Anchored Bootstrap

Vadim Efimov
ICG SB RAS, Novosibirsk, Russia
NSU, Novosibirsk, Russia
vmefimov@ngs.ru

Kirill Efimov
IHNA&NPh RAS, Moscow, Russia
efimov@phystech.edu

Vera Kovaleva
ISEA SB RAS, Novosibirsk, Russia
vkova@ngs.ru

***Abstract* – A new bootstrap algorithm is proposed. Unlike in the naive bootstrap, we assume that objects form a certain geometric configuration, are fixed in the Euclidean space (“anchored”), and their relative positions cannot be changed. This model embraces the time series, molecular sequences, and many other situations.**

Keywords – geometric configuration, time series, molecular sequences.

I. Introduction

Let us assume there is an “object–variable” matrix X of size N×M. A certain quantitative characteristic $\alpha = \alpha(X)$ is determined for this matrix. Table X is susceptible to some perturbations that might affect α. An algorithm to estimate the scale of perturbations of characteristic α needs to be elaborated.

If the number of matrices X were sufficiently large, their set could be regarded as general distribution approximation. By determining the characteristic α for each matrix, it would be possible to obtain its empirical distribution and all the distribution parameters. The problem is that ~~sometimes~~ the scale of perturbations needs to be estimated when there is only one matrix X.

For what applications is this algorithm needed? First of all, it can be used in all statistical problems where the validity of the results is assessed. It is usually assumed that rows x_i of matrix X are independent realizations of some random variable with an unknown general distribution F(x) that is the same for all rows. The characteristics can include the average values, variances, correlation coefficients of variables and correlation matrices, coefficients in regression equations of variables, predicted values for some variables depending on other ones, etc. The scale of perturbations is estimated using confidential intervals of characteristics at a certain significance level (p-value). In the classical problem setting, it is additionally assumed that the type of distribution F(x) is also known (typically the multivariate normal distribution), and only its parameters are unknown. The bootstrap approach can be used for any distribution F(x) regardless of its type and parameters. Due to this advantage, bootstrapping is widely used in modern research.

Therefore, let us consider the bootstrap approach [1]. In the naive implementation, a bootstrap copy of matrix X is formed according to the following rule: a random row is sampled N times from matrix X with a probability of 1/N and placed into the bootstrap copy X* of matrix X, while the original row is left on its place (*sampling with replacement*). Some rows will be sampled more than once, while others will not be sampled at all. Basically, the empirical distribution function (EDF) of rows x_i in the original matrix X, $F_X(x)$, is assumed to be a general function for rows. Therefore, no assumptions regarding the type and parameters of general distribution are required. All the information needed is taken from the original matrix.

Bootstrap sampling is repeated many times. The pool of bootstrap samples is regarded as a model of general distribution for the entire matrix X.

In naive bootstrapping, rows x_i are placed into X* in a random manner. If the computed characteristic α is invariant with respect to the order of rows in the matrix and allows permutation of rows or their replacement within the matrix, the bootstrap is working well.

If the initial assumptions are knowingly not fulfilled, this problem setting is invalid. For example, for time series it would be natural to expect that occurrence of an event strongly depends on the combination of preceding events. In the problems of molecular phylogenetics, rows x_i of matrix X correspond to text strings (nucleotide or amino acid sequences), while the parameters of phylogenetic trees (dendrograms including *every* row x_i) are the computed characteristics. Therefore, the hypothetical general distribution, as well as the assumption that rows of the original matrix are independent random realizations, should not be used. These rows are very likely to form a certain structure, and it is highly desirable to retain it during the bootstrapping process.

II. Algorithm

We assume that rows x_i are points in a multidimensional Euclidean space. However, unlike in the naive bootstrap, we will consider that these points form a certain geometric configuration, are fixed in the space (“anchored”), and their mutual arrangement must not be altered. This model embraces many situations, such as the time series or molecular sequences, etc.

But how should be bootstrapping performed in this case? Two classical bootstrap approaches are used.

<u>The first approach</u>. We assume that weight p_i equal to 1/N is assigned to each row x_i [1].

