Predicting El Niño Characteristics using Recursive Models, Data Trees, and an Evolutionary Algorithm

Nathaniel Kim

Palos Verdes Peninsula High School

February 17, 2016

**Abstract**

The objective of this project was to predict future values of El Niño characteristics using recursive equation models, data trees, and an evolutionary algorithm. In addition, the project attempted to find how number of generations and size of the tree influenced accuracy of the predictions.

In order to generate a final model and test its accuracy, a Python program was written and run. First, the program generated a number of models using a data tree. The program then tested all the potential models against training data. Poor models were removed, new models were generated as replacements, and random changes were carried out. This process of testing and changing the set of models, called a generation, was repeated various times. Finally, the best model throughout the entire process was outputted, and its accuracy was measured by comparing its predictions to testing data.

3 different characteristic values of El Niño were tested. These values measured air pressure, sea surface temperature, and precipitation levels each in different areas. For each value, the program was used to generate and test models after 10, 50, and 100 generations. Models created with different tree sizes were also generated.

Ultimately, this program had success in predicting future values of 2 El Niño characteristics, but also did not succeed at predicting the temperature characteristic. In addition, number of generations had little effect on accuracy, while increased tree size led to increased accuracy. The successes of the model did satisfy the objective and also indicate potential for improvements in future testing. Furthermore, the program’s generality indicates possible success both in predicting other El Niño characteristics and in predicting any other time series as well.

**Statement of Problem**

The objective of this study was to predict future values of El Niño indices by developing a data-tree-based model structure and an evolutionary algorithm to improve these models, and then using historical data to actually produce these models. The hypothesis is that these future values could be predicted to some reasonable degree of accuracy. A secondary objective was to determine whether the accuracy of the models increased with more generations simulated.

The objective of this project was to predict future values of El Niño characteristics using recursive equation models, data trees, and an evolutionary algorithm. In addition, the project attempted to find how number of generations and size of the tree influenced accuracy of the predictions.

Can future values of El Niño characteristics be predicted using recursive equation models, data trees, and an evolutionary algorithm? In addition, how do number of generations and size of the tree influence accuracy of the predictions?

**Introduction**

El Niño is a period in which ocean temperatures in the Equatorial Pacific are unusually warm and is also characterized by unusually high atmospheric sea level pressures in the western tropical Pacific and Indian Ocean regions, and unusually low sea level pressures in the southeastern tropical Pacific. El Niño causes increased rainfall across the southern tier of the US and in Peru, which has caused destructive flooding, and drought in the West Pacific, sometimes associated with devastating brush fires in Australia. Unfortunately, damages, such as that shown in Figure 1, linked to El Niño can cost billions of dollars, making methods of predicting El Niño so important.

One of the most important facets of El Niño predictions is the use of an index, a number scale in which all the individual factors needed to describe a complicated phenomenon are boiled down to a single number. There are many different El Niño indices used in practice today, including the Southern Oscillation Index (SOI) and sea surface temperatures (SSTs). The SOI is an index based on the sea level pressure differences between Tahiti and Darwin, Australia, and negative values of the SOI often indicate El Niño episodes. SSTs of regions in the Equatorial Pacific Ocean are also commonly used as El Niño indices. One such region which is extremely important is the Niño 3.4, given by the region shown in Figure 2. In addition, since heavy rainfall in some regions is also a characteristic of El Niño, rainfall levels are closely associated with El Niño (though do not necessarily predict it.)

There are many different methods which can be used to develop El Niño predictions, one of which is the use of an evolutionary algorithm. Such evolutionary algorithms usually follow this basic structure:

1) Generate many random models

2) Test each model for effectiveness

3) Create new models using characteristics of the better models of step 2

4) Apply mutations by randomly changing characteristics of present models or by adding random new models

5) Delete models with low effectiveness

6) Repeat steps 2 through 5 (known as a generation) until a desired effectiveness is reached

Data trees are a fundamental part of the model used in this project. Trees consist of many objects called nodes. One of these nodes is designated as a root. All other nodes are connected to by a directed edge from one, and only one, other node. This directed edge connects from the parent to the child. A node which has no children is called a leaf. Finally, the depth of a node is defined by the number of directed edges between the root and this node. A simple example of a data tree is shown in Figure 3.

A previous study has been done demonstrating the effectiveness of an El Niño prediction model which made use of a data tree and an evolutionary algorithm. However, this work is quite limited. First, the model only looked at the Japan Meteorological Agency (JMA) Index, the SST of a region in the tropical Pacific. Other El Niño prediction models use many other factors; thus, incorporation of additional factors into the model is important. In fact, by using different indices to create various models, insights regarding the relative importance of each factor in predicting El Niño may arise. Furthermore, the study only produced a model but did not test it for accuracy. Testing accuracy could be done by using previous data which excludes the last several years to develop the models, and then by comparing the models’ predictions during these several years to the actual data.

