IEICE TRANS. INF. & SYST., VOL.E89–D, NO.8 AUGUST 2006

INVITED PAPER *Special Section on Invited Papers from New Horizons in Computing*

The Bump Hunting Method Using the Genetic Algorithm with the Extreme-Value Statistics

Takahiro YUKIZANE[†a)], **Shin-ya OHI**[†b)], ***Nonmembers***, **Eiji MIYANO**[†c)], ***and*** **Hideo HIROSE**[†d)], ***Members***

SUMMARY In difficult classification problems of the z-dimensional points into two groups giving 0-1 responses due to the messy data structure, we try to find the denser regions for the favorable customers of response 1, instead of finding the boundaries to separate the two groups. Such regions are called the bumps, and finding the boundaries of the bumps is called the bump hunting. The main objective of this paper is to find the largest region of the bumps under a specified ratio of the number of the points of response 1 to the total. Then, we may obtain a trade-off curve between the number of points of response 1 and the specified ratio. The decision tree method with the Gini's index will provide the simple-shaped boundaries for the bumps if the marginal density for response 1 shows a rather simple or monotonic shape. Since the computing time searching for the optimal trees will cost much because of the NP-hardness of the problem, some random search methods, e.g., the genetic algorithm adapted to the tree, are useful. Due to the existence of many local maxima unlike the ordinary genetic algorithm search results, the extreme-value statistics will be useful to estimate the global optimum number of captured points; this also guarantees the accuracy of the semi-optimal solution with the simple descriptive rules. This combined method of genetic algorithm search and extreme-value statistics use is new. We apply this method to some artificial messy data case which mimics the real customer database, showing a successful result. The reliability of the solution is discussed.

key words: *data mining, data science, bump hunting, genetic algorithm, extreme-value statistics, trade-off curve, decision tree, bootstrap*

1. Introduction

Suppose that we are interested in classifying n points in a z-dimensional space into two groups according to their *responses*, where each point is assigned response 1 or response 0 as its target variable. For example, if a customer makes a decision to act a certain way, then we assign response 1 to this customer, and assign response 0 to the customer that does not. We want to know the customers' preferences presenting response 1. That is, who is likely to do this decision? We assume that their personal features, such as gender, age, blood type, living district, education, family profile, etc., are already obtained; the features can be plotted in the z-dimensional explanation variable space. Such a classification problem is fundamental in the data science field.

Many classification problems have been dealt with elsewhere to rather simpler cases using the methods of the linear discrimination analysis, the nearest neighbor, logistic regression, decision tree, neural networks, support vector machine, boosting, etc. (see [12], e.g.) as fundamental classification problems. In some real data cases in customer classification, however, the classification for the favorable customers is found to be difficult because of the messy data structure [13]. In such cases, we cannot separate the points of response 1 from those of response 0 clearly; we can find at most the denser regions to the favorable customers. Such regions are called the *bumps*, and finding the obscure boundaries from the noisy data is called *the bump hunting*.

Our Contributions. The primary objective of this paper is to find the bump regions under an optimization criterion (which will be precisely described later) in a high-dimensional space [14]. Let a pureness rate of a region R be the ratio of the number of points of response 1 to the total number of points in R. Also, let a capture rate of a region R be the ratio of the number of response 1 points in R to the whole number of ones. We can easily see that in general, the larger the pureness rate, the smaller the number of points in the bumps. In this paper, the situation we frequently encounter is formally modeled as the following optimization problem: Given n points with their responses 0/1 in the z-dimensional space and a rate p $(0 < p < 1)$, we find a region such that its pureness rate is at least p and it contains as the large number of points of response 1 as possible. Furthermore, our second objective is to find a trade-off curve of the capture rate to the specified pureness one when some appropriate bump hunting method is available. For the z-dimensional explanation variable space, an $O(n^{2z})$ algorithm was proposed in [1], [6]. Its running time might be polynomial of the instance size if we can be regarded z as constant. However, if the dimension is large, say, $z = 100$ or $z = 1,000$, then the running time could not be tractable. Hence, as a common alternative approach, we go into more efficient, simpler heuristics which have an approximation guarantee to the optimal value of the problem.

Even if we could obtain the free-shaped optimal boundaries for the bumpy regions, it would be difficult for us to make future actions to the customers directly using the information of the boundaries of the bumps. The bumps, having much simpler shapes of their boundary such as the union of z-dimensional boxes located parallel to some explanation variable axes, are more useful in doing the actions. Thus, as our basic strategy, we make use of the (binary) decision tree

Manuscript received February 17, 2006.
Manuscript revised April 17, 2006.

