12-1-1998

Interval Image Classification is NP-Hard

Alejandro E. Brito

Vladik Kreinovich
University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-98-34
Published in: *Proceedings of the Second International Workshop on Intelligent Virtual Environments*, Xalapa, Veracruz, Mexico, September 11-12, 1998, pp. 12-15.

Interval Image Classification is NP-Hard

Alejandro E. Brito[1,2] and Vladik Kreinovich[3]

[1]Division de Ingeneria y Ciencias
ITESM (Instituto Technologico de Monterrey)
Campus de Ciudad de Mexico

[2,3]Departments of [2]Electrical and Computer Engineering
and [3]Computer Science
The University of Texas at El Paso
El Paso, TX 79968, USA
emails [2]alexb@ece.utep.edu, [3]vladik@cs.utep.edu

Abstract

Feature extraction from images to perform object classification is a very hard problem for general solution. We prove that under interval uncertainty, *linear classification* is NP-hard.

Feature extraction and pattern recognition. One possible application of intelligent virtual environments is to train people to classify objects of certain type into several known classes (categories). To check how well they are trained, the system must be able to do this classification automatically. Thus, we must be able to design a system which can automatically classify objects of certain type into several known classes.

As a case study, we will take an automatic inspection problem, where, based on an image of a printed circuit board (PCB), we want to check whether a given Surface Mounted Device (SMD) is attached to this PCB or not [1, 2, 3]. In general, we measure (directly or indirectly) several characteristics $x_1, \ldots, x_n$ of the image, and we want to make our decision based on the results of these measurements; these characteristics are called *features*.

In the ideal case, these features should uniquely characterize the desired properties of the analyzed image. Feature extraction is usually problem dependent [4]; the right selection of the features can simplify greatly the job of the classifier [5]. In this paper, we will assume that the features have already been selected. For example, we can take as x_i, either the brightnesses values of different image pixels, or the results of applying functions or transformations to these brightness values.

Linear classifiers: the simplest case. Many of such classification problems have been successfully solved, by well-justified methods [5, 6]. In some cases, the solution is very computationally complicated (e.g., requires several thousand iterations on a neural network), but in many other cases, the solution is computationally very simple: we have *linear classifiers* that separate each pair of classes. A linear classifier is a pair consisting of a linear discriminant function $\ell(x) = c_1 \cdot x_1 + \ldots + c_n \cdot x_n$ and a threshold value c_0 such that $\ell(x) \geq c_0$ for all objects from the first class, and $\ell(x) \leq c_0$ for all images from the second class [5, 6].

Informal formulation of the problem. If linear classifiers are possible, then we can easily classify an arbitrary object by computing the values of the (easily computable) function $\ell(x)$. It is, therefore, desirable to check whether it is possible to separate two given classes by a linear classifier. If we already know that a linear classification is possible, then the next natural question is to find the coefficients c_i of the linear discriminant function. In this paper, we will analyze the computational complexity of these problems: i.e., the problem of checking whether a linear classifier is possible, and the problem of computing the coefficients of the classifier.

When measurements are absolutely accurate, these problems are easy to solve. Let us describe the above problems in precise terms. We have several examples of objects from the first class, and we have several examples of objects from the second class. Let us denote the number of known objects from the first class by p, and let us denote, for each i from 1 to p, the features of i-th object by $x^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)})$. Similarly, we will denote the total number of known objects from the second class by q, and the features of i-th object from the second class by $y^{(i)} = (y_1^{(i)}, \ldots, y_n^{(i)})$. The vectors which describe the features of known objects are usually called *feature vectors*, or *patterns* [5, 6]. If the measurements are accurate, then the possibility of a linear classifier is equivalent to the existence of the real numbers $c_1, \ldots, c_n, c_0$ for which the following inequalities are true:

$$c_1 \cdot x_1^{(i)} + \ldots + c_n \cdot x_n^{(i)} \geq c_0, \;\; 1 \leq i \leq p; \tag{1}$$

$$c_1 \cdot y_1^{(i)} + \ldots + c_n \cdot y_n^{(i)} \leq c_0, \;\; 1 \leq i \leq q. \tag{2}$$

(Of course, these inequalities are always true if we take $c_1 = \ldots = c_n = c_0 = 0$, so we must exclude this degenerate case.)

In algorithmic terms, the possibility of a linear classifier is equivalent to the solvability of the system of linear inequalities (1), (2). The problem of checking such solvability is well known in applied mathematics: it is a particular case of linear programming. There exist polynomial-time algorithms for solving linear programming problems; therefore, in the case of precise measurements, the problem of checking the existence of a linear classifier is computationally feasible (i.e., it can be solved in polynomial time [7]).