<u>The second approach</u>. When generating a next bootstrap sample, there is no need to physically sample a random row from the matrix. It can simply be specified how many times each row x_i was sampled. It is sufficient to use a single weight column B with length N for this purpose [2]. The sum of b_i obviously equals N. Of course, some b_i can be null.

The difference with naive bootstrap is not only that space for the bootstrap samples is saved. In naive bootstrap, some rows are not placed into a given bootstrap copy, while the remaining rows x_i are randomly placed into X*, which is an equivalent of additional permutation of these rows. If the geometric configuration corresponding to the original matrix contained any relevant information (e.g., information showing the similarity between the rows like it is in molecular phylogenetics or the sequence of their occurrence like it is in time series), this information is destroyed in the bootstrap sample when naive bootstrapping was performed. In the anchored bootstrap, the unclaimed rows are assigned zero weight, but the information remains in them and can be used for computing the characteristics of the matrix. Assigning weights does not alter the geometric configuration. Therefore, the scope of application of the anchored bootstrap is significantly broader compared to the native bootstrap.

For a set of points in the Euclidean space, it is always possible to compute a pairwise distance matrix. The converse is also true: the coordinates of points in a Euclidean space can be computed from the matrix of Euclidean distances between these points [3].

III. Properties

Properties of the anchored bootstrap are the same as those of naive bootstrap if the assumption that rows of the original matrix are independent random realizations is fulfilled. In this case, it is just a technique facilitating and simplifying the practical realization. It goes without saying that any quantitative characteristic $\alpha(X)$ should be replaced with its weighted analog (weighted average, weighted correlation, etc.).

IV. Cluster analysis

Let us consider the possible method for using anchored bootstrap in cluster analysis. We assume that rows x_i of matrix X are descriptions of the objects, with the Euclidean distances measured between each pair of objects. Principal components can always be computed for the Euclidean distance matrix by principal coordinates analysis [3]. Let there exist a set of clusters obtained using *any* method. Each cluster K is a subset of points x_i from a point cloud in a multidimensional Euclidean space and can be assigned by the Boolean membership vector A_K. Elementwise multiplication of A_K by the weight column B (B*) for matrix X (X*) yields weight columns B_K (B_K*) for each cluster both in the original matrix X and in its bootstrap samples X*. Therefore, we can compute various "weighted" characteristics of cluster K: centroid, variance, eugenvalues of the covariance matrix, the average and root-mean-square distances [4] within and between the clusters, etc., and the empirical distribution of each characteristic within the set of all bootstrap samples X*, which is the required result.

V. Example

Let us take the file Drosophila_Adh.meg from the folder Examples of the MEGA X software suite as the initial matrix X [5]. The evolutionary history was inferred using the UPGMA method [6]. Nine clusters were obtained (Fig. 1). The percentage of replicate trees in which the associated taxa clustered together in the bootstrap test (1,000 replicates) are shown next to the branches [2]. The evolutionary distances were computed using the p-distance method [7] and are shown in the units of the number of base differences per site. This analysis involved 11 nucleotide sequences without gaps and missing data. Codon positions included were 1st+2nd+3rd+Noncoding. There were a total of 762 positions in the final dataset. Evolutionary analyses were conducted in MEGA X [5].

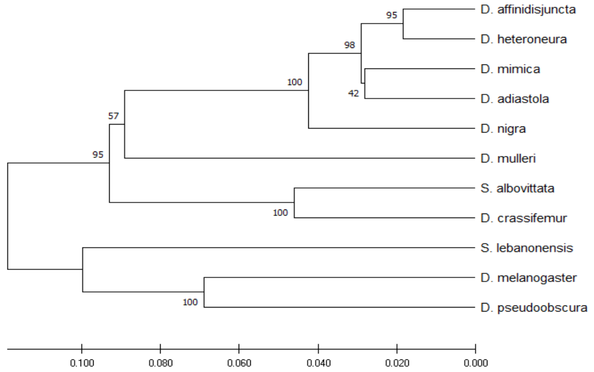

Fig. 1. A phylogenetic tree Drosophila_Adh.meg [5] with Felsenstein bootstrap support [2].

