A Discrimination Method Using Quasi-Convex Hull

Maria MIROIU
University of Pitesti, Computer Science and Mathematics Department
E-mail: mjaica@math.upit.ro

Abstract: *A discrimination method is proposed in which a class region is taken as a set of quasi-convex hulls. This comparing with a subclass method (Kudo and Shimbo, 1989) which approximates a class region by a hyper-rectangles in such a way that a hyper-rectangle includes training samples in the class maximally and excludes those of other classes. The experimental results show the effectiveness of the proposed method.*
Keywords: *discrimination, classifiers, subclass method, hulls.*

Introduction

Classifiers can be evaluated according to their description capability of discrimination boundaries. Usually, a classifier determines its discrimination boundary on the basis of a small number of training samples.

When a true discrimination boundary is complicated, classification with poor description capability, such as linear classifier, cannot sufficiently approximate the boundary or even explain the training samples well. On the other hand, classifiers with sufficiently high description capability, such as a nearest neighbor classifier, can fully approximate any complicated discrimination boundary and explain the training samples perfectly. However, such classifiers tend to depend too much on training samples to explain other unknown samples. Thus, a classifier with a medium degree of description capability is desirable.

In the following is described the subclass method of Kudo and Shimbo and the discrimination method using quasi-convex hulls.

The subclass method

The subclass method proposed by Kudo and Shimbo uses a set of hyper-rectangles for approximating the true region of a class. Let us consider a class, denote a training sample belonging to this class by $x \in S^+$ and denote a training sample belonging to other classes by $y \in S^-$. It calls x a positive sample and y a negative sample.

A sample $x = (x_1, x_2, ..., x_d) \in \mathbf{R}^d$ is converted to a binary vector $b = (b_1, b_2, ..., b_D)$, $b_i \in \{0,1\}$, $(\forall) i = \overline{1, D}$ ($D>d$) as follows. In the ith axis, $i = \overline{1, d}$ we determine a sequence of p thresholds $\{t_j^i\}$, $j = \overline{1, p}$ with an equivalent interval,

$$\min_{x \in S^+ \cup S^-} x_i = t_1^i < t_2^i < ... < t_p^i = \max_{x \in S^+ \cup S^-} x_i$$

where $t_j^i - t_{j-1}^i = \dfrac{t_p^i - t_1^i}{p-1}$, $j = \overline{2, p}$.

Using these thresholds, we convert x to b of $D=2pd$ bits by:

$$x \to b = (b_p^1, ..., b_1^1, \overline{b_1^1}, ..., \overline{b_p^1}, b_p^2, ..., b_1^2, ..., \overline{b_1^d}, ..., \overline{b_p^d})$$

where $b_j^i = \begin{cases} 1, & if\ x_i \ge t_j^i \\ 0, & \text{otherwise} \end{cases}$ and $\overline{b_j^i}$ is the negation of b_j^i. Here b_j^i is for determining a lower bound in the ith axis and $\overline{b_j^i}$ is for determining an upper bound in the construction of a hyper-rectangle as described below.

A binary vector b is said to be included in another binary vector c if $b_i \le c_i, (\forall) i$.

We convert a subset X of S^+ to a binary B as follows. We convert every positive sample $x \in X$ to a binary vector b and then merge all bs into B applying the binary operation "and" ($\wedge$) over all bs.

The subclass method finds such a subset family O of S^+ of which element X is maximal in inclusion relation and the corresponding binary vector B is not included in any binary vector c converted from some $y \in S^-$. We call $\Omega(S^+, S^-)$ a subclass family and each element X a subclass. The concrete procedure for finding subclasses has been described by Kudo et al. (1996). The procedure is an iteration of a random selection of positive samples. Starting with an empty set, we choose a positive sample at random, and we add it to the set if addition keeps the exclusiveness (it will de defined later) against all the negative samples, otherwise we discard it. This procedure is repeated until all positive samples are examined, and all trails of the procedures are repeated.

In addition, a minimum set of subclasses is chosen in such a fashion that it covers all positive samples and each subclass of it is the largest for at least one positive sample. Instead of all subclasses, the minimum set of subclasses is used in the experiments.

Subclass X and its binary expression B are connected with an axis-parallel hyper-rectangle, Rect(X), in which each side of this is specified by the position of the first bit of 1 in each block consisting of p bits in B. These rectangles are on the discrete domain:

$$\{t_1^1, t_2^1, ..., t_p^1\} \times \{t_1^2, t_2^2, ..., t_p^2\} \times ... \times \{t_1^d, t_2^d, ..., t_p^d\}$$

Such a hyper-rectangle Rect(X) includes positive samples maximally and excludes all negative samples.

An unknown sample x is assign to a class if x is included in at least one hyper-rectangle Rect(X) of the class. If x is included in more than one hyper-rectangle belonging to different classes, then the hyper-rectangle for which x is located the most inside is used for the assignment of x, where how deep x (or its binary expression b) is located inside of a hyper-rectangle Rect(X) (or its binary expression B) is measures by:

$$\sum_{i=1}^{d} \min\{\, H(B,b) \text{ in } (2i-1)th \text{ block of } p \text{ bits}, H(B,b) \text{ in } (2i)th \text{ block of } p \text{ bits}\} ,$$

where $H(B,b)$ is the Hamming distance between B and b. This calculation is done only if B is included in b.

If x is not included in any hyper-rectangle, then the nearest hyper-rectangle Rect(X) is used for the assignment of x, where the nearness is measured by $H(B, B \wedge b)$. When B is included in b, or, equivalently, when x is included in Rect(X), then this distance becomes zero.

