Supporting document - FSDCQ

Aparna Nayak, Bojan Bozic and Luca Longo

February 12, 2022

Appendix

Preliminaries

A input dataset D can have ‘N’ number of rows, ‘M’ features, and a class variable (K unique classes) as shown in Table 1. Consider that the measures are estimated from a dataset D organized in the feature-value format, as shown in Table 2.

Table 1: Sample dataset format

$\boldsymbol{C_1}$	$\boldsymbol{C_2}$	**...**	$\boldsymbol{C_M}$	**Class**
a_{11}	a_{12}	...	a_{1M}	c_1
a_{21}	a_{22}	...	a_{2M}	c_2
...	...	...	...	...
a_{N1}	a_{N2}	...	a_{NM}	c_K

Table 2: Sample feature set format

Example	$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	...	$\boldsymbol{A_m}$
D_1	a_{11}	a_{12}	...	a_{1m}
D_2	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...
D_n	a_{n1}	a_{2n}	...	a_{nm}

As shown in the input feature file format shown in Table 2, a dataset (D_i) is described by m features. Here the features represents data characteristics and quality. This dataset later is extended to add another feature which represents feature selection algorithm. Thus, the last column later can be considered as label in case of supervised machine learning algorithms.

Dataset Characterization Measures

1. Simple measures

Simple descriptors are properties derived from the attribute-value representation that describe the general characteristics of the datasets. The fundamental input meta-features are considered as follows:

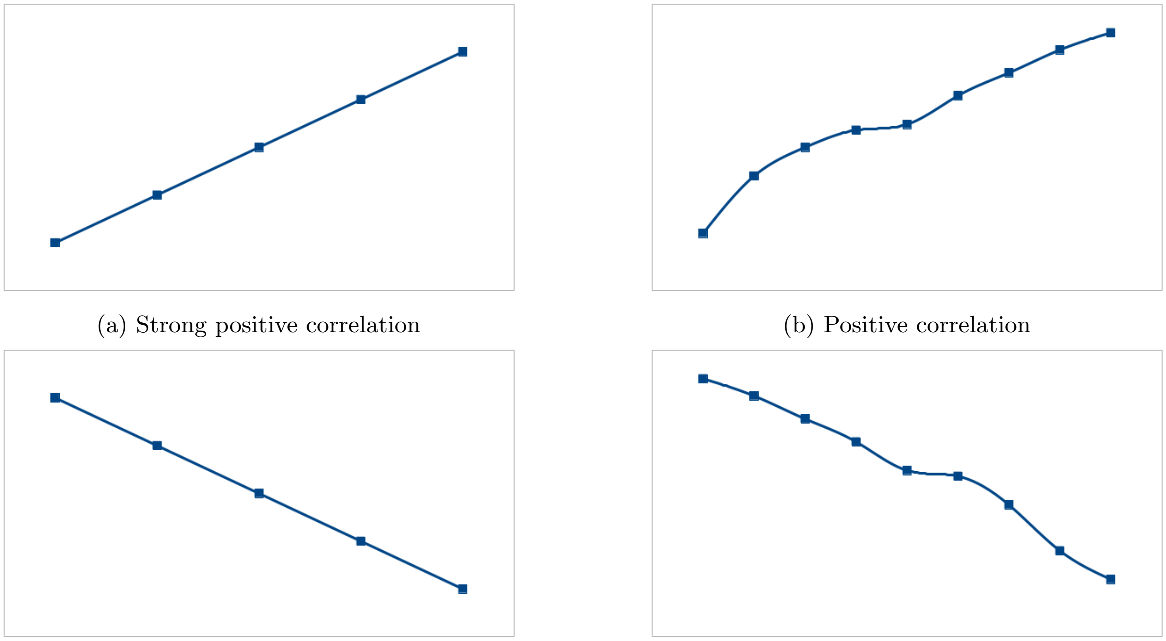
Number of features: Total number of columns (N)

Number of instances: Total number of rows (M)

Number of unique classes: Total number of unique feature classes (C)

2. Statistical measures

One of the most often used methods for characterizing a dataset is to extract input meta-features from a well-established statistical domain, descriptive statistics. Most of the descriptors have a low processing cost and are mostly used to validate the order and distribution of numeric data.



(a) Strong positive correlation

(b) Positive correlation

(c) Strong negative correlation

(d) Negative correlation

Figure 1: Generic categories of correlation

Correlation between attributes: Correlation is a statistical measure that expresses the extent to which two variables are linearly related. Equation 1, $\bar{x}$ refers to the average of the values x_i that constitute the attribute X. Similarly, $\bar{y}$ refers to the average of the values y_i that constitute the attribute Y.

The correlation coefficient is in the range [-1, 1]. The value 1 denotes a strong positive linear relationship, the value -1 represents a strong negative linear relationship. The value 0 represents no linear relationship between the attributes of the interest. The closer the coefficient is to 1 or 1, the stronger the linear association between two random attributes. Figure 1 represents a generic correlation using scatter plots.

