

論 文

Multitree 형상 인식 기법의 성능 개선에 관한 연구

正會員 金 泰 成* 正會員 李 正 熙* 正會員 金 聖 大*

A Study on the Improvement of Multitree Pattern Recognition Algorithm

Tae Sung KIM* Jung Hee LEE* Seong Dae KIM* *Regular Members*

要 約 본 논문은 [1]와 [2]에 의해 제안된 multitree 형상 인식 기법의 성능 개선에 관한 논문이다. Multitree 형상 인식 기법의 기본적인 생각은, Classifier 설계과정에서 각 특징별로 Binary Decision Tree를 구성하고, 이들의 탐색 순서를 결정하며, 인식 과정에서는 앞에서 정한 탐색 순서에 의거하여, BDT(Binary Decision Tree)를 탐색해 나간다는 것이다. 이때 BDT를 추가하여 탐색하기 전에 그때까지 얻은 정보를 이용하여 입력 물체를 인식할 수 있는지에 대한 여부를 결정하며, 인식이 가능한 경우 BDT의 탐색을 멈추고, 인식이 불가능한 경우 BDT의 탐색을 계속해 나간다. 이 방법은 BDT를 각 특징별로 만들기 때문에 새로운 특징의 삭제나 첨가가 상당히 용이하며 인식에 사용되는 특징의 갯수가 감소하게 된다. 따라서 이 알고리즘은 특징의 수가 많거나 class수가 많은 경우 쉽게 이용될 수 있다.

본 논문에서는 각 특징에서 구한 근사화된 확률 분포로부터 입력 특징값에 대한 확률값을 구해 인식에 이용하였으며, 이 값을 이용하는 여러가지 인식 방법을 제안하였다. 그리고 Branch and Bound 방법을 사용하여 특징의 선택 순서와 탐색 범위를 구하였다. 위에서 제안한 것들을 실험한 결과 기존의 multitree 형상 인식 기법보다 본 논문에서 제안한 기법의 성능이 향상되었다.

ABSTRACT The multitree pattern recognition algorithm proposed by [1] and [2] is modified in order to improve its performance. The basic idea of the multitree pattern classification algorithm is that the binary decision tree used to classify an unknown pattern is constructed for each feature and that at each stage, classification rule decides whether to classify the unknown pattern or to extract the feature value according to the feature order. So the feature ordering needed in the classification procedure is simple and the number of features used in the classification procedure is small compared with other classification algorithms. Thus the algorithm can be easily applied to real pattern recognition problems even when the number of features and that of the classes are very large. In this paper, the weighting factor assignment scheme in the decision procedure is modified and various classification rules are proposed by means of the weighting factor. And the branch and bound method is applied to feature subset selection and feature ordering. Several experimental results show that the performance of the multitree pattern classification algorithm is improved by the proposed scheme.

*韓國科學技術院 電氣 및 電子工學科
Dept. of Electrical Engineering Korea Advanced
Institute of Science and Technology.
論文番號 : 89-33(接受1989. 3. 4)

I. 서론

지금까지 알려진 형상 인식 기법으로 여러가지 알고리즘이 제안되었는데 인식에 소용되는 시간과 방법상의 관점에서 볼때 multistage방법인 hierarchical형상 인식 기법과 sequential 형상 인식 기법이 상당히 효율적인 것으로 알려져 있다. 그러나, hierarchical 형상 인식은 tree를 이용하여 인식하는 기법으로 tree구조의 선택, 각 nonterminal node에서 선택될 특징의 선택, 각 nonterminal node에서 인식을 수행할 classification rule의 선택에 관한 문제들이 서로 연관되어 있기 때문에 최적 decision tree를 구현한다는 것은 어려운 단점이 있다. 그리고 sequential 형상 인식 기법은 인식하는 과정에서 특징들을 순차적으로 한 개씩 사용하는 방법을 쓰기 때문에 인식에 필요한 특징의 갯수가 적다는 장점이 있으나, 각 단계에서 사용될 특징의 선택이나 각 단계에서 인식을 수행할 classification rule의 선택이 복잡하다는 문제점이 있다. 그래서 알고리즘의 최적화를 위하여 dynamic programming이나, gamma절차와 같은 접근 방식이 있으나 대체적으로 계산 시간이 많고, 계산에 필요한 메모리가 많이 필요하다.

이와 같은 문제점들을 해결하기 위해 [1]과 [2]에서 Multitree 형상 인식기법이 제안되었다. 이 기법은 모든 특징들을 동시에 고려하지 않고, 각 특징별로 BDT를 만들어서 입력 패턴을 인식하는데 사용한다. 그리고 설계된 BDT를 이용하여 인식을 수행하는 과정에서도, 구성된 BDT를 모두 탐색하는 것이 아니라, 인식될 때까지 BDT를 하나씩 추가하여 사용하기 때문에 사용되는 평균 특징갯수가 감소된다. 그리고 BDT를 각 특징별로 만들기 때문에 새로운 특징의 첨가나, 불필요한 특징의 삭제가 용이하다. 또 특징의 탐색 순서를 정하는 것이 BDT를 구성할 때 이루어지는 것이 아니라, BDT가 다 구성된 후 인식과정에 들어가기 전에 수행되기 때문에 특징의 탐색 순서를 정하는 문제나 특징의 탐색 범위를 정하는 문제가 쉽게 된다. 그런데 인식 과정에서, BDT의 terminal node에 속하는 class들에 대한 가중치를 모두

같도록 했고, BDT의 탐색순서를 정하는 방법에서 인식률에 대한 고려가 소홀하여 오인식률이 높아질 가능성이 있다.

그래서 본 논문에서는 이러한 문제점을 보완하기 위한 방법을 제시하였다. 즉 BDT의 terminal node에 속하는 class들에게 입력 패턴의 특징값에 따른 가중치들을 부여함으로써 좀더 정확한 인식이 되도록 하고, branch and bound방법을 사용하여 BDT의 탐색 순서를 결정하였다.

