L. Chmielewski
IPPT PAN

WYBÓR CECH I MINIMALIZACJA CZASU W WIELOPOZIOMOWYM,
MINIMALNO-ODLEGŁOŚCIOWYM ALGORYTMIE ROZPOZNAWANIA
Leszek Chmielewski
Instytut Podstawowych Problemów Techniki PAN

FEATURE SELECTION AND TIME MINIMIZATION
IN THE MULTI-LEVEL MINIMUM-DISTANCE RECOGNITION ALGORITHM

1. Introduction. The multi-level distance function

In the domain of pattern recognition, the effectiveness of a classifier is frequently viewed as the question of the least classification error and the possibility of preselecting the most distinctive features (e.g. [1], [2]). The time needed for classification results usually from the type of the classifier and the number and complexity of the features, already chosen on the grounds of the criterion of the best class separation.

In this paper it is proposed to lay the main stress on the minimization of the recognition time of the classifier, with the class separability condition used as a secondary criterion. The multi-level minimum-distance classifier, proposed by Chmielewski and Kosinski [3], will be discussed.

In the distance function approach (e.g. [2]) the decision function for assignment of a pattern (an object described by its features) to a class is the comparison of the distance (which is a measure of similarity between this pattern and a class prototype) with the threshold. However, as features can be of various kinds (e.g. continuous: area, perimeter; discrete: number of holes; boolean: existence of vertices in contour), it is not natural to construct a single distance function accounting for differences in all the features. Instead, a set of distance functions, or a multi-level distance function, is introduced, in which the distance on each level \( \text{Lev} \) represents the difference in some features of similar nature, between the recognized object \( \text{Obj} \) and the prototype of the class \( \text{Prt(Cla)} \) [3]. The decision criterion on a level \( \text{Lev} \) will be (for notations refer to Chapter 8):

\[
[\text{Obj } \in \text{ Cla}]_{\text{Lev}} \iff \text{DST}(\text{Obj}, \text{Prt(Cla), Lev}) \leq \text{Thr(Cla, Lev)} . \tag{1}
\]

It should be noted that now a class is represented by a prototype and a set of \( \text{NumLev} \) thresholds.
A decision criterion for a class is the conjunction of decision criteria for this class for all the levels:

\[ \text{Obj} \in \text{Cla} \iff \forall_{\text{lev}_k} \; k = 1, \ldots, \text{NumLev} \left[ \text{Obj} \in \text{Cla}_{\text{lev}_k} \right] \]  \( \text{(2)} \)

The classes should be separable by the whole decision process \( \text{(2)} \), i.e. by a criterion \( \text{(1)} \) for at least one level.

Let us consider only one distance level and omit the level notation for a while. If the distance functions have the properties of a metric, then the disjointness condition for a pair of classes (each defined by a prototype and a threshold) is as follows:

\[ \text{Cla}_i \cap \text{Cla}_j = \emptyset \iff \text{DSJK}(\text{Cla}_i, \text{Cla}_j) > 0 \quad \text{where} \]

\[ \text{DSJK}(\text{Cla}_i, \text{Cla}_j) = \]

\[ = \text{DST}(\text{Pro}(\text{Cla}_i), \text{Pro}(\text{Cla}_j)) - \text{Thr}(\text{Cla}_i) - \text{Thr}(\text{Cla}_j) \quad \text{(4)} \]

The recognition algorithm consists in updating a list of classes to which the recognized object can belong, which is called the candidate list. Initially the list contains all the known classes. Then the decision criteria based on the distance functions of subsequent levels are sequentially used to reject from the list those classes to which the object does not belong. In this way, only the distances between an object and the prototypes according to the first distance level are calculated for all the classes, because after processing each level the list is shortened.

The method proposed in this paper incorporates the search for the subset of most effective distance functions chosen from some large set of functions, and the minimization of recognition time by finding the proper sequence of distance functions within this subset.

