there is strong collinearity among candidate basis vectors.

4.2. **The up-and-down method**

We have seen from Example 2 that sometimes the top-down method performs better than the bottom-up method, and from Example 3 that sometimes the bottom-up method performs better than the top-down method. In the next example we will see that neither method works well.

**Example 4.** Consider the following nonlinear MA system

\[
y(t) = 3.0u(t - 1) + 5.0u(t - 3)^2 \\
+ 4.0u(t - 3)^3 + \varepsilon(t),
\]

where input signal \(u(t)\) is uniform random on \([0, 1]\) as the input signal and \(\varepsilon(t)\) is Gaussian with standard deviation 0.01. Again, using time delay polynomial models, choosing \(lag = 3\) and degree = 3 determines 20 candidate basis functions in the dictionary. We use 1000 data points to build the model.

Figure 5 shows that the description length of the model using the bottom-up method is smallest when the number of parameters selected is 4. The correct basis functions \(u(t - 1), u(t - 3)^2\) and \(u(t - 3)^3\) are included in this model. When the number of the parameters selected is four the description length of the model using the top-down method is the smallest. The basis functions in this model are \(u(t - 1), u(t - 3)^2, u(t - 3)^3\) and \(u(t - 1)u(t - 2)u(t - 3)\). Three of these are the correct choices but the other is not. So, neither of the methods work well in this example.

Here, we want to pay attention to the basis functions selected using the top-down method when the number of parameters changes from 4 to 3. When the number of parameters is four, the basis functions selected are \(u(t - 1), u(t - 3)^2, u(t - 3)^3\) and \(u(t - 1)u(t - 2)u(t - 3)\). Three of these are the correct choices but the other is not. However, when the number of parameters is three, which is the size of the original system, the basis functions selected are constant, \(u(t - 1)\) and \(u(t - 3)^2\). Two of these are the correct choices but the other is not, the correct basis function \(u(t - 3)^3\) is not selected at this stage. When the dictionary is just those basis functions that were selected when the model size was 4, then either selection models gives the correct model. Figure 6 shows the results. The description length in both methods is the smallest when the number of parameters selected is three. This suggests that it is sometimes advantageous to restrict the dictionary to only the previously selected basis functions rather than using the complete dictionary.

We now investigate a method that attempts to combine the strengths of the two methods and minimize their weaknesses. We combine the methods by
iterating between the bottom-up method and top-down method, using the best model of the previous iteration as the starting point of the next iteration. This leads us to the following up-and-down method (see the schematic diagram Fig. 7):

1. A model is built from the complete dictionary of basis functions using the bottom-up method. The model whose description length is the smallest is selected. Call this the current model.
2. Three models are built as follows:
   a. A model is built from the basis functions of the current model using the top-down method.
   b. A model is built from the basis functions of the current model using the bottom-up method.
   c. A model is built from the complete dictionary using the top-down method, but starting from the current model instead of the null model.

The model whose description length is the smallest of these three models is then selected; call this the refined model.

3. A model is built from the complete dictionary using the bottom-up method again, where the starting point is the refined model. If its description length is less than that of the refined...
Fig. 6. A plot of description length against size of model. (a) and (b) are the results using the bottom-up method, where (b) is an enlargement of (a). (c) and (d) are the results using the top-down method, where (d) is an enlargement of (c).

model, it replaces the refined model; otherwise, it is discarded.

(4) If the refined model is identical to the current model, it is taken as the final model and the algorithm terminates. Otherwise, the refined model becomes the new current model and we return to step 2.

It should be noted that the up-and-down method can do no worse than the bottom-up method.

5. Applications

Applying the up-and-down method in each of the previous four examples always obtains the correct model. So, in this section we now consider applying the up-and-down method to the real data set, the annual sunspot numbers, to find global linear models and global nonlinear radial basis models. Previously, Judd and Mees applied the bottom-up method to the annual sunspot numbers [Judd & Mees, 1995]. The following examples will compare the bottom-up method and the up-and-down method with this data.

