Cross Distance Minimization for Solving the Nearest Point Problem Based on Scaled Convex Hull

Qiangkui Leng[1(✉)], Erjie Jiao[1], Yuqing Liu[2], Jiamei Guo[1], and Ying Chen[1]

[1] School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China
qkleng@126.com

[2] College of Information Science and Technology, Bohai University, Jinzhou 121013, China

Abstract. In pattern classification, the geometric method often provides a simple and intuitive solution. In the case of linear separability, solving the optimization hyperplane problem can be transformed into solving the nearest point problem of the convex hulls between classes. In the case of nonlinear separability, the notion of scaled convex hull (SCH) is employed to reduce the initially overlapping convex hulls to become separable. Two classic nearest point algorithms, GSK and MDM, have been used as effective solvers for SCHs. However, their problem-solving speed is still a bit underperforming. This paper proposes a new solver called SCH-CDM, in which the CDM (cross distance minimization) algorithm is employed to calculate the nearest point pair of between-class SCHs. Experimental results indicate that the SCH-CDM algorithm can achieve faster convergence than the SCH-GSK algorithm and the SCH-MDM algorithm. In terms of accuracy, it also shows good competitiveness compared to the baseline methods.

Keywords: Pattern classification · Support vector machine · Geometric method · Cross distance minimization

1 Introduction

Geometric methods are proven effective in solving a pattern classification problem if the problem is itself a geometric problem or can be transformed into a geometric problem [1, 2]. The obtained solution is usually simple and intuitive. A typical paradigm is that the convex hull of the training point set is often used to learn a classifier [3, 4]. The well-known support vector machine (SVM) also follows this paradigm [5]. The geometric interpretation of SVM in the feature space is a consequence of the dual representation, i.e., the convexity of each class and finding the respective support hyperplanes that provide the maximal margin [6, 7].

In the case of linear separability, solving the optimization hyperplane problem can be transformed into the nearest point problem of finding the convex hull between classes [8]. Three nearest point algorithms, including Gilbert-Schlesinger-Kozinec (GSK) [9],

D.-S. Huang et al. (Eds.): ICIC 2022, LNAI 13395, pp. 198–208, 2022.
https://doi.org/10.1007/978-3-031-13832-4_17

Mitchell-Demyanov-Malozemov (MDM) [10], and cross distance minimization (CDM) [11] have been developed and have shown good potential in different aspects.

In the case of nonlinear separability, if we start from a purely geometrical point of view, then a proper convex hull transformation is required. Mavroforakis et al. [12, 13] investigated the geometric properties of reduced convex hull (RCH) and devised a mathematical framework to support RCH. Under the framework, the GSK algorithm is rewritten in order to show the practical benefits of the theoretical results that are derived herewith. In [14], a variant of the GSK algorithm that simultaneously updates pairs of points has been introduced to reduce the kernel operations. Moreover, López et al. [15] proposed an alternative clipped extension of the classical MDM algorithm that results in a simple algorithm with a good generalization ability.

Following the RCH scheme, scaled convex hull (SCH) was presented for improving the performance of the geometry-class SVM [16]. The main idea is to scale down the initial convex hulls by means of a parameter λ in such a way that the shape of the original hull is preserved. Besides, vertices in SCH are easier to calculate compared to RCH, leading to the easier application of nearest point algorithm to train the non-separable classifiers because the pair of nearest points depends directly on these vertices.

Under the framework of SCH, the GSK algorithm and the MDM algorithm have been used as solvers for the nearest point problem [17, 18]. Inspired by the above practice, we generalize the CDM algorithm into the framework of SCH in this paper. In our previous publication [19], we have discussed the GSK algorithm and the MDM algorithm. Although the MDM algorithm starts from a geometrical point of view, it involves many optimization processes, making the solving process a bit slow. While the GSK algorithm is geometrically intuitive, and it is significantly different from CDM, as shown in Fig. 1.

