A greedy ball algorithm for classification

Birgit Graf
Institut für angewandte Forschung, Fachhochschule Konstanz
Brauneggerstr. 55, D-78462 Konstanz, Germany
birgitg@fh-konstanz.de

and

Arnold Neumaier
Institut für Mathematik, Universität Wien
Strudlhofgasse 4, A-1090 Wien, Austria
neum@cma.univie.ac.at

Abstract. We propose a classification scheme that generates a set of balls separating points belonging to different points sets in n-dimensional space such that all points of a given training set are classified correctly. The method is very fast in training. Applications discussed include breast cancer prognosis and character recognition.

Keywords: supervised classification, pattern recognition, breast cancer prognosis, diabetes prognosis, character recognition, speech recognition.

1991 MSC Classification: primary 68T10, secondary 62H30.

1 Introduction

For supervised pattern recognition, a large number of classification algorithms are available in the literature. Good overviews are given in the recent books by Michie et al. [8] (elementary), Ripley [9] (advanced), and Devroye et al. [5] (probabilistic analysis). Recently, Bennett & Mangasarian [1, 2] introduced a recursive classification method where piecewise linear separators are constructed using linear programming techniques, and applied it very successfully to breast cancer prognosis (Mangasarian et al. [7]).

In the following, we show how to avoid the linear programming steps by using as separators balls instead of hyperplanes. This makes training very fast, even for large data sets. For example, the character recognition problem by HO & BAIRD [4] discussed later has 10^4 training vectors of dimension 2304, and 10^5 vectors in the test set.

We give a brief description of our training and test method: To separate points belonging to c different classes $S_\lambda, \lambda = 1, \ldots, c$ in an n-dimensional real space, we use a *training set* consisting of points whose class membership is known, and use it to determine a set of balls (or complements of balls) that separate all training points belonging to different classes. After training is completed, these balls are used to assign arbirtary points to some class.

We separate one class from the remaining one, i.e., for each $\lambda \in \{1, \ldots, c\}$ in turn, we put $\mathcal{A} = S_\lambda$ and $\mathcal{B}$ as the union of the other classes. Hence, separation of more than two classes is reduced to the case of two classes $\mathcal{A}$ and $\mathcal{B}$. Throughout this paper, only the separation of $\mathcal{A}$ is described.

Training produces a sequence $K_1, K_2^*, K_3^*, \ldots$ of balls containing no part of $\mathcal{A}$ and at least one point of $\mathcal{B}$ in the interior, touching a fixed number N of points of $\mathcal{A}$, called *touch points*. Points of $\mathcal{B}$ that are not inside the *first ball* K_1 are covered by a *correcting ball* K_i^*. This is done until all points of $\mathcal{B}$ are in the interior of (at least) one of these ball. In order to have better generalization properties, the radii are readjusted later. Points to be classified are considered to belong to $\mathcal{A}$ if they are in none of the readjusted balls.

The performance of the classifier is evaluated by computing approximate error rates (i.e., proportions of test samples that are misclassified). This is done by either using a specified test set or by repeated k-fld cross-validation, using for training only part of the data with known class membership.

The organization of this paper is as follows. In Section 2 we formulate the tasks of the main function `greedyball` and introduce a characterization of a ball, namely its *sign*. Then the main steps of the classifier are given. Section 3 treats details of `greedyball`, and gives a summary of the function `greedyball` that performs the training. How to classify new samples is explained in Section 4, and some computational results are reported in Section 5. Proofs of the results stated in Section 3 are given in the Appendix.

Notation. The pattern sets $\mathcal{A}$ and $\mathcal{B}$ consist of m and l points in the n-dimensional real space $\mathrm{I\!R}^n$, respectively.

$$\mathrm{mean}(\mathcal{A}) = \frac{1}{m} \sum_{y \in \mathcal{A}} y$$

is the mean vector of $\mathcal{A}$. $||y||$ is the Euclidean norm of a vector $y \in \mathrm{I\!R}^n$. For a ball K with center μ and radius r we define the *pseudo distance* of y to the boundary of the ball by $d_y := d_y(K) := ||y - \mu||^2 - r^2$, and $\mathrm{sgn}(d_y)$ denotes the sign of d_y.

$$\mathcal{S}' := \{y \in \mathcal{A} : d_y \neq 0\}$$

is the set of points of $\mathcal{A}$ that do not touch the ball and y_* is a point of $\mathcal{B}$ that has to be covered by a correcting ball.

2 The main steps of the classifier

The main ingredient of the classifier is a function $K =$ `greedyball`$(\mathcal{A}, y)$ that provides a center μ and radius r of a ball K having the following properties:

(P1) The interior of K is disjoint to $\mathcal{A}$.