Tong has described models of the sunspot time series that first transform the annual sunspot numbers \( s_t \) to a new time series \( y_t = 2\sqrt{s_t + 1} - 1 \) [Tong, 1990]. The purpose of this transform is to make the time series more like that of a linear system. When we apply the methods described in this section to this transformed data \( y_t \), then we obtain a minimum description length model that is linear, that is, has only linear basis functions (to be precise only the
In general, such a transform will not exist. Since our aim is to illustrate the general situation, we will build our model using the raw annual sunspot numbers.

Example 5. In this example we consider a dictionary containing only linear basis functions, which will give rise to a Reduced Auto-regressive (RAR) model [Judd & Mees, 1995]. The model will be built with lags up to 15, on the period 1700–1979 as the training set and the period 1980–2000 as the test set. See Fig. 8. The RAR model obtained using the bottom-up method is

\[ y_t = 8.0603 + 1.2059y_{t-1} - 0.5193y_{t-2} - 0.0304y_{t-5} + 0.1906y_{t-9}. \]

The RAR model obtained using the up-and-down method is

\[ y_t = 1.2104y_{t-1} - 0.4710y_{t-2} + 0.2475y_{t-9}. \]

To compare the performance of both models, short term noiseless simulations for 20 years of both models are investigated. Figure 9 shows the actual annual sunspot numbers and noiseless simulation of both models. The features of both models are similar to the actual annual sunspot numbers. However, the models do not follow the behavior of the actual annual sunspot numbers exactly because both models are linear. The details are described in [Judd & Mees, 1995].

Example 6. In this example the dictionary contains linear and radial basis functions. The period 1700–1979 of the annual sunspot numbers is used to build the models and the data is embedded in three dimensions with a lag of 3. The dictionary includes 10 linear basis functions and 100 radial basis functions. The radial basis functions are Gaussian basis functions \( f_i(x) = \exp(-(1/2)||x - c_i||^2/r_i^2) \), where the \( r_i \) are scalars uniformly distributed between \( \sigma/2 \) and \( 2\sigma \), where \( \sigma \) is the standard deviation of the...
time series, and the centers $c_i$ are *chaperons* generated by adding Gaussian random perturbations with standard deviation $\sigma/3$ to randomly selected data points [Judd & Mees, 1995]. Because the basis functions are generated randomly the best model will change when algorithm is repeated with a new dictionary. To evaluate the variability of the models we make 15 models from 15 different randomly generated dictionaries. We use both the bottom-up and the up-and-down methods with each of the dictionaries. Table 1 shows that the models using the up-and-down method have smaller description length and fewer number of parameters than those using the bottom-up method in 14 cases of 15 cases; model 11 was the same for both methods.

Figure 10 shows the actual annual sunspot numbers and short term simulations for 20 years of all models without observational or dynamical noise. All models follow the oscillation of the actual sunspot numbers well. Figure 11 shows the actual annual sunspot numbers and the mean simulation time series with the standard deviation. As there is noise in the value measured, it is very useful to decrease the influence of the noise by using the ensemble mean technique for the prediction. The behavior of both mean simulated time series is almost the...
same as the actual annual sunspot numbers. The behavior using the up-and-down method is closer to the actual annual sunspot numbers than that using the bottom-up method after 1996. Another advantage in using this technique is that we obtain an indication of the possible range of the predicted values. This is very important as it can provide indication of the reliability of the value predicted.

6. Summary and Conclusion

This paper has described algorithms for selecting the best basis functions from a dictionary for a model of a time series. The up-and-down method is a combination of the top-down method and the bottom-up method, which appears to compensate for the weak points of the bottom-up method and the top-down method, and make the best use of the advantages of both. The up-and-down method can do no worse than the bottom-up method.

The three methods were applied to artificial data sets such as nonlinear AR system, nonlinear MA system and a real data, the annual sunspot numbers. The up-and-down method obtained the correct model when applied to the identification of the artificial systems, when the bottom-up and top-down methods did not. The description length obtained when applying the up-and-down method to the annual sunspot numbers is smaller than that of applying the bottom-up method in most cases.

These results suggest that the up-and-down procedure is a more robust method.

References


