



Title	Construction of convex hull classifiers in high dimensions
Author(s)	Takahashi, Tetsuji; Kudo, Mineichi; Nakamura, Atsuyoshi
Citation	Pattern Recognition Letters, 32(16), 2224-2230 <a href="https://doi.org/10.1016/j.patrec.2011.06.020">https://doi.org/10.1016/j.patrec.2011.06.020</a>
Issue Date	2011-12-01
Doc URL	<a href="http://hdl.handle.net/2115/47979">http://hdl.handle.net/2115/47979</a>
Type	article (author version)
File Information	PRL32-16_2224-2230.pdf



[Instructions for use](#)

# Construction of Convex Hull Classifiers in High Dimensions

Tetsuji Takahashi, Mineichi Kudo and Atsuyoshi Nakamura

*Graduate School of Information Science and Technology*

*Hokkaido University*

*Kita-14, Nishi-9, Kita-ku, Sapporo 060-0814, JAPAN*

---

## Abstract

We propose an algorithm to approximate each class region by a small number of approximated convex hulls and to use these for classification. The classifier is one of non-kernel maximum margin classifiers. It keeps the maximum margin in the original feature space, unlike support vector machines with a kernel. The construction of an exact convex hull requires an exponential time in dimension, so we find an approximate convex hull (a polyhedron) instead, which is constructed in linear time in dimension. We also propose a model selection procedure to control the number of faces of convex hulls for avoiding over-fitting, in which a fast procedure is adopted to calculate an upper-bound of the leave-one-out error. In comparison with support vector machines, the proposed approach is shown to be comparable in performance but more natural in the extension to multi-class problems.

*Keywords:* Pattern Recognition, Convex Hull, Classifier Selection

---

**1 1. Introduction**

2 Determining class regions directly in a feature space and classifying a  
3 class-unknown sample to the class of which region is closest to the sample  
4 seems a promising approach to classification. However, the efficacy of this  
5 approach has not been established. In this paper, we address this issue,  
6 specifically for a set of convex hulls as a class region.

7 Usually, class regions are determined indirectly by discriminant functions  
8 or by an estimated decision boundary. Nevertheless, some classifiers are  
9 connected to certain types of regions. For example, a linear support vector  
10 machine (shortly, SVM) [1] is connected to the convex hulls of samples of  
11 two classes [2] and the nearest neighbor classifier is connected to the Voronoi  
12 diagram of samples [3]. Therefore, it is worth examining which type of re-  
13 gions is most effective for a wide range of classification problems and how  
14 such regions can be constructed. In this paper, we focus on a set of convex  
15 hulls that include training samples of a class maximally. Then, we assign  
16 a class label to a given test sample according to the distances of the sam-  
17 ple to the estimated class regions. This approach [4] using convex hulls has  
18 shown performance comparable with that of SVM in low-dimensional data.

<sup>19</sup> However, it is computationally difficult to construct the exact convex hull in  
<sup>20</sup> high-dimensional data [5]. The algorithm by McBride *et al.* [6] also suffers  
<sup>21</sup> from the same type of dimensionality problem. In this paper, therefore, we  
<sup>22</sup> use a set of polyhedral convex sets that are constructed in a linear order of  
<sup>23</sup> dimension.

<sup>24</sup> Zero training errors are achieved by using a sufficient number of convex  
<sup>25</sup> hulls in such a way that every training sample of a class is covered by at  
<sup>26</sup> least one convex hull and any other sample belonging to the other classes is  
<sup>27</sup> excluded, ensuring that no sample is shared by more than one class. Thus,  
<sup>28</sup> according to Occam's razor [7], we should choose the simplest classifier as  
<sup>29</sup> long as it attains the same degree of training error. In our case, we select  
<sup>30</sup> the smallest number of convex hulls with the smallest number of faces.

<sup>31</sup> There are three problems to be solved. First, construction of the convex  
<sup>32</sup> hull of a given sample set is computationally difficult. The time and space  
<sup>33</sup> complexities are exponential in dimension in the worst case. Second, more  
<sup>34</sup> than one convex hull is generally needed to cover all training samples exhaust-  
<sup>35</sup>ively. Indeed, as is well known, if each class sample set is covered by a single  
<sup>36</sup>convex hull, then any pair of classes is linearly separable, and the reverse is  
<sup>37</sup>also true. Finally, when we use convex hulls for classification, the distance

38 between a point and the boundary of each convex hull must be measured.

39 However, the computational cost is quite high (e.g., see [8]).

40 In this paper, we first describe an algorithm for constructing approximate

41 convex hulls that can cope with these three problems in a reasonable way and

42 then we describe a model selection procedure for choosing simpler classifiers.

43 The algorithm is based on the prototype version introduced in [8].

44 A similar idea is realized in [9] in which the distance between a given point

45 and the convex hull of each class is used for class assignment. However, only

46 a single convex hull is assigned for each class, and thus a non-linear kernel

47 is necessary for solving non-linearly separable problems. Good selection of

48 a kernel and its parameters thus becomes problematic as well as the case of

49 SVM. The distance between a point and a convex hull is calculated by solving

50 an SVM optimization problem by assuming the point as a singleton set. Since

51 this calculation is made for each testing point, the time complexity is very

52 high when the training sample set is large. Cevikap *et al.* [10] discussed linear

53 manifolds spanned by nearest neighbors of a query sample and constructed

54 an affine hull (a convex hull is contained in it) for classification, but the

55 nonlinearization also relies on a kernel. In contrast to these studies, the

56 classifier developed here does not need a kernel.

<sup>57</sup> **2. Definitions of Convex Hull and Reflective Convex Hull**

<sup>58</sup> The convex hull  $\text{conv}(S)$  of a given finite dataset  $S = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  is  
<sup>59</sup> defined as the intersection of all convex sets containing  $S$ . Here,  $\mathbf{x}_i \in R^m$  ( $i =$   
<sup>60</sup>  $1, \dots, n$ ) is a point in an  $m$ -dimensional Euclidean space. Since  $S$  is finite,  
<sup>61</sup>  $C = \text{conv}(S)$  is a polyhedron with at most  $n$  vertices.