(P2) The boundary of K contains $N \leq \min\{n+1, m-1\}$ points of $\mathcal{A}$, the *touch points*.

(P3) The ball covers y (if this optional parameter is given).

Every ball K produced by `greedyball` is characterized by a *sign* $\mathrm{sgn}(K)$ that determines the interior of the ball: We set $\mathrm{sgn}(K) = 1$ if $\mathrm{sgn}(d_y) > 0$ holds for all $y \in \mathcal{S}'$, otherwise we set $\mathrm{sgn}(K) = -1$. (Note that it is sufficient to check $\mathrm{sgn}(d_y)$ for one $y \in \mathcal{S}'$ because (P1) implies that all $y \in \mathcal{S}'$ are on the same side of the ball, hence $\mathrm{sgn}(d_y)$ is the same for all $y \in \mathcal{S}'$). The *interior* of K is then given by $\{y \in \mathrm{I\!R}^n : \mathrm{sgn}(K)\,\mathrm{sgn}(d_y) < 0\}$, i.e., a point y is *covered by a ball* K if $\mathrm{sgn}(K)\,\mathrm{sgn}(d_y) < 0$ holds.

With this, the main steps of the classifier `greedy` are:

STEP 1: K_1=`greedyball`$(\mathcal{A})$: Find a first ball K_1 with the above properties.

STEP 2: Find one point $y_* \in \mathcal{B}$ that is in none of the balls found so far. If no such point exists, the separation of $\mathcal{A}$ is completed and we continue with STEP 4.

STEP 3: K^*=`greedyball`$(\mathcal{A}, y_*)$: Find a correcting ball K^* that covers y_*.

STEP 4: To get a robust classification and to avoid problems caused by round-off errors, we readjust the radii in such a way that negative balls become larger and positive balls become smaller without losing points. This is done by taking as new radius

$$r_a := r + c(\overline{r} - r) \quad \text{for} \quad c \in (0, 1)$$

where $\overline{r}$ is given by $\min_{y \in E} ||y - \mu||$ for a negative ball and $\max_{y \in E} ||y - \mu||$ for a positive one, and E denotes the set of points of $\mathcal{B}$ lying in the interior of the ball.

Figures 1 and 2 show the original and readjusted balls produced by the classifier to separate two classes $\mathcal{A} = S_1$ and $\mathcal{B} = S_2$. For simplicity, only the balls that separate $\mathcal{A}$ from $\mathcal{B}$ are drawn. K is the first ball (with a negative sign) and K^* is a correcting ball (with positive sign).

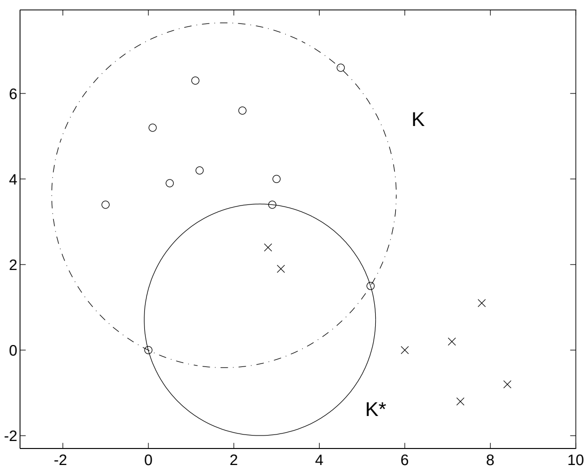


Figure 1: The original balls.

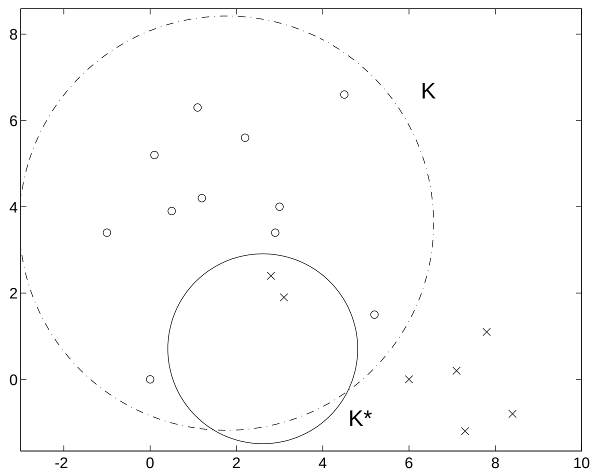


Figure 2: The readjusted balls.

(o: points of $\mathcal{A}$, x: points of $\mathcal{B}$.)

An arbitrary point y belongs to $\mathcal{A}$ if y is neither in the interior of the first ball (i.e., *outside* the dashed line), nor in the interior of a correcting ball belonging to this class. Hence, all points labeled 'o' belong to $\mathcal{A}$. The test methology is described in Section 4 in detail.