**CUT THIS DOWN**

**Hypothesis**

The objective of this study was to predict future values of El Niño indices by developing a data-tree-based model structure and an evolutionary algorithm to improve these models, and then using historical data to actually produce these models. The hypothesis is that these future values could be predicted to some reasonable degree of accuracy. A secondary objective was to determine whether the accuracy of the models increased with more generations simulated. The hypothesis is that as more generations were simulated, the accuracy would increase. Finally, a third objective was to determine the effects of number of total models per generation on accuracy. It was hypothesized that as this number increased, the accuracy of the models would also increase.

**Materials and Methods**

The materials needed were a computer used to run Python (a Windows 7 laptop was used), Python 3.4.3 to run a modeling program, and a Python development environment to actually develop the model (IDLE, Python’s default development environment, was used.) In addition, to actually create and test models, various sources of index data were used. These were monthly SOI values in PSI from January 1991 to January 2016, monthly Niño 3.4 values in °C from January 1950 to January 2016, and annual inches of rainfall in Los Angeles from 1877 through 2016.

A Python program, written in Python 3.4.3 for Windows 7, to generate models was developed and run. What follows are details about the program.

First of all, the program used a data tree to represent the set of models. Each node in the tree consisted of an operator and a value. The operator was a randomly chosen fundamental operator of arithmetic (namely +, -, \*, and /.) Values were either random floating-point numbers from -2 to 2 inclusive, or previous values of the relevant index ranging from f(t-1) to f(t-8) where f(t) is the value of the index at time t. The only exception for this was the root, which had no operation and a value of 1.

Each path directly from the root to a leaf represented a model. For example, if the nodes in such a path were “1”, “+f(t-1)”, "\*2", and "+f(t-3)", in that order, then the model was given by f(t) = 1 + f(t-1) \* 2 + f (t-3). This is shown in Figure 4.

In order to create a model, the program followed the following steps:

1. The program first took in input data of a given index. To develop the models, all values not from the past 5 years were used as training data, as the values from the past 5 years were instead used to later test the data.

2. The program then generated a tree containing between 90 and 110 random leaves, all of which had depth less than 10. Thus there were between 90 and 110 models.

3. For each mode, the program predicted values of the given index and tested these predicted values against the real values. It then calculated each model’s accuracy by taking the arithmetic mean of all percentage errors. The lower that this accuracy value was, the more accurate the model was. The most accurate model and its accuracy were recorded.

4. The program deleted the branches of the worst half of models (i.e., those which have the lowest accuracy).

5. To replace these models, the program duplicated about 35 models and created about 15 new models by attaching leaves to pre-existing nodes.

6. Next, the program performed approximately 5 crossovers. A crossover consisted of swapping an entire subtree with another subtree, while still keeping these subtrees and the rest of the tree intact.

7. The program performed approximately 5 mutations. A mutation consisted of replacing random nodes in the tree having sufficiently high depth with new, randomly generated nodes.

8. Steps 3 through 7 (a generation) were repeated a large number of times to carry out more generations.

9. Finally, the model with the lowest average percent error out of all models from all generations and this percent error were recorded. Furthermore, this model was compared against the actual data from the last 5 years, and its average percent error (used as accuracy) at predicting this data was recorded.

To now generate actual models and test accuracies and hypotheses, the aforementioned SOI, Niño 3.4, and Los Angeles precipitation data were used. For each value, 10 models were created using a tree of approximately 100 leaves and 10, 50, and 100 generations each. (These different number of generations were used to test the effect of number of generations on accuracy.) For each of these sets of 10 models, the average percent error in training and the accuracy at predicting “future” values were recorded. Furthermore, to test the effects of tree size on accuracy, 10 models was created using a tree of approximately 200 leaves and 50 generations. Again, the average percent error in training and the average percent error in predicting were recorded. (In all, 120 models were created.) In addition, for each of the 3 sets of data, the most accurate set of 10 models (out of the 4 sets) was used to predict 5 future values of the data.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Generations | Leaves | Average Percent Error | Accuracy |
| 10 | 100 | 33.73 | 58.45 |
| 50 | 100 | 33.19 | 59.20 |
| 100 | 100 | 32.40 | 55.60 |
| 50 | 200 | 32.60 | 66.31 |

Chart 1: Accuracy of the SOI Models

|  |  |  |  |
| --- | --- | --- | --- |
| Generations | Leaves | Average Percent Error | Accuracy |
| 10 | 100 | 10.51 | 3.78 |
| 50 | 100 | 10.51 | 3.74 |
| 100 | 100 | 10.51 | 3.74 |
| 50 | 200 | 14.34 | 3.18 |

Chart 2: Accuracy of the Niño 3.4 Models

|  |  |  |  |
| --- | --- | --- | --- |
| Generations | Leaves | Average Percent Error | Accuracy |
| 10 | 100 | 42.77 | 23.22 |
| 50 | 100 | 38.53 | 37.56 |
| 100 | 100 | 35.66 | 26.65 |
| 50 | 200 | 35.70 | 33.98 |

Chart 3: Accuracy of the L.A. Precipitation Models

The three charts above show the average percent error and the accuracy of each of the 10 sets of models. Also, for each value, the most accurate set of models (*i.e.*, the set with the lowest accuracy value) is shaded with yellow. As stated before, lower values of the right-hand column indicate more accuracy.