<sup>62</sup> By  $\partial C$ , we denote the boundary of  $C$  and divide it into  $q$ -faces according  
<sup>63</sup> to the dimensions. For example, 0-faces are the vertices of  $C$  and  $(m - 1)$ -  
<sup>64</sup> faces are the facets or hyper-planes. Let  $V(C)$  be the set of vertices of  $C$  and  
<sup>65</sup> let  $F(C)$  be the set of facets of  $C$ . A convex hull of a finite set of  $S$  can be  
<sup>66</sup> expressed by several ways. For example, the  $\mathcal{V}$ -representation of  $C$  is given by  
<sup>67</sup>  $C = \{\mathbf{y} = \sum c_{\mathbf{x}} \mathbf{x} \mid \sum c_{\mathbf{x}} = 1, c_{\mathbf{x}} \geq 0, \mathbf{x} \in V(C)\}$ . Another expression is the  
<sup>68</sup>  $\mathcal{H}$ -representation given by  $C = \{\mathbf{y} \mid \langle \mathbf{w}, \mathbf{y} \rangle \leq c, \forall (\mathbf{w}, c) \in F(C)\}$ , where  $\langle \cdot, \cdot \rangle$   
<sup>69</sup> is the inner product and facet  $(\mathbf{w}, c)$  is specified by a normal vector  $\mathbf{w}$  ( $\|\mathbf{w}\| =$   
<sup>70</sup> 1) and a constant  $c \in R$ . We need both expressions to handle  $C$  efficiently  
<sup>71</sup> in space complexity or in different goals. For example, an  $m$ -dimensional  
<sup>72</sup> cube has  $2m$  facets and  $2^m$  vertices, and an  $m$ -dimensional simplex has  $2m$   
<sup>73</sup> vertices and  $2^m$  facets. For determining if a sample is included in  $C$  or  
<sup>74</sup> not and for distance calculation to  $C$ ,  $\mathcal{H}$ -representation has an advantage of  
<sup>75</sup>  $\mathcal{V}$ -representation.

76 Unfortunately, we know that the number of facets can be of order  $n^{\lfloor \frac{m}{2} \rfloor}$  [5].

77 Therefore, we propose to use an approximated convex hull that has a reasonable number of facets. To do this, we consider another expression of a convex hull. We use *support functions* [11] for the new expression, which is similar to  $\mathcal{H}$ -representation but needs an infinite number of half-spaces. A 81 *support function* of a unit vector  $\mathbf{w}$  ( $\|\mathbf{w}\| = 1$ ) is given by

$$H(S, \mathbf{w}) \triangleq \sup\{\langle \mathbf{x}, \mathbf{w} \rangle | \mathbf{x} \in S\},$$

82 where “sup” denotes the supremum. With the set  $W_0$  of all possible unit vectors, the convex hull  $C$  is defined as

$$C = \text{conv}(S, W_0) \triangleq \bigcap_{\mathbf{w} \in W_0} \{\mathbf{y} | \langle \mathbf{y}, \mathbf{w} \rangle \leq H(S, \mathbf{w})\}.$$

84 Here,  $h(S, \mathbf{w}) = \{\mathbf{y} | \langle \mathbf{y}, \mathbf{w} \rangle = H(S, \mathbf{w})\}$  is called a *support plane*. The convex hull is an area that is surrounded by support planes  $h(S, \mathbf{w})$  for all  $\mathbf{w} \in W_0$ .  
86 An example of the convex hull constructed by support planes is shown in Fig. 1. As an example of the support plane,  $h(S, \mathbf{w})$  with angle  $\theta$  is also shown.  
88

89 We notice that a finite subset  $W \subset W_0$  gives an approximate convex hull 90  $\text{conv}(S, W)$  and thus a good selection of  $W$  derives a good approximation. Of course, instead of  $W$ , we can use the vectors corresponding to  $F(C)$ . Then

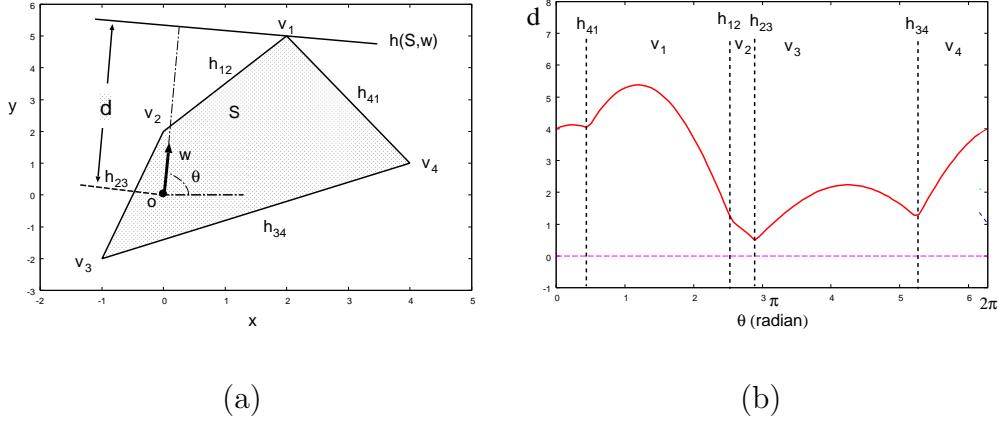


Figure 1: (a) The convex hull of a set  $S$ , including points  $v_1, v_2, \dots, v_4$ , represented by support planes. Only one support plane  $h(S, \mathbf{w})$  is shown for a directional vector  $\mathbf{w}$  with angle  $\theta$ . As the value of  $\theta$  increases, starting from this angle about  $\theta = 85^\circ$ , support planes will find edge  $h_{12}$  first and then vertex  $v_2$ , edge  $h_{23}$ , and so on. The behavior is shown in (b) as a graph of support function  $d = H(S, \mathbf{w})$  with the changing angle  $\theta$  of  $\mathbf{w}$ .

92 the exact convex hull is obtained. However, as described before, the number  
 93 of facets can grow exponentially in dimension. Therefore, we use a constant  
 94 number of facets by letting  $K = |W|$  be constant.