II. Multitree 형상 인식의 성능 개선

Multitree 형상 인식 기법의 전체 과정을 살펴보면 classifier를 설계하는 과정과 설계된 classifier를 이용하여 입력 패턴을 인식하는 과정으로 구성된다. 이 두 과정의 간단한 흐름도를 (그림 1)와 (그림 2)에 나타내었다.

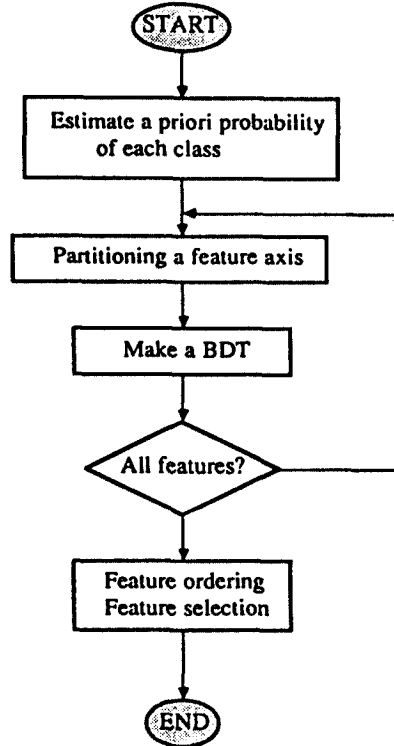


그림 1. Classifier 설계 과정
The design procedure

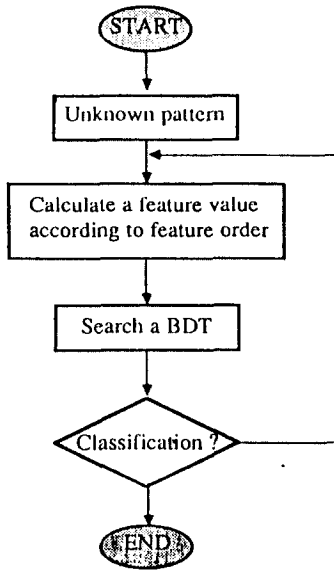


그림2. Classification 과정
The classification procedure

classifier를 설계하는 과정에서는(그림 1)에서 보듯이 특징 벡터를 각 특징축으로 투영시켜서 각 특징축에 대해 decision tree를 설계한다. 먼저 특징축을 여러 구간으로 분할하고 각 구간에 존재 가능한 class들을 구하여 구간과 class의 관계를 BDT의 형태로 구성한다. 그리고 classifier 설계의 마지막 단계에서는 인식과정에서 필요한 BDT 탐색 순서를 결정한다. 여기서 BDT의 탐색 순서는 오인식률에 영향을 미치게 되는데 그 이유는 multitree 형상 인식 기법에서는 sequential 형상 인식 기법에서처럼 인식이 될때까지 특징을 한 개씩 받아 들이기 때문이다.

위 과정으로 설계된 classifier를 사용하여 임의의 물체를 인식하는 과정은 (그림 2)에서 나타낸 것처럼 이미 정해진 특징의 순서에 따라 물체의 특징값들을 추출하고 이 값으로 BDT를 탐색해서 인식하게 된다.

[1]과 [2]에서 제안한 기법은 입력 물체의 특징 값에 대한 class들의 사후확률이 서로 다른데도 마치 같은 확률을 가지는 것처럼 취급한다는 단점이 있다. 만약 그 class들간의 확률값들을 쉽고 빠르게 알 수 있다면 인식에 상당히 도움이 될

수 있고 오인식률을 낮출 수 있을 것이다. 그래서 본 논문에서는 각 구간에서 존재할 수 있는 class들의 근사화된 확률 분포를 BDT에 포함시켜서 multitree 형상 인식 기법의 장점을 이용하면서 오인식률을 줄일 수 있는 방법들을 제안하였다.

그리고 앞서 설명한 바와 같이 BDT의 탐색 순서가 오인식률에 영향을 미치기 때문에 Branch and Bound 방법을 사용해서 특징의 탐색 순서를 정하였다.

II.1절에서 경계값 추출과 BDT의 구성에 관해 설명하고 II.2절에서는 제안한 여러가지 classification rule에 대해 설명한다. 그리고 마지막으로 II.3절에서는 BB(Branch and Bound) 방법을 사용하여 특징의 조사 순서를 정하는 방법에 관하여 설명한다.

II.1 BDT의 구성

본 논문에서 제안하는 것의 가장 기본을 이루는 것은 입력 특징값에 대한 각 class의 확률값을 쉽고 빠르게 추출하는 것이다. 즉 class에 대한 특징의 확률 밀도 함수를 각 구간별로 근사화시키고 BDT를 만들때 근사화된 밀도 함수를 나타낼 수 있는 parameter들을 포함시켜서, BDT를 탐색함으로써 각 class의 근사화된 확률값을 얻도록 하였다. 이를 위해 특징축을 여러 구간으로 나눌 필요가 있는데, 구간을 나눌 경계값은 아래의 조건을 만족하는 특징값 b 로 선택했다.

$$(1) \frac{dp(x_n | w_t)}{dx_n} \Big|_{x_n=b} = 0 \quad (2-1)$$

$$(2) \int_b^{\infty} p(x_n | w_t) dx < \alpha_n \quad (2-2)$$

$$\int_{-\infty}^{-b} p(x_n | w_t) dx < \alpha_n \quad (2-3)$$

여기서, $p(x_n | w_t)$ 는 class w_t 의 n 번째 특징 x_n 에 대한 조건 확률 밀도 함수이고, α_n 은 1보다 충분히 작은 양의 상수이다.