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{1}$$

Instead of using the average correlation value, which is irrelevant if the number of positive and negative samples are equal, the proposed model uses the average value in each category of correlation. There are some cases for non numeric data where it is not possible to calculate correlation, rather than excluding such cases another category is added which gives average number of attributes for which correlation is not computed. Average correlation in each category can be computed using the equation 2, where n is the total number of features in the dataset.

$$AvgCorrelation = \frac{Total\ correlated\ instances}{n * (n-1)} \tag{2}$$

As a result, the average correlation between attributes is projected to provide 5 values in table 2.

Implementation: Pandas correlation function neglects non-numeric data, therefore it is required to treat continuous and categorical data differently.

Heuristics : 1. benford's law

Need to come back

Symmetry of the attributes: In statistics, a distribution is said to be symmetric if the mean, median, and mode are all the same. Otherwise, the distribution will become asymmetric. If the right tail is longer, the distribution becomes positively skewed, with mean > median > mode. If the left tail is longer, we get a negatively skewed distribution in which the mean, median, and mode are all negative with mean < median < mode. Considering three different scenarios for symmetrical nature of attributes, in the proposed model 3 features are considered. Karl Pearson's Coefficient of Skewness is used to identify the symmetric nature of the attributes which is shown in equation 3.

$$S_k = \frac{3 \times (\bar{m} - \tilde{m})}{\sigma} \tag{3}$$

3. Information measures

Information-theoretical measures seek to characterize the nominal attributes and their relationship with the class attribute. In the following, we describe the information-based input metafeatures used for implementation.

Class entropy The class entropy of a dataset D, is computed as described in Equation 4. Entropy value reflects the randomness or "impurity" in these partitions. An entropy of zero indicates that the dataset contains only samples from one class; an entropy of one indicates that the dataset contains the maximum amount of entropy possible for a balanced dataset (approximately equal numbers of samples in each class), with values in between indicating levels between these extremes [3].

$$Entropy(\mathrm{D}) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{4}$$

In equation 4, p_i, $1 \leq i \leq n, representstheoccurrenceprobabilityofeach \mathrm{c}_i$ of the class C, given by the ratio between the number of samples that belong to c_i and the total number of examples. A log function to the base 2 is used, because the information is encoded in bits. Entropy(D) is the average amount of information needed to identify the class label of a tuple in D. The information is based solely on the proportions of tuples of each class.

Signal/noise ratio The signal/noise ratio is defined as the amount of ineffectual information of the dataset D. This can be calculated as shown in equation 5. It is computed by subtracting the average mutual information (AMI) between classes and attributes, from the mean attribute entropy (MAE) and then the result is divided by the average mutual information between classes and attributes [4].

$$SNratio(D) = \frac{MAE(D) - AMI(D)}{AMI(D)} \tag{5}$$

Mean attribute entropy (MAE(D)) is average entropy of all the attributes. It is computed as shown in equation 6 where M is number of attributes and A_i is i^{th} attribute of the dataset D.

$$MAE(D) = \frac{\sum\limits_{i=1}^{M} Entropy(A_i)}{M} \tag{6}$$

Mutual information of each attribute A_i with the class attribute C, is calculated to calculate average mutual information (AMI(D)) of the dataset as shown in equation 12.

$$AMI(D) = \frac{\sum\limits_{i=1}^{M} MI(C, A_i)}{M} \tag{7}$$

Conditional entropy can be interpreted as the uncertainty about Y when X is know, or number of bits needed to describe Y when X is known [5].

$$Entropy(Y|X) = \sum_{x \in X, y \in Y} p(x,y) log(p(x,y)) \tag{8}$$

Mutual information (MI(Y,X)) describes the reduction in the uncertainty of Y due to the knowledge of X, and can be described by equation 9.

$$MI(Y,X) = Entropy(Y) - Entropy(Y|X) \tag{9}$$

Equivalent number of attributes The equivalent number of attributes is computed as shown in Equation 10. It is the ration between the class entropy and the average mutual information (AMI) between classes and attributes for a dataset D.

$$ENA(D) = \frac{Entropy(D)}{AMI(D)} \tag{10}$$

Information gain refers to the process of determining which features/attributes provide the most information about the class. It is based on the concept of entropy and is used to quantify the decrease in entropy as a function of the values of a given attribute A as shown in Equation 11.

$$Informationgain(D) = Entropy(D) - Entropy(D, A) \tag{11}$$

Therefore, average mutual information for a dataset D with M attributes can be calculated as denoted by Equation 12.

$$AMI(D) = \frac{\sum_{i=1}^{M} Informationgain(A_i)}{M} \tag{12}$$

4. Quality measures

All the quality metric values range from 0 to 1. The value 0 for any metric is considered as good.

1. Classification quality metrics

Class overlap: Class overlap occurs when instances of more than one class share a common region in the data space. This can be computed using One-class SVM [2], clustering based on under sampling [1], k-means [6]. The proposed model makes use of k-means algorithm to identify class overlap as shown in algorithm 1. Centroids of each cluster is identified using k-means algorithm. If the distance between centroid and the data point is greater then $mean + 3 * stddev$, new distance is calculated with remaining centroids to check whether the data point overlaps with other clusters.

