Adaptive algorithm of classification on the missing data

Alexander V. Medvedev[1], Daniil A. Melekh[1],
Natalia A. Sergeeva[2], Olesya V. Chubarova[1]
[1] *Siberian Federal university, Krasnoyarsk, Russia*
[2] *LLC Rd-science, Krasnoyarsk, Russia*
e-mail: danilmelekh@gmail.ru,
n.sergeeva@rd-science.com,
kuznetcova_0@mail.ru

Abstract

The problem of classification by data with gaps, bypassing the stage of their filling, is considered. An adaptive restructuring of algorithms is proposed as a result of the introduction of corresponding indicators into them. The indicators take into account the flow of current information, on the basis of which a decision is made to change the algorithm and the data processing technology itself at each cycle. Computational procedures are based on non-parametric estimation, are given their settings and the results of numerical modeling.

Keywords: supervised learning, missing data, adaptive algorithm, non-parametric estimation of probability density, smoothing window, kernel function, numeric and nominal features.

Introduction

When solving practical problems, the fact of missing values in real data has traditionally been the case. It is possible to solve the problem of processing gaps in the data using different techniques, among which are both suggestions to form training samples only from completely filled objects and fill in the missing values with various approaches and methods. Any initial information about the object is of great value for the researcher, therefore, they most often resort to recovering the missing data, which is already a traditional stage of data preprocessing [1, 3].

The authors of the article were engaged in solving an applied problem related to the classification of objects with a teacher, having a very small sample of data, obtaining of which is also slow. Differences between objects affect the shift of their statistical characteristics in each class. Due to the bias of statistical evaluations of objects of different classes, the choice of tactics for restoring gaps in a training sample for building a classifier and in a new object entering to determine its class is difficult [2, 3].

Also, it was not possible to search for dependencies between features to fill in the gaps due to the smallness of the samples. Therefore, an adaptive classification algorithm was developed that will be able to process data with gaps without a procedure for filling them. The article presents the essence of the algorithm, identifies the conditions for its use, provides the results of numerical experiments on simulation data and widely known data (Fisher's Iris data set).

The adaptive nature of the classification algorithms is expressed in the reshaping of the training sample from the original, depending on the set of filled features. The computational algorithm changes at each step of the iterative procedure, depending on the completeness of the current information.

To construct adaptive classification algorithms, a modification multidimensional non-parametric probability estimate of the Rosenblatt-Parzen is applied [4].

1 Problem formulation

Let there be a set of objects $\{O_i, i = \overline{1,s}\}$, where s, which are described by a known set of features is the sample size $\{p_i, i = \overline{1,n}\}$, measured in numeric (n_1) and nominal (n_2) scales: $n_1 + n_2 = n$. . For each object there is an indication of the label (i.e., class): $O_i \in Z_l l = \overline{1,L}\}$, L - number of classes. We denote feature measurements for each object with a set of values $\{\left(z_i, x_i^j\right), i = \overline{1,s}, j = \overline{1,n}\}$, where x_i^j - value of p_j feature at O_i object, z_i - class, s_l - number of class objects Z_l, $\sum_{l=1}^{L} s_l = s$.

Object feature measurements contain omissions. It is necessary to build a classification algorithm that operates with data that contains gaps without a process of their filling, and to develop procedures for setting parameters.

2 Classification algorithm for objects with missing data

For each classified object O_t it is necessary to evaluate its belonging to each class. This procedure involves a comparison with the objects of the training set of each class. The basic idea of the algorithm is to use for evaluation only a set of features O_t that have initial values p. At each t step of the algorithm, the entire training set must be re-sorted relative to the existing set of features p of O_t, presented for classification. Then the size of the training data set may change: $\sum_{l=1}^{L} s_l^t = s^t \leq s$, where s_l^t - class data set Z_l, s^t - total data set after selection of non-empty attributes. Since the set of attributes for assessing the similarity with objects of each class will change, then we denote the number of numeric and nominal features n_1^t and n_2^t, respectively.

Let us demonstrate the idea visually using the example of 3 classes. In the tables below, the filled attributes are highlighted in gray, the features with missing values are displayed without highlighting.

Table 1: Features p of O_t without specifying a teacher

1	2	3	4	5	6	7	8	...	$p_{n_1+n_2}$
▨		▨		▨		▨	▨		

Table 2: Initial selection of objects with the teacher

№	Class	1	2	3	4	5	6	7	...	$p_{n_1+n_2}$
1	z_1	▨	▨	▨		▨	▨			▨
2	z_2	▨	▨		▨		▨	▨		▨
...	...									
s	z_s	▨	▨	▨	▨	▨		▨		▨

Subsequent lines of the training set are formed similarly.