위의 조건을 만족하는 경계값에서의 확률 밀도 함수값만을 이용하여, 아래와 같은 방법으로 임의의 특징값에서의 확률 밀도 함수값을 근사적으로 알 수 있다. n번째 특징에 대한 i번째 구간의 두개의 경계값 b_n^{i-1} , b_n^i 과 그에 해당하는 j번째 class의 확률 밀도 함수 값을 각각 $p(b_n^{i-1} | w_j)$, $p(b_n^i | w_j)$ 이었다고 하면 근사화된 확률값은 다음의 식으로 구해진다.

$$F(x_n | w_j) = \frac{p(b_n^i | w_j) - p(b_n^{i-1} | w_j)}{b_n^i - b_n^{i-1}} \cdot (x_n - b_n^{i-1}) + P(b_n^{i-1} | w_j) \quad (2-4)$$

임의의 특징 벡터가 주어졌을 경우 위의 근사화된 확률값을 구하는데 필요한 정보인 각 구간의 경계값과 그 구간에서 존재 가능한 class을 BDT가 포함하도록 구성한다. 이를 위한 tree의 non-terminal node와 terminal node에서의 data 구조를 (그림 4)의 BDT에서 살펴보자.

이 그림에서 d_k 에 해당하는 node들은 non-terminal node들으로써 child node들이 가질 수 있는 class의 subset을 둘로 나누는 경계값과 child node를 가르키는 pointer로 구성되며, T_k 은 terminal node를 나타내고 다음과 같이 구성 된다.

$$\begin{aligned} & \{ \{ C_1, P(b_n^{i-1} | C_1), P(b_n^i | C_1) \}, \\ & \{ C_2, P(b_n^{i-1} | C_2), P(b_n^i | C_2) \}, \\ & \dots, \{ C_m, P(b_n^{i-1} | C_m), P(b_n^i | C_m) \} \} \quad (2-5) \end{aligned}$$

- 여기서 b_n^{i-1} : 해당 구간의 왼쪽 경계값에서의 특징값
- b_n^i : 해당 구간의 오른쪽 경계값에서의 특징값
- $C_1 \in \{ w_1, w_2, w_3, \dots, w_c \}$
- m : terminal node T_k 에 포함될 수 있는 class들의 갯수
- $p(b_n^i | C_1)$: class C_1 의 특징값 b_n^i 에 대한 확률 밀도 함수값
- $p(b_n^{i-1} | C_1)$: class C_1 의 특징값 b_n^{i-1} 에 대한 확률 밀도 함수값.

경계값의 추정과 BDT의 구조를 살펴보기 위해 (그림 3)의 간단한 예를 살펴보자. 이 그림에서 앞에서 설명한 경계값들으로써 $b_n^1, b_n^2, \dots, b_n^6$ 이 구해진다.

(그림 3)에 나타난 확률 밀도 분포 함수에 대한 BDT를 (그림 4)에 나타내었다. (그림 4)에서 나타나는 BDT를 설명하면 다음과 같다. 어떤 nonterminal node에 해당하는 구간이 $[b_n^i, b_n^{i+k}]$ 라 하자. 이 구간안에 있는 k-1개의 경계값중, i번째 경계값에서 i+j번째 경계값까지의 모든 class의 누적 확률값 (cdf)의 합이 i번째 경계값에서 i+k 번째 경계값까지의 모든 class의 누적 확률값의 합이 $\frac{1}{2}$ 에 가까운 경계값을 2개의 child node로 분할할 경계값으로 선택한다.

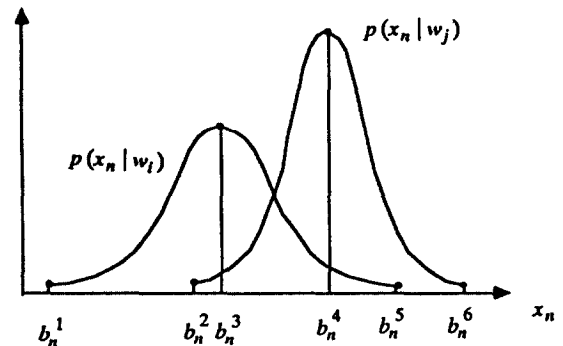


그림3. 두 class 경우에서의 경계값들
Partitioning points in a 2-class case

이런 방법으로 각 nonterminal에 경계값을 할당하고, 할당되지 않은 경계가 없도록 구성된 BDT는 왼쪽 child node에 해당하는 경계의 확률과, 오른쪽 child node에 해당하는 경계의 확률이 비슷하게 됨으로써 BDT의 평균 조사 경로가 최소가 되도록 한다. BDT의 구성할때 일반적인 기준은 최대 조사 경로의 길이에 대한 기준과 평균 조사 경로의 길이에 대한 기준의 두가지가 있는데 본 논문에서 사용한 BDT의 구조는 양쪽 child node의 확률을 같게 함으로써 평균 조사 경로의 길이를 줄이도록 하는 두번째 기준을 이용한 것이다.

T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇			
	C ₁	C ₁	C ₂	C ₁	C ₂	C ₁	C ₂	C ₃	
	p(b _n ¹ C ₁)	p(b _n ² C ₁)	p(b _n ³ C ₂)	p(b _n ⁴ C ₁)	p(b _n ⁵ C ₂)	p(b _n ⁶ C ₁)	p(b _n ⁷ C ₂)	p(b _n ⁸ C ₃)	
	p(b _{n_s} ¹ C ₁)	p(b _n ² C ₁)	p(b _n ³ C ₂)	p(b _n ⁴ C ₁)	p(b _n ⁵ C ₂)	p(b _n ⁶ C ₁)	p(b _n ⁷ C ₂)	p(b _n ⁸ C ₃)	