2. Time of recognition

For a given object \( \text{Obj} \), the criteria \( \text{(1)} \) are used to reject the wrong classes from the candidate list. After using the criterion with the first level function, the list which initially contained \( \text{NumCla} \) classes will be shorter by the number of rejected classes \( \text{NumRjtCla} \); therefore, the second level criterion applies to the shorter list:

\[ \text{LenLst}(\text{lev}_k) = \text{NumCla} \quad \text{for} \quad k = 1 \;
\]

\[ \text{LenLst}(\text{lev}_k) = \text{LenLst}(\text{lev}_k) - \text{NumRjtCla}(\text{lev}_k) \quad \text{for} \quad k > 1. \quad \text{(5)} \]

The general time of recognition depends on the time of checking the conditions \( \text{(1)} \) on levels \( \text{Tim}(\text{lev}_k) \) and the list lengths. Our goal is to minimize this time:

\[ \text{GenTim} = \sum_{k=1}^{\text{NumLev}} \text{LenLst}(\text{lev}_k) \cdot \text{Tim}(\text{lev}_k) = \min. \quad \text{(6)} \]

For a given set of distance functions the values of time for each level are constant and can be measured directly. The expected numbers of classes rejected on each level depend on the statistical properties of the decision criteria and on the sequences in which they are used. To find the time-optimal sequence, we shall search the space of possible sequences, where the equation \( \text{(6)} \) will be used as an optimization criterion. The speeds \( \text{Tim}(\text{lev}_k) \) and the effectivenesses \( \text{NumRjtCla}(\text{lev}_k) \) (Eq. \( \text{(5)} \) in Chapt. 5) of the decision criteria will be used to guide the search.
3. Candidate list length estimation

Let us concentrate our attention on an event which consists in rejecting a single class $Cl_{i}$ from the candidate list, by a decision criterion on the level $Lev_{k}$. We shall consider this event under the condition that the recognized object belongs to the class $Cl_{j}$. The list is shortened by this criterion by a unit (one class is rejected) if the class $Cl_{i}$ has not been rejected on the previous levels and it is rejected on the current one, and by zero with the probability of a complementary event. Let us denote the event consisting in rejecting the class $Cl_{i}$ from the list by the decision criterion on level $Lev_{k}$, provided that the recognized object belongs to the class $Cl_{j}$, by $r_{ij}^{k}$, and the event consisting in accepting this class by $a_{ij}^{k}$. Now, we can apply the general probability formula to find the expected number of classes rejected from the candidate list:

$$\left[\text{NumRjtClac}\mid Cl_{j}, Lev_{k}\right]_{Cl_{i}} =$$

$$= 1 \# \text{Prb}\left(r_{ij}^{k} \cap a_{ij}^{k-1} \cap a_{ij}^{k-2} \cap \ldots \cap a_{ij}^{2} \cap a_{ij}^{1}\right) +$$

$$+ 0 \# \text{Prb}\left(a_{ij}^{k} \cup r_{ij}^{k-1} \cup r_{ij}^{k-2} \cup \ldots \cup r_{ij}^{2} \cup r_{ij}^{1}\right). \quad (7)$$

By repetitive substitution of the formula for conditional probability: $\text{Prb}(x \cap y) = \text{Prb}(y \mid x) \cdot \text{Prb}(x)$ to Eq. (7) we obtain

$$\left[\text{NumRjtClac}\mid Cl_{j}, Lev_{k}\right]_{Cl_{i}} =$$

$$\text{Prb}\left(a_{ij}^{1}\right) \# \text{Prb}\left(a_{ij}^{2}\right) \# \text{Prb}\left(a_{ij}^{3}\right) \# \ldots \#$$

$$\# \text{Prb}\left(a_{ij}^{k-1}\right) \# \text{Prb}\left(r_{ij}^{k} \mid a_{ij}^{k-1} \cap \ldots \cap a_{ij}^{1}\right) =$$

$$= \prod_{l=1}^{k-1} \left[\text{Prb}\left(a_{ij}^{l}\right) \cup a_{ij}^{l-1} \cap \ldots \cap a_{ij}^{1}\right] \# \text{Prb}\left(r_{ij}^{k} \mid a_{ij}^{k-1} \cap \ldots \cap a_{ij}^{1}\right). \quad (8)$$