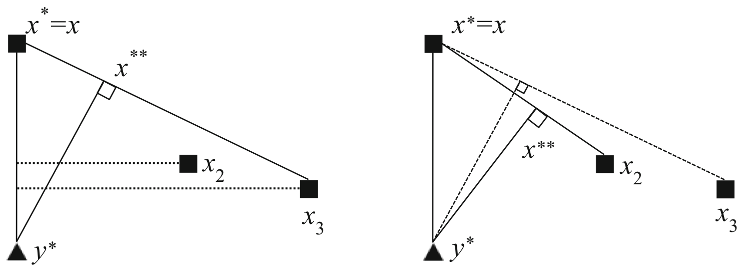

Fig. 1. Different updates of the GSK algorithm and the CDM algorithm.

In each iteration, the GSK algorithm updates only one point, either x^* or y^*, whereas the CDM algorithm updates both of them. In addition, they may pick different data points for updating the nearest pair due to their different calculation ways. In Fig. 1, the GSK algorithm chooses x_3 to update x^*, while the CDM algorithm chooses x_2 to update x^*. This paper introduces the CDM algorithm into the framework of SCH, which aims to speed up finding the nearest point pair between classes.

The rest of the paper is organized as follows: In Sect. 2, we give a brief introduction to the SCH notion. In Sect. 3, we present the SCH-CDM algorithm for solving the nearest point problem. Also, the corresponding algorithm is described with detailed comments. In Sect. 4, we evaluate the proposed SCH-CDM algorithm by comparative experiments. In Sect. 5, we give the concluding remarks with a brief discussion.

2 Preliminary on SCH

Let $\mathbf{R}^n$ be the n-dimensional Euclidean space. For a finite point set S in $\mathbf{R}^n$, its convex hull can be denoted as follows:

$$CH(S) = \{s|s = \sum_{1 \le i \le |S|} \alpha_i s_i, \sum_{1 \le i \le |S|} \alpha_i = 1, s_i \in S, \alpha_i \ge 0, \alpha_i \in \mathbf{R}\} \tag{1}$$

where $|S|$ represents the cardinality of S. For two point sets X and Y in $\mathbf{R}^n$, if they are linearly separable, the task of an SVM is to find the maximal margin hyperplane:

$$\mathrm{f(x) = w^*x + b^*} \tag{2}$$

where (w^*, b^*) is an optimal solution of the following quadratic model:

$$\begin{aligned} &\min \frac{1}{2}||w||^2 \\ &s.t.\ w \cdot x_i + b \ge 1, i \in |X|; w \cdot y_j + b \le -1, j \in |Y| \end{aligned} \tag{3}$$

It is well known that solving (3) is equivalent to solving a nearest point problem:

$$\min ||x - y|| s.t. x \in CH(X), y \in CH(Y) \tag{4}$$

If X and Y are nonlinearly separable (i.e., $CH(X) \cap CH(Y) \ne \Phi$), it is not feasible to directly compute the nearest point pair between their convex hulls because they are overlapping. In this case, the SCH notion is introduced to transform the nonlinear separable problem into a separable one.

$$SCH(X, \lambda) = \{x|x = \sum\nolimits_{1 \le i \le |X|} \alpha_i(\lambda x_i + (1-\lambda)m^+), \sum\nolimits_{1 \le i \le |X|} \alpha_i = 1, x_i \in X\} \tag{5}$$

$$SCH(Y, \lambda) = \{y|y = \sum\nolimits_{1 \le j \le |Y|} \beta_j(\lambda y_j + (1-\lambda)m^-), \sum\nolimits_{1 \le j \le |Y|} \beta_j = 1, y_j \in Y\} \tag{6}$$

where $m^+ = \sum_{i=1}^{|X|} x_i/|X|$ and $m^- = \sum_{j=1}^{|Y|} y_j/|Y|$ are the mean values (also called centroids) of all points of X and Y, respectively.