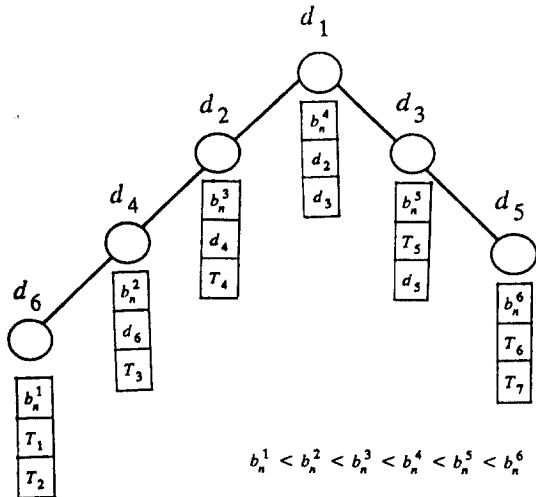


그림 4. BDT의 구조
The structure of BDT

II. 2 Classification rule

기존의 multitree 형상 인식 기법에서는 각 BDT를 탐색할때 입력 특징 벡터를 가질 수 있는 class들만을 구했고, n개의 BDT를 탐색했을때의 인식 방법은 가장 많이 찾아진 class를 입력 특징 벡터에 대한 class들로 인식하는 것이다. 그런데 본 논문에서는 입력 특징 벡터에 해당하는 확률 밀도 분포 함수 값을 근사적으로 구할 수 있기 때문에 여러가지 classification rule이 가능하다.

classification rule 1)

입력 특징 벡터(x₁, x₂, ..., x_d)가 주어 졌을 경우 각 class에 대한 사후확률은 다음으로 정의된다.

$$p(w_j | X) = \frac{p(X | w_j) P(w_j)}{P(X)} \quad (2-6)$$

여기서, $P(X) = \sum_{j=1}^c p(X | w_j) P(w_j)$

위 식에서 p(w_j | X)는 p(X | w_j) P(w_j)에 비해 하세 된다. 그런데 모든 특징들이 상호 독립이라고 가정하면

$$p(X | w_j) = \prod_{n=1}^d p(x_n | w_j) \quad (2-7)$$

이다. 그런데 F(x_n | w_j)는 p(x_n | w_j)로 근사화할 수 있으므로

$$p(w_j | X) \cong \prod_{n=1}^d F(x_n | w_j) P(w_j) \quad (2-8)$$

즉 입력 특징 벡터가 주어졌을 경우 각 class에 대한 사후 확률은 BDT를 탐색한 결과 찾아지는 값들로부터 근사적으로 구할 수 있다. 여기서 P(w_j)가 모든 class에 대해서 같다고 가정하고, d개의 특징중 중요한 n개의 특징만 고려한다면 사후 확률 F_n(w_j)는 다음으로 계산될 수 있다.

$$F_n(w_j) = \prod_{i=1}^n F(x_{n_i} | w_j) \quad (2-9)$$

$$j=1, 2, \dots, c$$

$$n_i \in \{1, 2, \dots, d\}$$

이때 F_n(w_j) (j=1, ..., c) 중에서 가장 큰 값과 두번째로 큰 값이 서로 비슷하다면 입력 특징 벡터가 어떤 class에 속하는지 인식할 수 없고, 그 두 값의 비가 크다면 가장 큰 F_n(w_j)를 갖는 class로 인식할 수 있다. 만약 특징의 순서가 정해져 있고, 그 순서에 따라 특징을 첨가시키면서 위의 근사화된

사후 확률을 계산하여 인식에 이용한다면 평균적으로 사용되는 특징의 수를 감소시킬 수 있다. 이러한 내용을 간단한 언어로 표시하면 다음과 같다.

while (all feature?)

$$F_n(w_j) = F_{n-1}(w_j) * F(x_n | w_j)$$

여기서, $j=1, 2, \dots, c$
 w_j : j 번째 class
 x_n : n 번째 feature

if ($(\max_{i=1, \dots, c} F_n(w_i) / \text{smax}_{j=1, \dots, c} F_n(w_j)) > t_{\text{mut}}$)
 return ($w_i | \max_{i=1, \dots, c} F_n(w_i)$)

else

{
 search a BDT
 $n=n+1$

calculate $F(x_n | w_j)$

return($w_i | \max_{i=1, \dots, c} F_n(w_i)$)

여기서, $\max_{i=1, \dots, c} F_n(w_i) : F_n(w_i) (i=1, 2, \dots, c)$

중에서 가장 큰 값

$\text{smax}_{j=1, \dots, c} F_n(w_i) : F_n(w_i) (i=1, 2, \dots, c)$ 중에서

두번째로 큰 값

위의 알고리즘에서 t_{mut} 의 값이 클 수록 더 많은 특징을 고려하게 되고 인식률도 높게 된다.

classification rule 2)

multitree 형상 인식에서는 입력 특징 벡터의 각 성분을 차례로 받아 들이면서 인식하는 방법을

쓰기때문에 $F(x_n | w_j)$ 를 연속적으로 곱하여 인식에 이용할 경우 오인식률이 크게 될 수 있다. 왜냐하면 각 특징축의 구간을 결정하는 과정에서 확률 밀도 분포 함수값이 작은 경우 0으로 근사화시켰기 때문에, 입력 특징 벡터의 한 성분에 대한 확률 밀도 함수값이 0이기만하면 다른 성분에 대한 값이 아무리 크다하더라도 결국 구해지는 $F_n(w_j)$ 는 0이기 때문이다. 그래서 근사화된 확률 밀도 분포 함수의 값을 연속해서 더하는 classification rule를 제시한다.

while (all feature?)

$$F_n(w_j) = F_{n-1}(w_j) + F(x_n | w_j)$$

if ($(\max_{i=1, \dots, c} F_n(w_i) - \text{smax}_{j=1, \dots, c} F_n(w_j)) > t_{\text{add}}$)
 return ($w_i | \max_{i=1, \dots, c} F_n(w_i)$)

else{

search a BDT
 $n=n+1$
 calculate $F(x_n | w_j)$

return($w_i | \max_{i=1, \dots, c} F_n(w_i)$)

classification rule 3)

BDT를 조사할때 각 terminal node에 해당하는 구간들에서 오인식될 확률을 구해보자. 어떤 terminal node에서 찾아진 class들을 $w_i(i=1, 2, \dots, m)$ 라고 할때 입력 특징이 이 구간에 있을 확률 P_a 는 다음과 같다.

$$P_a = \sum_{i=1}^m \int_{b_{n-1}^k}^{b_n^k} P(w_i) p(x_n | w_i) dx \quad (2-10)$$

그리고 이 구간에 대해 error로 판단될 확률 P_e 은 다음과 같다.

