

論文2000-37SP-2-5

다중 클래스 데이터를 위한 분류오차 최소화기반 특징추출 기법

(Optimizing Feature Extraction for Multiclass problems Based on Classification Error)

崔 義 善 * , 李 哲 熙 *

(Eui-Sun Choi and Chul-Hee Lee)

요 약

본 논문에서는 다중 클래스 데이터를 위한 특징 추출 방법을 최적화하는 기법을 제안한다. 제안된 특징 추출 기법은 분류 오차에 기반한 방법으로 특징 공간(feature space)을 탐색하여 가우시안 최대우도 분류기(Gaussian ML Classifier)의 분류오차(classification error)가 최소가 되도록 하는 특징벡터 집합을 구하는 방법이다. 제안된 방법은 임의의 초기 특징벡터를 설정한 후 steepest descent 알고리즘을 적용하여 분류오차가 감소하는 방향으로 초기벡터를 갱신시킨다. 본 논문에서는 순차탐색 및 전체탐색 두 가지의 방법을 제안하며 순차탐색은 추가로 특징벡터를 구하는 경우 이미 구해진 특징벡터를 포함하여 최소의 분류오차를 얻을 수 있는 특징벡터를 구한다. 반면에 전체탐색 방법은 추가의 특징벡터를 구할 경우 새로운 초기 특징벡터 집합을 설정하여 이미 구해진 특징벡터를 포함하는 제약은 받지 않는다. 실험결과 제안된 두 가지 방법은 기존의 특징추출 방법보다 우수한 성능을 보여주고 있다.

Abstract

In this paper, we propose an optimizing feature extraction method for multiclass problems assuming normal distributions. Initially, We start with an arbitrary feature vector. Assuming that the feature vector is used for classification, we compute the classification error. Then we move the feature vector slightly in the direction so that classification error decreases most rapidly. This can be done by taking gradient. We propose two search methods, sequential search and global search. In the sequential search, an additional feature vector is selected so that it provides the best accuracy along with the already chosen feature vectors. In the global search, we are not constrained to use the chosen feature vectors. Experimental results show that the proposed algorithm provides a favorable performance.

1. 서 론

패턴분류시 특징추출(feature extraction)은 분류기

* 正會員, 延世大學校 電氣·컴퓨터工學科

(Dept. Electrical and Computer Eng., Yonsei University)

※ 본 연구는 연세대학교 교내연구비 지원으로 이루어졌습니다.

接受日字 : 1999年2月18日, 수정완료일 : 2000年1月20日

(classifier)의 성능을 향상시키고 복잡도(complexity)를 감소시키는 등 효율적인 패턴분류를 가능하게 한다. 이와 같은 이유로 특징추출은 패턴분류나 패턴인식 문제에 있어서 매우 중요하게 다루어져왔던 주제이며 오랫동안 많은 연구자들에 의해 폭넓게 연구되어 왔다^[1-7].

일반적으로 패턴인식 시스템에서는 입력데이터의 효율적인 저장과 패턴 분류단(classification stage)으로의 실시간 전송 등을 위해 데이터의 크기를 줄이는 것이 바람직한데 이 경우 원래의 데이터로부터 분류에 필요

한 정보들만을 추출하게 된다. 즉 특징추출은 고차원의 데이터를 저차원의 데이터로 변환시키는 것으로 볼 수 있다^[1]. 일반적으로 패턴분류(pattern classification)문제에서 원래의 데이터가 아닌 적은 수의 특징을 사용한 분류는 분류정확도(classification accuracy)가 저하되지만 실제 패턴인식 시스템에서 많은 수의 특징을 사용하는 것은 분류비용(classification cost) 측면에서 볼 때 매우 비효율적이다. 간단한 선형 분류기(linear classifier)의 경우 계산량은 사용된 특징 수에 비례하고, 원격탐사 데이터 등과 같은 다중 스펙트럴(multispectral) 데이터에 자주 사용되는 가우시안 최대우도 분류기(Gaussian maximum likelihood classifier)의 계산량은 특징 수의 제곱에 비례한다^[2]. 따라서 효율적인 패턴인식 시스템 구현을 위해서는 가능한 최소의 특징들을 사용해야 하고 이와 함께 분류에 필요한 정보의 손실을 최소화하여야 한다.