3 The function greedyball

Here we specify the function `greedyball` that provides balls having the properties (P1)–(P3). We start with a ball having *one touch point* with $\mathcal{A}$ by choosing μ =mean(A) and $r = \max_{y \in \mathcal{A}} \parallel y - \mu \parallel$ for the first ball and $\mu = y_*$ and $r = \min_{y \in \mathcal{A}} \parallel y - \mu \parallel$ for a correcting ball (i.e., we begin with a negative first ball and a positive correcting ball). Then we move the center and modify the radius in such a way that the resulting new ball satisfies the following conditions:

(C1) We get at least one new touch point, and the touch points found so far are preserved.

(C2) The interior of the ball is disjoint to $\mathcal{A}$.

(C3) y_* and all $y \in \mathcal{S}'$ must lie on opposite sides of the ball.

A ball is called *valid* if these conditions are satisfied. Note that only correcting balls have to satisfy the third condition. Hence, a first ball is valid if (C1) and (C2) hold.

Notation. K denotes a valid ball with center μ and radius r having t touch points $y_1, \ldots, y_t$ with $\mathcal{A}$. The $n \times t$ matrix S is given by $S = (y_1 - \mu, \ldots, y_t - \mu)$, and $S_{\bullet k}$ denotes the k-th column of S. $\bar{\mu}$ and $\bar{r}$ are the center and the radius of a ball $\bar{K}$ having $t+1$ touch points with $\mathcal{A}$.
We now describe how to get a valid ball $\bar{K}$.

If we take $\bar{\mu} = \mu + \alpha p$ with a *search direction* $p \in \mathbb{R}^n$ and a *step size* $\alpha \in \mathbb{R}$, we get

$$\begin{aligned} \parallel y - \bar{\mu} \parallel^2 &= \parallel (y - \mu) - \alpha p \parallel^2 \\ &= \parallel y - \mu \parallel^2 - 2\alpha (y - \mu)^T p + \alpha^2 \parallel p \parallel^2 \end{aligned} \tag{1}$$

for some $y \in \mathbb{R}^n$. This will be used to get a condition for p, necessary to ensure that all touch points are preserved (Lemma 1) and to obtain the new radius $\bar{r}$ (cf. (5)).
For a touch point $y = y_k$ we get $\parallel y_k - \mu \parallel^2 = r^2$. Then (1) reduces to

$$\parallel y_k - \bar{\mu} \parallel^2 \; = r^2 - 2\alpha (y_k - \mu)^T p + \alpha^2 \parallel p \parallel^2$$

for touch points y_k. To preserve all touch points of the old ball K,

$$\| y_k - \bar{\mu} \|^2 = \bar{r}^2$$

has to be satisfied for all $k = 1, \ldots, t$. With $y_k - \mu = S_{\bullet k}$ we get the following condition:

$$\| y_k - \bar{\mu} \|^2 \quad = r^2 - 2\alpha S_{\bullet k}^T p + \alpha^2 \| p \|^2 = \bar{r}^2 \ . \tag{2}$$

The following result shows how to choose the search direction p so that this condition is satisfied.

Lemma 1 *If the search direction p satisfies*

$$S^T p = e, \tag{3}$$

then all touch points found so far are preserved. Here, $e \in \mathbb{R}^t$ is a vector of ones. In particular, (3) *is satisfied if we choose*

$$p = S(S^T S)^{-1} e. \tag{4}$$

In the following the search direction p is determined by (4). With (2) and (4) the new radius $\bar{r}$ is

$$\bar{r}^2 = r^2 - 2\alpha + \alpha^2 \| p \|^2 \ . \tag{5}$$

With this choice of p all touch points found so far are preserved, but to get a new touch point and to ensure (C2) and (C3), a suitable step size α has to be determined. This will be discussed in the following.

The step size. Points of $\mathcal{S}'$ must lie outside the new ball $\bar{K}$. This leads to the condition

$$y \in \mathcal{S}' \Rightarrow \quad \| y - \bar{\mu} \|^2 \leq \bar{r}^2$$

for a negative ball and

$$y \in \mathcal{S}' \Rightarrow \quad \| y - \bar{\mu} \|^2 \geq \bar{r}^2$$

for a positive ball. With (1) and (5) we get for a negative ball the condition

$$y \in \mathcal{S}' \Rightarrow \quad \| y - \mu \|^2 - 2\alpha (y - \mu)^T p + \alpha^2 \| p \|^2 \leq r^2 - 2\alpha + \alpha^2 \| p \|^2,$$

hence

$$y \in \mathcal{S}' \Rightarrow \quad \| y - \mu \|^2 - 2\alpha((y-\mu)^T p - 1) \leq r^2 \,. \tag{6}$$