Algorithm 1 Class overlap

```
$total\_clusters \leftarrow total\_unique\_classes$
Apply K-means on the dataset.
Compute mean and standard deviation (stddev) for each cluster
for <each datapoint in the dataset> do
    $distance \leftarrow$ distance between datapoint and cluster
    if $distance \geq (mean + 3 * stddev)$ then
        for <remaining clusters> do
            $distance \leftarrow$ distance between datapoint and cluster
            if $distance \geq (mean + 3 * stddev)$ then overlap+=1
            end if
        end for
    end if
end for
$overvlap\_perc$ = overlap / $(total\_rows(dataset) * total\_columns(dataset))$
```

Outlier detection: Outliers are extreme values that deviate from other observations of the as to arouse suspicion that they were generated by a different mechanism. Outliers are detected using K-means algorithm as shown in algorithm 2.

Class imbalance ratio: Imbalanced classification is the problem of classification when there is an unequal distribution of classes in the training dataset. This is computed by considering percentage of number of observations in each class. Percentage of additional samples in majority class is calculated to that of minority class as shown in equation 13. c_i represents i^{th} class of the dataset.

$$total(c_i, c_{i+1}) = \begin{cases} (perc(c_i) - perc(c_{i+1}),\ if\ (perc(c_i), perc(c_{i+1}))\ \geq\ 30 \\ 0,\ otherwise \end{cases} \tag{13}$$

Algorithm 2 Outlier detection

```
$total\_clusters \leftarrow total\_unique\_classes$
Apply K-means on the dataset.
Compute mean and standard deviation (stddev) for each cluster
for <each row in the dataset> do
    $distance \leftarrow$ distance between datapoint and cluster
    if $distance \geq (mean + 3 * stddev)$ then
        $outlier+ = 1$
    end if
end for
$outlier\_perc$ = outlier / $(total\_rows(dataset) * total\_columns(dataset))$
```

2. Intrinsic quality metrics

Completeness: Completeness refers to the degree to which required data are in the dataset. Equation 14 and 15 shows calculations that is used to identify completeness of the dataset. Here r_i refers to row, c_j refers to column. m and n refers to total number of rows and columns in the dataset.

$$nullValue(r_i, c_j) = \begin{cases} 1, \; if \; (r_i, c_j) \; = \; null \\ 0, \; otherwise \end{cases} \tag{14}$$

$$Completeness = \frac{1}{m * n} \sum_{i=1,j=1}^{i=m,j=n} nullValues(r_i, c_j) \tag{15}$$

Conciseness: Conciseness metric refers to uniqueness nature of the dataset. This metric ensures there are no duplication of the data values and is measured against all the records within a dataset. Equation and shows calculation of conciseness of the dataset.

$$Equalrows(r_i, r_{i+1}) = \begin{cases} 1, \; if \; (r_i == r_{i+1}) \; = \; True \\ 0, \; otherwise \end{cases} \tag{16}$$

$$Conciseness = \frac{1}{m} \sum_{i=1,j=1}^{i=m} equalrows(r_i, r_{i+1}) \tag{17}$$

Conciseness can also be calculated using k-columns validation. Here, k-columns are neglected to compare remaining values of the row. k can be any integer ranging from 1 to (n-1) where n is total number of columns.

Accuracy: Accuracy describes the degree to which the data correctly describes real world objects and confirms with a verifiable source. Accuracy verification can be syntax as well as semantic [?]. One of the metric to evaluate the syntax validity is detection of ill-typed literals. Regular expressions are used to identify the type of a column and values that do not conform the regular expressions are treated as ill-typed literals.

Algorithm 3 Accuracy

```
$invalid \leftarrow 0$
for <each column in the dataset> do
    Check datatype of column
    if $datatype = object$ then
        $Check\ datatype\ of\ each\ value\ using\ regular\ expressions$
        $Assign\ the\ maximum\ occurrence\ of\ datatype\ to\ the\ column$
    end if
    if $maxoccurrence \neq total\_rows$ then
        $invalid + 1$
    end if
end for
$invalid\_perc =$ invalid / $total\_columns(dataset))$
```

References

[1] B. Das, N. C. Krishnan, and D. J. Cook. Handling class overlap and imbalance to detect prompt situations in smart homes. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 266–273. IEEE, 2013.

[2] D. Devi, S. K. Biswas, and B. Purkayastha. Learning in presence of class imbalance and class overlapping by using one-class svm and undersampling technique. *Connection Science*, 31(2):105–142, 2019.

[3] J. Han, J. Pei, and M. Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[4] A. Kalousis. *Algorithm selection via meta-learning*. PhD thesis, University of Geneva, 2002.

[5] A. Orlitsky. Information theory. In R. A. Meyers, editor, *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 751–769. Academic Press, New York, third edition edition, 2003.

[6] N. H. Shrifan, M. F. Akbar, and N. A. M. Isa. An adaptive outlier removal aided k-means clustering algorithm. *Journal of King Saud University-Computer and Information Sciences*, 2021.