For any point $x'_i \in SCH(X, \lambda)$, it is a linear combination of the point $x_i \in X$ and the centroid m^+. Figure 2 shows the geometric interpretation of SCH. x_i is an original point in X, and x'_i is a point on $SCH(X, \lambda)$. Apparently, x'_i lies on the line connecting x_i and the centroid m^+, i.e., $x'_i = \lambda x_i + (1-\lambda)m^+$. Similarly, we can get $y'_j = \lambda y_j + (1-\lambda)m^-$. λ is a scaled reduction factor between 0 and 1. One advantage of SCH is that after scaled reduction, $SCH(X, \lambda)$ and X have the same number of vertices and these vertices are in a one-to-one correspondence.

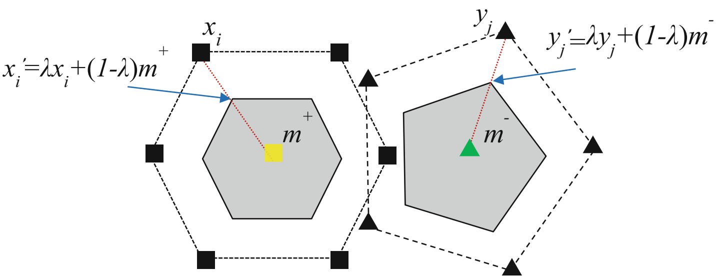

Fig. 2. Geometric interpretation of SCH.

If X and Y are nonlinearly separable in the initial state, as λ gradually becomes larger, $SCH(X, \lambda)$ and SCH(Y, λ) will gradually reduce and eventually become linearly separable. In [16], the author gives the criterion for judging separability. Assume that $r^+ = \max_{i\in|X|} ||x_i - m^+||$, $r^- = \max_{j\in|Y|} ||y_j - m^-||$, and $r = ||r^+ + r^-||$, $SCH(X, \lambda)$ and $SCH(X, \lambda)$ will be considered linearly separable when $\lambda r^+ + \lambda r^- \le r$. However, this criterion is somewhat rigid, as shown in Fig. 3. In practice, we can give a moderate value of λ.

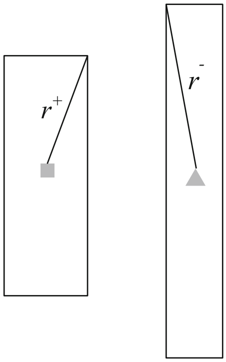

Fig. 3. Illustration on the moderate choice of λ.

3 The Proposed Method

The CDM algorithm was proposed to calculate a nearest point pair between two point sets X and Y in the linearly separable case [11]. It chooses any two points $x^* \in X$, $y^* \in Y$ as the initial nearest point pair. Then, y^* is fixed and a point $x^{**} \in CH(X)$ nearest to y^* is found, i.e.,

$$x^{**} = \arg\min_x\{||x - y^*||, x \in CH(X)\} \tag{7}$$

Next, x^* is fixed and a point $y^{**} \in CH(Y)$ nearest to x^* is found, i.e.,

$$y^{**} = \arg\min_y\{||x^{**} - y||, y \in CH(Y)\} \tag{8}$$

This process iterates until the stopping condition is satisfied:

$$||x^* - y^*|| - ||x^{**} - y^{**}|| < \varepsilon \tag{9}$$

where ε is a pre-determined precision parameter for controlling the convergence. If the nearest point pair (x^{**}, y^{**}) is found at a certain moment, the classification discriminant function $f(z)$ is constructed as follows:

$$f(z) = (x^{**} - y^{**})z + \frac{||y^{**}||^2 - ||x^{**}||^2}{2} \tag{10}$$

The geometric interpretation of the CDM algorithm is shown in Fig. 4. When $\delta = 1$, $x^{**} = x_1$ is a point in X. When $0 < \delta < 1$, $x^{**} = x_1 + \lambda(x_2\text{-}x_1)$ is actually the vertical point from y^* to the line segment connecting x_1 and x_2. This indicates if x^* is not the nearest point from y^* to $CH(X)$, there must exist another point x^{**} such that $dist(x^{**}, y^*) < dist(x^*, y^*)$.

