# Image Recognition Algorithms Based on the Representation of Classes by Convex Hulls

Anatoly Nemirko(✉)

Saint Petersburg Electrotechnical University “LETI”, Saint Petersburg 197376, Russia
apn-bs@yandex.ru

**Abstract.** Various approaches to the construction of pattern recognition algorithms based on the representation of classes as convex hulls in a multidimensional feature space are considered. This trend is well suited for biometrics problems with a large number of classes and small volumes of learning samples by class, for example, for problems of recognizing people by faces or fingerprints. In addition to simple algorithms for a point hitting a convex hull, algorithms of the nearest convex hull with different approaches to assessing the proximity of a test point to the convex hull of classes are investigated. Comparative experimental results are given and the advantages and disadvantages of the proposed approach are formulated.

**Keywords:** Intelligent systems · Computational geometry · Pattern recognition · Nearest convex hull classification · Linear programming · Automatic medical diagnostics

## 1 Introduction

Algorithms of *k*-nearest neighbors (*k*NN) [1] due to their simplicity and efficiency are successfully used in many multiclass problems of pattern recognition and image analysis. The *k*NN rule is that it assigns a test object to the most common class among its *k* nearest neighbors. Despite the great success achieved in many applied problems, its application is limited by the well-known drawbacks: the limitation of the dimension of the feature space and the number of objects in the training set. For *k*NN, the problems of the influence of noise and small training sets remain.

Another well-known method in machine learning is the support vector machines (SVM) [2]. Designed for binary classification, it is widely used in all kinds of pattern recognition problems. SVM is highly generalizable and handles high-dimensional data easily. However, it does not directly solve multiclass problems.

In this paper, we consider the nearest convex hull (NCH) classifier, which is conceptually related to both *k*NN and SVM. NCH is an intuitive geometric classification method that assigns a test point to the class whose convex hull is closest to it. The useful properties of NCH are:

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12665, pp. 44–50, 2021.
https://doi.org/10.1007/978-3-030-68821-9_4

1. NCH classifies multi-class problems easily in a straightforward way.
2. NCH is well suited for biometrics problems with a large number of classes and small volumes of learning samples by class: problems of recognizing people by faces [7] or fingerprints.
3. Since, NCH is resistant to the problems of small learning samples and noise, since the elimination of one point of an element of the convex set does not affect or only locally affects the entire convex hull.

This article is organized as follows. First, we give a definition of a convex hull, then a classifier is formulated based on the description of classes in the form of convex hulls in general, approaches to determining the proximity of a test point to the convex hull of a class are given, which are used in classification algorithms for the nearest convex hull. These approaches include: the SVM-based method used in [3], the Lightweight nearest convex hull method (LNCH), based on calculating the projection of the class onto the direction from the test point to the centroid of the class [4] and the method [6] based on the application of linear programming [5]. All of the above approaches provide only an approximate estimate of the proximity parameter. At the end of the article, a classification algorithm based on the use of linear programming is described, the results of experimental studies are given, and a conclusion is formulated, thanks and a list of references are given.

## 2 The Definition of a Classifier Based on the Convex Hull Representation of Classes

Let the learning set of one class have the form $\mathrm{X} = \{\mathbf{x}_i,\ \mathbf{x}_i \in R^n,\ \ i = 1, 2, \ldots, k\}$. Then the convex hull generated by this set is defined as

$$conv\,(\mathrm{X}) = \left\{ \mathbf{v} : \mathbf{v} = \sum_{i=1}^{k} a_i \mathbf{x}_i, \quad 0 \le a_i, \quad \sum_{i=1}^{k} a_i = 1, \quad \mathbf{x}_i \in X \right\}$$

where $a_i$ - are scalar non-negative coefficients. For $m$ classes, we have $m$ sets of $\mathrm{X}_i,\ i = 1, 2, \ldots, m$ and, accordingly, $m$ convex hulls $conv(\mathrm{X}_i),\ i = 1, 2, \ldots, m$. The simplest classifier for an arbitrary requested point $\mathbf{x}$ will use the following rule.

**Rule A**:
If $\mathbf{x}$ is inside $conv\left(\mathrm{X}_p\right)$, then it belongs to class $p$, otherwise it belongs to another class or its affiliation is undefined.