95 Next, let us consider separating a finite set  $S$  from another finite set  $T$   
 96 by the convex hull of  $S$  when they are linearly or non-linearly separated. A  
 97 support plane  $h$  of  $S$  might locate both  $S$  and  $T$  in the same side of the two  
 98 half-spaces separated by it. Apparently, such support planes are useless for  
 99 separating  $S$  from  $T$ . For separation of  $S$  from  $T$ , all we need is *reflective*

100    *support planes*, which are support planes separating  $S$  from  $T$  perfectly or  
 101   partly. A *reflective convex hull*,  $C_r = \text{conv}(S, W_r)$ , is the polyhedral convex  
 102   set specified by the (infinite) set  $W_r$  of all unit vectors generating reflective  
 103   support planes. Then the reflective convex hull  $C_r$  is formally defined by

$$C_r = \text{conv}(S, W_r) = \bigcap_{\mathbf{w} \in W_r} \{\mathbf{y} | \langle \mathbf{y}, \mathbf{w} \rangle \leq H(S, \mathbf{w})\}.$$

104   From the definition,  $C = \text{conv}(S, W_0) \subseteq C_r = \text{conv}(S, W_r)$  since  $W_r \subseteq W_0$ .  
 105   Intuitively speaking, the reflective convex hull of  $S$  is the polyhedral convex  
 106   set of  $S$  whose faces reflect rays emitted from points in  $T$ . An example of  
 107   the reflective convex hull is shown in Fig. 2. Note that usually a reflective  
 108   convex hull, unlike the convex hull that encloses the samples of a class, is  
 109   open on the side opposite the decision boundary.

110   We can also define the *margin*  $M(S, T, \mathbf{w})$  between  $S$  and  $T$  in direction  
 111    $\mathbf{w}$  as

$$M(S, T, \mathbf{w}) \triangleq -H(T, -\mathbf{w}) - H(S, \mathbf{w}).$$

112   Note that when  $S$  and  $T$  are linearly separable, there exists a sup-  
 113   port plane specified by a unit vector  $\mathbf{w}$  with positive margin  $M(S, T, \mathbf{w}) =$   
 114    $M(T, S, -\mathbf{w})$ . Now a reflective support plane can be defined formally as a

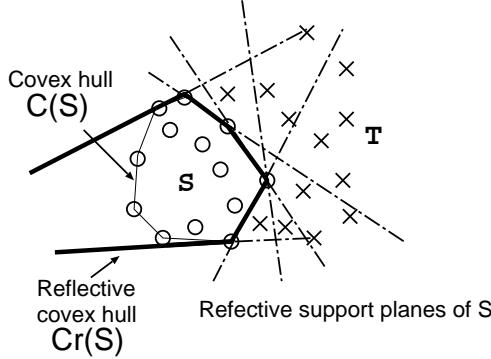


Figure 2: Reflective convex hull of the set of positive samples  $S$  against the set of negative samples  $T$ . A few reflective support planes are shown.

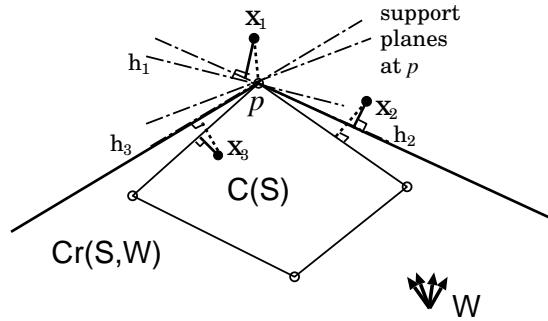


Figure 3: Distances  $D(\mathbf{x}, \partial\text{conv}(S, W))$  from  $\mathbf{x}$  to an approximated convex hull  $\text{conv}(S, W)$  with  $W$  are shown by solid lines and the exact distances  $D(\mathbf{x}, \partial\text{conv}(S))$  to the exact convex hull are shown by dashed lines.

<sup>115</sup> support plane  $h_r(S, \mathbf{w})$  with  $\mathbf{w}$  satisfying

$$H(T, \mathbf{w}) - H(S, \mathbf{w}) > 0.$$

<sup>116</sup> The (signed) distance between a point  $\mathbf{x}$  and the nearest boundary of a  
<sup>117</sup> convex hull  $\text{conv}(S, W_0)$  is given by

$$D(\mathbf{x}, \partial\text{conv}(S, W_0)) = \sup_{\mathbf{w} \in W_0} \{M(S, \{\mathbf{x}\}, \mathbf{w})\}.$$

<sup>118</sup> Here  $D$  takes a positive value for  $\mathbf{x}$  outside  $\text{conv}(S, W_0)$  and a negative  
<sup>119</sup> value for  $\mathbf{x}$  strictly inside  $\text{conv}(S, W_0)$ . The closer  $\mathbf{x}$  is to  $\partial\text{conv}(S, W_0)$ ,

120 the smaller the absolute value is. Note that the general calculation problem  
 121 of  $D(\mathbf{x}, \partial\text{conv}(S, W_0))$  is known to be NP-hard [12, 13]. In our case, since we  
 122 will use a finite set  $W \subset W_r$ , we can calculate the distance  $D(\mathbf{x}, \partial\text{conv}(S, W))$   
 123 in a linear order of  $|W|$  as

$$D(\mathbf{x}, \partial\text{conv}(S, W)) = \max_{\mathbf{w} \in W} \{M(S, \{\mathbf{x}\}, \mathbf{w})\}. \quad (1)$$

124 An example is shown in Fig. 3. The method for calculating the distance  
 125  $D(\mathbf{x}, \partial\text{conv}(S, W))$ , where  $\text{conv}(S, W)$  is the approximate convex hull speci-  
 126 fied by  $W$ , is shown in Fig. 3. When  $\mathbf{x}$  is outside  $\text{conv}(S, W)$  such as  $\mathbf{x}_1$  and  
 127  $\mathbf{x}_2$ , then the distance takes a positive value; otherwise, e.g.,  $\mathbf{x}_3$ , the distance  
 128 takes a negative value. It is noted that the distance calculation can be made  
 129 by Eq.(1) regardless of whether the point is inside or outside  $\text{conv}(S, W)$ .  
 130 The difference from the true distance becomes smaller if we use a large set  
 131 of  $W$ . For example, the nearest point of  $\mathbf{x}_1$  on  $\partial\text{conv}(S, W)$  is  $\mathbf{p}$ , but the  
 132 distance calculated by (1) is the distance to the support plane  $h_1$ .

### 133 **3. Approximation of a class region by reflective convex hulls**

134 In this section, we introduce an algorithm to find a set of approximated  
 135 reflective convex hulls for a class.

<sub>136</sub>    3.1. Algorithm

<sub>137</sub>    The following algorithm is applied class by class.