Fig. 1 implies that the pool of objects contains three core points, with the remaining objects being more or less clustered around them. The arrangement of objects on the PC1–PC2 plane (50.9% of the total variability) generally supports this situation (Fig. 2). However, the actual situation is more complex; these objects lie rather far away from the core points and in fact are appreciably independent (Fig. 3) (22.5% of the total variability).

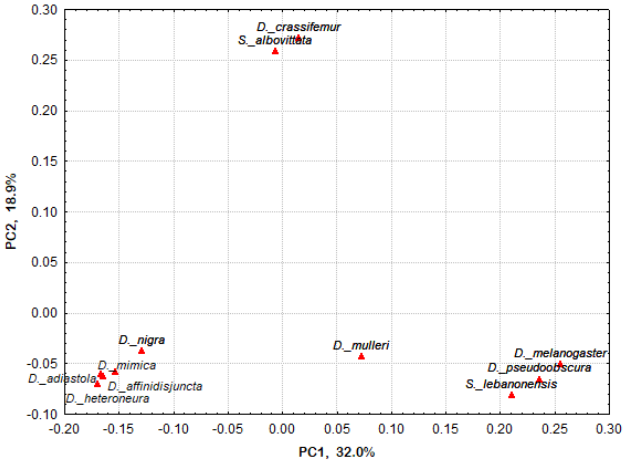

Fig. 2. Configuration of objects on the PC1–PC2 plane.

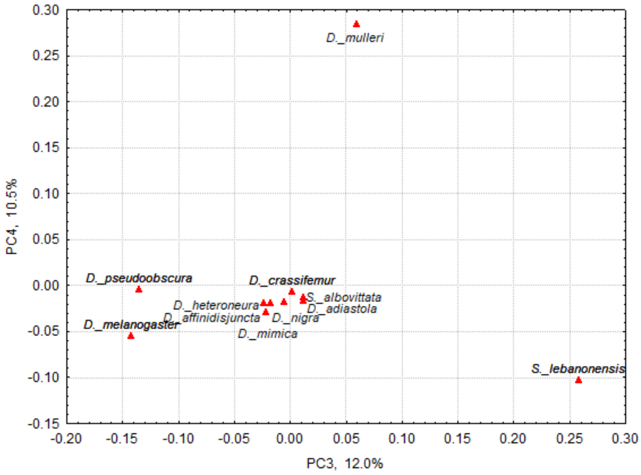

Fig. 3. Configuration of objects on the PC3–PC4 plane.

Thus, there arises a question regarding the validity of the resulting classification and specific criteria that should be used to assess cluster stability when the original matrix is exposed to perturbations. We emphasize that the classification and set of clusters are considered already specified, so only their stability is estimated. Since a hierarchical classification is being analyzed, for any two clusters there are only two variants: (1) one cluster contains another one or (2) these clusters are disjoint. In this study, we focus on the following criterion of a “good” cluster: the average distance between the objects within a cluster is shorter than the average distance between this cluster and any disjoint cluster within the classification being discussed.

In each bootstrap replicate, each object is assigned a random weight. Let us assume that the weight of a weighted distance between two objects is equal to a product of their weights. The weighted average intra- and intercluster distances will change; the intracluster distance can become longer than the intercluster one, and the cluster will no longer be “good”. The percentage of successful bootstrap replicates can be determined by repeating this process a sufficient number of times.

It was revealed during the computations that the proposed criterion sometimes cannot be determined. If cluster weight is zero, the cluster does not exist. If its weight is 1, the cluster contains a single object, but the intracluster distance still does not exist. Therefore, it is more reasonable to determine the percentage of successful bootstrap replicates only for the bootstrap tests where there are both average distances and this criterion can be computed.

VI. Results

Table 1 summarizes the results of computing the bootstrap supports (1000 replicates). All clusters turned out to be more or less “good”.