For each class, we enter the following value:

$$\alpha_l^t = \sum_{i=1}^{s_l^t} \prod_{i_1=1}^{n_1^t} \Phi\left(\frac{x_t^{j_1} - x_i^{j_1}}{C^{j_1}}\right) \prod_{j_2=1}^{n_2^t} 1(x_t^{j_2}, x_i^{j_2}), \quad l = \overline{1, L}, \tag{1}$$

which constructively repeats the Rosenblatt-Parzen multidimensional nonparametric estimator of probability density. As a bell-shaped function in (1), a triangular kernel, a truncated parabola, a Gaussian kernel, cos, the Sobolev function, and others can be used for features on an numeric scale. For nominal features, the Kronecker delta indicator is used:

$$1(x_t^{j_2}, x_i^{j_2}) = \begin{cases} 1, & \text{if } x_t^{j_2} = x_i^{j_2} \\ \delta_{j_2}, & \text{if } x_t^{j_2} \neq x_i^{j_2} \end{cases}. \tag{2}$$

where δ_{j_2}, $0 < \delta_{j_2} < 1$ — some threshold value for each feature. The specific value of the threshold is selected in the process of training the classifier. Kernel functions satisfy the convergence conditions for nonparametric estimates and are discussed in detail [4].

If $n_1^t = 0$ or $n_2^t = 0$, then the resulting value of the product is limited to some threshold value from the bottom of the entire product, in order to preserve the non-zero value of the other indicators. The more objects differ in the values of the nominal features, the closer the whole work tends to zero, reducing the total weight of the influence of points on the assignment of class values. But at the same time, the difference in only one attribute does not reduce the magnitude of the assessment of the general belonging of an object to a class. A lower bound on the result of the entire work is introduced:

$$0 < \delta < \prod_{j_2=1}^{n_2^t} 1(x_t^{j_2}, x_i^{j_2}) < 1. \tag{3}$$

The smoothing parameter is a vector (by the number of quantitative features). The optimization procedure for setting the parameters of the algorithm is carried out according to the number of points k under the bell-shaped function, the threshold values δ, δ_{j_2}.

The value α_l^t estimates the belonging of an object O_t to a class Z_l according to the initial set. For each new object O_t presented for classification, the volume of the training set for calculating relation (1) will vary due to a different set of unfilled features. Thus, the number of items to be calculated within one class of objects will change. This fact reflects the adaptive nature of the computational procedure, which uses for each object arriving to the classification the newly formed training set from the original.

The next step for deciding whether an object belongs to a particular class is the calculation of normalized estimates β_l^t based on the calculated ones α_l^t:

$$\beta_l^t = \alpha_l^t \Big/ \sum_{l=1}^{L} \alpha_l^t. \tag{4}$$

The probability that an object belongs to a class is proportional to the relative assessment of belonging. The closer it is to 1, the higher the probability of the truth of this class. This can be formulated as follows:

$$O_t \in Z_k | \quad \beta_k^t = \max_{l=\overline{1,L}}(\beta_l^t) \tag{5}$$

As a quality criterion of the classifier, an estimate of the area under the ROC-curve (AUC) is used.

3 Numerical experiment

The numerical study of the algorithm was carried out on three data sets. The first data set corresponded to two non-intersecting classes in the three-dimensional attribute space. The second data set had two intersecting classes in the space of 7 features: 2 in nominal and 5 in numeric scales. To assess the accuracy of the classification, cross-validation using the Monte-Carlo method was used to divide the training and test samples. The calculation results contain the AUC mean value for each sample.

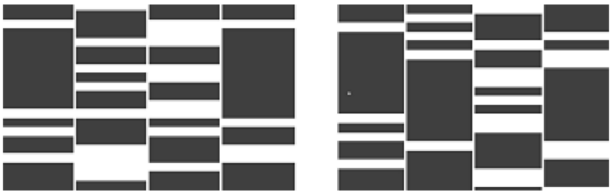

Figure 1: Example of a sample with random gaps

The first sample demonstrates how the algorithm works under favorable conditions. This sample was used with and without gaps. In the case of gaps, two approaches were used: random imitation of gaps (Fig. 1, left side) and random imitation of sequential pairs with disjointed gaps (Fig. 2, right side). The difference of these approaches is shown in Figure 1. Gaps were introduced in each class in equal numbers, similarly by features.

The share of gaps in both variants was 10% for each attribute, class. Table 1 contains the results of setting the number of points under the kernel function, δ due to the small number of features was set as following: $\delta = \delta_{j_2}^{n_2^t}$, $\delta_{j_2} = 0,01$.