$$P_e = \sum_{i=1}^m \sum_{j=1}^m P(x_n \in R_j, w_i) \quad (2-11)$$

$$\begin{aligned}
 &= \sum_{i=1}^m \sum_{j=1}^m \int_{R_j} p(x_n | w_i) P(w_i) dx \\
 &= P_a - \int_{b_n^{k-1}}^{b_n^k} \max_{i=1, \dots, m} p(x_n | w_i) P(w_i) dx \\
 R_j : &| x_n | b_n^{k-1} < x_n < b_n^k, p(x_n | w_j) P(w_j) \\
 &= \max_i p(x_n | w_i) P(w_i) |
 \end{aligned}$$

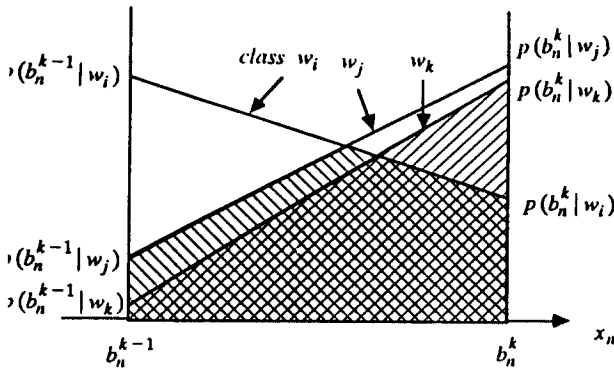


그림 5. 근사화된 확률 분포 함수에 대한 P_a, P_e의 영역
The areas of P_a and P_e in approximated p.d.f

P_e가 큰 영역에서 구한 사후 확률값은 작은 영역에서 구한 사후 확률값보다 입력 특징 벡터를 오인식할 가능성이 크다는 것을 의미하므로 P_e에 비례하는 가중치를 각 특징에 대해 주는 방법을 제시한다. 그런데 P_e는 구간의 크기에 상당히 관계가 있으므로 이런 영향을 줄이기 위해 P_a로 정규화시켜 classification rule에 이용한다. 여기서 P_a와 P_e는 앞에서 구한 확률 밀도 분포 함수를 근사화시켰기 때문에 쉽게 구할 수 있다. 확률 밀도 함수를 근사화시킬 경우 (그림 5)와 같은 분포가 되는데, m개의 class가 있는 n번째 특징의 k번째 구간에 입력 특징값이 있을 확률 P_a는 다음의 사다리꼴 면적 구하는 식으로 간단히 표현된다.

$$P_a = \sum_{i=1}^m (b_n^{k-1} + b_n^k) (p(b_n^{k-1} | w_i) + p(b_n^k | w_i)) / 2 \quad (2-12)$$

그리고 정확하게 판단될 확률 P_c도 식(2-12)과 비슷한 형태로 표현될 수 있다. 여기서 구해지는 P_e/P_a를 모든 특징과 모든 구간에 대해 구한 후 BDT의 terminal node에 기억시킨 후 classification에 이용한다. 그러면 classification rule는 다음과 같다.

while(all feature?)

$$\begin{aligned}
 &F_n(w_j) = F_{n-1}(w_j) + \frac{P_a}{P_e} F(x_n | w_j) \\
 &\text{if } ((\max_{i=1, \dots, c} F_n(w_i) - \min_{j=1, \dots, c} F_n(w_j)) > t) \\
 &\quad \text{return } (w_i | \max_{i=1, \dots, c} F_n(w_i))
 \end{aligned}$$

else

search a BDT
n = n + 1
calculate F(x_n | w_j)

return(w_i | max_{i=1, ..., c} F_n(w_i))

II.3 특징 순서와 특징 탐색 범위

Classification하는 과정에서 BDT를 순차적으로 조사하는 방법을 선택하였으므로 특징 순서가 classification error에 영향을 준다. 그리고 n개로 이루어진 특징 subset중 error가 적은 subset을 구하는 문제나 classification error의 허용치가 주어졌을 때 가장 적은 갯수의 특징을 포함하는 subset를 찾는 문제에서 선택된 특징과 classification error의 관계를 알아야 한다. 이러한 특징의 탐색순서와 특징 선택의 문제를 해결하기 위해 앞에서 설명하였던 BB(Branch and Bound)방법을 사용하였다. 일반적으로 BB 방법은, 임의의 함수값을 최소화시키는 변수의 조합을 찾는 문제에서, 변수의 조합의 경우의 수가 상당히 많을 경우 그 함수가 변수의 갯수에 대해 단조 성질을 가진다는 조건하에서 탐색해야 할 변수의 조합이 상당히 적게 해 준다.

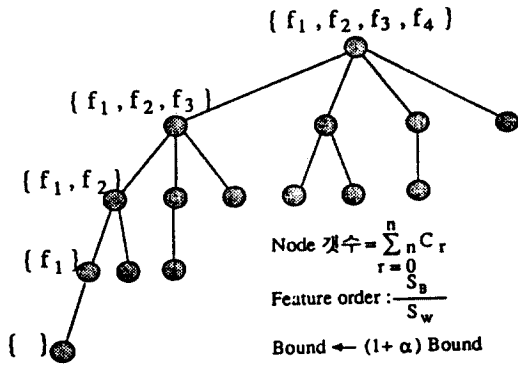


그림6. Branch and Bound 탐색 tree
Search tree used in Branch and Bound algorithm

그런데 해결해야 할 문제는, 많은 특징중 error 가 적게 되는 특징의 subset을 찾는 것이므로 BB 방법을 사용하여 해결할 수 있다. 이를 위해 (그림 6)의 tree를 구성하는데 이 tree의 각 node에는 특징 subset이 대응되고, parent node에 대응하는 특징 set에서 각각 한개의 특징을 뺀 subset들이 child node에 대응되도록 구성하고, Fisher criterion 값의 크기 순서로 각 subset 안에서의 특징의 순서를 정하도록 한 구조이다. Fisher Criterion 은 between scattering matrix를 S_B 라 하고, within scattering matrix를 S_w 라 하면, $\frac{S_B}{S_w}$ 의 값이다. 그러므로 root node에 대응되는 특징 set는 d개의 특징으로 이루어진 전체 set이고 child node로 감에 따라 특징 갯수가 한개씩 줄어 들어서, terminal node에 대응되는 특징 subset의 특징갯수는 한개이다.