[†]The authors are with the Department of Systems Innovation and Informatics, Kyushu Institute of Technology, Fukuoka-shi, 820–8502 Japan.
a) E-mail: yukizane@ume98.ces.kyutech.ac.jp
b) E-mail: ohi@ume98.ces.kyutech.ac.jp
c) E-mail: miyano@ces.kyutech.ac.jp
d) E-mail: hirose@ces.kyutech.ac.jp

DOI: 10.1093/ietisy/e89–d.8.2332

method in principle in the classification since it provides the clear descriptive rules. We will show that the Gini's index in the decision tree may discover the boundaries for the bumps if the density for the favorable customers has a rather simple or monotonic shape; this would not be the optimal solution, but we prefer to adopt this simpler shape because of its ease of use. We call this the semi-optimal solution in the following.

One of the crucial issues in the decision tree technique is how we should determine the sequence of explanation variables used to make decisions. Probably, the conventional decision tree algorithm do not always provide the large capture rate in the bumpy regions we satisfy, if the algorithm is left as it is; we have to modify the algorithm. Since the computing time and the search for the optimal trees will cost too much intuitively due to the NP-hardness of the optimal decision tree layout [16], some random search methods for the optimal tree are required; if the sample size is N, then we have to investigate cases of $\Omega(z^N)$ and the cost of computing would be exponential. We propose here to use the genetic algorithm adapted to the tree; see [7] for the genetic algorithm and [17] for the genetic programming, for example.

Some data cases have many local maxima for the number of captured points unlike the ordinary search results in the genetic algorithms; then, the *extreme-value statistics* will be useful to estimate the global optimum number of captured points in hyper rectangles. This will also guarantee the accuracy of the semi-optimal solution with the simple descriptive rules obtained by using the genetic algorithm. We apply this method in Sects. 4 and 5 to an artificial messy data case which mimics the real customer database, showing a successful result assured by the *bootstrap* method.

Relation to the Related Work. The bump hunting has been studied in the fields of statistics, data mining, and machine learning in this decade [1], [6], [9], [10]: Friedman and Fisher [9] were aware of the importance and difficulty of finding the bumpy regions in higher dimensions early, and proposed a method based on PRIM (patient rule induction method); this would still cost much in computing, although lesser by a greedy method. Gray and Fan [10] proposed a probabilistic bump finding method, a kind of genetic algorithm, to enhance the computational speed. To provide many genes, they used the decision tree method. However, these two references do not touch on the pureness-rate capture-rate trade-off curve mentioned above. Solutions obtained by the genetic algorithm primarily are not global optimal. Our proposed approach using the extreme-value statistics to many local solutions obtained by the genetic algorithm provides not only the estimate for the optimal solution but also the confidence interval of the solution; thus, the trade-off curve obtained has also a confidence band under the condition that we are using hyper rectangles. This combined method of the genetic algorithm search and extreme-value statistics use is new. Other topics regarding the bump hunting are studied by Becker and Fahrmeir [2], Hand and Heard [11], Kehl and Ulm [15], Muller and Sawitzki [18], and Yip et. al. [19]; they are related to the bump hunting to some extent but not so much related to our approach.

2. Problem and Objectives

Our Problem. In two-class classification problems, we usually try to find the boundaries to discriminate the points of response 1 from those of response 0 in the z-dimensional space which corresponds to the z explanation variables. Traditionally, the goodness of optimality in classification has been measured by the misclassification rate which means the ratio, q, of the misclassified number of points to the total number of points. The smaller the misclassification rate, the better the classification. If we are interested in searching for a certain class, say, response 1, to which the pureness rate of response 1 in the class is high to some extent, the provided boundaries would be satisfactory. A typical case is shown on the left in Fig. 1 where the misclassification rate is small and the pureness rate of response 1 is high.

However, we often experience cases in practical problems where the misclassification rate could not be made smaller no matter how carefully we search, due to the messy data structure. Such a case is shown on the right in Fig. 1 where the number of the points of response 1 with a high pureness rate would be very small. Rather, we require the number of points to some extent with an acceptable pureness rate. This is our primary objective in this paper. The misclassification rate, in such a case, can no longer be the criterion for the optimality of classification.