For a positive ball we get

$$y \in \mathcal{S}' \Rightarrow \quad \| y - \mu \|^2 - 2\alpha((y-\mu)^T p - 1) \geq r^2. \tag{7}$$

To get a new touch point, (6) or (7) have to hold with equality for some $y \in \mathcal{S}'$. Hence, for (C1) it is enough to compute

$$\alpha_y := \frac{1}{2} \frac{\| y - \mu \|^2 - r^2}{(y-\mu)^T p - 1} \tag{8}$$

for one $y \in \mathcal{S}'$ and to take $\alpha = \alpha_y$. But choosing this step size does not always lead to a ball satisfying (C2) and (C3). To avoid this problem, α_y will be evaluated *for all* $y \in \mathcal{S}'$. Doing this, we get a finite set $I = \{\alpha_y \mid y \in S'\}$ of possible step sizes α. In the following we show how to choose $\alpha \in I$ to get a valid ball. We note, that the search direction $p = S(S^T S)^{-1} e$ depends only on the existing touch points, so the new ball $\bar{K}$ is completely determined by the choice of α.

The distance of y to the boundary of the new ball $\bar{K}$ is $\bar{d}_y(\alpha) := \| y - \bar{\mu} \|^2 - \bar{r}^2$, where $\bar{r}$ is given by (5) and $\bar{\mu} = \mu + \alpha p$. $\bar{v}_y$ and v_y denote the sign of $\bar{d}_y(\alpha)$ and d_y, respectively. These distance functions have the following property:

Lemma 2 *For any $y \in \mathcal{S}'$, the function $\bar{d}_y(\alpha)$ is linear in α and the zero of $\bar{d}_y(\alpha)$ supplies a new touch point.*

The meaning of these distances will be clear by considering Figure 3. $\mathcal{S}'$ consists of the four points $\{y_0, y_1, y_2, y_3\}$ and the distances $\bar{d}_i(\alpha) := \bar{d}_{y_i}(\alpha)$ are depicted for all $y \in \mathcal{S}'$. Taking $\alpha = \alpha_3$ leads to a ball where y_0 and y_1 are on the same side, but y_2 is on the opposite side of the ball. An equivalent situation arises by taking $\alpha = \alpha_0$. Hence, these step sizes are invalid, and only $\alpha = \alpha_1$ or $\alpha = \alpha_2$ ensure (C2). In Figure 4 we can also see that a ball may change the sign, i.e., that a positive ball becomes negative or reverse. If we choose $\alpha = \alpha_2$ then $\bar{v}_y = 1$ holds for all $y \in \mathcal{S}'$. $\alpha = \alpha_1$ leads to a sign change of the ball since $\bar{v}_y = -1$ holds for all $y \in \mathcal{S}'$.

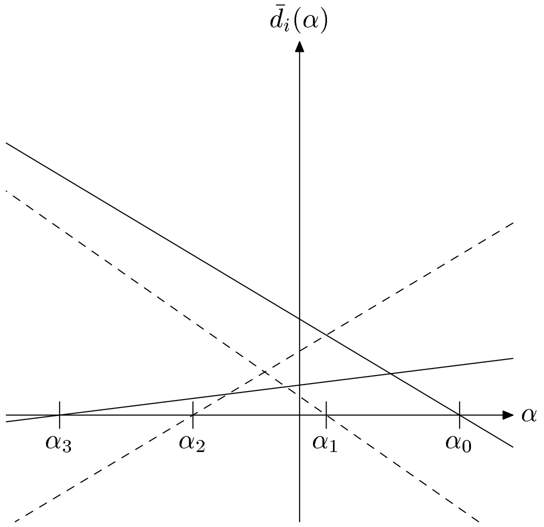


Figure 3: Valid and invalid step sizes.

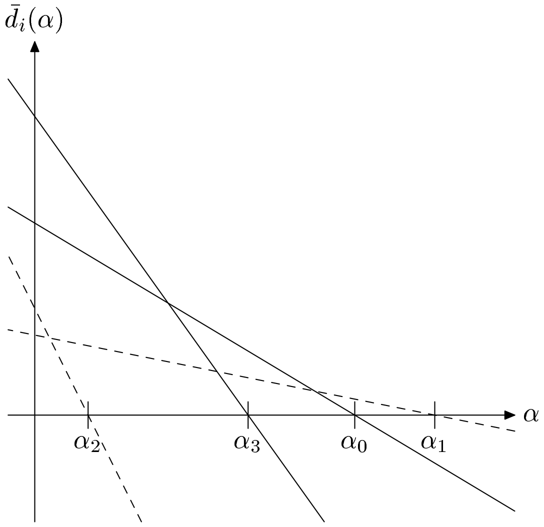


Figure 4: Sign change of a ball.

