Data Doppelgangers: What are they and how to Deal with them

Introduction to Data Doppelgangers

Machine learning classifiers function by identifying the features which are similar between positive examples and which are different from the negative examples in the training set. In some instances, the model will tend to memorize patterns that exist only in the training data, but which do not exist in the population being modelled in general. This problem is known as overfitting.

An overfitted model performs very well on the training set, but does poorly when tested on data that it has not trained on. A technique known as cross validation attempts to address this issue by evaluating the model on a separate set of examples which the model has not seen during training. Thus, cross validation attempts to test whether the model learned trends that generalize over examples outside those covered during training.

Nonetheless, cross validation only works under the assumption that the validation set is adequately dissimilar to the training set. If for whatever reason, the validation set is similar in feature space compared to the training set, then the model's accuracy will be falsely inflated. This is because an overfitted model can simply use the "patterns" it has memorized for the training data to infer the targets for the validation data, and because the data points are highly similar, the predictions are highly likely to be correct, leading to grossly inflated model accuracy. The pairs of data points in the validation and training set which are highly similar to each other are referred to as Data Doppelgangers (DDs).

A familiar everyday analogy can be used to explain the effect of DDs. In academics, examinations attempt to measure the performance of students. The performance can be quantified by counting the number of "problems" the students can solve. In Singapore, students frequently study for examinations by attempting past years exam papers and comparing their answers to the solutions provided.

In this analogy therefore, the population is the set of all possible problems in the real world. It is impossible to test students on all the possible problems at once , hence, examinations are developed which sample from that population. This is the dataset.
The current examination is the validation set, and the past year's exam papers are the training set.

Certain professors have a habit of reusing past years exam questions in current examinations. The DDs in this case are the reused exam questions. It is obvious that this habit leads to overestimated exam scores , because students can simply memorize the answers to past exam questions. In the same vein, validating on DDs leads to overestimated validation scores.

Furthermore, when current examination questions are simply the past years' but rephrased, the words of the question might be totally different, but the semantics of the problem remain unchanged and the exam scores remain overestimated. Similarly, DDs do not necessarily have to be similar in terms of features to have an adverse effect on validation scores. In real world datasets, the features which need to be similar between data points for them to be DDs are not well defined, and may not even be part of the measured set of features, thus, identifying DDs remains a challenge.

Based on the above analogy, it becomes quite clear that doppelgangers might not only affect biomedical science data, but potentially data in any other field as well. For example, data doppelgangers might arise when data is not adequately cleaned, resulting in duplicate records, or when efforts are not taken to ensure a representative sample (for instance, not accounting for non response bias). Furthermore, any instance where the dataset contains examples that are similar to each other when taking into account the diversity of the population, might lead to data doppelgangers.

In sales, repeat customers who buy the same products multiple times can cause DDs.
When text is mined during web scraping, for instance, by traversing social media links, datasets potentially can become enriched with celebrity profiles , or trending events, leading to extremely similar records where many different “observers” describe a similar topic in their own words. Finally, in evolutionary biology, where the genomes of different species are compared, failing to exclude repetitive sequences such as retrotransposons might cause highly similar genomes and produce DDs.

Factors which Affect the Effects of Data Doppelgangers

We further ask ourselves this question: While DDs can potentially arise anywhere, under what circumstances do the effects of DDs become more pronounced?

In answering this question, let us consider a simplified model:

Let us define an arbitrary classification problem, X, where the labels are generated by applying a decision tree process to a set of features (we call this the population tree), and that the set of features necessary to classify the population data points are all measured in the data set. Let us define the complexity of the problem as the minimum number of splits necessary to completely classify all the data points in the population. Based on this population tree, we can build a sample tree, where this time, the splits are defined by training data. Let as also assume that the criteria for the splits in the sample tree are identical to the population tree. Hence, at each level, the sample tree omits a split if the sample does not contain examples necessary to populate both branches , resulting in missing leaves in the sample tree compared to the population.

Using this model , one can obtain an intuition about how DDs affect the accuracy of a ML model, in terms of the representativeness of the dataset and the complexity of the problem.

Firsty, one can ask whether an inflated validation score due to DDs necessarily translates to poor performance for the general population. This is not necessarily the case, provided that the dataset remains representative of the population, and the complexity of the dataset is similar to the population. If the population tree requires 15 splits to produce pure samples, a sample tree containing at least 2^15 (or ~32000) examples can adequately predict the target, irregardless of whether the sample has DDs.