The above considered event takes place subsequently for every class (i.e., for $i = 1, \ldots, \text{NumClac}$), and every time the list is shortened by a value according to (8). Hence, we can find the expected number of classes rejected from the candidate list on this level by summing these values for all the classes:

$$\text{NumRjtClac}(Lev) = \sum_{j=1}^{\text{NumClac}} \left[\text{Prb}(Cl_{j}) \sum_{i=1}^{\text{NumClac}} \left[\text{NumRjtClac}\mid Cl_{j}, Lev\right]_{Cl_{i}}\right], \quad (9)$$

where to remove the conditional clause $\mid Cl_{j}$, the general probability formula with the a priori probabilities of the classes $\text{Prb}(Cl_{j})$ has been used. Substitution of Eq. (9) to (5) yields the sought formula for the expected list lengths on the levels.
4. Estimation of probabilities

Let us introduce the following notations:

\[
\Prb\left( a^k_{ij} \mid a^k_{ij} \wedge \ldots \wedge a^{k-1}_{ij} \right) = \PrbDms(m_{ij} \mid m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) ; \quad (10)
\]

\[
\Prb\left( a^k_{ij} \right) = \PrbDms(m_{ij}) = \PrbUcd(m_{ij}) ; \quad (11)
\]

For weakly correlated features it will not introduce large errors if instead the conditional probabilities, the unconditioned ones are used to evaluate the expression (9). The unconditioned probabilities are independent of the sequence in which the decision criteria are used, what frees us from recalculation them each time a new sequence is analyzed. The only risk of this considerable simplification is that we can find a slightly sub-optimal sequence instead of an optimal one.

If classes $Cl^i$ and $Cl^j$ are disjoint on level $Lev$ then and only then the conditional as well as unconditioned probability $\PrbDms(m_{ij} \mid m_{ij} \wedge \ldots \wedge m^{k-1}_{ij})$ is equal 0. On the other hand, when the classes are identical, this probability is equal 1. The disjointness function $DSJ$ (4) can be used to find out if the probabilities assume their upper or lower bounds. To calculate the probabilities accurately we shall define a subspace of the feature space, denoted as the range of features $RngFea(m_{ij})$ of the class $Cl^i$ on the level $Lev$, as follows:

\[
Fea(Obj) \in RngFea>m_{ij} \quad \Rightarrow \quad \left[ \begin{array}{c}
\text{Obj} \in Cl^i
\end{array} \right]_{Lev} . \quad (12)
\]

The subspace of the feature space occupied by the feature vector of the class $Cl^i$ will be denoted as

\[
RngFea(m_{ij}) = \bigcap_{k=1}^{numLev} RngFea(m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) . \quad (13)
\]

Let us assume that the probability density functions $PrbDms$ of the feature vectors of each class are known. The formulae for the probability of accepting or rejecting a class will be as follows:

\[
\PrbDms(m_{ij} \mid m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) = 1 - \PrbDms(m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) =
\]

\[
= \int_{V_c} \PrbDms(Tec(m_{ij})) \, d \, V_c , \quad (14)
\]

\[
V_c = \left\{ \text{Tec(m_{ij})} \in RngFea(m_{ij}) \cap \bigcap_{k=1}^{k} RngFea(m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) \right\} , \quad (15)
\]

and $d \, x$ is differential of $x$. The unconditioned probabilities are

\[
\PrbUcd(m_{ij} \mid m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) = 1 - \PrbDms(m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) =
\]

\[
= \int_{V_u} \PrbDms(Tec(m_{ij})) \, d \, V_u , \quad (16)
\]

\[
V_u = \left\{ \text{Tec(m_{ij})} \in RngFea(m_{ij}) \cap RngFea(m_{ij} \wedge \ldots \wedge m^{k-1}_{ij}) \right\} . \quad (17)
\]

It can be noted that

\[
V_c \subset V_u \quad \Rightarrow \quad \PrbDms \leq \PrbUcd \quad \Rightarrow \quad \PrbDms(\text{Cnd}) \leq \PrbUcd(\text{Cnd}) . \quad (18)
\]
In many cases the probability density functions of the features PrbDMS can be approximated. It is also possible to estimate the probabilities (10) and (11) with the frequencies of the appropriate events, on the basis of sufficiently large sets of training patterns.

