* 1. Hi every one, I am Yun-Hao yang, form the Institute of Industrial Engineering of NTU. I am glad I can be here to share the topic, which is about variable encoding for helping the data with numerous binary features.
  2. as most of you may already know, variable encodings help us when we stumble upon categorical data while doing Data pre-processing, most of them assigning numerical value to each category in the feature, like ordinal, frequency, and target encoding.

However, One Hot encoding is also one of the most commonly used Among all methods, for its convenience and efficiency. One Hot encoding creates dummy features for each category. Like the table shows, if we have 4 unique types in the city features, it created 4 dummy features to describe it

It might seem ok when the number of types are limited

Yet, many problems might come along with it when the types increased.

* 1. Unfortunately, If the categorical feature has numerous types, one hot encoding will largely expand the dimension of the data, and might trigger the curse of the dimensionality, causing ML model perform poorly.

The reason why is mainly due to too many features encumber the ML model in making decision, making model overfitting to the training data. And also, the numerous 0 value would make the boosting algorithm invalid while calculating the gradient.

* 1. The common way to prevent this is simply encode with others methods, Instead of presenting data in numerous binary columns. if we receive this kind of data, which has many grouped binary features, we can reverse from one hot encoded data back to categorical, then choose another method to solve this problem.
  2. But, what if the data we got in first hand has no default group for those binary features? What if there are too many binary features to deal with? Is there a better way to describe, or encode this kind data?
  3. As the traditional encoding method is only for categorical part. In the research, we try to develop a variable encoding method built for binary data, which helping encode them as numerical.

By column Grouping, Sequencing, and binary-coded decimal encoding, we can now represent the binary data in a better way via the method we develop. The following animation will show how these steps works.

* 1. As we gather the non-grouped binary data, there might be some relationship between columns, yet we don’t know about it.
  2. The thinking is by the feature selection or extraction method can help to find this relationship. In this case, we successfully group the columns by their animal class, under mammal, fish and Amphibian
  3. Secondly, for generating a better encoded numerical value, we need to rearrange the column sequencing in group by certain rule. The detail and the reason why will be explained later on
  4. F
  5. In the last step, with the binary – code decimal encoding, we can represent column groups in into numerical values. And that’s it, the process is ended. We have generated a both understandable and tidy dataset with numerical value form the non-grouped binary data.
  6. Here is a little example. Assume we are a zookeeper who have to check of animal's health status, and here is the checklist, showing the sample’s animal type, size, color, and health status. Each checkbox in same group is exclusive to each other.

After collecting all 300 animals in your zoo, the result is at the right, presenting how many times each checkbox was checked, and what kind of health status the checked animal was.

Like, in the size: mid checker, 100 out of 300 samples are checked, and half of them are in the good status, other half of them don’t.

* 1. First, we try to find the relationship of these 18 checkers, for now, we can follow the default group from the data if we have it, or by the PCA or correlation clustering. In this case we using PCA.

By viewing the weight of 3 Principle components, PCA helped us makes 3 different columns groups, The main purpose using PCA here is by flittering out the less importance features, We can create groups that contain the most importance ones.

* 1. After grouping the features by PCA, we now can rearrange the columns within three groups. We had been trying using column sum, impurity, and Feature importance to sequence the columns.
  2. In the right hand side we can see the columns been rearrange with its total columns sum decreasingly.

But at this point, you must be thinking why bother the sequence. What's the difference between good sequencing and the bad one?

* 1. In fact, sequencing is decisively importance to the encoded output value. If a certain sequence can make good and bad these two types be clearly sperate in encoded numerical value. Will makes classifier easily to find the cut point in the new feature. In a nut shell, a good sequence can make our encoded numerical data easier to classify.

The example shows if we sequence the columns well, than the value in the new features would be much easier to classify.

* 1. But how to transform multiple binary value into numerical by Binary-code-decimal, After sequencing? In the right is the how the BCD works. It basically is by decimalize a sample's grouped columns as a binary value. Like the sample here, 2 power 5 plus 2 power 0 is 33 in total.

As you can see, if we can put a more Informative column at front, it can affect BCD value more than other columns do. Hence putting informative column at front can help us in output a better numerical value in separating two types.

* 1. That's the main steps of whole process, the goal of the research is focus on how to find a better grouping and sequencing methods, in order to generate a more easily classified data.
  2. In case study we will go through the simulated continuous data, and a Kaggle dataset. To demonstrate our method, and compare with the traditional encoding method in the same time.
  3. These shape data were generated procedurally, with around 3 thousand samples and ten to one ratio in two types, and has several noises in the majority. The following I will only be using shape 1 as an example due to the time constraints.
  4. First, we binarize each axis of the shape data, and use both our method and the traditional variable encoding to numericize the data. Then compare the classification result.

Here can see the 3 axes of shape1, and type was showing by the color. And the right shows after binarize each axis to 10 categories.

* 1. Here is a little demonstration of PCA grouping before sequenced by column sum. After binarized the data into 30 binary features in total, PCA makes them into 5 different groups. Principle components will selecting the columns gradually, avoiding the columns had been selected from the former PC.

in each PC we can see columns sliced from different default group, and also, compare PC1 to PC5, PC5's columns has higher type purity, yet less in sample size. Meaning those binary features in PC1 has more samples can explain more variance to the whole data.

The BCD coding result is reasonable, the informative seems dwindling by PC sequencing.

And we can clearly see the grouped columns sort by the column sum. The disturb here is because the data was split into training and testing part at first, our method was trained by the training part, yet here shows the whole data.

* 1. This shows difference in sequencing randomly or with type impurity. In the bottom right can clear see the type red is more condense at the left on the number line. Yet cannot observe such trend at top right, which represent the encoded value of randomly sorted.
  2. The method we develop can also use as a way of dimension reduction, visualize the encoded data in to the dimension we wanted to.

We can also see the difference of sequencing, All these three data was follow the default group, but sequencing differently.

It’s not hard to image what kind of data will be benefit whiling training the classification model.

* 1. The line chart shows the k-folds validation f1 score average under different binary features we binarized from the continuous data. All dataset performs alike when the number of features is low. Yet, as the features increased, the variance became wider, meaning the grouping and sequencing getting more importance. ~~We can also discover that the PCA group has the worst f1 score among all groups.~~
  2. Bar chart shows the average f1 score under different binary features. We can see the score of purity and the feature importance is much higher than the random in all groups, and slightly higher than most variable encoding method, except for target encoding, which is a strong supervised encoding method. Our conclusion here is grouping and sequencing will definitely effect on classification result.

The numbers below show the data’s number of features. One hot and binary encoding has no fix number of features, they variate with how we binarize the continuous data.

~~For default group and traditional encoding method, we know the binary data was originally from the 3 dimensions. However, in correlation or PCA grouping, we assume the group~~

~~knowledge is not given, hence setting 5 as the group we wanted to.~~

* 1. Move on to the Kaggle dataset. We selected some of features as the read-in data. And it contains multiple data types, includes numerical, categorical, and binary. After one hot encoded, it will generate 75 dummy features.
  2. We can clearly see the default grouping is the best grouping method compare to correlation or PCA. And default with purity can match the result of target encoding.
  3. Although two the grouping method we try cannot outperform the default group or some traditional encoding methods, in facing non-grouped binary data, those method can’t be used anymore. And there we can see the advantage of the method we use. By compress the 83 binary features into 5 numerical, and maintain a same or slightly better classification score.
  4. In conclusion, we formulate a way to encode non-grouped binary data into numerical, and can preserve the classification score with certain sequencing method. Or the method can be used as a dimension reduction method for binary data.

That’s all, thank you.