However , if the dataset is not representative or diverse, we end up with some population tree leaves having zero examples in the training data, and some leaves having multiple. Thus, population instances falling into the leaves with zero training examples fail to be correctly classified. Additionally, if the problem complexity is very high, it requires a very large number of splits to model it , then most of the leaves will be empty of training data. When DDs exist, therefore, each pair of DDs is likely to be segregated into the same leaf in the training data, and there will be less population leaves that are covered by the dataset. Thus, DDs exert

their effects by reducing the diversity of the dataset, and that reduces the knowledge of the model. This effect is irregardless of whether the DDs are sorted into the validation or training sets.

Ideally, the investigator should detect if DDs reduce the diversity of the training set. This is when cross validation is done . If a datapoint from the validation set falls into a population tree leaf that fails to be defined by the training data, then it is likely to be misclassified. If the validation data is representative of the population, then the probability that it will fall into an undefined leaf reflects the probability that an example from the population will fall. Hence the validation score is reflective of the real world performance. However, consider the case when the validation data consist entirely of DDs from the training set. Then all the validation data will fall into a leaf that is already covered by the training data, and lead to a perfect model score.

Based on the above arguments,I argue that DDs are more of a problem in fields where the problem to be modelled is very complex, or when the dataset is not diverse or does not adequately represent the population.

Biological data science problems are prime examples of where DDs can be particularly deleterious. When considering gene/protein sequences, omics data and image data, the feature space is incredibly high dimensional (if we assume the identities of amino acids for each residue are nominal categorical variables, the space for a 100 residue proteins on the order of (2^20)^100). Furthermore, the class boundaries are likely to be complex, high dimensional and not regular at all. With regards to sequence data, evolution produces protein, functional RNA and genes/regulatory element families that are similar in sequence and function, and thus, we end up with sparse clusters of similar labels in a very large feature space. So, therefore, the positive examples end up to be very similar to one another and negative examples (those from other families) end up so distant from one another such that there are few , if any, representative examples in the feature space between families. Thus, no matter how the investigator splits the train and validation sets, positive examples will always end up in the same evolutionary branch(clusters) , or in cases where divergent evolution has occurred, one of only a few branches, and most examples will end up having a very similar pair in the same branch of evolution.

On the other hand, in fields where the target variables are simple linear or nonlinear functions of the features, like in most physics or engineering problems, accurate models can be trained on datasets containing few data points, and the presence of DDs is not much of a concern.

Avoiding and Minimizing the Effects of Data Doppelgangers

I earlier pointed out that data doppelgangers arise when the feature space covered by the sample is small compared to the population. Therefore, efforts must be taken during sample collection to ensure that the data set is as diverse as possible. DDs arise when such efforts are ignored: for example when using inbred laboratory mice for tissue samples , or by using biased sample preparation techniques , the chances of DDs arising increase. For example, TargetFinder, the algorithm that Fullwood et. al [1]identified to have an artificially inflated performance when predicting promoter enhancer interactions,had training data samples where the most of enhancer promoters pairs overlapped with one another, leading to poor coverage of the enhancer-promoter pair space.

Furthermore, the investigator has to explicitly define the scope of the model, i.e, the population in which the model is to be applied on during production, and ensure that the training and validation data exhaustively cover all segments of that population as much as possible. It might benefit the investigator, therefore, to reduce the scope of the model. With regards to the sequence function prediction problem, for example, instead of aiming to predict the function of any protein sequence, a more accurate model can be produced aiming to predict whether a certain protein is a hydrolase, given that it shares certain percentage similarity to known homologs.

In addition, particular attention needs to be placed on the techniques used to obtain the dataset. With regards to drug discovery for instance, library generation techniques based on chemical synthesis often generate biased molecules (i.e certain conformations/structures cannot be synthesized using that particular chemistry, leading to the examples being synthesised being identical to one another in one particular dimension). The investigator has to identify the limitations of the sampling technique, and reduce the scope of the model to take them into account.

In public datasets, the same samples are frequently reused in to produce data by multiple investigators. This leads to the occurrence of DDs. Checks can be done to verify that each sample is not a duplicate of another, for example, by verifying the source of the sample (i.e from which patient and tissue it was from) , that sample identifiers are unique, or checking that if the data points themselves are not identical (for example, by doing a MD5 checksum). This technique removes duplicate data points, but it does not remove data points from independent sources that are highly similar .