Optimization Criterion. We propose a new criterion for classification, i.e., a trade-off relationship, $T(p, c(p))$, between the pureness rate, p, and the maximum capture rate, $c(p)$, where p and $c(p)$ denote the pureness rate of response 1 in some region and the maximum capture rate in the region when the pureness rate is given, respectively; we try to find the largest region $R(p)$ when a pre-specified pureness rate is provided, and $R(p)$ becomes the optimal region, in other words, the optimal bump. Another trade-off relationship, $T(p, m(p))$, between the pureness rate and the maximum captured points, $m(p)$, may also be useful; $m(p)$ is expressed as

$$m(p) = \sup_{R \subset \Omega} m_R(p), \tag{1}$$

where Ω denotes the total region. Our second objective is to

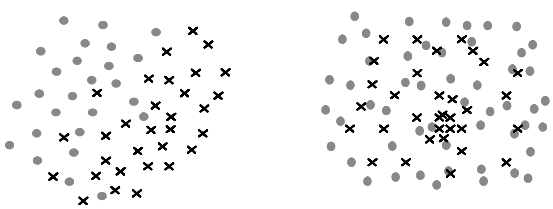


Fig. 1 Different optimality criterion is required.
Left: misclassification rate Right: pureness rate

construct the trade-off curve. Figure 2 shows an example of this optimal trade-off curve of $T(p, c(p))$.

Why We Use the Decision Tree. As as understood easily, the region $R(p)$ is not uniquely determined, if we perturb the boundary surface $S(p)$ of the region $R(p)$ a little bit; $c(p)$ can remain the same value with this perturbation. Even if the region $R(p)$ is determined somehow, it would be difficult how we use this boundary information. We cannot simply interpret the boundary information to the concrete explanation variable terms. On the contrary, the rules made by using the decision tree method are descriptive, and they can be easily understood by the (*if-then*) rules due to the binary splitting algorithm. The region made by the decision tree is a kind of box located parallel to the explanation variables. We prefer the simple shaped box to the optimum region with the free surface boundary, even though the optimization is spoiled a little. Thus, semi-optimization is pursued here. In Fig. 2 an example of the semi-optimal trade-off curve of $T^*(p, c(p))$ is superimposed.

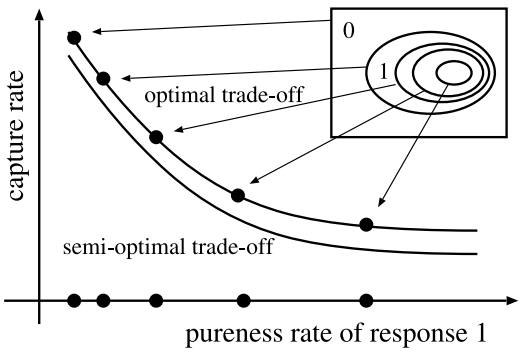


Fig. 2 Optimal and semi-optimal trade-off between the pureness rate and the capture rate.

Another merit using the decision tree is that we can circumvent the curse of dimensionality (see [3]); the decision tree method requires a one-dimensional search (for the splitting point). Whether the decision tree can detect the bump is important next, and this is discussed in the following section.

3. Can the Decision Tree Detect the Bump?

The Gini's index pursues to find the best explanation variable and the splitting point in the sense that the two split children nodes contain the purer classes than the parent node. If the shape of the marginal density function of an explanation variable is simple, such as being monotonic or unimodal, the Gini's index can detect the possible change points of the density function. We do not know exactly where the index splits the region of a certain explanation variable, but somewhere the change in the density shape can be caught.

Let us see how the decision tree can find the boundaries for the bumpy region in the one-dimensional case. Figure 3 shows a one-dimensional example in which a major group with response 0 is uniformly distributed, and a minor group with response 1 is almost uniformly distributed except in the bumpy region. The major group has a mass about five times larger than the minor group. By moving the threshold x from the left to the right to set the splitting point, we can find the optimal point in the sense of the Gini's impurity. This point coincides with a boundary point for the bumpy region, even if the mass of the bump is very small. Thus, the bump hunting can be achieved by the decision tree algorithm