Similarly, the problem of computing the values c_i for which (1) and (2) are true can also be solved by known linear programming algorithms and therefore, this problem is also computationally feasible.

Entering interval uncertainty. In many practical problems, measurements are not precise: For each object, the measured values $\tilde{x}_i$ of the features differ from the (unknown) actual values x_i of these features. Usually, we know the *upper bound* Δ_i on the corresponding measurement errors $\Delta x_i = \tilde{x}_i - x_i$; as a result, from the measurement result $\tilde{x}_i$, we can conclude that the actual value x_i can take any value from the interval $\mathbf{x}_i = [\tilde{x}_i - \Delta_i, \tilde{x}_i + \Delta_i]$ [8].

In some cases, in addition to upper bounds, we also know the probabilities of different measurement errors, but in many practical problems, we do not know these probabilities. With these problems in mind, in this paper, we will consider the classification problem under interval uncertainty.

Towards the exact formulation of the interval classification problem. Each object is characterized by n different features $x_1, \ldots, x_n$. So, if we take interval uncertainty into consideration, the result of measuring features for each object is a sequence of intervals $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. We know such sequence $\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_n^{(i)})$ for each of p objects from the first class, and we know the corresponding sequences $\mathbf{y}^{(i)} = (\mathbf{y}_1^{(i)}, \ldots, \mathbf{y}_n^{(i)})$ for each of q objects of the second class. We say that a linear classifier is possible if there exist coefficients $c_1, \ldots, c_n, c_0$, and values $x_j^{(i)} \in \mathbf{x}_j^{(i)}$ and $y_j^{(i)} \in \mathbf{y}_j^{(i)}$ for which the inequalities (1) and (2) are both true. We will show that checking this condition is computationally intractable (NP-hard) [7].

Definition.

- *Let an integer n be given; this integer will be called the number of features.*
- *A sequence $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of n intervals will be called an (interval) pattern.*
- *By an interval classification problem, we mean a pair $\langle F, S \rangle$, where F and S are finite sets of interval patterns.*
- *We say that for an interval classification problem, a linear classifier is possible if there exist real numbers $c_1, \ldots, c_n, c_0$ which satisfy the following three conditions:*
  - *these numbers are non-degenerate (i.e., $c_i \neq 0$ for some i from 1 to n);*
  - *for every pattern $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ from the set F, there exist values $x_i \in \mathbf{x}_i$ for which $c_1 \cdot x_1 + \ldots + c_n \cdot x_n \geq c_0$; and*
  - *for every pattern $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ from the set S, there exist values $y_i \in \mathbf{y}_i$ for which $c_1 \cdot y_1 + \ldots + c_n \cdot y_n \leq c_0$.*

Theorem. *The problem of checking whether a linear classifier is possible for an interval classification problem is NP-hard.*

Proof. We will show that if we can check, for every interval classification problem, whether a linear classifier exists or not, then we would be able to solve a *partition* problem which is known to be NP-hard [7, 9]. The partition problem consists of the following: given N integers $s_1, \ldots, s_N$, check whether exist N integers $c_i \in \{-1, 1\}$ for which $c_1 \cdot s_1 + \ldots + c_N \cdot s_N = 0$.

To solve this problem, let us formulate the following interval classification problem. In this problem, $n = N + 1$, i.e., we have $N + 1$ features $x_1, \ldots, x_N, x_{N+1}$. We will form the sets S and F as follows:

- First, we add an interval pattern $([s_1, s_1], \ldots, [s_N, s_N], [1, 1])$ both to F and to S.
- Next, for each i from 1 to N, we add:
  - a pattern $([0, 0], \ldots, [0, 0], [-1, 1]$ (i–th place), $[0, 0], \ldots, [0, 0])$ to F;
  - a pattern $([0, 0], \ldots, [0, 0], [1, 1]$ (i–th place), $[0, 0], \ldots, [0, 0])$ to S;
  - a pattern $([0, 0], \ldots, [0, 0], [-1, -1]$ (i–th place), $[0, 0], \ldots, [0, 0])$ also to S.
- Finally, for $i = N + 1$, we add a pattern $([0, 0], \ldots, [0, 0], [1, 1])$ both to F and to S.

