Potential Solutions for The Doppelganger Effect

Yaning Zhang

The doppelganger effect is a problem in which the results of ML applications seem to perform well due to similar data. The author gives some examples of the doppelganger effect and attributes it to the similar features or functions of molecules, such as erroneously speculating proteins belonged to the same ancestor and the QSAR model failed to detect variations on similar molecules (Wang et al., 2021). From a quantitative perspective, this effect is similar to the wrong split of the test set and validation set, which may cause high accuracy on the training set but performs poorly on the test set. The difference is that for the doppelganger effect, predictions perform well not because of the number of training sessions or the model fit on the set, but because the training set is highly similar to the test set, resulting in good classification or prediction results on the test set.

The paper demonstrates some recommendations for avoiding doppelganger effects. The first way is cross-checking using meta-data as a guide. With this method and meta-data, it can find potential doppelgangers and divide the dataset into training and test ones. The second method is to stratify the dataset instead of evaluating the whole one, and to do more independent validation checks is another proposed way to improve performance (Wang et al., 2021). The Doppelanger effect is not limited to biomedical data, and it happens because of similar features of the dataset, so I think there are two solutions. The first solution is to cluster the datasets based on similar features, and make an analysis for each group. The second solution is to divide the validation set and training set reasonably, and make the process more random to avoid data similarity problems. It divides the dataset into layers based on similarities, and I also learned some methods to split the dataset. I think cross-validation may be a good way to divide training and test set, which includes Leave-one-out cross-validation (LOOCV) and k-fold CV methods. LOOCV is the method that considers one individual as the test set, the remaining observers as the training set. The ML model can be trained and fit on (n-1) observers, and make a prediction on one observer. The process repeats n times, and it rarely happens that every observer is similar (James et al., 2013). However, LOOCV is unsuitable for large data sets because its calculation is too large and will take too much energy and time. K-fold CV is another method to replace LOOCV. It divides the dataset into k-fold, which is similar to stratification. Each time one fold is used as a validation set, and the remaining (k-1) folds are used as the training set to fit the model (James et al., 2013). The prediction will be made for one fold. This process will repeat for k times. This method can avoid the validation data being similar to training data, since it is hard for K folds to be all similar.

Reference

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.

Wang, L. R., Wong, L., & Goh, W. W. B. (2021). How doppelgänger effects in biomedical data confound machine learning. *Drug Discovery Today*.