패턴 분류 및 패턴 인식에 있어서 대부분의 특징추출 알고리즘들은 2개의 클래스에 대하여 정의된 결정기준함수를 사용한다. 예를 들어 Fisher의 알고리즘은 가장 큰 클래스 분리도(separability)를 얻을 수 있는 결정기준함수를 사용하며^[3] 최근 발표된 결정경계 특징추출 알고리즘은 결정경계특징행렬(decision boundary feature matrix)을 기준함수로 사용하고 있다^[4]. 다중클래스(multiclass)문제의 경우 두 개의 클래스에 대하여 정의된 결정기준함수들은 확장되어 사용되는데 일반적으로 두 개의 클래스를 기반으로 만들어진 특징추출 방법을 다중 클래스 문제로 확장시킬 경우 엄밀한 의미에서 최적의 해를 기대할 수 없다. 다중 클래스문제에 적용되는 canonical analysis방법도 모든 클래스의 평균 벡터(mean vector)와 공분산 행렬(covariance matrix)을 이용하여 구한 결정기준함수를 사용하여 비록 많은 경우 만족할 만한 성능을 보여 주지만, 어떠한 의미에서도 최적의 해라고 할 수 없다^[5]. 즉 현재까지 수많은 특징추출기법들이 제안되었지만 다중클래스 문제에 대해서는 어떠한 기법도 최적의 해라고 할 수 없는 상황이다. 따라서 본 논문에서는 다중 클래스 문제의 최적 특징추출에 대하여 근본적인 고찰을 위해 전체 특징공간을 탐색하여 특징추출을 최적화하는 기법을 제안한다.

본 논문의 구성은 제II단락에서 특징추출의 개요와 이와 관련된 연구들을 소개하고 문제점과 개선점 등을 살펴보면 제III단락에서는 본 논문에서 제안하는 순차

탐색(sequential search)과 전체탐색(global search) 알고리즘에 대하여 구체적으로 기술한다. 제IV단락에서는 실험결과 및 기존 알고리즘들과의 패턴 분류시 성능을 비교평가하고 마지막으로 제V단락에서 결론을 맺는다.

II. 특징추출의 개요와 관련연구

패턴분류 문제에서 취급하는 데이터들은 일반적으로 서로 다른 통계적분포를 가지며 다중 스펙트럴(multispectral)정보를 포함하는 클래스를 형성하고 있다. 클래스 데이터를 N 차원 유클리드 공간에서 벡터로 표현하면 다음과 같다.

$$\bar{x} = [x_1, x_2, \dots, x_N]^T$$

특징추출은 N 차원의 데이터 \bar{x} 를 $M(M < N)$ 차원의 데이터 \bar{y} 로 변환하는 것으로 볼 수 있으며 이러한 과정을 설명하기 위해 매핑함수 $\mathcal{W}(\bar{x})$ 를 생각할 수 있다. 일반적으로 $\mathcal{W}(\bar{x})$ 는 선형변환 함수이며 기존의 많은 특징추출 알고리즘들이 이와 같은 선형변환 함수를 사용하고 있다. 선형 특징추출(linear feature extraction)은 식 (1)과 같이 선형변환 행렬 Φ_M 를 이용하여 특징을 추출하게 되며 이 경우 행렬 Φ_M 는 M 개의 특징벡터 ϕ_i ($i=1, 2, \dots, M$)로 구성된다. 여기서 ϕ_i 는 $N \times 1$ 열 벡터이다^[2].

$$\bar{y} = \Phi_M^T \bar{x} \quad \Phi_M: N \times M \text{ matrix} \quad (1)$$

Principal components analysis방법에서는 전체 클래스 데이터에 대하여 공분산행렬(covariance matrix) $\Sigma_{\bar{x}}$ 를 구하고 이 행렬의 고유치(eigenvalues) λ_i 와 고유벡터(eigenvectors) ϕ_i 를 계산하여 고유치가 큰 순서에 해당하는 고유벡터를 특징벡터로 선택한다^[2].

$$\Phi = [\phi_1, \phi_2, \dots, \phi_M, \dots, \phi_N] : \Sigma_{\bar{x}} \text{의 고유벡터 집합}$$

$$J = \{\lambda_1 > \lambda_2 > \dots > \lambda_M > \dots > \lambda_N\}$$

$$\Phi_M = [\phi_1, \phi_2, \dots, \phi_M]$$

여기서 J 는 principal components analysis방법에서 특징벡터를 추출하기 위해 사용되는 결정기준이다. 그러나 그림 1에서 볼 수 있듯이 단순히 전체 클래스 데이터들의 분산정보를 기초로 하여 고유치가 큰 고유벡

터를 특징벡터로 선택할 경우 패턴분류 측면에서 볼 때 반드시 효율적이지는 않다^[6].