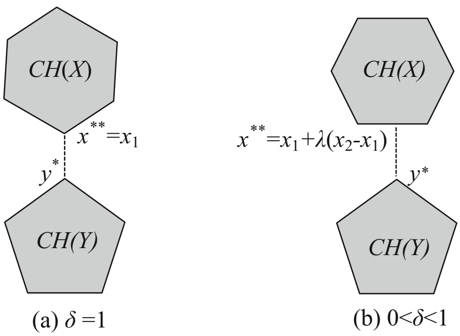

Fig. 4. The geometric interpretation of the CDM algorithm

In the nonlinearly separable case, the CDM algorithm will no longer work. As mentioned above, we can generalize it to the SCH framework for converting nonlinearly separable problem into a linearly separable one. By introducing reduction factor λ, the scaled point sets can be denoted as $X' = \{x_i', i = 1, 2, \ldots, |X|\}$ and $Y' = \{y_i', j = 1, 2, \ldots, |Y|\}$, where $x_i' = \lambda x_i + (1 - \lambda)m^+$ and $y_i' = \lambda y_j + (1 - \lambda)m^-$. We present the SCH-based CDM algorithm (SCH-CDM for short) in Algorithm 1.

Algorithm-1. SCH-CDM
Input: Two scaled sets X' and Y', precision parameter ε.
Step1: $x^{**} \in X$, $y^{**} \in Y$;
Step2: $x^{*} = x^{**}$, $y^{*} = y^{**}$;
Step3: x^{**}=argmin$_t\{dist(t, y^*)|t= x^*+\delta_1(x_2- x^*)\}$, where $x_2 \neq x^*$, $x_2 \in X$, and δ_1=

$$\min\{1, \frac{(x_2 - x^*)(y^* - x^*)}{(x_2 - x^*)(x_2 - x^*)}\} > 0$$

Step4: y^{**}=argmin$_t\{dist(x^*, t)|t= y^*+\delta_2(y_2- y^*)\}$, where $y_2 \neq y^*$, $y_2 \in Y$, and δ_2=

$$\min\{1, \frac{(y_2 - y^*)(x^* - y^*)}{(y_2 - y^*)(y_2 - y^*)}\} > 0$$

Step5: If $dist(x^*, y^*)$-$dist(x^{**}$-$y^{**}) \geq \varepsilon$, goto Step2;
Step6: $w^{**}=x^{**}-y^{**}$, $b^{**}=(\|y^{**}\|^2-\|x^{**}\|^2)/2$;
Output: $f(z)=w^{**}z+b^{**}$.

It should be noted that the scaled reduction factor λ does not appear in Algorithm 1 because we take X' and Y' as inputs, which have an implicit scaling factor. The SCH-CDM algorithm has a rough time complexity $O(I(\varepsilon)(|X'| + |Y'|))$, where $I(\varepsilon)$ represents the number of total iterations related to ε. Furthermore, if we want to use kernel function in the SCH-CDM algorithm, then we can take the same strategy as in [19].

Table 1. Datasets used for experiments.

Datasets	Key	n	d	Source	λ
Australian	aus	690	14	Libsvm	0.9
Autistic Spectrum Disorder	aut	104	21	UCI	0.95
Breast Cancer	bre	683	10	Libsvm	0.7
Coronary Artery Disease	cor	303	59	UCI	0.6
Fourclass	fou	862	2	Libsvm	0.95
German.numer	ger	1000	24	Libsvm	0.9
HCC Survival	hcc	165	49	UCI	0.75
Immunotherapy	imm	90	8	UCI	0.9
Ionosphere	ion	351	34	Libsvm	0.85
Chronic Kidney	kid	400	25	UCI	0.85
Mammographic Mass	mam	961	6	UCI	0.85
Mushrooms	mrs	8124	112	Libsvm	0.95
Musk (Version 1)	mus	476	166	UCI	0.95

(*continued*)

Table 1. (*continued*)

Datasets	Key	n	d	Source	λ
Pima Indians Diabetes	pim	768	8	UCI	0.65
Sonar	son	208	60	Libsvm	0.8

4 Experiments

In this section, we evaluate the proposed SCH-CDM algorithm by experiments on fifteen benchmark datasets. These datasets are from UCI repository [20] and LIBSVM [21], and their details are listed Table 1. n is the number of total points. d is the dimensionality of a dataset. For convenience, we use a key to abbreviate a dataset.

