

Hello everyone,

From my research on feature neutralization and feature exposure I've prepared a basic introduction note for myself and think it may help start-off anyone researching the topic, hope it's helpful.

References: this post is basically a mash-up of the following links

1. [Our Experience with Numerai. Introduction to Numerai | by Saahil Barai | Analytics Vidhya | Medium](#)
2. [Model Diagnostics: Feature Exposure - Data Science - Numerai Forum](#)
3. [What exactly is neutralization? - Data Science - Numerai Forum](#) see comment by akak2021
4. [Feature Exposure Clipping Tool, and working code to deploy locally | Numerai FN Special Part 3 - YouTube](#)

Feature neutralisation and feature exposure explained:

Text in quotes - taken from references

In our models we want to reduce feature exposure, a model with high feature exposure (high correlation with particular features), will result in inconsistent predictions over time:

(Ref 2) The idea behind feature exposure is as follows: Any supervised ML model from a very high level perspective, is a function that takes an input feature vector (X) and outputs a prediction (y). At training time, the model learns a mapping between input features and the predictions. With the numerai data, the underlying process is non stationary. i.e features that have great predictive power in one era might not have any predictive power, or perhaps might even hurt the model's performance in another era. A model that attributes too much importance to a small set of features might do well in the short run, but is unlikely to perform well in the long run. Feature exposure (more specifically, max feature exposure) is a measure of how well balanced a model's exposure is to the features. Models with lower feature exposures tend to have more consistent performance over the long run

We can reduce this exposure by standard methods such as regularisation (see ref 2) or by another method: feature neutralisation:

Feature neutralisation consists in subtracting from our predictions the linear relation between one of our features and the target, "neutralising" that feature - eliminating the component that the feature contributes alone, leaving only the interactions with other features - the intuition behind this process is as follows:

(Ref 3) Neutralization of a prediction for risky features is the first order approximation of the operation that removes the component that the risky feature contributes alone, leaving only the interactions with other features.

For simplicity, we will consider neutralizing for just one feature x_1 .

Without loss of generality, we can assume that the true target value y is deterministically determined by the following function

(Eq.1)

Please note that $f(x_1)$ is the component that only x_1 contributes to y .

Under the assumption of ignoring terms above the second order, the neutralization for x_1 is equivalent to deleting $f(x_1)$.

This result can be obtained through a calculation to find α and β that minimize the squared error of Eq.(1) and

. (Unless my algebraic calculations are wrong...)

Since it is only a first-order approximation, this argument does not hold if the absolute value of the feature value is large.

In order to do this, the numerai example scripts propose using the [Moore Penrose matrix](#), which is a way of finding a matrix's inverse (it's pseudo inverse more exactly). As the matrix we are trying to invert has more rows than columns it is not invertible, with Moore-Penrose's inverse we find the pseudoinverse that minimises least squared error.

(Ref 1)

[

5

709×192 5.75 KB

](https://forum.numer.ai/uploads/default/original/2X/b/ba292906323c92d2950f60bcc8c46e3a736f49d7.png)

Feature neutralization begins by taking the entire dataframe, the column to neutralize, the features to neutralize by, and the

neutralization proportion as inputs. The first and second lines isolate the column to neutralize (scores) and the features to neutralize (exposures) respectively. For the third line, the code is reducing the neutralization column by a vector multiplied by the proportion specified. The vector being used is computed by first taking the dot product of the pseudo inverse of the exposures with the scores and second taking a dot product of the resultant and the exposures.

[

3

488×688 26 KB

](<https://forum.numer.ai/uploads/default/original/2X/6/6c5ced5a91bf80a4a0851cb832c643a6ccfd380b.png>)

In the context of our problem above the Moore Penrose matrix is represented by the result of `np.linalg.pinv(exposures)`. The vector `y` can be thought of as the scores represented by the `.dot(scores)`. Lastly, `x` can be thought of as a vector of beta values. It is important to keep in mind that the Moore Penrose solution is not an exact solution because of the “ $m > n$ ” constraint placed on the problem. However, if $m = n$ we could obtain an exact solution. The Moore Penrose solution produces a solution with the least squared error and this is why we can think of the `x` vector as a beta vector where beta represents the coefficients of a least squares linear solution. Once these beta values are computed we take another dot product this time multiplying the exposures and beta values. This produces what we can think of as a prediction from the least squares solution. This prediction is then multiplied by the desired proportion and subtracted from the original score vector to create a new score vector. Finally the new score vector is divided by its standard deviation to rescale it and then returned. The goal of this process was to reduce feature exposure.

In numerai’s advanced example script we only neutralise the 50 features the script identifies as “riskiest features”, in other examples, such as those discussed in refs 1 and 2 all features are neutralised but a proportionality factor is used to scale down this neutralisation.

There is a trade off between neutralisation (which increases consistency) and correlation (from ref1):

[

4

667×532 57.1 KB

](<https://forum.numer.ai/uploads/default/original/2X/d/dd63fbedce9a1a82b0b5806ec936c94b32d77ed7.png>)

Looking for the sweet spot is key in Numerai models, by choosing which features to neutralise or by tuning neutralization with a parameter (proportion) or both.