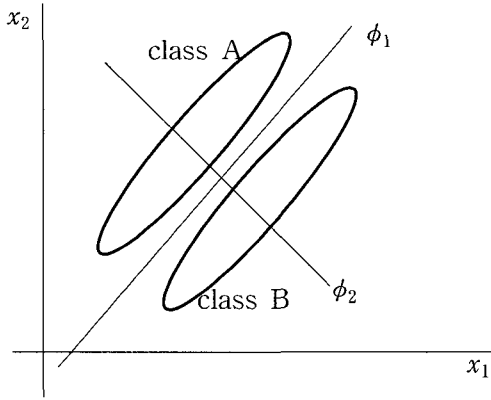


그림 1. Principal component analysis (2 클래스, 2차원 예제 $\lambda_1 > \lambda_2$)

Fig. 1. Principal component analysis (2 class, 2 dimension example $\lambda_1 > \lambda_2$).

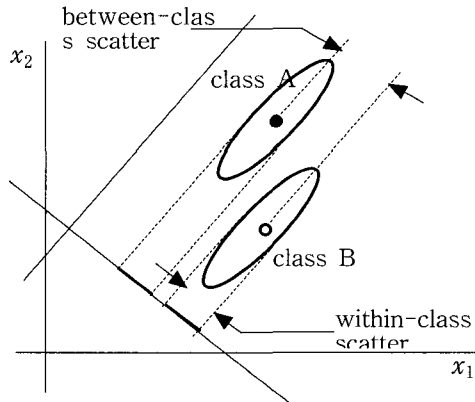


그림 2. Canonical analysis (2 클래스, 2차원 예제)

Fig. 2. Canonical analysis (2 class, 2 dimension example).

Canonical analysis 방법은 이와 같은 문제에 대하여 보다 개선된 방법을 제시한다^[2]. 즉, Canonical analysis 방법은 다음과 같은 결정기준함수 J 의 값을 최대로 하는 특징벡터 d 를 구함으로써 클래스들간의 분리도가 최대가 되는 특징벡터를 추출하게 된다. 그림 2는 이와 같은 과정을 보여주고 있다.

$$J = \frac{\sigma_b^2}{\sigma_w^2} = \frac{\text{between class variance(scatter)}}{\text{within class variance(scatter)}}$$

$$= \frac{d^T \Sigma_b d}{d^T \Sigma_w d}$$

위 식에서

$$\Sigma_w = \sum_i P(\omega_i) \Sigma_i$$

(within-class scatter matrix)

$$\Sigma_b = \sum_i P(\omega_i) (\mu_i - \mu_0)(\mu_i - \mu_0)^T$$

(between-class scatter matrix)

$$\mu_0 = \sum_i P(\omega_i) \mu_i.$$

$\mu_i, \Sigma_i, P(\omega_i)$ 는 각각 클래스 ω_i 의 평균벡터와 공분산행렬 그리고 선행(prior) 확률이다. Canonical analysis 방법은 대부분의 패턴분류 문제에 있어서 효과적이지만 클래스들간의 평균차이가 크지 않을 경우 이 방법으로 구해진 특징벡터는 신뢰도가 떨어진다. 더욱이 Σ_w 와 Σ_b 는 각 클래스의 평균벡터와 공분산행렬에 가중치를 곱하여 얻게되고 이 값들로부터 구한 특징벡터는 주어진 문제에 대하여 최적의 해가 된다는 아무런 보장이 없게 된다^[6]. 이와 같은 최적성 결여의 문제는 현재 다른 특징추출기법에서도 나타난다. 즉 현재까지 제안된 여러 특징추출기법들이 비록 대부분의 경우 만족할 만한 성능을 보여주지만 엄밀한 의미에서 최적의 해라고 볼 수 없다. 따라서 본 논문에서는 이와 같은 패턴인식의 근본문제를 고찰하기 위하여 전체 특징벡터 공간을 탐색하여 다중 클래스 패턴분류 문제의 특징 추출 기법을 최적화하는 알고리즘을 제안한다. 탐색 알고리즘으로는 순차탐색과 전체탐색을 제안하며 다음에서 상세히 기술한다.