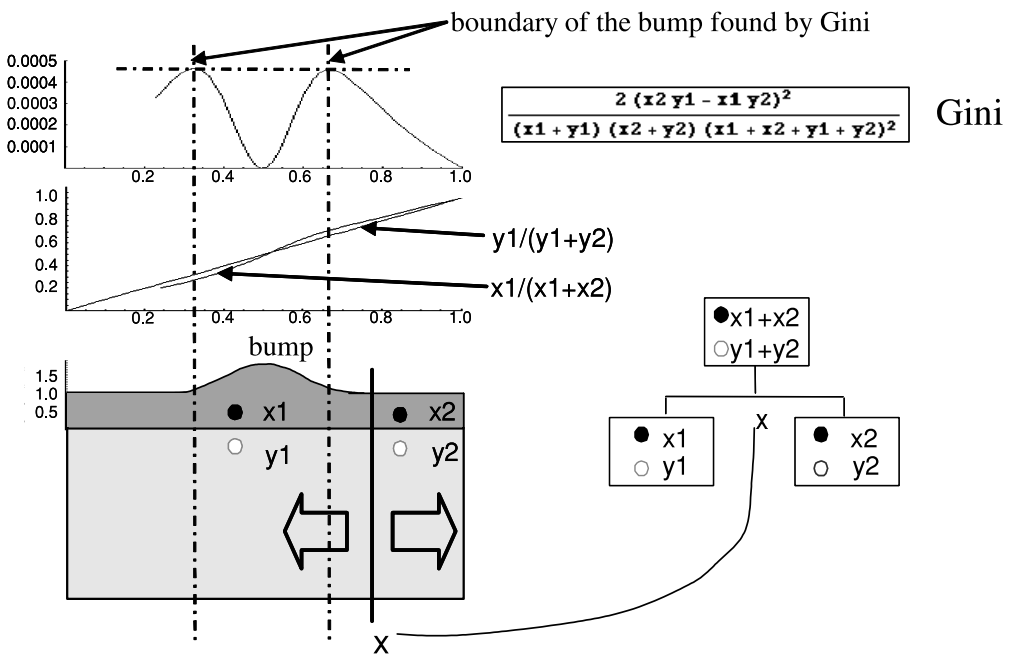


Fig. 3 How the Gini's index detect the bumpy region.

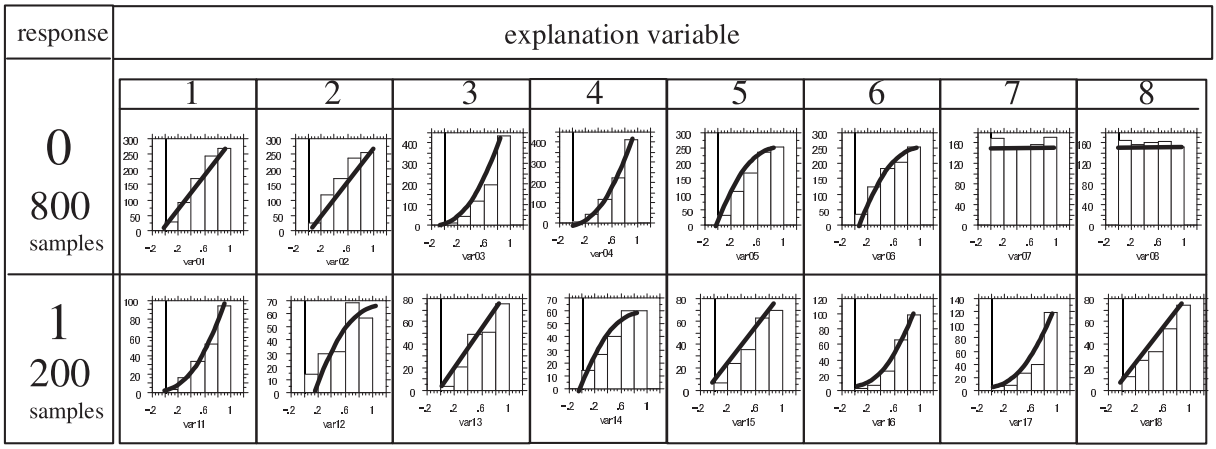


Fig. 4 Marginal density functions of eight explanation variables to response 1 and 0.

with the Gini's impurity. These splitting points may differ according to the volume of response 0, but the amount of fluctuation in the splitting points are small.

Even in a typical two-dimensional case of 0/1 responses, where 1,000 points of response 1 are uniformly distributed in xy-plane with a spread of $[-5.0, 5.0]$ for each axis, and 200 superimposed points are normally distributed on a bumpy region with mean 0.0 and a standard deviation of 1.0, and the response 0 are 5,000 uniformly distributed points in xy-plane with a spread of $[-5.0, 5.0]$ for each axis, the decision tree detects a bumpy region around $x, y \in [-a, a]$, where $1.5 \leq a \leq 2.0$. This is satisfactory.

4. The Random Search Method

In our optimization criterion, whether the conventional decision tree can give a semi-optimal solution is unknown. However, any simple example can be a counter example to the semi-optimal solution; this will be shown later in an example case. This fact proves that the conventional automatic tree algorithm will not provide the globally semi-optimal solution. Then, the random assignment method of the explanation variables has the possibility of providing the globally optimal solution.