그런데 multitree 패턴 인식 기법은 특징 갯수가 늘어 남에 따라 classification error가 줄어드는 성질을 가지나 뒤의 III절에서 보듯이 정확히 그러하지는 않다. 그러나 이런 예외성을 없애기 위해 BOUND의 값을 약간 늘려서 약간의 node를 더 탐색하는 방법을 썼다. 그러므로 단조 성질을 가져야만 BB 방법을 사용할 수 있다는 가정을 위의 tree가 만족한다. 이렇게 구성된 tree에 대해 error 허용치가 주어질 경우 tree 탐색은 다음과 같다. root node로부터 아래 node

로 가면서 그 node들에 해당하는 특징 subset에 대해 error값을 얻는다. 어느 node에서의 error 값이 허용치를 넘어섰을때 그 node 아래의 node들을 error값이 더 클것이 확실하므로, parent node에 속하는 다른 node들로 탐색을 계속한다. 이때 탐색하지 않는 child node들 때문에, 목적하는 바인 적은 수의 특징 subset만의 error 값을 계산하면서 error값이 적은 특징 subset를 찾을 수 있다. 이렇게 찾아진 특징 subset의 의미는, 적은 특징 갯수로 error값의 허용치에 만족되는 인식을 할수 있기 때문에 좀 더 빠른 인식이 된다는 것이다.

그리고 구해진 특징 subset들로 특징의 탐색 순서를 정할 수 있다. 즉 n-1개와 n개의 특징으로 이루어진 가장 작은 error를 가지는 두개의 subset사이에 완전한 포함 관계를 이루지는 않지만, 거의 대부분이 포함되므로, 적은 갯수의 subset에 포함되는 특징을 먼저 탐색하도록 하여 특징 탐색 순서를 정하면 된다.

III. 실험 및 결과

본 논문에서 제안한 Multitree 형상 인식 기법의 성능 개선에 대해 알아보기 위해 두가지 data set에 대해 실험을 하였다.

실험에 사용한 첫 번째 data set는 5개의 class에 대한 7가지 특징 분포를 임의로 지정하여 본 논문에서 제안한 세가지 classification rule에 대하여 실험하였고, 다른 논문과 기존의 multitree를 이용한 형상 인식방법과 본 논문의 성능 비교를 위해서는 손으로 쓴 26의 영문자로부터 얻은 특징 분포를 이용하였다. 이때 특징들이 모두 gaussian 분포라고 가정하고, 손으로 각각의 영문자를 네번씩 써서 그것들에 대한 mean과 variance를 얻어서 사용하였다. (표 1)에 5개의 class에 대한 7가지 특징 분포의 data set를 나타내었고, (표 2)에 영문자에 대한 특징의 이름과 fisher criterion 값에 따른 순서로 특징의 평균값을 나타내었다.

표 1. (1) 첫번째 data set에서 사용한 특징들의 평균값
(1) The mean of the features used for first data set

feature \ class	0	1	2	3	4
0	1000	1000	1000	1000	1000
1	2000	2000	2000	2000	2000
2	2000	3000	4000	5000	6000
3	1000	1000	1000	1000	1000
4	1000	1000	2000	3000	4000
5	6400	5000	3000	3456	20000
6	9400	3000	4000	9238	9878

표 1. (2) 첫번째 data set에서 사용한 특징들의 variance
(2) The variance of the features used for second data set

feature \ class	0	1	2	3	4
0	1000	1100	1200	1300	1400
1	1000	1100	1200	1300	1400
2	1000	1000	1000	1000	1000
3	1000	1000	1000	1300	1500
4	1000	1100	1200	1200	1200
5	1301	1500	1600	1450	1400
6	2000	1950	1600	2109	2340

표 2. (1) 영문자 인식에서 사용한 특징들
(1) The features used in the english character recognition

1	Area	of object
2	Perimeter	of object
0	Compactness	of object
3	Average	of object
4	Minimum Radius	of object
5	Maximum Radius	of object
6	Area	of convex hull
7	Perimeter	of convex hull
8	Compactness	of convex hull
9	Average	of convex hull
10	Minimum Radius	of convex hull
11	Maximum Radius	of convex hull
12	Area of holes	
13	Area of bays	
14	Area of con-obj	
15	Area of obj-hol	

실험에서 성능 평가는 오인식률로써 정하였는데, 앞에서 구한 확률 밀도함수에 근거하여 각 class의 특징값들을 random 하게 발생시켜 test sample들을 여러개 만들고 설계된 classifier로 인식해서 오인식된 갯수를 전체 test sample수로 나눈값을 오인식률로 하였다.

(그림 7)은 본 논문에서 제안한 세가지 classification rule에 대한 error 분포를 나타내고

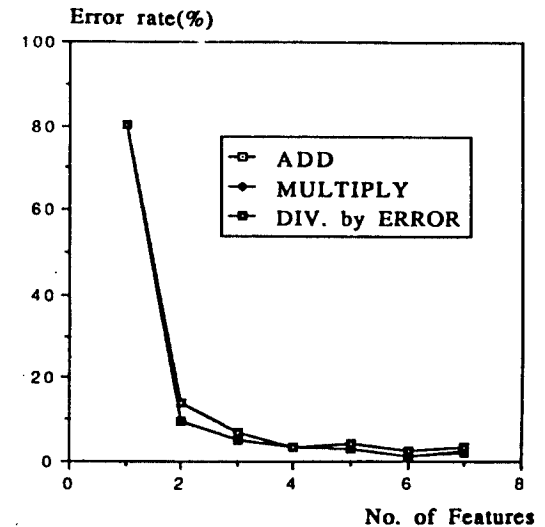


그림 7. 제안한 여러가지 criterion에 대한 오인식률
Error rate of the proposed classification algorithm

