Report for Q3

How doppelganger effects emerge

Data doppelgängers occur when independently derived data are very similar to each other, causing models to perform well regardless of how they are trained. It is well established in ML that, when assessing the performance of a classifier, the training and test data sets should be independently derived. However, independently derived training and test sets could still yield unreliable validation results. For example, models trained and validated on data doppelgängers (where training and validation sets are highly similar because of chance or otherwise) might perform well regardless of the quality of training. When a classifier falsely performs well because of the presence of data doppelgängers, we say that there is an observed doppelgänger effect. Data doppelgängers that also generate a doppelgänger effect (confounding ML outcomes) are termed functional doppelgängers.

According to my understanding, when there is an inevitable similarity between the data, no matter how scientifically divides the training set and the validation set, it may have a misleading impact on the results of machine learning. Thus, I think the reason for the doppelganger effects is the similarity of the original data. As the given paper shows, they observed a high proportion of PPCC data doppelgängers in the RCC data set (half of the samples are PPCC data doppelgängers with at least one other sample). They cannot say for sure whether this is a problem because of PPCC itself or because the transcriptional profile of genes is, for the most part, positively correlated.

Examples of doppelganger effects in different data types

Drug discovery: quantitative structure–activity relationship (QSAR) models are classification and regression ML models trained to predict the biological activities of molecules from their structural properties. QSAR models assume that structurally similar molecules have similar activities. In most instances, this assumption is true (cases of data doppelgängers). Sorting similar molecules with similar activities into

both training and validation sets (by chance during time-split validation or random test set selection) confounds model validation because poorly trained models (trained on uninformative structural properties) might still perform well on these molecules.

Protein function prediction: proteins with similar sequences are inferred to be descended from the same ancestor protein and thereby inherit the function of that ancestor (i.e., the two proteins are presumed to be similar in function). This naïve application of abductive reasoning is true in most cases (cases of data doppelgängers), giving us a false impression of highly accurate predictions. However, on greater inspection, they realize that this approach would be unable to correctly predict functions for proteins with less similar sequences but similar functions, such as twilight-zone homologs and enzymes that are dissimilar in sequence overall but with similar active site residues.

Gene sequencing: as the paper *The Doppelganger Effect: Hidden Duplicates in Databases of Transcriptome Profiles* says, whole-genome analysis of cancer specimens is commonplace, and investigators frequently share or re-use specimens in later studies. Duplicate expression profiles in public databases will impact re-analysis if left undetected, a so-called doppelganger effect. They propose a method that should be routine practice to accurately match duplicate cancer transcriptomes when nucleotide-level sequence data are unavailable, even for samples profiled by different microarray technologies or by both microarray and RNA sequencing. They demonstrate the effectiveness of the method in databases containing dozens of datasets and thousands of ovarian, breast, bladder, and colorectal cancer microarray profiles and of matching microarray and RNA sequencing expression profiles from The Cancer Genome Atlas (TCGA). They identified probable duplicates among more than 50% of studies, originating in different continents, using different technologies, published years apart, and even within the TCGA itself. Finally, they provide the doppelgangR Bioconductor package for screening transcriptome databases for duplicates. Given the potential for unrecognized duplication to falsely inflate prediction accuracy and confidence in differential expression, doppelganger-checking should be a part of standard procedure for combining multiple genomic datasets.

Doppelganger effects are not unique to biomedical data

Since the doppelganger effect is caused by the similarity of data, I think this effect is not limited to biomedical data types. As long as the original data structure has inevitable similarity, it is possible to produce doppelganger effect. For example, in the field of face recognition, twins with similar facial features are doppelganger pairs in the data set. It is also very important to train machine learning models for these similar faces and guard against the risks caused by this loophole.

Ways of avoiding or checking for doppelganger effects

Limited methods:

1.Enforced colocation of doppelgängers in either training or validation sets are suboptimal solutions. When all PPCC data doppelgängers are placed together in the training set, the doppelgänger effect is eliminated. This provides a possible way of avoiding the doppelgänger effect. However, constraining the PPCC data doppelgängers to either the training or validation set are suboptimal solutions. In the former, when the size of training set is fixed (thus, each data doppelganger that gets included causes a less similar sample to be excluded from the training set), it leads to models that might not generalize well because the model lacks knowledge. In the latter, you might end up with spectacular winner-takes-all scenarios (the doppelgängers will all either be predicted correctly or wrongly.)

2. DoppelgangR. In studies in which the PPCC outlier detection package, doppelgangR was used for the identification of doppelgängers, PPCC data doppelgängers could be removed to mitigate their effects. However, this approach does not work on small data sets with a high proportion of PPCC data doppelgängers, such as RCC, because the removal of PPCC data doppelgängers would reduce the data to an unusable size.

Suggested method:

1.Perform careful cross-checks using meta-data as a guide. Here, we used the meta-data in RCC for constructing negative and positive cases. This allowed us to anticipate

PPCC score ranges for scenarios in which doppelgängers cannot exist (different class; negative cases) and where leakage exists (same-patient and same-class based on replicates; positive cases). The plausible data doppelgängers that warrant concern are samples arising from same class but different patients. With this information from the meta-data, we are able to identify potential doppelgängers and assort them all into either training or validation sets, effectively preventing doppelgänger effects, and allowing a relatively more objective evaluation of ML performance.

2.Perform data stratification. Instead of evaluating model performance on whole test data, we can stratify data into strata of different similarities. Assuming each stratum coincides with a known proportion of real-world population, we are still able to appreciate the real-world performance of the classifier by considering the real-world prevalence of a stratum when interpreting the performance at that stratum. More importantly, strata with poor model performance pinpoint gaps in the classifier.

3.Perform extremely robust independent validation checks involving as many data sets as possible. Although not a direct hedge against data doppelgängers, divergent validation techniques can inform on the objectivity of the classifier. It also informs on the generalizability of the model despite the possible presence of data doppelgängers in the training set.

My idea:

Inspired by the method proposed in this paper, I suppose that the doppelgänger effect can be solved by setting more validation sets. There are two ways to implement more validation sets: one is to get more data from metadata and set them as different validation sets (not put them into only one validation set); the other is to divide the existing validation sets into independent subsets as new validation sets. The accuracy and effectiveness of the machine learning model are tested in these validation sets. My starting point is that the validation set is as many and random as possible, which can weaken the similarity between the validation set and the training set, so as to reduce the proportion of doppelgängers pairs and thus reduce the doppelgänger effect. In the initial stage, the worse the performance of the validation set, the lower the similarity with the training set, but the most effective validation set for the whole progress. We can keep

the validation set with poor performance as validation set, discard other validation sets with good performance or add them to the training set to train, and train the machine learning model again. We can be confident that the performance of this validation set is reliable.