Permutation Decision Tree Project Report - 2

For Partial Fulfillment of CS F266

Pranav Srinivas
2020A7PS1694G
f20201694@goa.bits-pilani.ac.in

Ravi Sanker S
2020A7PS0142G
f20200142@goa.bits-pilani.ac.in

I. Introduction

In this report, we aim to generate an extensive Auto-Regressive Dataset using the following two equations and use it as input for the Permutation Decision Tree we had implemented in Project Report-1 [1].

$$x(t+1) = c_1 * x(t) + c_2$$

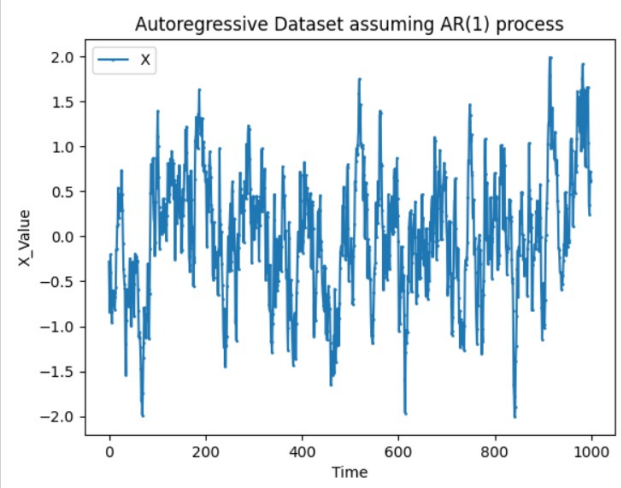
$$y(t+1) = b_1 * y(t) + b_2 * x(t) + b_3$$

Using Scikit-learn's TimeSeriesSplit for Cross-Validation, we wish to study the performance of the Permutation Decision Tree with Auto-Regressive Datasets and see if it is able to correctly classify causes and effects.

II. Methodology

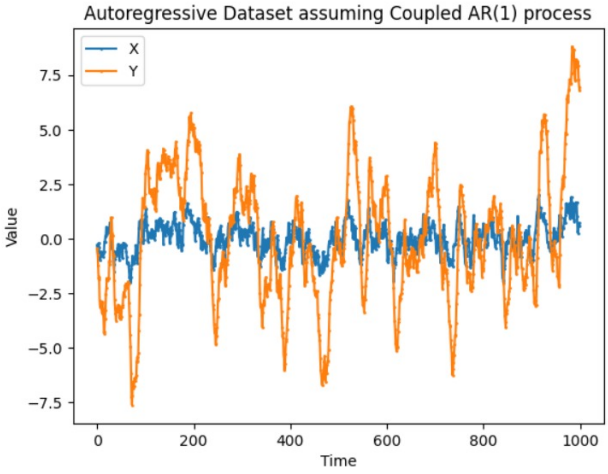
A. Generating the Auto-Regressive Data-set

Initially, we have assumed the data-set contains 10 features. However, note that this can be generalised if required. Each feature follows the equations described above independently. The constants mentioned in the equation have been hard-coded for now, except for c_2 and b_3 which have been taken as noise and are generated randomly. They are generated from a normal distribution with $mean = 0$ and $variance = \frac{1}{3}$. A quick visual inspection check for an Auto-Regressive process is to check if the data varies about a mean. Following is a plot of one of the features with time:



B. Coupled Auto-Regressive Process and the Coupling Coefficient

A coupled Auto-Regressive (AR) process is a statistical model that is useful for modeling and analyzing the relationships between multiple time series variables that can influence each other over time. The primary idea is that each time series variable is regressed on its own past values as well as the past values of other related time series variables. We have represented this using the variable y which not only depends on its past values, but also on the past values of variable x. Following is a plot of the Coupled Auto-Regressive Processes $x(t)$ and $y(t)$:



The coupling coefficient determines the strength and direction of the relationship between the variables in the system at different lags. It is represented by b_2 in our equations and has been hard-coded to the value $b_2 = 0.6$.

C. Feeding the Data-set to the Permutation Decision Tree

Before using the data to train the Decision tree, we have to remove the first few data instances to make sure we have no transient data. Transience refers to the behavior where the influence of past values on the current value of the time series gradually diminishes as you go further back in time. This means that current time series values do not depend heavily on the transient data values. For this report, we have omitted the first 100 data instances.

In the data-set that we have generated using the above-mentioned equations, $x(t)$ represents the *cause* and $y(t)$ rep-

resents the *effect*. We generated appropriate labels to train the Permutation Decision tree classifier where *Class-0* represents *cause* and *Class-1* represents *effect*.

D. Using TimeSeriesSplit

Due to the temporal nature of the data-set, we had to use a specific method called TimeSeriesSplit to split the data-set into Training and Testing. This is because unlike the regular TrainTestSplit method, TimeSeriesSplit takes care to ensure that the temporal structure of the data is not disturbed in order to ensure that there is no bias involved when training the classifier.
We made 10 splits and evaluated the performance of the Permutation Decision Tree on different possible temporal arrangements of the data-set.

III. Experiments

A. Alternating Cause and Effect Permutation

In this Permutation, we arranged the data such that every data instance from the *cause* class was immediately followed by a data instance from the *effect* class. We took 200 data instances from each class giving us a total data-set size of 400 rows and 10 columns.

We then made 10 splits of this data-set using the TimeSeriesSplit and evaluated the performance of our model using SkLearn's *accuracy score* which gives us the percentage of correctly predicted data instances.

B. Permutation With Causes Followed by Effect

In this Permutation, we arranged the data such that the first 200 data instances are from the *cause* class and the next 200 data instances are from the *effect* class. This again gives us a total data-set size of 400 rows and 10 columns.

We then made 10 splits of this data-set using the TimeSeriesSplit and evaluated the performance of our model using SkLearn's *accuracy score* which gives us the percentage of correctly predicted data instances.

IV. Observations

In the case of *Permutation A*: The classifier was unable to construct a tree because of the fact that the parent dataset had a very low ETC value as compared to the children which resulted in a $-ve$ value of Information Gain.

This happens because when the labels are perfectly alternating in a dataset, the NSRPS algorithm will require only one pass to completely compress all the data but this will not be true in case of the children nodes because they would be some non-empty subset of the parent, and so would not have a perfectly alternating sequence of labels.

In the case of the *Permutation B*: Using ten-fold cross-validation, we get the following accuracy values:

Split Number	Accuracy (in Percentage)
1	100.0
2	100.0
3	100.0
4	100.0
5	44.44
6	16.67
7	27.78
8	22.22
9	41.67
10	55.56

V. Conclusion

We get an average accuracy of 60.83 (in percentage) in classifying cause and effect data instances generated using the AR(1) model.

References

[1] Harikrishnan, N.B., Nithin Nagaraj (2023). Permutation Decision Tree. arXiv:2306.02617v2.