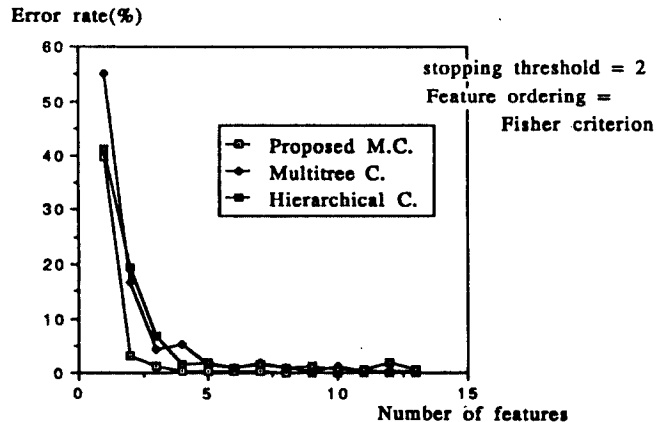


그림 8. 다른 classifier와의 성능 비교
Comparison of various classification algorithms and the proposed algorithm

표 2. (2) 영문자 인식에서 사용한 특징들의 평균값
(2) The means of the features in the english character recognition

class number	feature 0	feature 14	feature 6	feature 3
0	3004.450928	9315.793945	3730.662109	4744.491699
1	4077.757080	2432.067627	5700.098145	801.369385
2	1319.649292	5787.633301	5288.442871	1391.652832
3	1628.983887	2967.060303	978.425293	1716.804321
4	4678.319336	7980.007813	5955.737305	1488.415527
5	2913.915527	6254.990234	5975.168457	4928.208008
6	2110.538574	3811.238281	1494.470947	4732.016602
7	816.655945	7589.956055	652.051697	3451.143799
8	631.838684	5785.574707	5222.347168	2801.684082
19	1068.060791	7183.243164	3593.512695	3664.521729
10	4688.399902	1821.337891	5878.979980	739.993835
11	1690.934814	328.951538	1783.587769	3929.243896
12	740.950073	4422.777832	2737.450684	916.851807
13	3277.432617	6907.109863	6784.402344	5278.881836
14	3513.071289	1508.214722	5604.307129	3029.494629
15	3850.359131	2543.074219	5579.993164	728.997986
16	4604.192383	5581.898926	6730.319336	4102.489746
17	4001.869385	7116.873047	2710.174561	2003.842041
18	3321.792725	4562.151367	1877.412354	3833.073486
29	1754.767578	6088.234375	6225.916504	2118.976807
20	1542.702637	1118.849976	4363.290039	4591.152344
21	2839.909668	9502.815430	6622.411621	2107.973877
22	3713.103027	2839.058838	2927.566650	728.623718
23	141.399475	2820.342773	4928.317871	331.735168
24	1432.421875	2558.284668	3865.634033	5306.071777
25	4528.793457	5920.846680	5683.450195	3169.170654
Fisher Criterion	36.40	32.34	21.45	9.92

있다. 이 표에 의하면 가중치를 더 하는 방법이 많은 수의 feature를 사용할 경우 error가 가장 적었으며, 적은 수의 특징을 사용할 경우 error가 컸다.

(그림 8)은 addition classification rule에서 가중치를 모두 1로 할 경우와 확률로부터 얻은 가중치로 classification할 경우에 대한 classification error이다. 이를 보면 확률로부터 얻은 가중치로 인식하는 경우의 error가 가중치를 모든 class에 대해 같게 하여 얻은 error보다 상당히 적었다. 그리고 hierarchical 형상 인식의

실험 결과도 나타내었다. 이때 결과가 나오기까지의 시간을 보면 hierarchical 형상 인식 기법이 가장 빠르고, 그 다음이 기존의 multitree 형상 인식 기법, 그리고 제안하는 multitree 형상 인식 기법의 순이었다.

(그림 9)는 addition classification rule에서 stopping rule의 문턱값 t값에 따른 classification error의 변화를 나타내었다. 결과를 보면 임계값이 커질수록 error는 작아졌다. 즉 임계값이 커질수록 더 확실히 classify될때까지 feature를 조사하기 때문에 더 많은 feature를 조사하므로

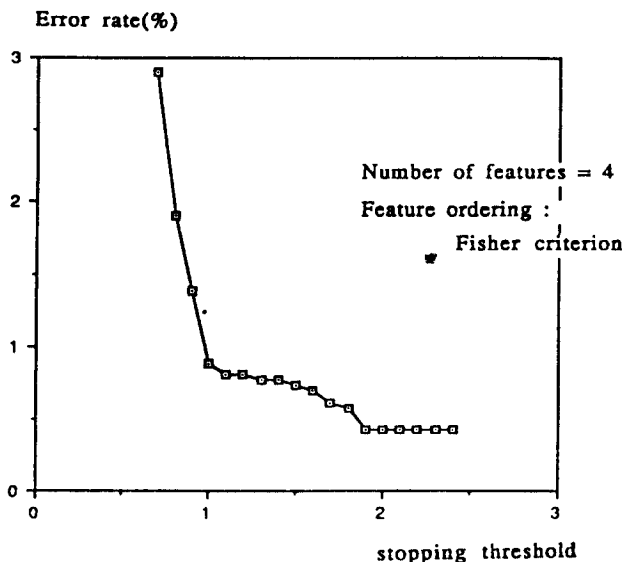


그림 9. Stopping threshold와 오인식률의 관계
The relation between stopping threshold and error rate

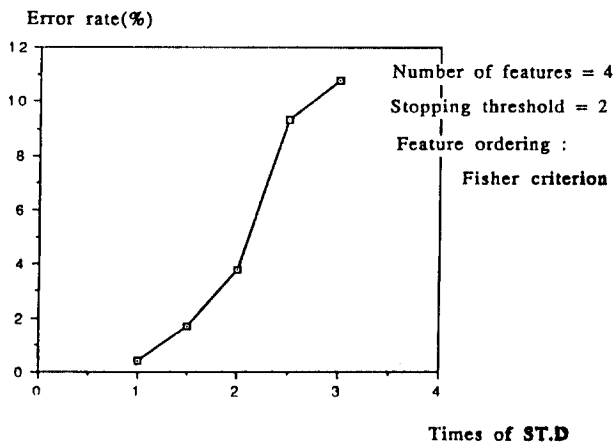


그림 10. 특징의 표준 편차와 오인식률의 관계
The relation between standard deviation and error rate

classification 에 걸리는 시간은 길어지지만 error 는 작아진다.