To glance at how the numbers of the captured points are distributed by randomly assigned explanation variables to each splitting node, we here define the *tree-gain plot*. We assign an identification number, *id*, to the tree by this formulation,

$$id = \sum_{i=1}^{k} (a_i - 1) z^{k-i}, \tag{2}$$

where a_i is the explanation variable number, k is the number of splitting nodes, and z is the number of explanation variables. For example, we observe the case in Fig. 4; in the figure, each histogram corresponds to a density of the explanation variable for responses 0 and 1. We regard the points in one dimensional space as those projected from a high dimensional space. The number of simulated random

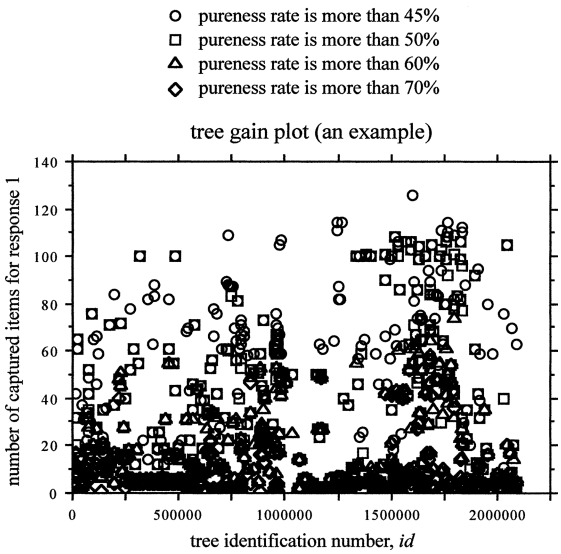


Fig. 5 Tree-gain plot (an example case).

points are 800 and 200 for responses 0 and 1, respectively. We assume eight explanation variables here; the shapes of the density functions are mimicked by the real data case of some customer database. 500 randomly generated trees and corresponding numbers of captured points of response 1 are plotted in Fig. 5. We call this the tree-gain plot. It appears that we cannot find any key features in this distribution, so we will next consider the random search method such as the genetic algorithm.

In applying the genetic algorithm to the tree structure, we are supposed to modify the crossover method to preserve the good inheritance property. To execute the crossover in the tree structure, we, for example, consider parents A and B, and we preserve the left hand side of tree A with the top node and preserve the right hand side of tree B without the top node, combining them to create a new tree having good inheritance. This genetic algorithm method is adapted to the tree structure. Then, to pursue the semi-optimal solution, the detailed algorithm of the genetic algorithm adapted to

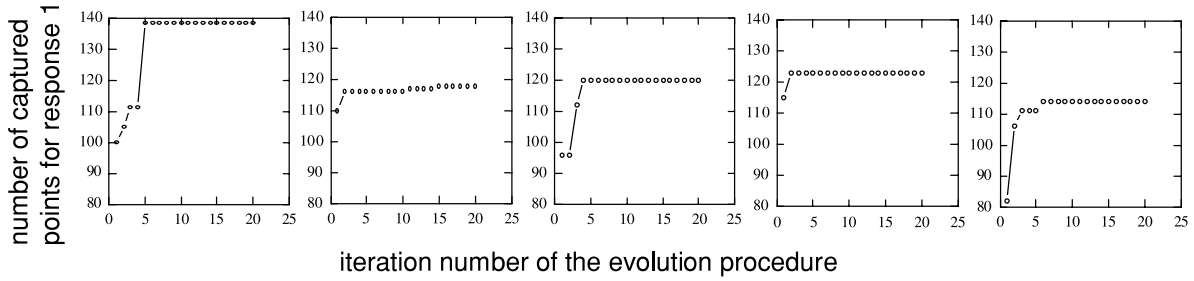


Fig. 6 The number of captured points for response 1: 5 evolution procedures with 5 different initial conditions.

the tree is as follows:

Step 1. Specify the pureness rate for the bumpy region.

Step 2. Generate some number of trees randomly, and from each tree, count the number of points in the nodes in which the pureness rates are larger than the pre-specified value; then, sort the trees in descending order of the captured number of points.

Step 3. Select the top ten trees as they are, and the rest of the trees are changed by crossover with good inheritance property.

Step 4. Some of the trees are changed by mutation. The mutation percentage is 1% through 20%, because the tree-gain plot shows quite a randomness, and the mutation procedure may pick up unfavorable trees.

Step 5. Continue the evolution procedure according to the genetic algorithm.

Step 6. Pick up the converged tree for the specified density, and memorize the number of points in the bumpy region.

Even if one solution is obtained by using the genetic algorithm with some initial value, the number of captured points is larger than any when using the simple random search method of 500 cases in some example case, which proves that the genetic algorithm works well.

5. Extreme-Value Statistics Assist the Genetic Algorithm

As Fig. 6 shows the ten cases of the iteration procedures in the previous example with ten different initial conditions, the converged solution may differ from each other when the initial value is set to a different value. The genetic algorithm may have an inclination of searching for the local maxima because the tree-gain plot shows the randomness of the captured points and suggests the existence of many local maxima. From 20 trials of the genetic algorithm, the maximum value of the captured points is 138. On the contrary, the conventional decision tree algorithm provides only 47 captured points; this is a typical counter example.