Leaving aside the question of how to determine the location of $\mathbf{x}$ "inside $conv\left(\mathrm{X}_p\right)$", you need to decide what to do if $\mathbf{x}$ is inside several convex hulls at the same time (the case of their intersection) or $\mathbf{x}$ is not in any convex hull. In these cases, a way out can be found by introducing the concept of the distance from $\mathbf{x}$ to the convex hull of the $i$-th class $d_i(\mathbf{x}, conv(\mathrm{X}_i))$. Then the classification algorithm can be formulated as follows.

**Rule B**

- If $\mathbf{x}$ is inside only one convex hull $conv\left(\mathrm{X}_p\right)$, then it belongs to class $p$.

- If **x** is outside the convex hulls of all classes, then it belongs to the class $i$, distance $d_i(\mathbf{x}, conv(\mathrm{X}_i))$ to which is less.
- If **x** is inside several convex hulls, then its membership is undefined.

In this case, two points present difficulties. It is necessary to determine how to check the location of the test vector **x**: inside $conv(\mathrm{X})$ or outside it. In addition, it is necessary to remove the uncertainty of the membership of the vector **x** in the case when it enters into two or more convex hulls.

## 3 Determining the Distance from the Test Point to the Convex Hull

If **x** is not inside $conv(\mathrm{X})$, the minimum distance $d(\mathbf{x}, conv(\mathrm{X}))$ can be determined as in [3], using the support vector machines (SVM). In the case when **x** is inside $conv(\mathrm{X})$, the SVM method does not give an exact solution due to the dependence of the obtained weight vector on classification errors.

In papers [4, 6], approaches to the approximate definition of $d(\mathbf{x}, conv(\mathrm{X}))$ are proposed regardless of the location of the test point.

### 3.1 Lite Distance Determination Method

The paper [4] describes a lightweight method for determining the distance regardless of the location of **x** in relation to $conv(\mathrm{X})$. For data **x** and $conv(\mathrm{X})$, the method consists in analyzing the projections of nodes $conv(\mathrm{X})$ onto the unit vector **u**, directed from test point **x** to centroid **c** of class X, i.e. $\mathbf{u} = (\mathbf{c} - \mathbf{x}) \big/ norm(\mathbf{c} - \mathbf{x})$. The analysis is carried out by the following algorithm.

**DISTANCE algorithm**

```
Input: $\mathbf{u}, \mathbf{x}, \mathrm{X}_i$ {direction vector, test point, set of elements of the $i$-th class}
Output: $\mathrm{F}_i$, $d_i\left(\mathbf{x}, conv(\mathrm{X}_i)\right)$ {intersection mark, distance to the convex hull of the
$i$-th class}. Distance $d_i\left(\mathbf{x}, conv(\mathrm{X}_i)\right)$ is defined along the direction vector $\mathbf{u}$.
    1. For the set $\mathrm{X}_i$, form the matrix $\mathbf{X}_i$
    2. If $\min\left(\mathbf{u}^T\mathbf{X}_i\right) \le \mathbf{u}^T\mathbf{x} \le \max\left(\mathbf{u}^T\mathbf{X}_i\right)$
               $\mathrm{F}_i = 1;\ d_i\left(\mathbf{x}, conv(\mathrm{X}_i)\right) = \left|\mathbf{u}^T\mathbf{x} - \min\left(\mathbf{u}^T\mathbf{X}_i\right)\right|$
          else if $\mathbf{u}^T\mathbf{x} \le \min\left(\mathbf{u}^T\mathbf{X}_i\right)$
               $\mathrm{F}_i = 0;\ d_i\left(\mathbf{x}, conv(\mathrm{X}_i)\right) = \left|\mathbf{u}^T\mathbf{x} - \min\left(\mathbf{u}^T\mathbf{X}_i\right)\right|$
          else $\mathrm{F}_i = 0;\ d_i\left(\mathbf{x}, conv(\mathrm{X}_i)\right) = \left|\mathbf{u}^T\mathbf{x} - \max\left(\mathbf{u}^T\mathbf{X}_i\right)\right|$
          end
```

As a result of executing this algorithm for a given test point and $m$ classes we obtain $m$ pairs $(\mathrm{F}_i,\ d_i),\ i = 1, 2, \ldots, m$, where $\mathrm{F}_i$ – is the $i$-th class intersection mark with $\mathbf{x}$ ($\mathrm{F}_i = 0$, if $\mathbf{x}$ is outside $conv(\mathrm{X}_i)$, and $\mathrm{F}_i = 1$, if $\mathbf{x}$ is inside $conv(\mathrm{X}_i)$), $d_i$ - parameter of distance to $conv(\mathrm{X}_i)$. Further, the following rule is used for classification.