<sub>138</sub>    1. Let  $S$  be the positive sample set of a target class and  $T$  be the negative

<sub>139</sub>    sample set of other classes. Let  $\mathcal{C} = W = \emptyset$ . Let  $L$  be an upper bound

<sub>140</sub>    of the number of convex hulls and  $K$  be the number of normal vectors,

<sub>141</sub>    that is, the number of facets of a convex hull.

<sub>142</sub>    2. Find random  $K$  pairs of  $\mathbf{x}$  ( $\in S$ ) and  $\mathbf{y}$  ( $\in T$ ) and put  $\mathbf{w} = \frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|}$  in  
<sub>143</sub>    set  $W$ .

<sub>144</sub>    3. Repeat  $L$  times the following steps 4–5.

<sub>145</sub>    4. Let  $U = \emptyset$ . According to a random presentation order of positive  
<sub>146</sub>    samples, add positive samples  $\mathbf{x}$  to  $U$  as long as  $\text{conv}(U \cup \{\mathbf{x}\}, W) \cap T =$   
<sub>147</sub>     $\emptyset$ .

<sub>148</sub>    5. Add the obtained  $\text{conv}(U, W)$  into  $\mathcal{C}$ , unless it is already in  $\mathcal{C}$ .

<sub>149</sub>    6. Select a subset of  $\mathcal{C}$  by a greedy set cover procedure for all positive  
<sub>150</sub>    samples. That is, the largest member of  $\mathcal{C}$  is chosen first and then the  
<sub>151</sub>    member including the largest number of uncovered samples is chosen,  
<sub>152</sub>    and so on.

<sub>153</sub>    By this procedure, we have at most  $L$  approximated convex hulls that

<sub>154</sub>    include samples of one class only. It should be noted that each convex hull

155 includes the positive samples maximally in Step 4.

156 Let us analyze the time complexity first. The dominant step is Step

157 4 in which we add a positive sample  $\mathbf{x} \in S$  to the current subset  $U$  in

158 order to expand the previous convex hull to  $\text{conv}(U \cup \{\mathbf{x}\}, W)$ . This needs

159  $O(K)$  steps. Then we check if this convex hull is still exclusive, that is, if it

160 does not include any negative sample  $\mathbf{y} \in T$ . To do this, we need  $O(Kn^-)$

161 steps for  $n^-$  negative samples ( $n^- = |T|$ ). Since we need to scan every

162 positive sample,  $O(Kn^+n^-)$  is necessary in Step 4 to obtain one exclusive

163 and maximal convex hull for  $n^+$  positive samples ( $n^+ = |S|$ ). In Step 3,

164 we repeat these procedures  $L$  times, so that the total order is  $O(LKn^+n^-)$ .

165 Here the time complexity with respect to dimensionality  $m$  is omitted, but

166 it is linear because we can obtain all necessary values by the inner product

167 between two vectors in an  $m$ -dimensional space. Regarding  $L$  as a constant

168 and regarding  $n^+$  and  $n^-$  as the same complexity as the total sample number

169  $n$ , we have  $O(Kn^2m)$ . It is also noted that the complexity to measure the

170 distance of a query sample to one convex hull with  $K$  facets is  $O(Km)$ . In

171 summary, this algorithm needs  $O(Kn^2m)$  for training and  $O(Km)$  for testing.

172 Since the worst-case complexity of SVM procedure with  $p$  support vectors is

173  $O(n^3m)$  ( $O(nmp)$  with SMO solver) for training and  $O(mp)$  for testing [14],

<sub>174</sub> the proposed method is comparable with SVM.

<sub>175</sub> The space complexity is the memory amount required to hold all the  
<sub>176</sub> convex hulls. It is  $O(LK + mK)$ , because one convex hull is expressed by  
<sub>177</sub>  $K$  values to specify the distances from the origin in  $K$  directions, and each  
<sub>178</sub> direction is specified by an  $m$ -dimensional vector.

<sub>179</sub> A controllable nature of complexity is a characteristic of the proposed  
<sub>180</sub> algorithm. By decreasing the value of  $K$  we can reduce the time and space  
<sub>181</sub> complexities at the expense of approximation precision to the exact convex  
<sub>182</sub> hull. As will be discussed later, approaching to the exact convex hull is  
<sub>183</sub> not necessarily recommended. We need to choose the best model in sample-  
<sub>184</sub> limited situations.

<sub>185</sub> One problem may arise when noisy samples are included in the training  
<sub>186</sub> data. It is easily understood that only one negative sample greatly breaks  
<sub>187</sub> down a convex hull covering many positive samples if it appears inside the  
<sub>188</sub> convex hull. Any kind of region-based algorithm shares the same problem. In  
<sub>189</sub> the proposed method, we cope with this problem by a simple technique. We  
<sub>190</sub> carry out the above algorithm twice over all classes. The first one is carried  
<sub>191</sub> out in order to find noisy samples and the second one is carried out in order  
<sub>192</sub> to find class regions. After the first round, we regard the samples that are

193 included in only small convex hulls as *noisy samples* and remove them for the  
194 second round. The second round is carried out without such noisy samples.  
195 To distinguish noisy samples from normal samples, we use a threshold  $\alpha$ . If  
196 a convex hull is smaller than  $\alpha$  in the *size*, that is, if the ratio of the number  
197 of samples included in the convex hull to the number of positive samples is  
198 less than  $\alpha$  ( $= 1\%$  in the following experiments), all samples included in it  
199 are judged as noisy.

200 To emphasize the number of facets, we call an approximated reflective  
201 convex hull with  $K$  directional unit vectors a *K-directional approximated*  
202 *reflective convex hull* (in short,  $K$ -ARCH). A convex hull might have less  
203 than  $K$  facets, but we use this terminology whenever  $|W| = K$ . The number  
204 of vectors  $K$  is controllable, so the (time) complexity can be decreased if  
205 we use  $K$ -ARCHs instead of the exact (reflective) convex hulls. Roughly  
206 speaking, as  $K$  increases, the corresponding  $K$ -ARCH approaches the true  
207 reflective convex hull. Class assignment of an unknown sample is carried out  
208 on the basis of distance to the nearest boundary of  $K$ -ARCHs.

