The reports of doppelganger data

0. Introduction

Due to the widespread availability of doppelganger data, this phenomenon can even have a direct inflationary effect on Machine Learning accuracy. This can probably reduce the usefulness of Machine Learning for phenotype analysis and subsequent identification of potential drug leads. However, there is no standard answer to the question of how to identify the phenomenon and how to eliminate it. Therefore, the focus of the problem should be on how to identify the data for this similar phenomenon, process the data and purify the data for separation. Simply removing similar data would result in a greater loss of valid information, which is clearly not feasible given the limited sample size.

Given paper discusses the causes, identification and implications of the doppelganger effect on the training of machine learning models and suggests that machine learning classifiers will exaggerate the accuracy of the models when highly correlated doppelganger data are present in the training and test sets. The article identifies the doppelganger data in valid cases by setting the maximum value of the negative cases as the threshold and puts them into different classifiers for training. The classifier will exaggerate the training effect of the model when there are more doppelganger data. The article concludes with three suggestions as follows: the article proposes three solutions, one by cross-validation, two by data stratification, and three by robust independence tests on as much data as possible to roughly measure generalizability.

Doppelganger effects manifest themselves as a phenomenon where the training and test sets are highly similar, resulting in an inflated accuracy of the learner. This phenomenon is not only seen in biomedical data but also in stock data. Not only are the data similar, but also the patterns, i.e. the overall trend of the line segments are similar. Although this phenomenon can have a serious impact on the accuracy of the learners in machine learning, fundamentally similar data is a more reliable reference. Based on these ideas, the given paper proposes solutions from three perspectives. The first one is the random sampling of the data, the second one is the improvement of the relevance measure, and the last is the addition of a discriminative algorithm for similar data in the text domain as a reference.

1. Random sampling of data

Based on the consideration of the effective accuracy of machine learning, the best practice is as follows:

[1] The first step used the article to identify doppelganger data, and similar determination criteria were used to filter out the doppelganger data. Dividing the data into non-doppelganger data and doppelganger data.

[2] In the second step, non-doppelganger data and doppelganger data are randomly sampled separately and then composed into a new data set, where a smaller sample of doppelganger data can be taken to further reduce doppelganger effects.

Segmentation like this may bring a degree of robustness due to the randomness and thus reduces the impact of these data.

2. Basic metrics for similar data

2.1 Metric based on the correlation coefficient

2.1.1 Pearson

The Pearson correlation coefficient is calculated as follows, with Cov(x,y) being the covariance:

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

However, the Pearson correlation coefficient has the following assumptions:

[1] The relationship between the two variables is linear and both are continuous data.

[2] The overall of the two variables is normally distributed, or a near-normal single-peaked distribution.

[3] The observations of the two variables are paired, with each pair of observations being independent of the other.

Obviously, the basic Pearson correlation coefficient is relatively demanding, i.e. it works well with data that are normally and continuously distributed and independent of each other, and when the sample distribution is indistinguishable the Pearson correlation coefficient loses its advantage. At this point, the non-parametric idea of Spearman, and Kendall Rank should be introduced.

2.1.2 Spearman

Suppose two random variables are X and Y (which can also be regarded as two sets), both of which have N elements, and the i-th (1<=i<=N) values taken by the two random variables are denoted by Xi and Yi respectively. Sort X and Y (both in ascending or descending order) to obtain two sets x and y, where the elements x_i and y_i are the ranking of X_i in X and the ranking of Y_i in Y, respectively. The elements in the set x and y are correspondingly subtracted to obtain a ranking difference set d, where $d_i = x_i - y_i$, $1 <= \mathrm{i} <= \mathrm{N}$. The Spearman rank correlation coefficient between the random variables X and Y can be calculated from x, y or d, which is shown below.

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}$$

Spearman's rank correlation coefficient is not as stringent as the Pearson correlation coefficient. As long as the observations of the two variables are paired rank ratings or are transformed from observations of continuous variables, Spearman's rank correlation coefficient can be used to determine the correlation, regardless of the overall distribution pattern of the two variables and the size of the sample size.

The calculation of the Kendall correlation coefficient will not be described in detail due to space constraints.

2.2 Distance-based metrics

Therefore, in addition to similarity measures based on correlation coefficients, similarity can also be measured by distance, but the only drawback is that there is no relevant definition of similarity and some experimentation may be required to arrive at a criterion for expressing similarity by distance.

2.2.1 Minkowski Distance

$$Dist(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

When $p = 1$, the above equation becomes the Manhattan distance; when $p = 2$ the above equation is the Euclidean distance; when $p \to \infty$, the above equation becomes the Chebyshev distance.

The disadvantage of these distances is that they do not take into account the dimensionality between the features, and they do not take into account the expectation and variance of the distribution of the features.

2.2.2 Jaccard similarity coefficient

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard coefficient is mainly used to calculate the similarity between individuals on a symbolic or Boolean measure, because the attributes of individuals are identified by symbolic or Boolean values, so it is not possible to measure the size of the specific value of the difference, but only to obtain the result "whether they are the same". The Jaccard coefficient is therefore only concerned with the consistency of features shared by individuals. The Jaccard distance is thus introduced as a measure of distance.

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Jaccard distance measures the differentiation of two sets by the proportion of different elements to all elements in the set.

2.2.3 Cosine similarity

$$\cos(\theta) = \frac{a^T b}{|a| \cdot |b|}$$

Cosine similarity uses the cosine of the angle between two vectors in a vector space as a measure of the magnitude of the difference between two individuals. Compared to the distance metric, cosine similarity focuses more on the difference between two vectors in terms of direction rather than distance or length. The cosine of the angle takes values in the range [-1,1]. A larger cosine of the angle means that the angle between the two vectors is smaller, and a smaller cosine of the angle means that the angle between the two vectors is larger. When the directions of the two vectors coincide the cosine of the angle takes the maximum value of 1, while when the directions of the two vectors are exactly opposite the cosine of the angle takes the minimum value of -1. A major drawback of cosine similarity is that it does not take into account the magnitude of the vectors, but only their directions.