We compare the SCH-CDM algorithm with the SCH-GSK algorithm and the SCH-MDM algorithm, which are two other classical nearest point algorithms and are used as baselines. We randomly split each dataset into two halves for 10 times, one half for training, and the other for testing. The RBF kernel is employed for each algorithm.

$$K(x, y) = e^{||x-y||^2/2\sigma^2} \tag{11}$$

Two parameters, C and γ respectively from the candidate sets $\{2^i|-4, -3, \ldots, 3, 4\}$ and $\{2^i|-7, -6, \ldots, 4, 5\}$, are determined by 5-fold cross-validation on the training set. The precision parameter ε is set to 0.0001. All experiments are conducted on a PC with I7-8700 3.20 GHz CPU, 8 GB memory, and Windows 10 operating system.

Table 2. Accuracy comparison of SCH-CDM with SCH-GSK and SCH-MDM (%)

Datasets	SCH-GSK	SCH-MDM	SCH-CDM
aus	87.14 ± 0.77	**87.17 ± 0.75**	86.79 ± 1.49
aut	91.70 ± 2.21	91.70 ± 2.21	**92.83 ± 1.95**
bre	96.43 ± 0.94	96.43 ± 0.94	**96.58 ± 1.01**
cor	**99.28 ± 1.87**	96.91 ± 7.47	**99.28 ± 1.87**
fou	**99.63 ± 0.22**	**99.63 ± 0.22**	99.61 ± 0.27
ger	**75.14 ± 2.28**	75.08 ± 1.92	74.54 ± 2.40
hcc	72.17 ± 4.59	71.69 ± 4.55	**73.37 ± 4.83**
imm	77.39 ± 4.83	77.61 ± 3.56	**78.91 ± 4.81**
ion	92.67 ± 2.68	92.44 ± 2.48	**93.18 ± 1.54**
kid	**99.65 ± 0.67**	**99.65 ± 0.67**	**99.65 ± 0.67**
mam	**79.61 ± 1.69**	79.52 ± 1.72	79.38 ± 1.71
mrs	**99.99 ± 0.02**	**99.99 ± 0.02**	99.89 ± 0.09
mus	89.21 ± 1.95	**89.46 ± 2.22**	89.41 ± 1.54
pim	75.39 ± 1.54	**75.68 ± 1.44**	75.16 ± 2.37
son	83.71 ± 3.25	83.71 ± 3.25	**84.10 ± 2.62**

Table 2 shows the accuracy comparison of the SCH-CDM algorithm with the SCH-GSK algorithm and the SCH-MDM algorithm. The SCH-CDM algorithm obtains the highest accuracies on eight datasets (aut, bre, cor, hcc, imm, ion, kid, and son). In pairwise comparisons, it has the similar win-loss ratios with the SCH-GSK algorithm (7:6) and the SCH-MDM algorithm (7:7).

Table 3 provides the training time in seconds and the testing time in milliseconds. The SCH-CDM algorithm has obvious advantages compared to the SCH-GSK algorithm and the SCH-MDM algorithm. For example, on the mam dataset, it executes a training process consuming 7.59 s, whereas the SCH-GSK algorithm and the SCH-MDM algorithm take 2796.12 and 84.62 s, respectively. In terms of testing.