209 Extension to multi-class classification is straightforward. Changing the  
210 role of positive samples and negative samples class by class, we obtain  $c$  sets  
211 of  $L$   $K$ -ARCHs for  $c$  classes. Since the classification of a query sample is

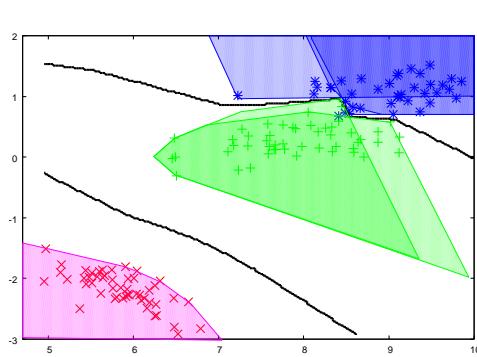


Figure 4: Approximated class regions by  $K$ -directional reflective convex hulls ( $K = 50$ ). The dataset is 3-class 2-dimensional K-L expanded `iris` dataset. The decision boundary is also shown.

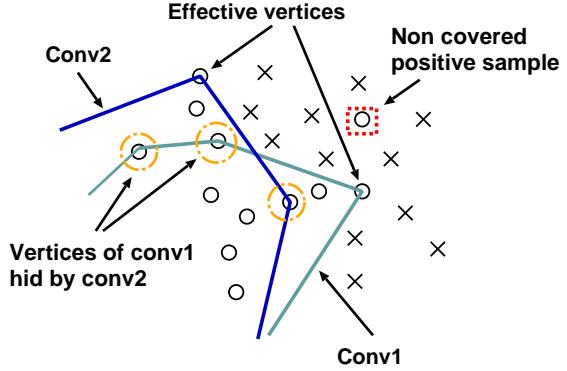


Figure 5: Effective vertices necessary for LOO upper bound. The circled vertices are not necessary.

212 made by the closest convex hull, the time complexity is  $O(cLKn)$ .

213 An example is shown in Fig. 4. Figure 4 shows the approximate class  
214 regions for 2-dimensional `iris` data consisting of two principal components.

215 The original data is taken from a database [15]. The value of  $K$  is 50. There  
216 are five  $K$ -ARCHs covering all samples. In Fig. 4, we can confirm that 1) the  
217 approximated convex hulls are larger than the corresponding exact convex  
218 hulls, 2) the convex hulls are often unbounded, and 3) the decision boundary  
219 is taken to keep the maximum margin locally.

220 **4. Model Selection**

221 In reference [8], we used a fixed value of  $K$  for each dataset. It is expected  
222 to raise the performance of the proposed  $K$ -ARCH algorithm by choosing the  
223 optimal value of  $K$ . Unfortunately, it is not so easy to solve this optimiza-  
224 tion problem because the performance evaluation is theoretically difficult.  
225 Therefore, in this paper we propose a model selection procedure to find a  
226 suboptimal value of  $K$ . In the following, we will show the efficacy of this  
227 model selection procedure in the process speed and in the classification per-  
228 formance of the resultant classifier.

229 *4.1. Estimation of generalization error*

230 As a measure of testing error, we use the LOO (Leave-One-Out) error  
231 rate. Leaving one sample out, we construct a classifier from the remaining  
232 samples to test the left-out sample, and continue this procedure to estimate  
233 the classification performance. As is well known, the LOO rate is almost  
234 unbiased (Luntz and Brailovsky Theorem [16]), but it requires the building  
235 of  $n$  classifiers for  $n$  samples. Hence, we consider an upper bound of LOO  
236 that can be easily obtained without reconstruction of classifiers. Let  $\epsilon_{LOO}$   
237 be the LOO error rate,  $V$  be the set of vertices of all convex hulls and  $Z$  be

238 the set of samples that are outside all convex hulls. If a single convex hull is  
 239 taken in each class,  $\epsilon_{LOO}$  is bounded by the sum of  $|V|$  and  $|Z|$  divided by  $n$ ,  
 240 because removal of samples inside a convex hull does not affect the classifier  
 241 design. However, in the case of more than one convex hull being taken in a  
 242 class, a vertex of one convex hull can be covered by another. Figure 5 shows  
 243 such a case. Such vertices can be safely ignored when counting for possible  
 244 errors. We call vertices “*effective vertices*” if they are not covered by the  
 245 other convex hulls. Let  $V_e$  be the set of effective vertices on the boundary  
 246 of the approximated class region as the union of the convex hulls. Then we  
 247 have an upper bound by

$$\epsilon_{LOO} \leq \frac{|V_e| + |Z|}{n}. \quad (2)$$

248 Here, in the numerator we count the number of samples that can change the  
 249 classifier if one of them is left out of training.

250 We use the value on the right-hand side of (2). Obviously, there is a trade-  
 251 off between  $|V_e|$  and  $|Z|$ . The greater is the number of convex hulls, the larger  
 252 is the size of  $|V_e|$ , while the smaller is the number of samples that are not  
 253 included in any convex hull, the smaller is the size of  $|Z|$ . In addition, a large  
 254 number of  $K$  increases the number of approximate convex hulls because a  
 255 more acute angle is allowed in a convex hull with a large variety of direction

Table 1: The statistics of datasets.

Dataset	#classes	#attributes	#samples
balance-scale	3	4	625
diabetes	2	8	768
ecoli	8	8	336
glass	6	10	214
heart-statlog	2	13	270
ionosphere	2	34	351
iris	3	4	150
sonar	2	60	208
wine	3	13	178

256 vectors. We therefore use the right-hand side term of (2) as our criterion.

257 *4.2. Experiments*

258 To construct  $W$  of unit vectors, we used  $n_p$  positive samples and  $n_p(c-1)$   
 259 negative samples, so that  $K = n_p^2(c-1)$  unit vectors were chosen randomly,  
 260 where  $c$  is the number of classes. In the following, we changed the value of  
 261  $n_p$  in  $[1, 50]$ , thus,  $K$  in  $[c-1, 2500(c-1)]$ .