(그림 10)는 test pattern의 분포를 더 넓혀서 classification error를 조사하였다. sample들에 대한 각 특징의 mean은 고정시켜 놓고, variance를 증가시켜서 classifier를 설계하고, error를 위한 sample들도 같은 만큼 variance를 증가시켰

다. 이때 variance가 크다는 것은 data의 모양이 상당히 많이 변한다는 것을 의미하고, variance가 작다는 것은 data의 모양이 마치 영문자의 인쇄체같이 변화가 거의 없다는 것을 의미한다. 그림에서 보듯이 error와 test sample의 variance는 근사적으로 비례관계에 있다는 것을 알수 있었다.

(표 3)에는 BB방법으로 얻은 결과를 나타내었다. 여기로부터 특징의 탐색순서가 0,14,6,15,1,2,4 였고, Fisher criterion으로부터 얻은 결과가 0,14,6,15,1,13,2였다.

위의 실험에서 가중치들을 각 class마다 다르게 추출하여 인식과정에 사용함으로써 classification error를 줄일 수 있었고, BB 방법을 사용하여 error허용치에 잘맞는 feature subset을 구할 수 있었다. 그리고 인식을 중지하기 위한 stopping rule의 문턱값을 크게 함으로써 더 정확한 인식이 가능하도록 하였다.

표 3. BB 방법에 의한 결과
The result of BB method

특징 subset	Error rate
0	39.76
0, 14	3.1154
0, 14, 6	1.1923
0, 14, 6, 15	0.4231
0, 14, 6, 15, 1	0.2492
0, 14, 6, 15, 2, 4	0.0769

BB방법에 의한 특징순서

0, 14, 6, 15, 1, 2, 4

Fisher criterion에 의한 특징순서

0, 14, 6, 15, 1, 13, 2

IV. 결 론

위의 실험에서 가중치들을 각 class마다 다르게 추출하여 인식과정에 사용함으로써 classification error를 줄일 수 있었고, BB 방법을 사용

하여 error허용치에 잘맞는 feature subset을 구할 수 있었다. 그리고 인식을 중지하기 위한 stopping rule의 문턱값을 크게 함으로써 더 정확한 인식이 가능하도록 하였다.

결론적으로 제안한 multitree 형상 인식 기법을 기존의 형상 인식 기법과 비교하여 성능을 평가하면, 샘플의 종류나 사용되는 특징의 종류에 따라 다르겠지만, 본 논문에서 사용한 특징으로는 특징의 수가 3개일때 오인식율이 Hierarchical 형상 인식 기법에 비해서 약 20%, 기존의 방법에 비해서 약 30%가 감소하였다.

REFERENCE

1. 김 병무, 이 정환, 김 성대, 김 재균, "Binary Decision Tree를 이용한 형상 인식 알고리즘," 통신 학회 학술 대회 논문집, vol. 9, NO.2, pp.699-702, 1986.
2. 이 정희, 권 중장, 김 성대, "투영을 이용한 Pattern Classification Algorithm에 관한 연구," KIEE 추계 종합 학술 대회, vol. V-10, 1987.

3. D.E. Gustafson and Gelfand, "A Nonparametric Multiclass Partitioning Method for Classification," IEEE Proc. pp.645-659. 1980.
4. E. G. Henrichon, JR. and K.S. Fu, "A Nonparametric Partitioning Procedure for Pattern Classification," IEEE Trans. Comput., vol.C-18, NO.7, pp.614-624, JULI.1969.
5. J.H. Friedman, "A Recursive Partitioning Decision Rule for Nonparametric Classification," IEEE Trans. Comput., vol.C-18, pp.770-775, APR.1977.
6. J. Schuermann and W. Doster, "A Decision Theoretic Approach to Hierarchical Classifier design," Pattern Recognition, vol. 17, NO. 3, pp.359-369, OCT.1984.
7. K.S. Fu, Y.T. Chien and G.P. Cardillo, "A Dynamic Programming to Sequential Pattern Recognition," IEEE Trans. Comput., vol.EC-16, NO.6, pp.790-803, DEC.1967.
8. M.W. Kurzynski, "The Optimal Strategy of a Tree Classifier," Pattern recognition, vol.16, NO.1, pp.81-87.1983.
9. P.M. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Selection Subset Selection," IEEE Trans. Comput., vol.C-26, pp.917-922, SEP.1977.



金 泰 成 (Tae Sung KIM) 正會員
 1964年10月8日生
 1983年3月~1987年2月: 延世大學校工
 大 電子工學科 卒業(工學士)
 1987年~1989年2月: 韓國科學技術院電
 氣 및 電子工學科(工學碩士)



金 聖 大 (Seong Dae KIM) 正會員
 1953年12月26日生
 1977年2月: 서울大學校 電子工學科 學
 士學位 取得.
 1979年2月: 韓國科學技術院電氣 및 電
 子工學科 碩士學位 取得.
 1983年12月: E. N. S. E. I. H. T 博
 士學位取得
 1984年9月~現在: 韓國科學技術院電氣
 및 電子工學科 助教授. 主

關心分野는 영상인식, 영상부호화 및 영상처리 3 차원물체 인식, 이동물체추적 등임.



李 正 熙 (Jung Hee LEE) 正會員
 1963年10月21日生
 1982年~1986年: 漢陽大 工科學 電子
 工學科(工學士)
 1986年~1988年: 韓國科學技術院 電氣
 및 電子工學科(工學碩
 士)