The following result shows the connection between v_y and $\bar{v}_y$.

Lemma 3 *For all $y \in \mathcal{A}$,*

$$\bar{v}_y = v_y \ \operatorname{sgn}(1 - \frac{\alpha}{\alpha_y}).$$

In the following we set $I := \{\alpha_y \mid y \in \mathcal{S}'\}$ and write I^+ and I^- for the set of positive and negative $\alpha_y \in I$, respectively. If all $\alpha_y \in I$ have the same sign, we define

$$\alpha_1 := \max\{\alpha_y \mid \alpha_y \in I^k\} \text{ and } \alpha_2 := \min\{\alpha_y \mid \alpha_y \in I^k\}, \tag{9}$$

where I^k is I^+ if $I^- = \emptyset$ and I^- if $I^+ = \emptyset$. Otherwise, α_1 and α_2 are given by

$$\alpha_1 := \min\{\alpha_y \mid \alpha_y \in I^+\} \text{ and } \alpha_2 := \max\{\alpha_y \mid \alpha_y \in I^-\}. \tag{10}$$

We first show which step sizes lead to (C1) and (C2).

Lemma 4

(i) (C1) holds with an arbitrary $\alpha \in I$.

(ii) The only step sizes leading to the satisfaction of (C2) are $\alpha = \alpha_1$ or $\alpha = \alpha_2$.

(C3) is only needed for a correcting ball, hence Lemma 3.4 gives the valid step sizes for the first ball.

We now investigate the step sizes that ensure (C3). We first consider the case that all α_y have the same sign.

Lemma 5 (Valid step sizes if all $\alpha_y \in I$ have the same sign).
Let $|I| \geq 2$. To ensure (C3), the valid step sizes are given by

$$\alpha = \begin{cases} \alpha_2 & \text{if } \alpha_{y_*} < 0 \qquad (\alpha_{y_*} \in]\alpha_1, 0[\ ) \\ \alpha_1 & \text{if } \alpha_{y_*} \in]0, \alpha_2[\quad (\alpha_{y_*} > 0 \ ) \\ \alpha_1 \text{ or } \alpha_2 & \text{if } \alpha_{y_*} \in]\alpha_1, \alpha_2[. \end{cases} \tag{11}$$

where α_1, α_2 are given by (9) and the conditions in parenthesis belong to the case $I^- \neq \emptyset$.

Lemma 6 (Valid step sizes if there are $\alpha_y \in I$ with different signs)
Let $|I| \geq 2$ and α_1, α_2 are given by (10). To ensure (C3) , the valid step sizes are given by

$$\alpha = \begin{cases} \alpha_2 & \text{if } \alpha_{y_*} \in \ ]0, \alpha_1[\\ \alpha_1 & \text{if } \alpha_{y_*} \in \ ]\alpha_2, 0[\\ \alpha_1 \text{ or } \alpha_2 & \text{if } \alpha_{y_*} \notin]\alpha_1, \alpha_2[. \end{cases} \tag{12}$$

Summary of the function greedyball. We now give a brief summary of the function `greedyball`. t denotes the number of existing touch points.

Step 1: Determine the ball with center $\mu(A)$ having one touch point y_1 with $\mathcal{A}$.
For this ball, the squared distances $D(y) = ||y - \mu||^2$ are evaluated for all $y \in \mathcal{A}$.
Step 2: If $t \geq N$, stop.
Step 3: If $D(y) = r^2$ for some $y \in \mathcal{S}'$, then y is also a new touch point. Set $t = t + 1$ and continue with Step 2.
Step 4: Set $S = (y_1 - \mu, \ldots, y_t - \mu)$, where y_i are the touch points with $\mathcal{A}$.
Step 5: Compute the search direction $p = S(S^TS)^{-1}e$ and $||p||^2$.
Step 6: For all $y \in \mathcal{S}'$: Evaluate $P(y) = (y - \mu)^T p$ and α_y with (8).
Step 7: Select the step size α using Lemma 4 for the first ball and Lemma 5 or Lemma

6 for a correcting ball.

STEP 8: The new center is $\mu = \mu + \alpha p$ and the new radius is $r^2 = r^2 - 2\alpha + \alpha^2 \parallel p \parallel^2$. Set $t = t + 1$. y_t denotes the new touch point.

STEP 9: Update the distances $D(y)$ by $D(y) = D(y) - 2P(y) + \alpha^2 \parallel p \parallel^2$ for all $y \in \mathcal{S}'$.