262 We used 9 datasets taken from the UCI machine learning repository [15].  
263 In Table 1, the number of classes, the number of attributes (the dimension-  
264 ality of the feature space), and the number of samples are shown. It is noted  
265 that in some of them the dimensionality  $m$  is too large (34 in `ionosphere`  
266 and 60 in `sonar`) to find the exact convex hull of these sizes of the train-  
267 ing sample set. We increased the value of  $K$  until  $K$  reached the maximum  
268 value. For each value of  $K$ , we repeated the algorithm 10 times to reduce  
269 the effect of other random factors. Among 10 trials with a fixed value of  $K$ ,  
270 we chose the best case in which the LOO estimate (2) took the minimum  
271 value. The recognition rate was estimated by 10-fold cross validation. The  
272 loop number  $L$  of the algorithm (Steps 4 and 5) was set to  $L = 20$ . That is,  
273 the number of convex hulls was limited to 20 in each class. The K-ARCH  
274 algorithm was implemented in *C++* and SVMTorch algorithm [17] was used  
275 for SVMs. Both algorithms were executed on a PC with Intel Core 2 Quad  
276 Q8200 2.33GHz CPU and 3GB RAM.

277 *4.3. Results*

278 We compared the proposed *K*-ARCH algorithm with an SVM in which an  
279 RBF (Radial Basis Function) kernel with the default values of parameters  
280 (standard deviation  $\sigma = 10.0$  and soft margin parameter  $\gamma = 100.0$ ) was

Table 2: Recognition rates of SVM and  $K^*$ -ARCH where  $K^*$  is optimal in our model selection criterion and  $|V_e|$  is the number of effective vertices.

Dataset	Classifier		#SV or $ V_e $	
	SVM	$K^*$ -ARCH ( $K^*$ )	SVM	$K^*$ -ARCH
balance-scale	<b>93.2</b>	90.3 (98)	255.0	200.1
diabetes	64.1	<b>75.0</b> (2401)	1310.0	483.3
ecoli	79.8	<b>83.0</b> (252)	385.7	144.6
glass	<b>66.3</b>	63.6 (180)	336.9	138.0
heart-statlog	59.3	<b>63.7</b> (2401)	479.4	201.8
ionosphere	<b>94.0</b>	90.9 (196)	132.5	331.2
iris	<b>98.0</b>	95.3 (18)	54.0	20.2
sonar	77.4	<b>80.4</b> (121)	214.0	429.3
wine	72.5	<b>87.0</b> (3200)	447.2	67.7
average	78.3	<b>81.0</b>	401.6	224.0

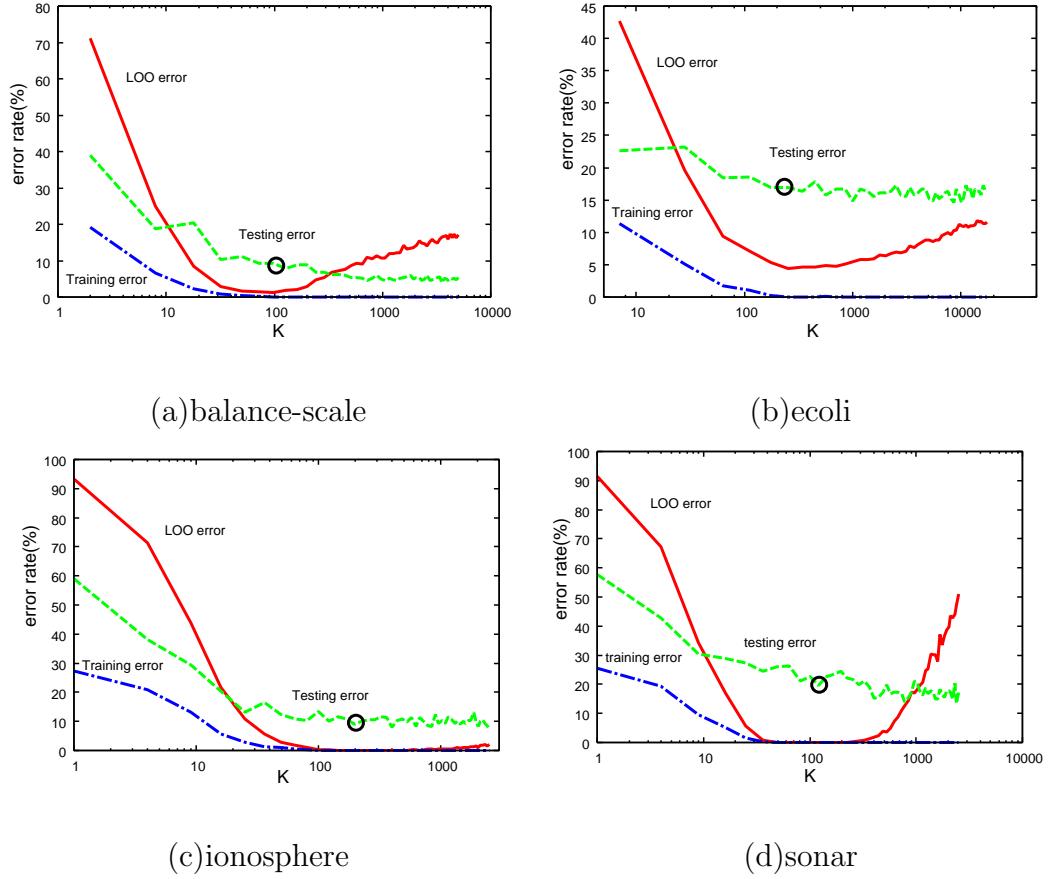


Figure 6: Error rate of  $K$ -ARCH as the number  $K$  of facets increases on four datasets. The three curves show the estimated LOO error (right-hand term of Ineq. (2)), the training error and testing error. The circled testing-error corresponds to the optimal value of  $K$  chosen by the minimum LOO error.

281 used [17]. For the  $K$ -ARCH algorithm, we used  $K^*$ -ARCHs for classification,  
282 where  $K^*$  is the value of  $K$  attaining the minimum LOO upper bound of (2).  
283 The results are shown in Table 2.

284 In Table 2, it is noted that  $K^*$ -ARCH outperforms SVM in more than  
285 half of the cases (5/9). Note that three problems of the remaining 4/9 cases  
286 are all easier or well-separated class problems. Indeed, in these three prob-  
287 lems (`balance-scale`, `iris` and `ionosphere`), the recognition rates are over  
288 90%. This might mean that  $K$ -ARCH tends to generate a slightly more com-  
289 plicated decision boundary compared with that of SVM. Note that a large  
290 number  $K^*$  is chosen for more difficult problems. This implies that  $K$ -ARCH  
291 formed a complex boundary. Note also that the number of (effective) vertices  
292 is often less than the number of support vectors. This means that  $K^*$ -ARCH  
293 often has higher sparsity than SVM.