The histogram shown in Fig. 7 is made from the 20 converged values in the example case, using the genetic algorithm with 20 different initial values. As each value can be considered to be the maximum value in some domain, the distribution for these values will follow an extreme-value type distribution, such as the gumbel, the Weibull, and the Fréchet distributions. If the mother distribution function is a normal, exponential, log-normal, gamma, gumbel, or Rayleigh type distribution, then the limiting distribution of the maximum values from the mother distribution follows the gumbel distribution (see [5] e.g.). Here, the gumbel fitting to the maximum values obtained by the genetic algorithm is first applied.

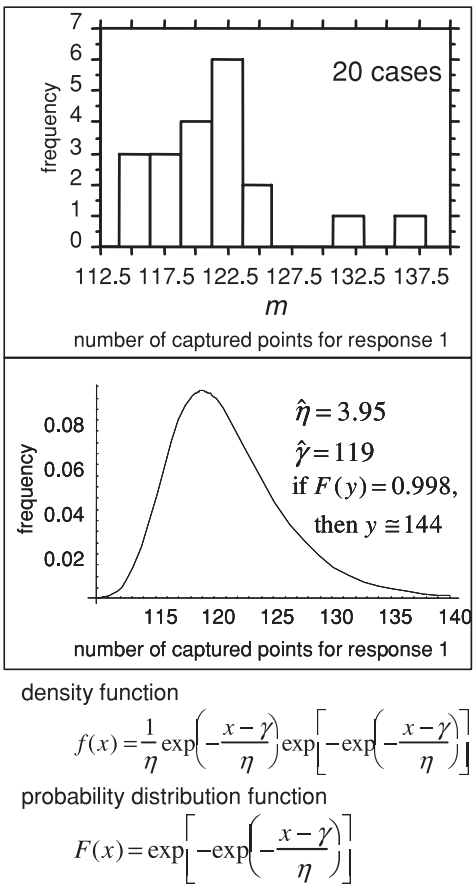


Fig. 7 Frequency distributions of the maximum number of captured points for response 1, and the fitted gumbel density function.

By fitting the gumbel distribution to these 20 data, we can obtain the fitted density distribution function and the

cumulative distribution function with the estimated location parameter 119, and scale parameter 3.95. Using these values, we can predict the maximum value of the captured points in other numbers of searching cases. Although we have only obtained the maximum value and its corresponding rule out of 20 genetic algorithm cases, we may predict where this maximum value is located. If we try 500 genetic algorithm cases, and we want to obtain the maximum converged value, then, it is estimated to be 144, and 138 is the 96 percentile point if we regard 144 to be the maximum value. We think that this result is satisfactory. We can use the number 138 and the corresponding descriptive rule as a semi-optimal solution.

To obtain the reliability of these estimated values, we have done the the bootstrap method (see [8]) using 20 genetic algorithm results. The number of trials is 1,000. Then, for the number 144, the 95% confidence interval is (135, 150). To confirm that this is true, we have computed 500 genetic algorithm cases, and the result is satisfactory.

By doing this procedure to the cases of 50%, 60%, and 70% pureness rates of response 1 in a similar manner, we can obtain the semi-optimal trade-off curve between the pureness rate and the number of response 1 captured points in the bumpy regions, which is shown in Fig. 8.

6. Discussions

Other Extreme-Value Type Distribution. Other extreme-value type distribution functions with the threshold to the right such as the Weibull distribution may also be applied to this problem instead of using the gumbel distribution. Due to the inclusion of the location parameter, the confidence interval to the threshold parameter may become large if the number of samples is not so large. Thus, the three-parameter estimation to find the endpoint of the distribution is not treated here. Rather, finding the higher percentile point is recommended.

Cases for Easier Classification Problems. We try to solve problems that are very hard to classify using the newly introduced classification criterion, which is the trade-off curve between the pureness rate and the capture rate. The capture rate strongly depends on the pureness rate as shown in Fig. 8. This dependency will become more gentle as the separability for response 0 and 1 can be obtained much easier. Figure 9 shows two other trade-off curves as well as the case in Fig. 8. The curve's upper side corresponds to an easily separable case for the explanation variables; this reminds us the figure on the left in Fig. 1. The curve in the middle corresponds to a case with mid-class separability.