Table 3: First data set, non-intersecting classes

k	AUC			
	No gaps	Random gaps	sequential pairs of disjointed gaps	filled with a median
5	1	0.89	0.9096	1
10	1	0.991	0.994	1
20	1	0.99997	0.9999	1
30	1	0.99956	0.9995	1
Max	1	0.99845	0.9987	1

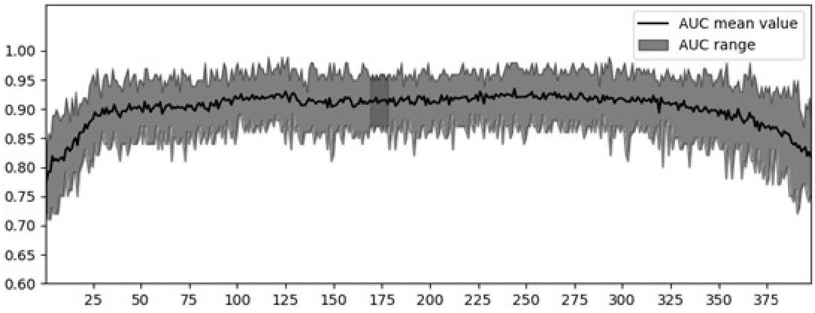

Figure 2: Dynamics of changes in AUC depending on k

The second data set with the intersection of classes contained about 7% of volume of the initial sample of common points in the intersection area. The results of the classification for this sample are presented in Fig. 2, which shows the dynamics of AUC (range and average value) depending on the number of points k under the nuclear function. The average value of accuracy has a stable position, and taking into account the spread of AUC, the best values of k are in the range [169; 178].

The Fischer's Iris appears as the third data set. Table 2 shows the results of the algorithm on accuracy of two Iris classes. Classes with maximum intersection of versicolor and virginica are selected.

To compare the accuracy of the algorithm under the same conditions, in addition to Table 5, the algorithm was tested on a sample with gaps filled with a median by

Table 4: The Fischer's Iris without gaps

k	AUC minimum value	AUC maximum value	AUC mean value
1	0.55	1	0.8358
2	0.59	1	0.84305
3	0.7	1	0.91355
4	0.7	1	0.9141
5	0.7	1	0.9225
6	0.75	1	0.92725
7	0.75	1	0.9338
8	0.8	1	0.9428
9	0.8	1	0.9528
10	0.8	1	0.9563
11	0.8	1	0.95525
12	0.8	1	0.95435

class (Table 6). The filling was done before the stage of dividing the data set into training and test sets, simulating the process of analyzing the labeled data. But in real conditions for newly received unlabeled data it is not possible to use the principle of filling with a median by class.

During the launches of the algorithm, with a gap proportion of 0.1, its accuracy deteriorated by 0.02035 relative to the sample with gaps filled with the median by class and the original sample (Table 6). However, medians for the Fischer's Iris artificially overestimate the accuracy. They contribute to the separation of classes in the original space, because the medians differ from each other. This effect is more clearly observed with an increase in the median filling percentage. Also Table 3 shows the change in the accuracy of the algorithm with different percentage of gaps for each feature. As a result, the algorithm was able to maintain an accuracy of 0.9 under the conditions of 120 gaps from 400 values.

Conclusions

The proposed algorithm can be applied to solve the classification problem if the initial data have gaps. In this case, the algorithm does not require the step of filling the gaps, therefore, its result is not affected by the bias of estimates of the restored gap values. A comprehensive study of the algorithm features requires a comparison with traditional approaches. Also, the further development of the algorithm is supposed to be directed to the formulation of criteria for identifying significant features or their sets for the classification with the teacher in case of having gaps in initial data.

Table 5: First data set, non-intersecting classes

k	Average AUC		
	Percentage of gaps 0.1	Percentage of gaps 0.2	Percentage of gaps 0.3
1	0.70205	0.6639	0.6726
2	0.7374	0.691	0.711
3	0.8847	0.80545	0.80595
4	0.8897	0.81185	0.8336
5	0.9075	0.8474	0.88205
6	0.9136	0.85805	0.8869
7	0.93275	0.87895	0.8969
8	0.9344	0.891	0.89745
9	0.93575	0.9078	0.8987
10	0.93595	0.9155	0.9008
11	0.93785	-	-
12	0.93745	-	-

References

[1] Garcia-Laencin P.J., Sanch-Gomes J.L., Figueiras-Vidal A.R. (2009). Pattern classification with missing data/ *Neural Comput & Applic*, DOI 10.1007/s00521-009-0295-6, Springer-Verlag.

[2] Li Y.Y., Parker L.E. (2008). Classification with missing data in a wireless sensor network *IEEE SoutheastCon 3-6 April*, pp. 533-538.

[3] Zhang S., Jin Z., Zhu X. (2011). Missing data imputation by utilizing information within in-complete instances *Journal of Systems and Software*. Vol. **84**, pp. 452-459.

[4] Härdle W, Linton O. (1994). Applied Nonparametric Methods *Cowles foundation for re-search in economics at Yale university*. 44 p.