Table 1. Clusters and bootstrap supports for them (1,000 replicates)

Species	Cl1	Cl2	Cl3	Cl4	Cl5	Cl6	Cl7	Cl8	Cl9
D._melanogaster	0	0	0	0	0	1	0	0	1
D._pseudoobscura	0	0	0	0	0	1	0	0	1
S._lebanonensis	0	0	0	0	0	0	0	0	1
S._albovittata	0	0	0	0	1	0	0	1	0
D._crassifemur	0	0	0	0	1	0	0	1	0
D._mulleri	0	0	0	0	0	0	1	1	0
D._affinidisjuncta	1	0	1	1	0	0	1	1	0
D._heteroneura	1	0	1	1	0	0	1	1	0
D._mimica	0	1	1	1	0	0	1	1	0
D._adiastola	0	1	1	1	0	0	1	1	0
D._nigra	0	0	0	1	0	0	1	1	0
Is disrupted	169	208	217	224	148	154	224	117	246
Does not exist	234	241	12	1	255	249	1	0	89
Is accomplished	597	551	771	775	597	597	775	883	665
Support, %	78	73	78	78	80	79	78	88	73

VII. Discussion

Let us mention a very important property of the anchored bootstrap. It allows one to *avoid* bootstrapping across the *variables* themselves [2]. It is sufficient to have an object distance matrix generated either using *variables* or without them.

Why is it important? The classical (naive) bootstrap method was invented by B. Efron (1979) [1]. He proposed to perform bootstrapping for objects, since objects can independently and randomly occur in a sample. Therefore, robust confidence intervals can be obtained for many sample characteristics.

Molecular biology arose in the middle of the latter half of the 20th century. Scholars started applying cluster analysis to nucleotide sequence pools and building phylogenetic trees. So the confidence estimation methods had to be found. The conventional methods have failed from the very beginning: the phylogenetic trees are too complex structures to apply normal distribution for them. For this very reason, although cluster analysis has existed for more than a century, it still has not been used in mathematical statistics and is regarded simply as a set of heuristic algorithms.

Bootstrap was found to be a more suitable candidate. However, there were several seemingly intractable obstacles that made it impossible to directly apply bootstrapping for clusterization. Bootstrap analysis assumes that a set of analyzed objects is a random independent sample drawn from some general distribution. A phylogenetic tree is typically based on the assumption that all the analyzed objects have not randomly occurred, but rather evolved from a hypothetical common ancestor. The phylogenetic tree shows the history of their emergence and divergence.

Furthermore, the bootstrap method was intended for sets of numbers, while molecular biologists are dealing with text strings. In each bootstrap sample, some objects are lost (~ 1/3 of the sample), while the remaining objects are shuffled. This is absolutely unacceptable for phylogenetic trees. A bootstrap sample of a phylogenetic tree must retain all the objects and not disrupt their mutual proximity.

In 1985, J. Felsenstein elegantly overcame all of these difficulties [2]. He suggested performing bootstrap by variables instead of objects (in this case, by positions in nucleotide sequences). In addition, he suggested that bootstrap support (the percentage of each cluster of the original tree in all bootstrap samples) should be also computed. Furthermore, J. Felsenstein has elaborated a software tool capable of doing it and started sending it to everyone interested.

From the very beginning, this approach seemed rather dubious when applied to variables. Even if variables are sufficiently homogeneous (i.e., have the same dimension or are dimensionless), they still are intercorrelated. Therefore, variables cannot be independent. For this very reason, a heated debate on this topic has lasted several years.

However, B. Efron, the author of the bootstrap method, supported J. Felsenstein in 1996 [8]. The discussion was immediately stopped. Today, it is the gold standard of data analysis in this field: “The usefulness, simplicity and interpretability of this method made it extremely popular in evolutionary studies, to the point that it is generally required for publication of tree estimates in a wide variety of domains (molecular biology, genomics, systematics, ecology, epidemiology, etc.). Felsenstein’s article has been cited more than 32,000 times and is ranked in the top 100 of the most cited scientific papers of all time.” [9].