Versatility of the Proposed Method. In Sects. 1 and 2, we introduced that the target of our problem is to solve the problems where the misclassification rate cannot be large but the capture rate can be obtained to some extent with the specified pureness rate. However, the method proposed here will work for any problem if we are interested in acquisition

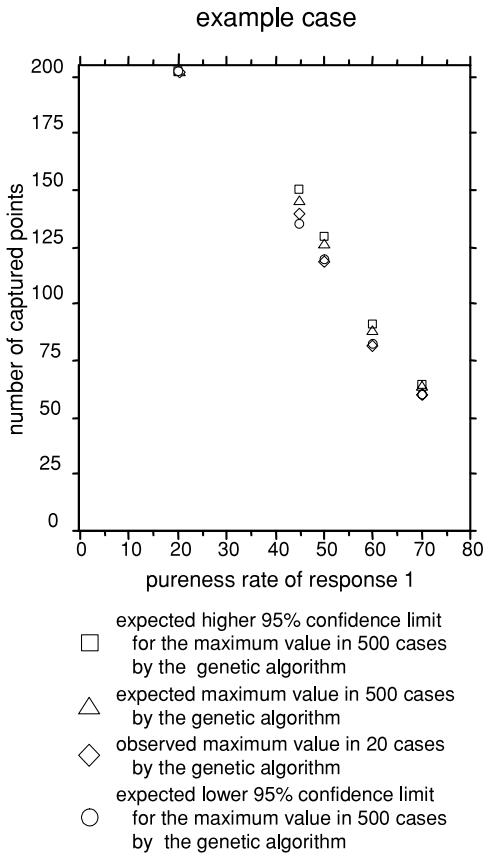


Fig. 8 Semi-optimal trade-off curve between the pureness rate and the number of captured points.

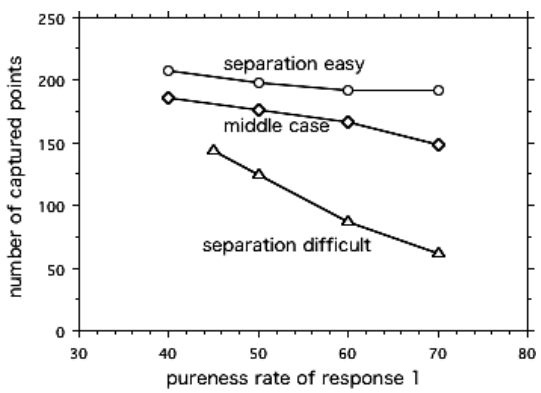


Fig. 9 Three types of trade-off curves between the pureness rate and the number of captured points.

of one class items. Thus, the proposed method is versatile as the method includes the conventional splitting algorithm such as the CART by Breiman et. al. [4]. We have experienced such cases and the proposed method has worked well.

Reliability of the Solution. In discussing classification accuracy, we often use misclassification error fluctuations by many learning-test data combinations, e.g., ten-fold cross-validation. This method is useful in prediction; if we used all the data as learning data in classification problems, then an

over-fitted solution adapted to the very learning data would result; that is, a bias to the true model might be produced. Another method to assess the accuracy of the misclassification rate is to use the bootstrap. The non-parametric bootstrap resamples the points from the original data with equal probability to every point in drawing for many trial cases. The statistical property of all the results given by each trial shows the accuracy of the estimated misclassification error. This method essentially uses the original data as the learning data, and has the possibility of producing the model bias. However, if the number of data points is large enough, such a bias would be reduced.

To assess the accuracy of the solution in our approach, we basically use the bootstrap method. We first select trees at random as the gene seeds for the genetic algorithm from the very original data. Then, we obtain a number of semi-optimal local solutions, say 20 solutions, which are converged capture rates using the genetic algorithm. Finally, we estimate the semi-optimal global solution using the extreme-value statical method based on the maximum likelihood principle. If we use the parametric bootstrap method to the underlying probability distribution for the solution, we can assess the accuracy of the solution similarly to the non-parametric bootstrap method.

Although we do not construct the solution tree by the learning data and do not assess the capture rate by the test data because such a method requires extremely large computing time, the proposed treatment using the combination of the genetic algorithm and the extreme-value statistical approach will provide an almost equivalent accuracy for the trade-off curve to the accuracy by using the learning-test data combined method. The difference between the two is whether it is sampled earlier or later. The reliability of the confidence band for the trade-off curve depends on the number of the extreme-value statistics samples, but the mean location of the band would not be affected by the number of samples. In addition, the number of data points in our real customer database model is large enough, e.g., 200, 000 points, so why we used the bootstrap method to obtain the reliability for the trade-off curve in this paper.