Table 3. Comparison on training time (seconds) and testing time (milliseconds)

Datasets	SCH-GSK	SCH-MDM	SCH-CDM
aus	1328.28 (17.71)	27.92 (9.53)	**6.74 (7.40)**
aut	23.15 (8.32)	0.39 (0.89)	**0.30 (0.12)**
bre	921.68 (6.58)	8.17 (6.03)	**3.42 (1.94)**
cor	153.61 (2.79)	2.30 (1.14)	**1.22 (0.15)**
fou	2315.36 (24.17)	32.09 (4.77)	**5.24 (1.01)**
ger	2233.18 (23.99)	67.00 (20.95)	**20.74 (10.10)**
hcc	107.63 (6.48)	1.90 (2.41)	**1.28 (0.70)**
imm	21.82 (0.63)	0.33 (0.45)	**0.17 (0.08)**
ion	526.89 (6.35)	7.20 (3.79)	**3.38 (0.89)**
kid	404.25 (2.98)	5.03 (3.16)	**3.06 (0.58)**
mam	2796.12 (10.28)	84.62 (9.35)	**7.59 (2.45)**
mrs	2.84×10^4 (3705.91)	7389.71 (3649.70)	**1271.85 (594.92)**
mus	1251.68 (60.55)	47.84 (34.39)	**24.23 (9.79)**
pim	2113.51 (10.26)	27.19 (9.41)	**5.89 (2.94)**
son	213.23 (11.41)	3.72 (4.28)	**2.04 (0.81)**

Table 4. Comparison on the number of support vectors

Datasets	SCH-GSK	SCH-MDM	SCH-CDM
aus	330	330	**313**
aut	39	39	**36**
bre	148	147	**96**
cor	30	27	**23**
fou	115	103	**78**

(*continued*)

Table 4. (*continued*)

Datasets	SCH-GSK	SCH-MDM	SCH-CDM
ger	404	404	**297**
hcc	69	69	**61**
imm	36	35	**35**
ion	92	92	**65**
kid	53	51	**41**
mam	318	313	**154**
mrs	3988	3994	**746**
mus	186	185	**144**
pim	245	242	**154**
son	74	73	**62**

time, it takes about 594 ms on the mrs dataset, which is far less than the 3705 ms spent by the SCH-GSK algorithm and the 3649 ms spent by the SCH-MDM algorithm. It should be noted that the training time includes the time to perform 5-fold cross-validation for choosing optimization parameters. Less running time indicates faster decision response, thus in some real-time systems, the SCH-MDM algorithm can be considered for prioritization.

Table 4 summarizes the numbers of support vectors. On all 15 datasets, the SCH-CDM algorithm obtains fewer support vectors than the SCH-GSK algorithm and the SCH-MDM algorithm. On the bre dataset, the SCH-CDM algorithm obtains 96 support vectors, whereas the SCH-GSK algorithm and the SCH-MDM algorithm obtains 148 and 147 support vectors, respectively. More remarkably, on the mrs dataset, SCH-CDM obtains only 746 support vectors, which is far less than 3988 by the SCH-GSK algorithm and 3994 by the SCH-MDM algorithm. In general, less support vectors indicates the corresponding method has a simple decision model, which may be more effective in the complicated task, meeting the criterion of Occam's razor.

5 Conclusion

Following the notion of SCH, we proposed a geometric method called SCH-CDM to solve the nearest point problem. By introducing a scaled reduction factor, it can transform the nonlinearly separable case into a linearly separable one. It builds on the CDM algorithm and has a pretty good iterative update strategy.

Experimental results on benchmark datasets show that the proposed SCH-CDM algorithm achieves faster training and faster testing than the other two nearest point algorithms, i.e., the SCH-GSK algorithm and the SCH-MDM algorithm. In terms of support vectors, it obtains the least number among the three comparison methods. In some real-time systems, it has the potential to be prioritized.

However, there is one issue that deserves attention. For the scaled reduction factor, the moderate setting of its value needs further study. Some work related to convex hull-based classification may be used as references.