294 We can see the details in some datasets in Fig. 6. From Fig. 6, we see that  
295 after reaching the optimal value  $K^*$ , the testing error is no longer significantly  
296 reduced. In general, a model selection criterion is expected to form a valley  
297 to simulate the testing error, but this is not the case. This implies that  
298  $K$ -ARCH does not change its decision boundary even if the model becomes  
299 more complicated than necessary. We can interpret this phenomenon as

follows. Even if the facets increase more than necessary, they are limited in  
 the location opposite to the decision boundary. Such a situation is shown  
 in Fig 7. As can be seen in Fig 7, such a redundant non-reflective support  
 plane can be generated by some noisy samples. In this sense, we have to be  
 careful about the value of  $\alpha$  used for the judgment of noisy samples. The  
 curve of the testing error fluctuates a little as  $K$  increases. This is because  
 small convex hulls with very acute angles can be generated when  $K$  is large.

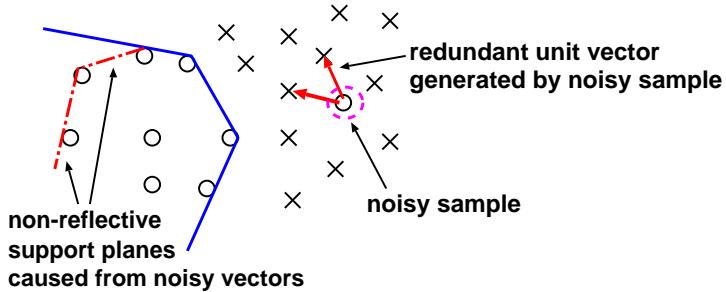


Figure 7: Redundant faces generated by noisy vectors.

307

## 308 5. Comparison with SVM

309 It is worth comparing the proposed  $K$ -ARCH algorithm with SVM to  
 310 make clear the advantages and disadvantages of  $K$ -ARCH algorithm. They  
 311 are summarized in Table 3.

Table 3: Comparison between the proposed  $K$ -ARCH and SVM. In below,  $n$  is the number of training samples and  $m$  is the dimensionality.

Evaluation item	Classifier	
	SVM	$K$ -ARCH
Maximization of margin	in the kernel space	in the original space
Adaptation to nonlinearity	kernel	multiple $K$ -ARCHs
Extension to multi-class	one-against-other	natural
Classification performance	high	high
Training time	medium ( $O(nm) - O(n^3m)$ )	slow ( $O(n^2m)$ )
Testing time	fast ( $O(m)$ )	fast ( $O(m)$ )

- 312      The largest difference is the space where the margin is taken. An SVM  
 313    uses a kernel to solve non-linearly separable problems. It is guaranteed to  
 314    keep the maximum (linear) margin in the mapped space with the kernel, but  
 315    the (non-linear) margin in the original feature space is not always maximized.  
 316    On the other hand, the  $K$ -ARCH algorithm keeps (locally linear) maximum  
 317    margins in the original space, though the margins are measured between  
 318    closest pairs of  $K$ -ARCHs of two different classes. As a result, in some cases,  
 319    the latter finds a better decision boundary than the former as shown in Fig. 8.  
 320    Note that the margin is not maximized by SVM in Fig. 8.

Table 4: Execution time of K-ARCH and SVM in training and testing phases. The time (seconds) is the results of 10 trials of the same task.

Dataset	K-ARCH		SVM	
	Training	Testing	Training	Testing
balance	30.40	0.04	1.59	0.07
diabetes	25.03	0.02	1.71	0.22
ecoli	29.36	0.20	0.84	0.11
glass	7.83	0.06	0.95	0.05
heart	3.07	0.01	0.44	0.10
ionos	6.36	0.02	0.44	0.11
iris	1.38	0.00	0.41	0.05
sonar	2.73	0.01	0.50	0.10
wine	2.32	0.02	0.56	0.06

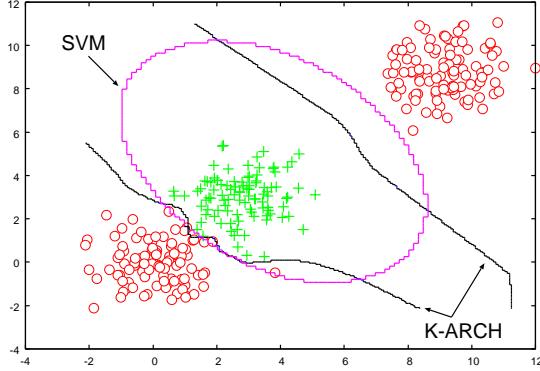


Figure 8: Decision boundaries by SVM with RBF and  $K$ -ARCH algorithms. Samples of class 1 are denoted by circles and samples of class 2 by crosses.

321     The  $K$ -ARCH algorithm is also advantageous to SVM in extension to  
 322     multi-class problems. Typically one-against-other strategy is adopted to use  
 323     two-class SVM for multi-class problems, though some multi-class SVMs have  
 324     been studied (e.g., see [18]). It is known, however, that such an extension  
 325     does not work in some cases [19]. On the contrary,  $K$ -ARCH algorithm can  
 326     naturally deal with multi-class problems because it finds a set of  $K$ -ARCHs  
 327     in each class and classify a sample to the class of the closest  $K$ -ARCH.

328     Note that we can see the example of Fig. 8 as a 3-class problem by re-  
 329     garding two clusters of one class (denoted by circles) as two different classes.  
 330     Then the problem of inappropriate decision boundary of SVM still remains  
 331     because of the one-against-other strategy.

332 In the usage of SVMs, the choice of a kernel and its parameters can be  
333 problematic. In this respect,  $K$ -ARCH is advantageous because the critical  
334 parameter is only the number  $K$  of facets.

335 In time complexity, SVM is superior to  $K$ -ARCH algorithm, especially  
336 in the training phase. In the testing phase, they are almost the same. The  
337 execution time on the nine datasets is shown in Table 4, from which we can  
338 confirm that  $K$ -ARCH is comparable to SVM and sometimes even faster than  
339 SVM in the testing time. It is also noted that parallelization of  $K$ -ARCH  
340 algorithm is easy due to its simple procedure.