5. The formula for the effectiveness of distance criteria

After substituting (8) to (9) we obtain the formula for the expected candidate list length reduction on the level Lev in the form

\[ \text{NumRjtCl}(\text{Lev}_k) = \sum_{j=1}^{\text{NumCla}} \left[ \text{PrbCl}(\text{Lev}_k) \right] \times \]

\[ \times \left[ \prod_{l=1}^{k-1} \left( 1 - \text{PrbRjtCl}^{(\text{Lev}_l)} \right) \right] \]

\[ \times \left[ \text{PrbRjtCl}^{(\text{Lev}_k)} \right] \] (19)

where as PrbRjt either the CndPrbRjt according to (10), (14) can be substituted to obtain the accurate formula, or the UcdPrbRjt according to (11), (16) to obtain the simplified, approximate one.

From the result of the formula (19) calculated for all the distance functions which could be chosen for a specified level the following inferences on the criteria based on these functions can be drawn:

1. The more classes can a function reject, the more class separating power it has, and the more effective it is. It should be noted that not the global effectiveness, but the effectiveness at the current stage of the recognition process is considered.

2. If for some function the effectiveness NumRjtCl is 0 then this function has no discriminatory power for the considered set of classes or, if \( \text{Lev} = 1 \), the classes which could be rejected by this function had already been rejected by the previously used functions. Such a function should be omitted in the choice, without adverse effect on class disjointness. In this way, not only the time effectiveness can be optimized, but also the most effective functions can be chosen from an abundant set.

3. If all the classes are mutually disjoint, then the final expected list length (Eq. (5)) is equal 1. If it is greater than 1, then not all the classes can be separated by the classifier.

6. Optimization algorithm

In cases with large number of distance levels the systematic search for an optimal sequence of functions is impossible. To guide the search, the following heuristics based on the formula (19) are proposed:

1. The shorter the list, the less speedy functions can be applied.

2. From the choice of the most speedy functions, the most effective one should be used.
3. If the list is very short (\(\text{LenLst} < 2\)), then consider the effectiveness of the functions as more important than their speed.

If, instead of promoting the criterion of speed of functions, the criterion of list length reducing efficiency is put first, we arrive at the concept of one, complex distance function and miss all the advantages of the hierarchical recognition algorithm.

The methodology of choosing the optimal sequence of distance functions will be presented by means of an example.

7. Numerical example

The distance functions were chosen from the following set:

\[
\begin{align*}
\text{DST}_1 & (\text{Obj}, \text{Prt}) = |\text{NumHol}(\text{Obj}) - \text{NumHol}(\text{Prt})|; \\
\text{DST}_2 & (\text{Obj}, \text{Prt}) = |\text{Per}(\text{Obj}) - \text{Per}(\text{Prt})|; \\
\text{DST}_3 & (\text{Obj}, \text{Prt}) = |\text{Are}(\text{Obj}) - \text{Are}(\text{Prt})|; \\
\text{DST}_4 & (\text{Obj}, \text{Prt}) = \left[\left(I_{11}(\text{Obj}) - I_{11}(\text{Prt})\right)^2 + \left[I_{22}(\text{Obj}) - I_{22}(\text{Prt})\right]^2\right]^{1/2}.
\end{align*}
\]

(20)

where \(I_{11}, I_{22}\) are the main area moments of inertia.

Times were (list modifying operations included, feature finding excluded; Turbo Pascal; AT 286 at 12 MHz):

1: 0.0741 ms (difference of integers)
2: 0.0420 ms (difference of reals)
3: 0.0420 ms (difference of reals)
4: 0.8188 ms

The 6 disjoint classes with equal a priori probabilities were: 1 coin, 1 tubular rivet, 1 washer and 3 nuts of different sizes.

The guiding data obtained with the formulae (3), (4) and (19) are given in the Tables 1 and 2.