3. Recognition algorithms for doppelganger data

3.1 Shingle Algorithm

The core idea of Shingle's algorithm is to convert the data similarity problem into a set similarity problem. The two main similarity measures for sets are resemblance and containment, which are defined as follows.

$$Rw(f1,f2) = \frac{|shingle(f1,w) \cap shingle(f2,w)|}{|shingle(f1,w) \cup shingle(f2,w)|}$$

$$Cw(f1,f2) = \frac{|shingle(f1,w) \cap shingle(f2,w)|}{|shingle(f1,w)|}$$

When the number of shingles is large, the system overhead, both in terms of memory and CPU resources, is high if all shingles are processed for similarity. There are three main shingle sampling approaches namely Min-Wise, Modm and Mins. The Min-Wise technique is used to reduce the space and time computational complexity of the shingle set by mapping the length *w* of the shingle to an integer value. The Modm technique consists of selecting all the shingles in the same common mapping set as Min-Wise with a modulo *m* of 0. The Mins technique similarly maps the shingle to the set of integers and then selects the smallest *s* elements to form the sampled set. In addition, it is possible to use the hash value of a shingle to represent the shingle for similarity calculation, saving some computational overhead.

3.2 Simhash Algorithm[1]

Shingle's algorithm has high spatial and temporal computational complexity and will be difficult to apply to such problems with large data sets. the core idea of the Simhash algorithm is to represent the feature values of a file with a b-bit hash value and then use the Hamming distance between Simhashes to measure similarity. the Hamming distance is defined as the number of different corresponding bits in two binary sequences. Simhash is calculated as follows.

[1] Initialize a b-dimensional vector V to 0 and a b-bit binary number s to 0;

[2] For each shingle, use the hash function (e.g. MD5, SHA1) to compute a b-bit signature h. For i=1 to b, if the i-th bit of h is 1, add that signature weight to the i-th element of V; otherwise, subtract that signature weight from the i-th element of V;

[3] The i-th bit of s is 1 if the i-th element of V is greater than 0, otherwise it is 0.

[4] Output s as Simhash.

Compared to traditional hash functions, Simhash has the remarkable feature that the more similar the files are, the more similar the Simhash values are, i.e. the smaller the Hamming distance. Obviously, Simhash uses only the b-bit hash value to represent the characteristics of a file, saving a lot of storage overhead; the Hamming distance calculation is simple and efficient, and Simhash uses the Hamming distance to measure similarity, reducing computational complexity significantly. In short, the Simhash algorithm effectively solves the high dimensional space and time computational complexity of Shingle's algorithm by reducing the dimensionality of the file features. However, the accuracy of the Simhash algorithm is also subject to lose and is related to the number of bits of the Simhash, b, the larger the b, the higher the accuracy.

3.3 Bloom filter Algorithm[1]

Similar in nature to the Simhash algorithm, the core idea of the Bloom filter algorithm is also focused on the dimensionality reduction of document features, it uses the Bloom filter data structure to represent feature values. bloom filter is a space-efficient data structure, it consists of a bit array and a set of hash mapping functions. bloom filter can be used to retrieve whether an element is in a set. It has the advantage of being far more spatially efficient and taking far longer to query than the usual algorithms, and the disadvantage of having a certain misidentification rate and deletion difficulties. Using the Bloom filter for similar data detection can compensate for the high computational and storage space overhead caused by applying feature set intersection to calculate document similarity in Shingle, striking a balance between performance and similarity matching accuracy. bloom filter is constructed as follows:

[1] Construct an m-bit bloom filter data structure bf, and initialize all bits to 0,

[2] Two hash functions are selected as mapping functions, hash1, hash2,

[3] For each shingle, apply hash1 and hash2, respectively, and for bf the

corresponding bit position 1,

[4] Output bf as file eigenvalue

In this way, the similarity calculation of two files is converted into the similarity calculation of two bloom filters, with the more similar files having more 1' in common in their bloom filters. The Bloom filter can also be used to measure similarity using the Hamming distance, as well as *Cosine, Overlap, Dice, Jaccard*, etc. The formulas for the last four methods are described as follows.

$$Cosine(x,y) = \frac{\mathrm{dot(x,y)}}{\mathrm{sqrt(|x|.|y|)}}$$

$$Overlap(x,y) = \frac{dot(x,y)}{min(|x|,|y|)}$$

$$Dice(x,y) = \frac{2 * dot(x,y)}{|x| + |y|}$$

$$Jaccard(x,y) = \frac{dot(x,y)}{|x| + |y| - dot(x,y)}$$

Here, $\mathrm{dot(x,y)} = \sum x_i \cdot y_i$, it corresponds to the number of bits in both Bloom filter data structures that are 1 at the same time; |x| denotes the number of bits in the bloom filter data structure that are 1.

Even though the above algorithms are listed, the core of the problem should focus on the determination of similarity[3] and the separation of the data. The separation of data should be the most important core and the starting point for further research.

References

[1] Li, K., Liu, Q., &amp; Fan, C. (2021). Comparison and optimization of text similarity algorithms in data cleaning. Communication Management and Technology.

[2] L. Waldron, M. Riester, M. Ramos, G. Parmigiani, M. Birrer, The Doppelgänger effect: hidden duplicates in databases of transcriptome profiles, J Natl Cancer Inst 108 (2016) djw146.

[3] Han, S., & Sun, L. (2020). A data cleaning method based on improved K-Means clustering and error feedback. *Power System and Clean Energy*.