References

1. Peng, X., Wang, Y.: Geometric algorithms to large margin classifier based on affine hulls. IEEE Trans. Neural Netw. Learn. Syst. **23**(2), 236–246 (2011)
2. Leng, Q., Wang, S., Qin, Y., et al.: An effective method to determine whether a point is within a convex hull and its generalized convex polyhedron classifier. Inf. Sci. **504**, 435–448 (2019)
3. Ding, S., Nie, X., Qiao, H., et al.: A fast algorithm of convex hull vertices selection for online classification. IEEE Trans. Neural Netw. Learn. Syst. **29**(4), 792–806 (2017)
4. Carmichael, I., Marron, J.S.: Geometric insights into support vector machine behavior using the KKT conditions. Electron. J. Stat. **15**(2), 6311–6343 (2021)
5. Chen, X., Xiao, Y.: Geometric projection twin support vector machine for pattern classification. Multimedia Tools Appl. **80**(15), 23073–23089 (2020). https://doi.org/10.1007/s11042-020-09103-1
6. Bennett, K.P., Bredensteiner, E.J.: Duality and geometry in SVM classifiers. In: Seventeenth International Conference on Machine Learning (ICML 2000), pp. 57–64 (2000)
7. Barbero, A., Takeda, A., López, J.: Geometric intuition and algorithms for Eν–SVM. J. Mach. Learn. Res. **16**, 323–369 (2015)
8. Almasi, O.N., Rouhani, M.: A geometric-based data reduction approach for large low dimensional datasets: Delaunay triangulation in SVM algorithms. Mach. Learn. Appl. **4**, 100025 (2021)
9. Franc, V., Hlaváč, V.: An iterative algorithm learning the maximal margin classifier. Pattern Recogn. **36**(9), 1985–1996 (2003)
10. Keerthi, S.S., Shevade, S.K., Bhattacharyya, C., et al.: A fast iterative nearest point algorithm for support vector machine classifier design. IEEE Trans. Neural Netw. **11**(1), 124–136 (2000)
11. Yujian, L., Bo, L., Xinwu, Y., et al.: Multiconlitron: a general piecewise linear classifier. IEEE Trans. Neural Netw. **22**(2), 276–289 (2011)
12. Mavroforakis, M.E., Theodoridis, S.: A geometric approach to support vector machine (SVM) classification. IEEE Trans. Neural Netw. **17**(3), 671–682 (2006)
13. Mavroforakis, M.E., Sdralis, M., Theodoridis, S.: A novel SVM geometric algorithm based on reduced convex hulls. In: 18th International Conference on Pattern Recognition (ICPR 2006), pp. 564–568 (2006)
14. Mavroforakis, M.E., Sdralis, M., Theodoridis, S.: A geometric nearest point algorithm for the efficient solution of the SVM classification task. IEEE Trans. Neural Netw. **18**(5), 1545–1549 (2007)
15. López, J., Barbero, Á., Dorronsoro, J.R.: Clipping algorithms for solving the nearest point problem over reduced convex hulls. Pattern Recogn. **44**(3), 607–614 (2011)
16. Liu, Z., Liu, J.G., Pan, C., et al.: A novel geometric approach to binary classification based on scaled convex hulls. IEEE Trans. Neural Netw. **20**(7), 1215–1220 (2009)
17. Liu, Z., Liu, J.G., Chen, Z.: A generalized Gilbert's algorithm for approximating general SVM classifiers. Neurocomputing **73**(1–3), 219–224 (2009)
18. López, J., Barbero, Á., Dorronsoro, J.R.: An MDM solver for the nearest point problem in Scaled Convex Hulls. In: 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2010)
19. Li, Y., Leng, Q., Fu, Y.: Cross kernel distance minimization for designing support vector machines. Int. J. Mach. Learn. Cybern. **8**(5), 1585–1593 (2016). https://doi.org/10.1007/s13042-016-0529-8

20. Frank, A., Asuncion, A.: UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 21 Dec 2021
21. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 1–27 (2011)