### Choice for level 1

<table>
<thead>
<tr>
<th>function</th>
<th>time ms</th>
<th>NumRjtClna</th>
<th>LenLst after the step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.57</td>
<td>1.6667</td>
<td>4.3333</td>
</tr>
<tr>
<td>2</td>
<td>0.04</td>
<td>4.7167</td>
<td>1.2833</td>
</tr>
<tr>
<td>3</td>
<td>0.64</td>
<td>4.9250</td>
<td>1.0750</td>
</tr>
<tr>
<td>4</td>
<td>0.92</td>
<td>5.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Comment: The single function 4 is enough to discriminate all the classes; functions 1-3 are surplus. Using function 4 would finish the recognition; however, function 3 is much quicker than 4 and still effective.

Chosen function: 3.

Intermediary result: time = 3.85 ms; list length = 1.0750.
Choice for level 2

<table>
<thead>
<tr>
<th>function</th>
<th>time ms</th>
<th>NumRjClα</th>
<th>LenList after the step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.57</td>
<td>0.0000</td>
<td>1.0750</td>
</tr>
<tr>
<td>2</td>
<td>0.64</td>
<td>0.0350</td>
<td>1.0400</td>
</tr>
<tr>
<td>4</td>
<td>0.82</td>
<td>0.0750</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Comment: List is very short now. Choosing the quickest function 2 would not make it much shorter before running function 4.

Chosen function: 4.

Final result: time = 4.94 ms; list length = 1.0000.

The sequence 1-2-3-4 ordered according to decreasing speed gives time = 6.01 ms. The results of various sequences are given in the Table 3. In this example the decreasing speed sequence, which is the common-sense guess for the time-optimal one, happened to be nearly the slowest. The above analysis of the choice of the sequence 3-4 makes it clear that surplus functions can improve the speed of the algorithm.

Comparison of various sequences of functions

<table>
<thead>
<tr>
<th>no</th>
<th>sequence</th>
<th>time ms</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-4</td>
<td>4.84</td>
<td>optimal</td>
</tr>
<tr>
<td>2</td>
<td>3-2-4</td>
<td>5.50</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5.62</td>
<td>one function; not optimal</td>
</tr>
<tr>
<td>4</td>
<td>4-3</td>
<td>6.16</td>
<td>seq.1 reversed</td>
</tr>
<tr>
<td>5</td>
<td>4-3-2-1</td>
<td>7.30</td>
<td>seq.7 reversed</td>
</tr>
<tr>
<td>6</td>
<td>1-4</td>
<td>7.43</td>
<td>seq.1 with fun.1 instead of 3</td>
</tr>
<tr>
<td>7</td>
<td>1-2-3-4</td>
<td>8.01</td>
<td>decreasing speed sequence</td>
</tr>
<tr>
<td>8</td>
<td>1-4-2-3</td>
<td>8.71</td>
<td>the worst possible sequence</td>
</tr>
</tbody>
</table>

8. Conclusion

The presented method of optimizing the set of functions which compose the multi-level distance function used in the recognition algorithm makes it possible to select the subset of most effective distance functions from some large set of functions, and to find a sequence of the selected functions which yields the minimal recognition time for a randomly chosen object.

The method is particularly useful as a development tool of simple and effective recognition systems with large sets of classes. It has been successfully used in a picture analysis system designed in the Technology Development Center of the Institute of Fundamental Technological Research, PAS. This system named LOOK is aimed at recognizing 2D silhouettes of small machine parts in binary images.
9. Notations

Names of variables are composed of abbreviations (according to [4]):

Acc: acceptance; accepted  Gen: general  Prb: probability
Are: area  Hol: hole  Prt: prototype
Cla: class  Len: length  Rjt: rejection; rejected
Cond: conditional  Lev: level  Rng: range (subspace)
DNS: density (function)  List: list  Thr: threshold
DSJ: disjointness (function)  Num: number of  Tim: time
DST: distance (function)  Obj: object  Ucd: unconditioned
Fea: features (vector)  Per: perimeter

References

4 Chmielewski L.J., "Proposition of the standard for naming variables in the picture processing programs" (In Polish), ibid., p. 80-92.

WYBÓR CECH I MINIMALIZACJA CZASU
W WIELOPOZIOMOWYM, MINIMALNOODŁEGŁOŚCIOWYM ALGORYTMIE ROZPOZNANIA

Streszczenie