Q3 Answer Report

This report is a reflection on the reading of paper ‘How doppelgänger effects in biomedical data confound machine learning’ (hereafter referred to as the original paper). From a personal perspective, I will express my views and discussions on ‘whether the doppelganger effect is unique to biomedical data’, ‘how to check and avoid the negative effects of doppelganger effect in machine learning model’ and other issues as well as some contents of the original paper. All contents in this report are my personal opinions only, and any similarities are purely coincidental.

1. The universality of doppelganger effect in various fields

1.1 Functional doppelgangers don't just exist in biomedical models

Although the original paper only mentioned some biomedical-related doppelganger effects, for example, both data set used in the training of quantitative structure-activity relationship (QSAR) model and renal cell carcinoma (RCC) proteomics data taken from the NetProt software library have data doppelganger to some extent, I believe that from the perspective of the generation principle of the doppelganger effect, models in other fields may also have functional doppelganger.

To prove this, I'd like to make some connections between the doppelganger effect and the overfitting phenomenon. Anyone who knows about machine learning knows that overfitting is a very common problem in the process of training models. It means that the model is overly accurate in catering to all the training data in order to minimize the error between the predicted value and the label (actual value). After over-fitting, although the model can fit the training data well, it cannot describe the overall characteristics of the data, resulting in poor adaptability to the new test data.

Now, looking back at the "data doppelganger does not necessarily produce the doppelganger effect" mentioned in the introduction of the original paper, it may be able to find the law of producing the doppelganger effect when combined with the

phenomenon of overfitting. I suspect, as stated in the original paper, that the data doppelganger does not necessarily cause the doppelganger effect. However, when a machine learning model has a slight or serious overfitting problem, and some data in the training set and test set are highly similar, the doppelganger effect is very likely to occur.

The reason is that, if a model overfits the training set, it means that its prediction error for the data in the training set is very small. When it is used to predict the test set data which is highly similar to the training set, the error is also relatively small. However, when new data that is less similar to the training set are predicted, the error shown increases. Therefore, when predicting test sets with data doppelgangers, the accuracy of the model would be higher than it actually was, which was called "overstated".

If the above reasoning can be established, we can find that there are two conditions for the emergence of the doppelganger effect: 1. There is a certain proportion of similar data in the training set and the test set, that is, data doppelganger. 2. A certain degree of overfitting occurs during the training of the model.

Furthermore, if all conditions are ideal and known, it may be possible to calculate the inflation of model performance caused by the doppelganger effect by the above conclusions. Let's assume that the proportion of data doppelganger in the test set is P, and the fitting accuracy of the model to the training set after over-fitting is A1, and the fitting accuracy to other new data (non-doppelganger data) is A2. Then, the prediction accuracy of the model for the training set after the inflation of the doppelganger effect is $(P * A1 + (1-P) * A2)/100\% = A2 + P * (A1-A2)$. Compared with the actual prediction accuracy A2, the difference of accuracy improvement brought by the doppelganger effect is P * (A1-A2).

So, going back to the original question, is there a doppelganger effect of machine learning in other areas as well as in biomedical area? The obvious answer is yes. There are always some test sets that are similar to the training set (whether due to the leakage of the training set or the problem of data acquisition), and there is no guarantee that there will not be over-fitting in the training model. When these two conditions occur at the same time, there is a doppelganger effect.

1.2 The doppelganger effect may indeed show up more frequently in biomedical models

While the data doppelganger effect may occur in machine learning models in any field mentioned above, I think it will happen more frequently in the biomedical field. The reason is that, from a biological perspective, the structure of each part of the same species among different individuals (such as organs, etc.) are similar or the same, so the data from different individuals, such as organization shape, cell number, hormone levels and so on are likely to be similar. The similar data are recorded in the data set and as features while training, then the accuracy of the model will be affected when testing. In contrast, datasets in other fields are less likely to be constructed with a large number of highly similar features, and the data doppelganger will be less obvious.

2. Possible ways to avoid or minimize the effects of the doppelganger effect on the model

2.1 Before model training

First of all, for some specific data sets, if the data can be dimensionally reduced and visualized in certain ways (such as principal component analysis), it is necessary to check the sample distribution of the data set before training the model. By analyzing the similarity of sample distribution of training set and test set, we can predict whether the model is affected by data doppelganger in advance. If the sample distribution is difficult to visualize, other methods of showing data relevancy, such as Pearson's correlation coefficient mentioned in the original paper,

should be used to determine the possibility of data doppelganger.

2.2 After model training

As mentioned in the third recommendation in the original paper, it is also a good method to use as many data sets as possible for divergent validation after model training is completed. It can test the effect of the model in dealing with different data sets and test the robustness of the model.

In fact, I once encountered a doppelganger effect problem when constructing a natural language processing model. At that time, I wanted to conduct a sentiment analysis on a batch of shopping reviews from a shopping website. I pre-processed all the review data properly, and converted them into vectors using word2vec and doc2vec, and then put these vectors into an SVM-based classifier to judge the sentiment polarity of different reviews. After the training, I tested the model by means of 10-fold cross validation, and the results showed that the accuracy of the model was very good. But out of caution, I used an additional corpus from another site, and the accuracy dropped from 87% to 73%, which was baffling.

Later, through careful analysis, I found the reason. In this batch of shopping comments, there are a certain number of default positive and negative reviews from shopping websites. Although these statements are preprocessed and stop words are removed, they are also mixed and distributed in the training set and test set. The model has a high accuracy in predicting these default statements, thus improving the overall accuracy of validation. However, after the replacement of the corpus, the model accuracy returned to the actual level due to the absence of these default statements.

It was due to the divergent validation by the extra corpus that I realized the shortcoming of the model performance and the doppelganger effect (I didn't know it was called the doppelganger effect at the time).

2.3 During model training

The above methods for examining functional doppelganger have been mentioned in the original paper, and I agree that they may indeed be useful for identifying functional doppelganger. But in addition to before and after the model training, I think take some measures during the model training can also play a good effect. In the previous section, I guessed that overfitting might be the cause of the functional doppelganger for the data doppelganger, so by inference, some methods to avoid overfitting might be used to avoid the doppelganger effect.

For some traditional machine learning models, L1 regularization can be added when the loss function is defined. L1 regularization is used to generate a sparse weight matrix, that is, to generate a sparse model. It is equivalent to the feature selection of the original data set to screen out those features that have a greater impact on the results. In this way, the weight of features that exist in both the training set and the test set and have little influence on the actual results will be reduced, reducing the potential negative impact of the data doppelganger.

For deep learning and neural networks, 'Dropout' can be added while constructing the models. During the model training, due to the random elimination of neurons, the network has a certain sparsity, thus reducing the synergistic effect between different features and the joint adaptability between neurons, thus enhancing the generalization ability and robustness of the whole model.

3. Conclusion

In general, I think the doppelganger effect is common in all fields of machine learning models, but it may be more frequent in biomedical models. In addition, before, during and after model training, some methods could be used to check the data doppelganger or avoid the doppelganger effect.