The technique used in the paper [2] to identify DDs relies on Pairwise Pearson Correlation Coefficient (PPCC).The PPCC measures the degree in which the features of two samples are similar to each other. If the PPCC is higher than a certain threshold between two samples, they might be considered DDs. The value of the threshold itself is ambiguous and depends on the context of the study. Nevertheless, the authors suggest a method of estimating the threshold by comparing the PPCC distributions between different categories : category 1: pairs of datapoints that share a similar class labels but from different samples ; category 2: pairs of datapoints with different class labels; and category 3: pairs of datapoints from the same sample (i.e technical replicates). Additionally, the paper defines functional doppelgangers (FDs) as DDs that inflate the model's accuracy. Category (3) is assured to consist entirely of FDs since they have (more or less) identical features and the same label, while category (2) is assured to not contain FDs at all because even if their features are identical, their class labels are different, hence, for every pair, partitioning one into the training set and partitioning the other into the validation set will result in an inaccurate prediction, thus lowering the accuracy of the model and preventing inflated accuracy scores.

The authors observed that there is a cut off, above which there were no pairs with higher PPCC in category (2), and this cutoff coincided with the vast majority of the pairs in (3). Pairs in category (1) fell on a spectrum which overlapped both the ranges in (2) and (3).
Thus, they concluded that the data points in (1) that had higher PPCC than the cutoff should be considered as FDs based on the observation that this cutoff cleanly separated the true FDs and the definitely true non-FDs.

A limitation of this approach that the authors discuss is that this method relies on metadata, this implies that the sources of the samples need to be known and samples are able to be partitioned into categories where FDs are impossible or always the case, based on the metadata. In instances where metadata is not available, this might not be possible. Furthermore, the PPCC measures feature similarity irrespective of whether the feature contributes towards the model prediction. In the case where only a small subset of

features informs the label, two data points which are identical in the predictive features but vastly dissimilar in all other non informative features will end up to be FDs, but with a low PPCC. I therefore suggest that the PPCCs be weighted based on feature importances, where informative features (as determined by a model) are given a higher weighting towards the PPCC than non informative features.

Comprehensive feature engineering and feature selection approaches also need to be applied to identify the features which can discriminate negative classes from positive ones, such that an appropriate distance metric can be developed, and similarity thresholds be set based on the distance metric. As the data at hand is seldom adequate to identify the effects of potentially hidden features, hypothesis driven experiments are necessary to test the effect of each putative feature as well as to discover new ones.

To minimise the effect of DDs, the authors suggest either dropping them entirely, or partitioning all DDs into either the training and validation set. Doing either of these might result in a training or validation set that is unusable in terms of size. Instead of the extreme approach (segregating the DDs entirely) , a more calibrated approach can be taken to minimise the number of doppelganger pairs which span across the train- validation set boundary. Data doppelgangers form a network, where the nodes are the data points and an edge connects two datapoints which are doppelgangers of each other. An edge crosses the train-validation set boundary if one sample in the pair is in the training set, and the other is in the validation set. Graph theory algorithms can be developed to partition the datapoints such that the number of crossing edges is minimised, while at the same time, maintaining a constraint on the sizes of each set. This will result in adequate train/validation set sizes, while minimizing the DDs between the sets.

Conclusion

DDs are datapoints which are highly similar to one another compared to the diversity of the population. When the training and validation datasets contain DDs , the validation score of the model tends to be artificially inflated. Apart from causing overestimated validation scores, DDs reduce the diversity of the dataset, as well as reduce the representativeness of the sample, thus leading to a reduction in model generalizability. This problem is particularly profound in fields where the class boundaries are complex and the feature space is large. Potential techniques to reduce the effects of DDs include selectively partitioning them into training and validation sets, removing duplicated datapoints, and by defining thresholds for similarity using metrics such as PPPCs to identify doppelgangers. Attention needs to be paid to defining the scope of the model, as well as to potential limitations of sample collection which may reduce sample diversity , in order to avoid DDs..

References

[1] Cao, F., Fullwood, M.J. Inflated performance measures in enhancer–promoter interaction-prediction methods. *Nat Genet* 51, 1196–1198 (2019). https://doi.org/10.1038/s41588-019-0434-7

[2]Wang LR, Wong L, Goh WWB. How doppelgänger effects in biomedical data confound machine learning. Drug Discov Today. 2021 Oct 28:S1359-6446(21)00455-4. doi: 10.1016/j.drudis.2021.10.017. Epub ahead of print. PMID: 34743902.