Finding Relationships in Numeric Data with the Numeric Correlation Algorithm

Christopher Palazzolo

Finding Relationships in Numeric Data with the Numeric Correlation Algorithm

Abstract

The Numeric Correlation Algorithm is an algorithm I have created for determining the most likely target in a numeric data set, and which features influence the target. This is the first step in abstract data science and allowing the data to tell us what the relationships are instead of us telling the data what the relationships should be.

Introduction

Given any numeric data set, I intend to create an algorithm which can determine the most likely target variable, and which features have the most amount of influence on the target. This is the first algorithm in abstract data science, and the first step towards context independence.

Background

During the research process, both the Pearson correlation coefficient, and the Spearman's correlation coefficient were used. The Pearson correlation coefficient yielded the best results and will be the one which will be used in this publication. The formula for the Pearson correlation coefficient is as follows.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

Methodology

The goal of the algorithm is to determine which variable in a data set is the target, which are features of the target, and which variables are auxiliary variables (variables which have little to no influence on the overall target).

This algorithm follows the following steps.

1. Find the correlation between every variable and every other variable.
2. Find the mean absolute coefficient for each column between every other column. For example, the mean absolute coefficient for *A* would be the mean of *A* and *B*, *A* and *C*, and finally *A* and *D*.
3. Use a threshold (*k*) to determine which variables are relevant.
4. Look back at the correlation between each relevant variable, and the determined target variable to learn how each dependant variable influences the target variable. In this case, we care more about the sign of the correlation than the actual correlation.

To better explain the algorithm, I will be using the following small data set to run the algorithm. Note that there is no context given with this data set, and that is the point of this branch of data science. Data without context.

	A	B	C	D
0	1	4	8	5
1	2	1	10	3
2	5	3	9	8

This data set was specially engineered to have a relationship in it for testing purposes. Variables A, B, and C are random numbers between 1 and 10. Variable *D* is calculated by adding variables *A* and *B*. In this case, variable *C* is the auxiliary variable.

The first step is to find the correlation between every variable and every other variable. This has been done in the following table. Numbers have been rounded to three decimal places.

A	B	-0.052
A	C	0.240
A	D	0.795
B	C	-0.982
B	D	0.564
C	D	-0.397

This process has shown that there is a very strong, inverse correlation between *B* and *C*. While *C* is our auxiliary variable, remember that the goal is to find hidden relationships in unknown data. While our small sample data set was engineered to have *D* as the target variable, in the generated data it is entirely possible that *C* ends up being a better target than *D*. That is the advantage to removing any context at all from data problems, and Abstract Data Science as a whole. We may learn something about our data which we had never anticipated.

If we look back at our table of correlations, the second strongest correlation is between *A* and *D*. Finally, the third strongest correlation is between *B* and *D*.

Our next step is to determine the mean absolute correlation for each column. In the case of *A*, the mean absolute coefficient for *A* would be the mean of *A* and *B*, *A* and *C*, and finally *A* and *D*. We take the absolute value because with correlation, both a 1, and a -1 indicate a strong correlation. If $cor(A, B) = 1$ and $cor(A, C) = -1$ and we did not take the absolute value, the mean correlation would be calculated as 0. This would cause two perfect correlations to be detected as non-existent. However, by first taking the absolute value first, the mean would become 1 which would best describe the relationship.

We can calculate the mean correlation for each column as follows. Numbers have been rounded to three decimal places.

A	0.363
B	0.533
C	0.540
D	0.585

From the above table, *D* has the highest mean absolute correlation. This makes it most likely to be the target variable of the data set.

Knowing *D* is our most likely target variable, we can look at the correlation between *D* and every other variable to determine which variables *D* depends on in some way. The following is a list of the correlation between *D* and the other variables:

A	D	0.795
B	D	0.564
C	D	-0.397

We can now use our threshold (*k*) to eliminate variables which may not affect *D*. If we were to select a value of 0.5, then any variable with an absolute correlation which is less than 0.5 would be eliminated. This would leave us with *A* and *B*. Both of these variables are positive which indicates as *A* or *B* increase, *D* will also increase. If you recall the formula to calculate *D* used to generate our data ($D = A + B$) this conclusion holds true.

Weisstein, E. W. (2021, April 15). *Pearson's Correlation Coefficient*. Statistics Solutions. https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/pearsons-correlation-coefficient/.

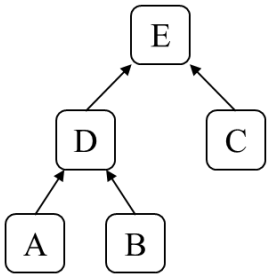
Weisstein, E. W. (n.d.). *Spearman Rank Correlation Coefficient*. from Wolfram MathWorld. https://mathworld.wolfram.com/SpearmanRankCorrelationCoefficient.html.

Results

In conclusion, the algorithm works as we expected it to. Unfortunately, there is no way I can think of to verify the accuracy of this form of algorithm because its goal is to find things that we cannot. I wrote a script to perform this algorithm in Python and ran a data set of 10,000 instances and 7 features which was engineered in a similar way (with a more complex calculation to get the target variable) and the result was similar to what I had expected. This does not say the algorithm works, but it does not say the algorithm does not work. More research will be needed to determine a verification system.

The algorithm appears to be able to accomplish the goal it was designed to.

There are some limitations with this algorithm. The more auxiliary variables exist in the data set, the less accurate the results will be. Also, the algorithm only allows us to find a single target variable. There will very likely be data sets which contain many targets, some of which are features of a larger target as illustrated in the following figure:



Despite this limitation, this algorithm is a good starting point. Further use of this algorithm will give us a better understanding of the efficacy of the algorithm.

Conclusion

In conclusion, the proposed 4 step algorithm appears to work, and work well, however, due to limited testing algorithms, we can not say yes or no for certain. The algorithm detected the relationship we created for it to find, but it is possible it missed a totally different relationship in the data which not even we know about.

This requires more research in testing the algorithm with real data sets, verification to score the algorithm, and refinements to the algorithm and new algorithms which can get around the mentioned limitations.