Quasi-convex hulls for subclasses

A set of hyper-rectangles of the original subclass method is insufficient for representing class regions in some cases. Therefore, we consider another monotonic expression of subclasses other than hyper-rectangles. Here, an expression of a set of samples is said to be monotonic when the expression always includes the expression of its subsets.

In the subclass method, a hyper-rectangle that is the minimum enclosure of some set of training samples includes any hyper-rectangle that is the minimum enclosure of its one subset. This is a necessary property for finding subclasses efficiently in the subclass method (Kudo et al., 1996).

A convex hull also has monotonicity. If we adopt a convex hull for expressing a subclass, we can represent rather complicates class regions efficiently. That is, the number of convex hulls needed to approximate a class region is less than that of hyper-rectangles needed to approximate the same class region.

We realize this idea approximately using technique of transformation of the domain, i.e., by an addition of features. After adding some features, the ordinal procedure of the subclass method is applied to the samples expressed by the extended features.

Let's explain this procedures by taking a simple example in a two-dimensional Euclidean space (i.e. d=2). A transformation

$x_3 = \cos q \cdot x_1 + \sin q \cdot x_2$

of the sample (x_1, x_2) gives a new feature x_3, which is the projection of the sample point to a line l with an angle ? passing through the origin. By this transformation, the domain is expanded from d=2 of (x_1, x_2) to d=3 of (x_1, x_2, x_3). In addition, by introducing

$$x_4 = \cos q' \cdot x_1 + \sin q' \cdot x_2 ,$$

we can expand the original space of 2 features to another space of 4 features of (x_1, x_2, x_3, x_4).

We determine the p thresholds on l and l', respectively. Now we can express a subclass by a hexagonal shape instead of a rectangle. Y increasing the number of different ls, we can approximate a convex hull correctly.

For the case of higher dimensions (i.e. d>2) we should consider every possible combination of d features. The number 2^d-1 of possible combinations will soon become infeasible even for a small number of d. Therefore, for a certain constant n, we consider only the combinations of at most n features. If n=1, it correspond to the original subclass method. The number of possible combinations is

$$\sum_{i=1}^{d} C_d^i .$$

For each subspace consisting of selected i ($i \le n$) features, we consider several straight lines l passing through the origin as shown in the two-dimensional case. We, furthermore, set up equally spaced p thresholds on each l, which determine p parallel hyperplanes orthogonal to l. Then we expand the domain by the projections of points on different ls. By $m(i)$ is denoted the number of such ls.

We can approach subclasses to convex hulls as closely as we desire by increasing n, $m(i)$ and p. New features are then generated by the projections to 2^i lines with direction vectors $(\underbrace{\pm 1, \ldots, \pm 1}_{i \text{ times}})$ and $m(i)=2^i$.

The length d of a binary vector x becomes $(2d + 2^2 C_d^2) \cdot p$ for n=2 and $(2d + 2^2 C_d^2 + 2^3 C_d^3) \cdot p$ for n=3.

Some experiments can be made for artificial or real data. To calculate the recognition rate, when the sample set is large enough, the whole sample set can be divided into a training set and a test set independently. If the sample set is relatively small, it can be used a 10-fold cross validation technique in which the whole sample set id divided into 10 almost equal-sized subsets randomly, and one subset is used for test and the remaining nine subsets for training and the average of the recognition rates of ten trials is calculated.

In vowel recognition, which is a speech recognition problem, there are five classes corresponding to the five vowels: "a", "e", "i", "o", "u". In total, it was been taken from the database ETL-WD-1-1 600 vowel samples. All the samples were divided into two sets: a training set consisting of 1000 samples (200 for each vowel), and a test set of 5000 samples (1000 for each vowel). From 18 features consisting of six sets of a peak frequency [Hz], peak bandwidth [Hz] and peak power [dB], there were selected six features by a feature selection procedure (Kudo and Shimbo, 1993). The recognition rate with the subclass method was 86.6% and with the quasi-convex hull method was 87.1%.

In character recognition, the dataset is of 26 alphabetical handwritten uppercase letters from A to Z from the database ETL3. Each character is expressed by $324 = 19 \times 18$ bits (1:black pixel, 0:white pixel). The training sample set consisted of 2600 characters (100 for each character) and the test sample set considered of another 2600 characters. There were selected 30 principal features by applying Karhunen-Loeve expansion to 81 features of characteristic loci as proposed by Glucksman (1969). The recognition rate with the subclass method was 97.9% and with the quasi-convex hull method was 98.5%.

Still now, it is not clear in which cases the proposed method is effective more than the other methods. In general, this is a difficult question. It depends on the problems, i.e. the number of training samples, the dimensionality, and the shape of the underlying distributions. The approach with quasi-convex hull is strongly related to the shape of the underlying distributions of samples. Efficient ways are desired in order to construct a convex hull covering a given subset of the training samples and to find the convex hull in which a given sample falls. However, it is not still clear whether or not the use of true convex hull as subclasses is effective more than the proposed approach. As discussed above, it depends on the shape of the underlying distribution of samples as well as number of training samples.

Still, compared with the original method using a set of hyper-rectangles, this method improved the discrimination rate in several experiments.

References

ETL3: *ETL character database*, Electrotechnical Laboratory, Japanese Technical Committee for Optical Character Recognition, 1993.

Kudo, M. et al., *Construction of class regions by randomized algorithm. A randomized subclass method*, Pattern Recognition, 29, 1996.

Kudo, M., et al., *Construction of class regions by a randomized algorithm: A randomized subclass method,* Pattern Recognition 29, 1996.