Felsenstein's approach has been extended to any distance matrices and added to many statistical packages. Today, not only molecular biologists are expected to compute bootstrap supports, but zoologists and botanists (i.e., researchers dealing with the regular quantitative variables that are usually intercorrelated) as well. Field biologists deal with much less variables than molecular biologists do; therefore, the bootstrap support is rather weak in this case. Building evolutionary trees also lies beyond the scope of their work, and cluster analysis is used only to show similarity between the objects. Therefore, field biologists absolutely do not understand why they should compute bootstrap supports. (In fairness, it should be noted that J. Felsenstein himself believes that bootstrap by variables should be used only for molecular sequences and discrete morphological characters – personal communication).

The use of variables means that the bootstrap results obtained for the same distance matrix will depend on which variables are used for its computation. Let us conduct the following experiment. We take the same file Drosophila_Adh.meg, increase its length via concatenation, and perform the same bootstrap analysis. The distances remained the same, but the bootstrap support increased. Now we duplicate the objects three times to analyze 33 rows instead of 11 rows. Nothing has changed: neither the distance matrix nor the bootstrap support. But this situation is not correct.

We believe that bootstrap analysis of the distance matrix should use only the distance matrix itself as suggested in this study.

Of course, this is just the beginning. The proposed idea can be developed in several directions. The idea of a “good” cluster was adapted from [10], but there is no reason not to test other variants as well. Bootstrapping is very useful for performing decomposition of time series or non-numerical sequences [11], etc., into principal components.

Researchers should not limit themselves to polynomial weights as it is for naive bootstrap. If it is allowed to average rows in smoothed bootstrap, smoothing can also be performed for anchored bootstrap. It is easy to see that the smoothed bootstrap method already employs the idea of geometrically arranging objects on a numeric axis. It is fair to add that this idea originates from the empirical distribution function. To feel the difference, it would be enough to imagine how an EDF of a circular data set will look like.

Acknowledgments

This work was supported by the Russian Foundation for Basic Research [#19-07-00658-a] and by the Russian State Budget Project [#0324-2019-0040-C-01].

References

[1] B. Efron, “Bootstrap Methods: Another Look at the Jackknife,” The Annals of Statistics, vol. 7, pp. 1–26, 1979.

[2] J. Felsenstein, “Confidence limits on phylogenies: an approach using the bootstrap,” Evolution, vol. 39, pp. 783–791, 1985.

[3] J. C. Gower, “Some distance properties of latent root and vector methods used in multivariate analysis,” Biometrika, vol. 53, pp. 325–338, 1966.

[4] P. L. Odell, B. S. Duran, Cluster Analysis: A Survey. Heidelberg Berlin: Springer. 1974.

[5] S. Kumar, G. Stecher, M. Li, C. Knyaz, and K. Tamura, “MEGA X: Molecular Evolutionary Genetics Analysis across computing platforms,” Molecular Biology and Evolution, vol. 35, pp. 1547–1549, 2018.

[6] M. Nei, S. Kumar, Molecular evolution and phylogenetics, Oxford: University press, 2000.

[7] P. H. A. Sneath, and R. R. Sokal, Numerical Taxonomy, San Francisco: Freeman, 1973.

[8] B. Efron, E. Halloran, and S. Holmes, “Bootstrap confidence levels for phylogenetic trees,” PNAS, vol. 93, pp. 13429–13429. 1996.

[9] F. Lemoine, J. B. D. Entfellner, E. Wilkinson, D. Correia, M. D. Felipe, T. de Oliveira, and O. Gascuel, “Renewing Felsenstein’s phylogenetic bootstrap in the era of big data,” Nature, vol. 556, pp. 452–456, 2018.

[10] V.L. Kupershtokh, B.G. Mirkin, V.A. Trofimov, “Sum of internal proximities as a criterion of classification quality”, Autom. Remote Control, vol. 3, pp. 133–141, 1976.

[11] V. M. Efimov, K. V. Efimov, and V. Y. Kovaleva, “Principal Component Analysis and its generalizations for any type sequence (PCA-Seq),” Vavilov Journal of Genetics and Breeding, vol. 23, pp. 1032–1036, 2019.