The possibility of having a linear classifier for these interval pattern is equivalent to the existence of the coefficients $c_1, \ldots, c_n, c_0$ for which:

$$c_1 \cdot s_1 + \ldots + c_N \cdot s_N + c_{N+1} \geq c_0; \tag{3a}$$

$$c_1 \cdot s_1 + \ldots + c_N \cdot s_N + c_{N+1} \leq c_0; \tag{3b}$$

for each i from 1 to N,

$$z_i \cdot c_i \geq c_0 \text{ for some } z_i \in [-1, 1]; \tag{4}$$

$$c_i \leq c_0; \tag{5}$$

$$-c_i \leq c_0; \tag{6}$$

and finally, for $i = N + 1$, we have

$$c_{N+1} \geq c_0; \tag{7a}$$

$$c_{N+1} \leq c_0. \tag{7b}$$

This system of inequalities can be simplified if we take into consideration that two inequalities (3a) and (3b) are equivalent to the equality

$$c_1 \cdot s_1 + \ldots + c_N \cdot s_N + c_{N+1} = c_0, \tag{3}$$

and similarly, (7a) and (7b) are equivalent to a single equality

$$c_{N+1} = c_0. \tag{7}$$

So, we have to satisfy conditions (3), (4), (5), (6), and (7).

If the partition problem has a solution $c_i \in \{-1, 1\}$, $1 \leq i \leq N$, then we can easily show that these values c_i, together with $c_{N+1} = c_0 = 1$ and $z_i = c_i$, satisfy the conditions (3)–(7).

Vice versa, let a non-degenerate vector c_i satisfy the conditions (3)–(7). Then, for $i = N + 1$, from (7), we conclude that $c_{N+1} = c_0$, and therefore, (3) takes the form

$$c_1 \cdot s_1 + \ldots + c_N \cdot s_N = 0. \tag{8}$$

For $i \leq N$:

- On one hand, from (5) and (6), we conclude that $|c_i| \leq c_0$ and therefore, $c_0 \geq 0$; we cannot have $c_0 = 0$, because otherwise $c_i = 0$ for all i, and we will have a degenerate vector, so $c_0 > 0$.
- On the other hand, from (4), we conclude that $|z_i| \cdot |c_i| \geq c_0$, i.e., that $|c_i| \geq c_0/|z_i|$. Since $|z_i| \leq 1$, we conclude that $|c_i| \geq c_0$.

Combining the two inequalities $|c_i| \le c_0$ and $|c_i| \ge c_0$, we conclude that $|c_i| = c_0$, i.e., that either $c_i = c_0$, or $c_i = -c_0$. So, if we take $\tilde{c}_i = c_i/c_0$, we conclude that $\tilde{c}_i \in \{-1, 1\}$. Dividing both sides of (8) by c_0, we conclude that

$$\tilde{c}_1 \cdot s_1 + \ldots + \tilde{c}_N \cdot s_N = 0,$$

i.e., that the values $\tilde{c}_1, \ldots \tilde{c}_N \in \{-1, 1\}$ form a solution of a partition problem.

Thus, the constructed interval classification problem has a linear classifier if and only if the original partition problem has a solution. Since we know that the partition problem is NP-hard, we can thus conclude that the problem of checking whether a linear classifier is possible is also NP-hard.

Acknowledgments. This work was partially supported by a 1995 Texas Advanced Technology Program Grant, by ARO Grant DAAH04-95-1-0494, by NASA under cooperative agreement NCC5-209, by NSF under grant No. DUE-9750858, by the United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518, and by the National Security Agency under Grant No. MDA904-98-1-0564.

The authors are thankful to Sergio D. Cabrera for valuable discussions.

References

[1] G. Carrillo, S. D. Cabrera, and A. A. Portillo, "Inspection of Surface-Mount-Device Images using Wavelet Processing", *Proceedings of SPIE Applied Imagery Pattern Recognition*, SPIE Vol. 2982, pp. 213–225, Washington D.C., October 1996.

[2] A. A. Portillo, A. E. Brito, and S. D. Cabrera, "Quantifying Improvements in the Preprocessing Stage of a Classification System for Surface-Mounted-Device Images", *Proceedings of SPIE Applied Imagery Pattern Recognition*, SPIE Vol. 3240, pp. 105–115, Washington D.C, October 1997.

[3] A. E. Brito, E. Whittenberger, and S. D. Cabrera, "Segmentation Strategies with Multiple Analysis for an SMD Object Recognition System", *Proceedings of Southwest Symposium on Image Analysis and Interpretation*, pp. 59-64, Tucson AZ., April 1998.

[4] O. G. Selfridge and U. Neisser, "Pattern Recognition by Machine", *in Computers and Though, Edited by E. A. Feigenbaum and J. Feldman*, pp. 237–250, MacGraw-Hill, New York, NY, 1963.

[5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons-Interscience, New York, NY, 1973.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, San Diego, CA, 1990.

[7] M. R. Garey and D. S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-Completeness*, W.H. Freemand and Company, San Francisco, CA, 1979.

[8] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[9] V. Kreinovich, A. V. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1997.