341 **6. Discussion**

342 It is well known, in the case of two-class problems, that the hyper-plane  
343 of linear SVM is equivalent to the bisector between the closest points on  
344 the boundaries of the convex hulls, each of which encloses the training sam-  
345 ples of one class, if the training sample sets are linearly separable [2]. Re-  
346 cently a similar relationship was reported even for soft-margin SVM using  
347 *reduced convex hulls* [20, 21]. Our classifier is almost identical to SVM when  
348 two classes are linearly separable (e.g., see Fig 4) in which one convex hull  
349 is sufficient for one class. An advantage of our approach is the capability of

350 handling non-linearly separable cases in which more than one convex hull is  
351 required for one class. Even in such cases, our  $K$ -ARCH algorithm maximizes  
352 the margin locally with a relatively smooth decision boundary (see Fig. 8).  
353 In this respect, our classifier is one of the maximum margin classifiers.

354 One drawback of the proposed algorithm is that it includes many ran-  
355 dom factors producing different classifiers. The direction vectors in  $W$  are  
356 randomly chosen and the set of  $K$ -ARCHs has a randomness because of the  
357 random algorithm. This randomness does not affect much to the resultant  
358 classifier, but needs an appropriate control to reduce its bad effect.

359 **7. Conclusion**

360 In this paper, a model selection procedure for a family of polyhedron clas-  
361 sifiers has been proposed. The family is based on polyhedral class regions  
362 close to the convex hulls of some parts of training samples. In the polyhedron  
363 family here, complexity mainly comes from the number of facets and vertices  
364 of each polyhedral region. The selection procedure employs an upper bound  
365 of the LOO error for time reduction and showed satisfactory results in se-  
366 lecting a good model. However, more sophisticated, hopefully theoretically  
367 established, procedure to choose the optimal value of  $K$  is desired. Another

368 direction of study is to find another criterion instead of margin maximization.  
369 Once nonlinear margins are taken into consideration, it is not easy even to  
370 define the nonlinear margins appropriately. As long as considering the ge-  
371 ometric margin between training samples of two classes, the largest margin  
372 is kept by the nearest neighbor rule. However, it is not the best classifier  
373 when only a limited number of samples is given for training. The authors  
374 are currently considering a way to widen the margin taken by the proposed  
375 algorithm. The result would be helpful for seeking a better criterion.

376 **Acknowledgment**

377 This work was partly supported by Grant-in-Aid for Scientific Research  
378 (C) (No. 10213216) of the Japan Society for the Promotion of Science.

379 **References**

- 380 [1] V. N. Vapnik, The nature of statistical learning theory, Springer, 1996.  
381 [2] D. Zhou, B. Xiao, H. Zhou, R. Dai, Global geometry of svm classifiers,  
382 Institute of Automation, Chinese Academy of Sciences, Technical report,  
383 AI Lab.

- 384 [3] B. V. Dasarathy, Nearest neighbor(NN) norms: NN pattern classifica-
- 385 tion techniques, IEEE Computer Society Press, Los Alamitos, 1991.
- 386 [4] M. Kudo, M. Shimbo, Approximation of class region by convex hulls,
- 387 Technical report of IEICE. PRMU 100 (507) (2000) 1–6, (In Japanese).
- 388 [5] E. Jeff, New lower bounds for convex hull problems in odd dimensions,
- 389 SIAM Journal of Computing 28 (4) (1999) 1198–1214.
- 390 [6] B. McBride, G. L. Peterson, Blind data classification using hyper-
- 391 dimensional convex polytopes, in: Proceedings of the 17th International
- 392 FLAIRS Conference, 2004, pp. 520–525.
- 393 [7] A. Blumer, A. Ehrenfeucht, D. Haussler, W. M.K., Occam’s razor, Read-
- 394 ings in machine learning (1990) 201–204.
- 395 [8] M. Kudo, I. Takigawa, A. Nakamura, Classification by reflective con-
- 396 vex hulls, Proceedings of the 19th International Conference on Pattern
- 397 Recognition (ICPR2008), Tampa, Florida, USA.
- 398 [9] G. I. Nalbantov, P. J. F. Groenen, J. C. Bioch, Nearest convex hull
- 399 classification, Econometric Institute Report.
- 400 [10] H. Cevikalp, D. Larlus, M. Neamtu, B. Triggs, F. Jurie, Manifold based

401 local classifiers: Linear and nonlinear approaches, Journal of Signal Pro-  
402 cessing Systems (2008) 1–13.

403 [11] P. K. Ghosh, K. V. Kumar, Support function representation of convex  
404 bodies, its application in geometric computing, and some related rep-  
405 resentations, Computer Vision and Image Understanding 72 (3) (1998)  
406 379–403.

407 [12] O. L. Mangasarian, Polyhedral boundary projection, SIAM Journal on  
408 Optimization, 9 (4) (1999) 1128–1134.

409 [13] P. Gritzmann, V. Klee, Computational complexity of inner and outer  
410  $\bar{\jmath}$ -radii of polytopes in finite-dimensional normed spaces, Mathematical  
411 Programming 59 (1) (1993) 163–213.

412 [14] D. Decoste, B. Schölkopf, Training invariant support vector machines,  
413 Machine Learning 46 (1) (2002) 161–190.

414 [15] A. Asuncion, D. Newman, UCI machine learning repository (2007).

415 [16] A. Luntz, V. Brailovsky, On estimation of characters obtained in statis-  
416 tical procedure of recognition, Technicheskaya Kibernetika 3 (6).

- 417 [17] R. Collobert, S. Bengio, SVMTorch: support vector machines for large-  
418 scale regression problems, Journal of Machine Learning Research 1  
419 (2001) 143–160.
- 420 [18] K. Crammer, Y. Singer, On the algorithmic implementation of multi-  
421 class kernel-based vector machines, The Journal of Machine Learning  
422 Research 2 (2002) 265–292.
- 423 [19] Y. Lee, Y. Lin, G. Wahba, Multicategory support vector machines, the-  
424 ory, and application to the classification of microarray data and satellite  
425 radiance data, Journal of the American Statistical Association 99 (2004)  
426 67–81.
- 427 [20] S. Theodoridis, M. Mavroforakis, Reduced convex hulls: A geometric  
428 approach to support vector machines, IEEE Signal Processing Magazine  
429 24 (3) (2007) 119–122.
- 430 [21] B. Goodrich, D. Albrecht, P. Tischer, Algorithms for the computation of  
431 reduced convex hulls, AI 2009: Advances in Artificial Intelligence (2009)  
432 230–239.