**Rule B**

- If $\mathbf{x}$ is inside only one convex hull $conv(\mathrm{X})$, then it belongs to this class,
- otherwise, if it is inside several convex hulls, then it belongs to the class into which it entered most deeply, i.e. for which $d(\mathbf{x},\ conv(\mathrm{X}))$ is maximum,
- otherwise ($\mathbf{x}$ is outside the convex hulls of all classes) it belongs to the class for which $d(\mathbf{x},\ conv(\mathrm{X}))$ is minimal

### 3.2 A Method Based on Linear Programming

In paper [5], the optimal solution $z^*$ of the following LP1 linear programming problem was considered. Given set $\mathrm{X} = \{\mathbf{x}_i,\ \mathbf{x}_i \in R^n,\ \ i = 1, 2, \ldots, m\}$ and point $\mathbf{b} \in \mathrm{R}^n$, and the origin must be inside $conv(\mathrm{X})$ ($\mathrm{P} = conv(\mathrm{X})$).

LP1

$$z = \min \sum_{i=1}^{m} \lambda_i$$

Provided that

$$\sum_{i=1}^{m} \lambda_i \mathbf{x}_i = \mathbf{b},$$

$$\lambda_i \geq 0,\ i = 1, \ldots, m,$$

where $\mathbf{b}$ is an arbitrary nonzero vector.

For this problem, the following statement was formulated and proved [5].

If $z^*$ is an optimal solution to LP1 problem for some $\mathbf{b} \neq 0$, then

1. $z^* < 1$, if and only if $\mathbf{b}$ is inside P
2. $z^* = 1$, if and only if $\mathbf{b}$ is on the boundary of P
3. $z^* > 1$, if and only if $\mathbf{b}$ is outside P

This can be illustrated with the help of Fig. 1. The above statement creates the prerequisites both for determining that the vector $\mathbf{x}$ is inside the convex hull and for estimating the proximity of a given point to the convex hull of the class based on solving a linear programming problem [6]. Theoretical considerations and experiments show that the ratio of the distance from a point to the convex hull D (along the ray from the origin to the test point $\mathbf{b}$) to the length of the vector $\mathbf{b}$ is equal to the ratio $|z^* - 1|$ to $z^*$. That is, it is true regardless of the location of the test point ($\mathrm{F} = 1$ or $\mathrm{F} = 0$)

$$\frac{\mathrm{D}}{\|\mathbf{b}\|} = \frac{|z^* - 1|}{z^*}$$

whence follows

$$\mathrm{D} = \|\mathbf{b}\| \cdot \left|z^* - 1\right| / z^*$$

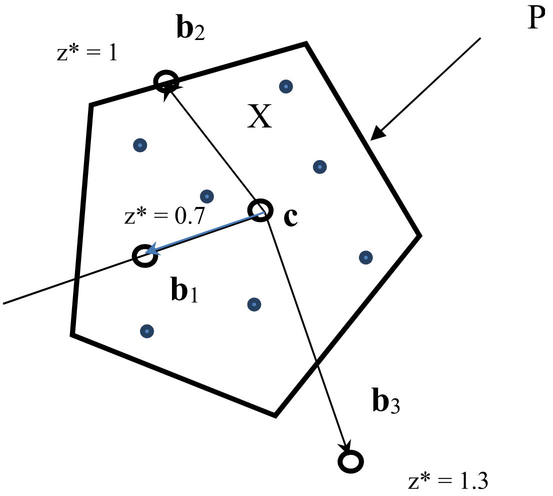

**Fig. 1.** The figure shows: a set of points X in a multidimensional space, P is a convex hull, **c** is the origin, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ are test points (vectors), $z^*$ are the optimal values of the objective function $z$ for these points (examples).

Since the ray used is not perpendicular to the facet it "pierces", the determined distance D is an approximation of the true distance to the convex hull. In order to increase the accuracy of such approximation, it is advisable to align the origin of coordinates with the centroid of the set X before measuring.