Chart 4 shows the accuracy of the models with approximately 100 leaves and illustrates the effect of generations on accuracy. Also included are average accuracies of the 3 models with 10 generations, of the 3 models with 50 generations, and of the 3 models with 100 generations.

Chart 5 shows the accuracies of each set of the 50-generation models. Again, averages are also included.

Charts 6, 7, and 8 show the real values and model-predicted values of each of the characteristics for the last 5 time intervals (months for SOI and Niño 3.4; years for L.A. precipitation.) In addition, for each characteristic, the values for each of the next 5 time intervals are shown as predictions.

**Discussion and Conclusions**

The first three charts show the average percent error and the accuracy of each of the 10 sets of models. Also, for each value, the most accurate set of models (*i.e.*, the set with the lowest accuracy value) is shaded with yellow. As stated before, lower values of the right-hand column indicate more accuracy. From Chart 1, the SOI was predicted with a 55.60% error, which is very high. However, from Chart 2, Niño 3.4 was predicted with an astoundingly low 3.18% error. Finally, from Chart 3, Los Angeles annual precipitation levels were predicted with 23.22% accuracy, a fairly low percentage error.

Charts 4 and 5 represent the information presented in Charts 1-3, but in graph form instead. Chart 4 illustrates the effect of generations on accuracy by showing the accuracy of the models with approximately 100 leaves but different numbers of generations. Also included are average accuracies of models with a given number of generations.

Chart 5 shows the effect of number of leaves on accuracy by displaying accuracies of each set of the 50-generation models. Again, averages are also included.

Charts 6, 7, and 8 show the real values and model-predicted values of each of the characteristics for the last 5 time intervals (months for SOI and Niño 3.4; years for L.A. precipitation.) In addition, for each characteristic, the values for each of the next 5 time intervals are shown as predictions.

CONCLUSIONS

Chart 1 shows that the model did not predict the SOI well, while Chart 2 shows that the model did predict Niño 3.4 very well. One explanation for this accuracy is that values for Niño 3.4 do not change by much from month to month, and this was especially true for the 5 values used as testing data. These smaller changes indicate more stability of the Niño 3.4 values, thus making them easier to predict. Finally, Chart 3 shows that the model had some success predicting the Los Angeles precipitation data. In all, the model had mixed results in predicting various indices and characteristics of El Niño, but its successes do demonstrate potential and support the first hypothesis.

As stated before, Chart 4 illustrates the relationship between number of generations and accuracy. The average results indicate that the 10-generation models were most accurate, followed closely by the 100-generation models. The 50-generation models were least accurate. However, each of the three models illustrates different trends in this regard, suggesting that number of generations actually has little effect on accuracy. This result was quite counter-intuitive and went against the relevant hypothesis, but can be accounted for by the fact that the most accurate model with regards to the training data was not necessarily the most accurate model with regards to the testing data.

According to Chart 5, in Niño 3.4 and L.A. precipitation predictions, the set of models with 100 leaves was much more accurate than the set with 200 leaves. However, this is not the case with the SOI models. Based on the majority and the fact that Niño 3.4 and L.A. precipitation models were much more accurate, however, it can be concluded that the models with 200 leaves are more accurate than those with 100, the same result that the average values indicate and that the hypothesis predicted. One possible explanation for this trend is that the increased number of leaves led to a larger pool of potential models at each generation and overall, allowing for more accurate models to be created.

Predictions made by Chart 6 are probably unreliable, as the models failed to predict the test data of SOI well. However, Charts 7 and 8 are reliable based on the models’ accuracy at predicting the test data. Chart 7 indicates a steady dropping of the Niño 3.4 in coming months, while Chart 8 indicates relatively low annual rainfall in Los Angeles in future years.

**Future Research**

Many improvements can be made to this study. Potential improvements in accuracy include use of more than just one independent and dependent variable, curve smoothing to reduce random variation of the input data, priority of operation for nodes in order to circumvent order of operations, weighting of recent values more than earlier ones, and larger sample sizes of the models. Unfortunately, the program is actually quite slow and inefficient, so this larger size may cause the program to take too long. Thus improvements in efficiency would be needed.

Future research could incorporate these potential improvements. Also, more indices and characteristics could be tested in order to build a more robust model and to find out what makes an index or characteristic more predictable.

It is very important to note that although the models generated in this study predicted El Niño characteristics, this is not the study’s only use. Instead, the program presented in the study can actually be used to generate models which predict any time series, such as ocean tides, sunspots, and stock market prices. It would be interesting to have the program generate and test models for a wide range of time series. Furthermore, comparing accuracies of models for many different time series could lead to further insights on predictability.. someuture values could be predicted with a reasonable degree of accuracty