4 Classifying a new sample point

To assign an arbitrarily $y \in \mathbf{R}^n$ with the balls $\{K_i^\lambda, i = 1, \ldots, n_\lambda, \lambda = 1, \ldots, c\}$ produced by training to one class S_λ, a c-dimensional vector $\mathbf{\Lambda} \in \mathbf{N}^c$ is used, where we set $\mathbf{\Lambda}_\lambda = \lambda$ if the balls of S_λ provide the information that y belongs to S_λ, otherwise we set $\mathbf{\Lambda}_\lambda = -\lambda$. Remember that points in the interior of the first ball K_1^λ or of a correcting ball $K_j^\lambda, j \in \{2, \ldots, n_\lambda\}$ do not belong to S_λ, and points outside the first ball K_1^λ belong to S_λ if and only if they are not covered by a correcting ball K_j^λ.

With this in mind, classification of y proceeds as follows: For $\lambda = 1, \ldots, c$ we set $\mathbf{\Lambda}_\lambda = \lambda$ if and only if $y \notin K_j^\lambda$ holds for all $j = 1, \ldots, c$ and $\mathbf{\Lambda}_\lambda = -\lambda$ if y is inside the first ball K_1^λ or $y \in K_j^\lambda$ holds for one $j \in \{2, \ldots, n_\lambda\}$. (Note, that if y is in the interior of K_1^λ we need not look for correcting balls $K_j^\lambda, j \in \{2, \ldots, n_\lambda\}$). If $\mathbf{\Lambda}_\lambda > 0$ holds for exactly one $\lambda \in \{1, \ldots, c\}$, then y can definitely be assigned to S_λ. In the other cases we assign y to S_λ, where λ is the lexicographically first index with smallest $d_y(K_1^\lambda)$ (*post processing*).

Figure 5 shows three balls that separate two classes S_1, S_2, where K_1^1 and K_1^2 are the first balls of S_1 and S_2 respectively (both having negative signs), and K_2^1 is a (positive) correcting ball belonging to S_1. The above described test method leads to:

$$\begin{aligned} \Lambda_A &= (1, -2), \quad \Lambda_B = (1, 2), \qquad \Lambda_C = (-1, 2), \\ \Lambda_D &= (-1, 2), \quad \Lambda_E = (-1, 2), \quad \Lambda_F = (-1, -2), \quad \Lambda_G = (-1, -2). \end{aligned}$$

Hence, A will be assigned to S_1, C, D, E to S_2 and the remaining points will be assigned using the above described rule for postprocessing.

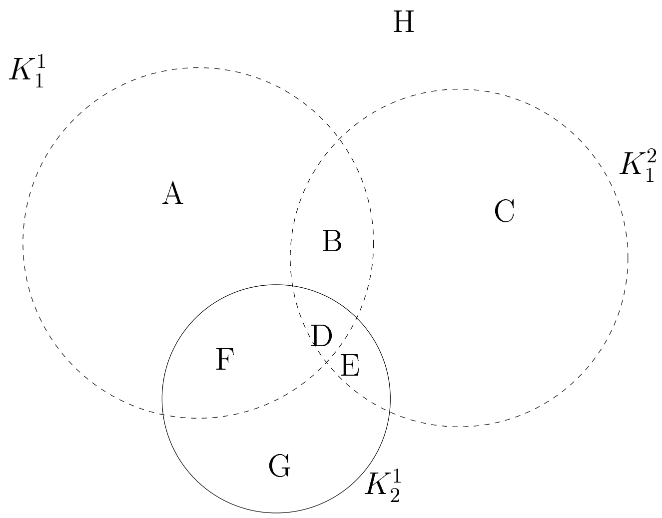


Figure 5: Classifying new points.

5 Numerical results

We now present computational results obtained with the classifier `greedy`. We consider four different problems: breast cancer prognosis, diabetes diagnosis, speech recognition (distinguishing between nasal and oral vowels), character recognition (distinguishing between c and e). In the first three cases we used 3 touch points, for the character recognition problem we used 10 touch points.

The *Wisconsin Breast Cancer Database* consists of 8 groups of patient data collected between Jan. 1989 and Nov. 1991 by Dr. William H. Wolberg at the University of Wisconsin Hospital in Madison (Bennett & Mangasarian [1]). The dataset is available per `ftp` from `ftp.cs.wisc.edu` in the directory `/math-prog/cpo-dataset/machine-learn/cancer1/datacum`. As in Benett & Mangasarian [1], we removed 16 points with missing attributes. The aim is to distinguish whether a cancer is benign or malignant.

Using the first group (consisting of 353 vectors of dimension 9) as training set and the other groups as test set, nearest-neighbor achieved an error rate of 1.8% and greedy an error rate of 1.5%. To compare the results for `greedy` with those in Bennett & Mangasarian [1] we made a 3-fold cross validation, i.e., we divided the whole set randomly into 3 sets and used each set in turn as test set and the remaining two as training set. Repeating this ten times, the averaged error rate of greedy was 3.2%. In Bennett & Mangasarian

[1] the averaged error rates are between 2.6 and 6.1%.