## 4 The Nearest Convex Hull Algorithm Based on the Linear Programming Method

The principle of measuring the distance from the tested (test) point **x** to the convex hull described above can be used to construct a multi-class classifier of the nearest convex hull. Before solving the LP1 problem for each class of points $X_i$, it is necessary to place the origin at the centroid point of this class. After finding the optimal solution to LP1 problem, we introduce a label F, which shows the location of **x**: inside or outside the given convex hull. F = 0 if $z^* \geq 1$ (point outside or on the boundary of the convex hull), and F = 1 if $z^* < 1$ (point inside the convex hull). Then for a given **x** and $X_i$ we get a pair (F, $D_i$).

For a given **x** and $m$ classes $X_i$, $i = 1, 2, \ldots, m$ we get $m$ pairs ($F_i$, $D_i$), $i = 1, 2, \ldots, m$. Further, the classification is carried out according to the following decision rule.

1. If no pair contains F = 1, then the number of the recognized class is given by formula $class(\mathbf{x}) = \underset{i=1,2,\ldots m}{\arg\min}\, d_i(\mathbf{x},\ conv(X_i))$.
2. If only one pair contains F = 1, then the number of the recognized class is equal to the index of this pair.

3. If several pairs (possibly all) contain F = 1 and the indices of these pairs form a set $G$, then the number of the recognized class is chosen from these classes so that $class(\mathbf{x}) = \underset{i \in G}{\arg\max}\, d_i(\mathbf{x},\ conv(\mathrm{X}_i))$. The penetration of the test point into this class turns out to be the greatest.

As a description of classes, it is better to use not the original sets $\mathrm{X}_i$, but only the vertices of their convex hulls. To do this, it is necessary to calculate the set of vertices of the convex hulls of the classes from the initial learning set. This is an easier task. This significantly reduces the running time of the linear programming algorithm. Using the vertex list at the learning stage makes it more accurate to determine the center of the convex hull, which can affect the accuracy of determining the distance D.

## 5 Experimental Research

The described algorithm for the classification of the nearest convex hull using linear programming was tested on a two-class problem of medical diagnostics of breast cancer [8]. A sample of 683 people was used (444 cases of healthy and 239 cases of sick). To form the control sample, the classes were divided in half. Convex hulls were constructed from the first halves of the samples. The second half of the samples were used as test samples. The average classification error on test samples was 2.15%, which is better than the same indicator when solving this problem with other classification algorithms [6].

## 6 Conclusion

This paper discusses approaches to pattern recognition algorithms based on the representation of classes as convex hulls in a multidimensional feature space. For classification algorithms based on the nearest convex hull, different approaches to assessing the proximity of a test point to a convex hull are considered: based on the analysis of the directed penetration depth and based on the use of linear programming. Both approaches estimate the distance from the test point to the convex hull along the ray from the class centroid to the test point. As shown by the results of experiments, the method using linear programming is characterized by high recognition quality. It is easy to implement, does not require any parameter setting, and can be easily used to solve multiclass problems, especially biometrics problems with a large number of classes and small volumes of learning samples by class.

**Acknowledgments.** The study is performed with partial support from the Russian Foundation for Basic Research (RFBR), grants 19-29-01009 and 18-29-02036.

## References

1. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)

2. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Knowl. Discov. Data Min. **2**, 121–167 (1998)
3. Nalbantov, G., Smirnov, E.: Soft nearest convex hull classifier. In: Coelho, H. et al. (eds.) Proceeding of the 19th European Conference on Artificial Intelligence (ECAI-2010), (IOS Press, 2010), pp. 841–846 (2010). https://doi.org/10.3233/978-1-60750-606-5-841
4. Nemirko, A.P.: Lightweight nearest convex hull classifier. Pattern Recogn. Image Anal. **29**(3), 360–365 (2019). https://doi.org/10.1134/s1054661819030167
5. Dulá, J.H., Helgason, R.V.: A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space. Eur. J. Oper. Res. **92**(2), 352–367 (1996). https://doi.org/10.1016/0377-2217(94)00366-1
6. Nemirko, A., Dulá, J.: Machine learning algorithm based on convex hull analysis. In: 14th International Symposium «Intelligent System» , INTELS 2020, 14–16 December 2020, Moscow, Russia (in press) (2020)
7. Zhou, X., Shi, Y.: Nearest neighbor convex hull classification method for face recognition. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2009. LNCS, vol. 5545, pp. 570–577. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01973-9_64
8. Breast Cancer Wisconsin (Original) Data Set. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)