7. Concluding Remarks

In difficult classification problems of the z-dimensional points into two groups giving 0-1 responses due to the messy data structure, we try to find the bumpy regions for the favorable customers of response 1, instead of finding the boundaries to separate the two groups. By specifying the pureness rate of response 1 to the total, we can find the largest semi-optimal boxes parallel to some of the explanation variables using the decision tree both with the Gini's index and with the genetic algorithm method. Then, we can obtain a trade-off curve between the pureness rate and the number of points of response 1. Due to the existence of many local maxima unlike the ordinary genetic algorithm search results, the extreme-value statistics work to estimate the semi-optimal solution with the guarantee of the accuracy of the descriptive rules. The reliability of the solution can be obtained using the bootstrap method.

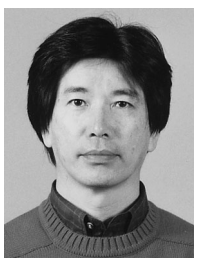
References

[1] D. Agarwal, J.M. Phillips, and S. Venkatasubramanian, "The hunting of the bump: On maximizing statistical discrepancy," SODA'06, pp.1137–1146, 2006.
[2] U. Becker and L. Fahrmeir, "Bump hunting for risk: A new data mining tool and its applications," Computational Statistics, vol.16, pp.373–386, 2001.
[3] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.
[4] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, Classification and Regression Trees, Wadsworth, 1983.
[5] E. Castillo, Extreme Value Theory in Engineering, Academic Press, New York, 1988.
[6] D.P. Dobkin, D. Gunoplos, and W. Maass, "Computing the maximum bichromatic discrepancy, with applications to computer graphics and machine learning," NeuroCOLT Tech Rep Series NC-TR-95-008, 1995.
[7] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Professional, 1989.
[8] B. Efron, The Jackknife, the Bootstrap and Other Resampling Plans, Society of Industrial and Applied Mathematics, Philadelphia, 1982.
[9] J.H. Friedman and N.I. Fisher, "Bump hunting in high-dimensional data," Statistics and Computing, vol.9, pp.123–143, 1999.
[10] J.B. Gray and G. Fan, "Target: Tree analysis with randomly generated and evolved trees," Technical Report, The University of Alabama, 2003.
[11] D.J. Hand and N.A. Heard, "Finding groups in gene expression data," J. Biomedicine and Biotechnology, vol.2, pp.215–225, 2005.
[12] T. Hastie, R. Tibshirani, and J.H. Friedman, Elements of Statistical Learning, Springer, 2001.
[13] H. Hirose, "A method to discriminate the minor groups from the major groups," 2005 Hawaii International Conference on Statistics, Mathematics and Related Fields, pp.317–318, Honolulu, Jan. 2005.
[14] H. Hirose, "Optimal boundary finding method for the bumpy regions," IFORS2005 (International Federation of Operational Research Societies) Triennial 2005 Conference, FD-19-3, Honolulu, July 2005.
[15] V. Kehl and K. Ulm, "Responder identification in clinical trials with censored data," Computational Statistics and Data Analysis, vol.50, pp.1338–1355, 2006.
[16] L. Hyafil and R.L. Rivest, "Constructing optimal binary decision trees is NP-complete," Inf. Process. Lett., vol.5, pp.15–17, 1976.
[17] J.R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.
[18] D.W. Muller and G. Sawitzki, "Excess mass estimates and test for multimodality," J. American Statistical Association, vol.86, pp.738–746, 1991.
[19] A.M. Yip, C. Ding, and T.F. Chan, "Dynamic cluster formation using level set methods," IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.6, pp.877–889, 2006.

Takahiro Yukizane was born on February 19, 1983 in Japan. He received M.E. degree from Kyushu Institute of Technology in 2006. He is currently a graduate student of Systems Innovation and Informatics Department at Kyushu Institute of Technology.

Shin-ya Ohi received B.E. degree from Kyushu Institute of Technology in 2005. He is currently a graduate student of Systems Innovation and Informatics Department at Kyushu Institute of Technology.

Eiji Miyano is Associate Professor of Systems Innovation and Informatics Department at Kyushu Institute of Technology. Received B.E., M.E., and D.E. degrees in computer science from Kyushu University in 1991, 1993, and 1995, respectively. His research interests are in the area of algorithms and complexity theory. He is a member of IPSJ and ACM.



Hideo Hirose graduated from Kyushu University (Mathematics in 1977) and from Nagoya University (Dr. Eng. in 1988). He worked for Takaoka Electric Manufacturing Co., Ltd. from 1977 to 1995, and was Vice Research Director there from 1988 to 1995. He was Professor at Hiroshima City University from 1995 to 1998, and has been Professor at Kyushu Institute of Technology since April 1998. His interests include data mining, statistical lifetime analysis, and reliability engineering. He is a member of IEEE, ASA, IMS, AMS, SIAM, MPS, ACM, IEEJ, IPSJ, ORSJ, REAJ, MSJ, JSS, JSAS, JSCS, and JSIAM.