The *Pima Indian Diabetes Database* consists of 768 patient data vectors of dimension 8. The data set is available per `ftp` from `ftp.ics.uci.edu` in the directory `/pub /machine-learning-databases/diabetes`. The aim is to diagnose whether a patient shows signs of diabetes.
Using 576 training instances, the error rate of `greedy` is 27%. Nearest neighbor and ADAP (SMITH et al.[10]) achieve an error rate of 35 and 24% respectively.

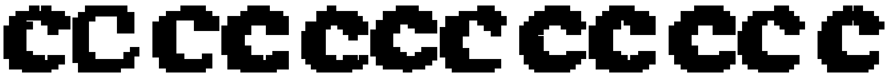
The *phoneme database* contains 5404 vectors of dimension 5, representing vowels coming from isolated syllables, where 5 different attributes were chosen to characterize each vowel. The aim is to distinguish between nasal and oral vowels. The data set is available per `ftp` from `ftp.dice.ucl.ac.be` in the directory `/pub/neuralnets/ELENA/databases /REAL/phoneme`.
The file `phoneme.txt` contains a description of the dataset and classification results. k-nearest neighbor with $k = 20$ gives an error rate of 14.2%. (c.f. `phoneme.txt`). We performed with `greedy` a random 20−fold cross validation three times and achieved an averaged error rate of 12.7%.

Character Recognition. HO & BAIRD [4] studied the influence of unrepresentative training data on the accuracy of an isolated character recognition problem, namely to distinguish the symbols c and e. The dataset can be requested by the authors of [4]. The characters are given as 48×48-bit matrices. As in HO & BAIRD [4], we used 10^4 points as training set and 10^5 points as test set.
In HO & BAIRD [4] the error rates reported are 0.21% (nearest-neighbor with Hamming distance), 0.12% (nearest-neighbor with normalized Hamming distance) and 0.32% (decision-tree). With `greedy`, we obtained an error rate of 0.19% where 102 cs and 91 es are misclassified. Some characters misclassified by `greedy` are illustrated below; their display suggests that adding some additional shape features would almost certainly reduce the classification error further.

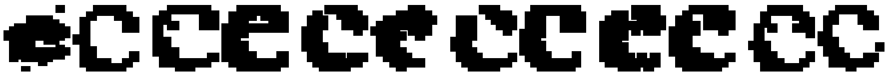
It is remarkable that, although we didn't select the data sets for this, greedy outperformed in each case nearest neighbors, a method that is provably nearly optimal (see e.g. Ripley

[9]) but for large data sets much slower than most other classification methods.

Some $c's$ classified as $e's$



Some $e's$ classified as $c's$



6 Conclusion

We presented a fast classification scheme that separates points of a given training set correctly. Our scheme exploit geometrical constructions to determine balls as separating surfaces. The algorithm has been tested on a number of real-life test problems, and achieved comparable or even superior results to other classification methods.

Appendix: Proofs

Proof of Lemma 1. If p satisfies $S^Tp = e$, (2) does not depend on k and $\| y_k - \bar{\mu} \|$ is the same for all y_k, $k = 1, \dots, t$. Thus all touch points found so far are preserved. In particular, if we choose $p = S(S^TS)^{-1}e$, we get $S^Tp = (S^TS)(S^TS)^{-1}e = e$, hence p satisfies (3).

Proof of Lemma 2. With $\bar{\mu} = \mu + \alpha p$ and (5) we get

$$\begin{aligned} \bar{d}_y(\alpha) &= \| y - \mu - \alpha p \|^2 - r^2 + 2\alpha - \alpha^2 \| p \|^2 \\ &= \| y - \mu \|^2 - 2\alpha(y-\mu)^T p + \alpha^2 \| p \|^2 - r^2 + 2\alpha - \alpha^2 \| p \|^2 \\ &= -2\alpha((y-\mu)^T p - 1) + \| y - \mu \|^2 - r^2 \, . \end{aligned} \tag{13}$$

Hence, these functions are linear in α. The zeros are given by (8), hence they supply a new touch point.

Proof of Lemma 3. With (8) and (13) we get

$$\bar{v}_y = \mathrm{sgn}(-2\alpha((y-\mu)^T p - 1) + \| y - \mu \|^2 - r^2$$

$$
\begin{aligned}
&= \operatorname{sgn}(-2\alpha\Big(\frac{\| y-\mu \|^2 - r^2}{2\alpha_y}\Big) + \| y-\mu \|^2 - r^2) \\
&= \operatorname{sgn}((\| y-\mu \|^2 - r^2)(1-\frac{\alpha}{\alpha_y})) \\
&= v_y \ \operatorname{sgn}(1-\frac{\alpha}{\alpha_y}) \ .
\end{aligned}
$$

Proof of Lemma 4. (i) is clear. To prove (ii), we first show that these step sizes ensure (C2). The old ball K is valid, hence v_y is the same for all $y \in \mathcal{S}'$ and we assume w.l.o.g. that $v_y = 1$ holds for all $y \in \mathcal{S}'$. With Lemma 3 we have to show that $\bar{v}_y = \operatorname{sgn}(1-\frac{\alpha}{\alpha_y})$ is the same for all $y \in \mathcal{S}'$. This is obvious if all α_y have the same sign, hence we consider the case that both I^+ and I^- are nonempty. Choosing $\alpha = \alpha_2 = \max\{\alpha_y \mid \alpha_y \in I^-\}$ we get $\alpha_y < \alpha < 0$ for all $\alpha_y \in I^-$, hence $\operatorname{sgn}(1-\frac{\alpha}{\alpha_y}) = 1$. Thus $\bar{v}_y = 1$ for all y with $\alpha_y \in I^-$. For $\alpha_y \in I^+$ we get $\alpha < 0 < \alpha_y$, hence $\operatorname{sgn}(1-\frac{\alpha}{\alpha_y}) = 1$. Therefore $\bar{v}_y$ is the same if we choose $\alpha = \alpha_2$. The case $\alpha = \alpha_1$ can be done analogous.
Now we show that there exists no other valid step sizes. We assume w.l.o.g. that all $\alpha_y \in I$ are positive, and that $v_y = 1$ holds for all $y \in \mathcal{S}'$. If $\alpha \in]\alpha_2, \alpha_1[$ leads to (C2), then we get $\bar{v}_1 = \operatorname{sgn}(1-\frac{\alpha}{\alpha_1}) = 1$ and $\bar{v}_2 = \operatorname{sgn}(1-\frac{\alpha}{\alpha_2}) = -1$. Hence y_1 and y_2 are on opposite sides, which contradicts the assumption.

Proof of Lemmas 5 and 6. Let I^+ and $I^- \neq \emptyset$ and $\alpha_{y_*} \in (0, \alpha_1)$. We assume first that $\alpha = \alpha_1$ is valid and that $v_{y_*} = -1$ and $v_y = 1$ holds for all $y \in \mathcal{S}'$. With $\alpha = \alpha_1$ we get $\bar{v}_y = \operatorname{sgn}(1-\frac{\alpha_1}{\alpha_y}) = 1$ for all $y \in \mathcal{S}'$ and $\bar{v}_{y_*} = -\operatorname{sgn}(1-\frac{\alpha_1}{\alpha_{y*}}) = 1$. Hence, y_* and $y \in \mathcal{S}'$ are on the same side of the ball, and α_1 is an invalid step size.
On the other hand, if we take $\alpha = \alpha_2$, we get $\bar{v}_{y_*} = -1$ and $\bar{v}_y = 1$ for all $y \in \mathcal{S}'$, hence, $\alpha = \alpha_2$ leads to (C3).
The other cases and Lemma 5 can be proved similarly.

References

[1] K.P. Bennett and O.L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets", *Optimization Methods and Software*, Vol. 1, 1992, pp.23–34.

[2] K.P. Bennett and O.L. Mangasarian, "Multicategory discrimination via linear programming", *Optimization Methods and Software*, Vol. 3, 1994, pp. 29–39.

[3] K.P. Bennett and O.L. Mangasarian, "Serial and Parallel Multicategory Discrimination", *SIAM J. Optimization*, Vol. 4, 1994, pp. 722–734.

[4] T.K. Ho and H.S. Baird, "Asymptotic accuracy of two-class discrimination", to appear in: *Proc. Third Ann. Symp. Document Anal. Information Retrieval*, Las Vegas, 1994, pp. 275–288.

[5] L. Devroye, L. Gyvrfi and G. Lugosi, "A Probabilistic Theory of Pattern Recognition" Springer, New York, 1996.

[6] K. Fukunaga, "Introduction to Statistical Pattern Recognition", Academic Press, New York, 1972.

[7] O.L. Mangasarian, W.N. Street and W.H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming", *Operations Research*, Vol. 43, 1995, pp. 570-577.

[8] D. Michie, D.J. Spiegelhalter and C.C. Taylor, "Maschine Learning, neural and Statistical Classification", Ellis Horwood, New York, 1994.

[9] B.D. Ripley, "Pattern Recognition and Neural Networks", Cambridge Univ. Press, 1996.

[10] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, and R.S. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus", *Proc. Symp. Comput. Appl. Medical Care*, IEEE Computer Society Press, 1988, pp. 261–265.