III. 제안된 알고리즘

본 논문에서는 패턴 분류 시 다중 클래스 문제에 대하여 분류오차를 감소시켜 식 (1)의 선형 변환행렬 ϕ 를 구하는 알고리즘을 제안한다. 패턴 분류시 분류 오차는 추출된 특징 벡터의 유효성을 검증하기 위한 최적 결정 조건(optimal criterion)이 된다. 이러한 의미에서 분류 오차에 직접적으로 기반하는 특징 추출 기법은 최적이라고 할 수 있다^[7, 8]. 제안된 알고리즘에서는 전체 특징 공간을 steepest descent algorithm을 사용하여 탐색한다. 제안된 방법은 순차탐색과 전체탐색 두가지 방법으로 임의의 특징벡터를 사용하여 패턴분류를 수행하고 분류오차를 계산한 후, 분류오차가 가장 많이 감소하는 방향으로 특징벡터를 갱신한다.

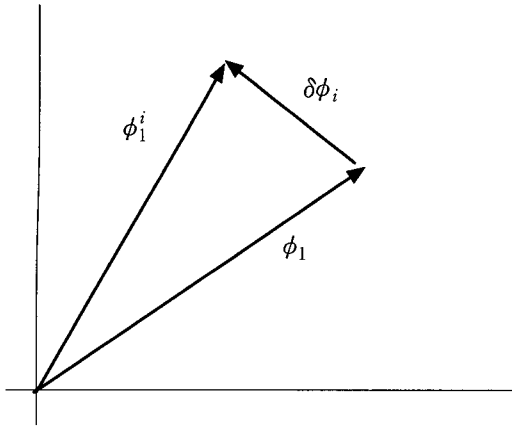


그림 3. 초기 특징 벡터의 이동

Fig. 3. Movement of initial feature vector.

1. 순차탐색

특징벡터 공간을 탐색하여 최소의 분류오차를 가지는 특징벡터를 추출하기 위하여 N 차원 유클리드 벡터 공간에서 직교기저 벡터집합 $\Phi_N = \{\phi_1, \phi_2, \dots, \phi_N\}$ 을 고려한다. 여기서 ϕ_i 는 $N \times 1$ 단위 기저벡터이다. 먼저 특징벡터 한 개를 추출하는 경우 ϕ_1 을 초기 특징벡터로 가정하여 분류오차를 계산하고 그 다음 특징벡터 ϕ_1 을 식 (2)와 같이 나머지 기저벡터 방향으로 조금씩 이동시킨다.

$$\phi_1^i = \phi_1 + \delta\phi_i \quad (i=2, \dots, N) \quad (2)$$

여기서 δ 는 양의 상수이며 그림 3은 위의 과정을 보여준다. 이동된 벡터 ϕ_1^i 을 사용하여 분류오차를 계산하고 다음의 식 (3)을 이용하여 ϕ_i 방향으로의 이동에 따른 오차의 변화율(gradient)을 계산한다.

$$r_i = \frac{\Delta \varepsilon}{\delta} = \frac{\varepsilon(\phi_1^i) - \varepsilon(\phi_1)}{\delta}, \quad \text{for } i=2, \dots, N \quad (3)$$

여기서 $\varepsilon(\phi_1^i)$ 와 $\varepsilon(\phi_1)$ 는 각각 특징벡터 ϕ_1^i 과 ϕ_1 을 사용하여 가우시안 최대우도 분류기로 패턴분류를 수행하였을 때의 분류오차이며, 가우시안 최대우도 분류기는 다음의 식 (4)와 같이 정의된다. 즉, N 차원 유클리드 공간 E^N 에서 관측 데이터를 X 라고 할 때, 정의된 두개의 클래스 w_1, w_2 에 대하여

$$P(w_1)P(X|w_1) > P(w_2)P(X|w_2) \text{ 이면}$$

클래스 w_1 으로 결정,

$$P(w_1)P(X|w_1) < P(w_2)P(X|w_2) \text{ 이면}$$

클래스 w_2 으로 결정.

여기서

$$\text{함수 } h(X) \text{를 } h(X) = -\ln \frac{P(X|w_1)}{P(X|w_2)}, \quad t = \ln \frac{P(w_1)}{P(w_2)}$$

라고 할 때, 클래스 w_1, w_2 에 대하여 정규분포를 가정하면 함수 $h(X)$ 는 다음 식과 같이 나타낼 수 있다^[2] 기.

$$h(X) = \frac{1}{2}(X - \mu_1)' \Sigma_1^{-1}(X - \mu_1) + \frac{1}{2} \ln |\Sigma_1| - \frac{1}{2}(X - \mu_2)' \Sigma_2^{-1}(X - \mu_2) - \frac{1}{2} \ln |\Sigma_2| \quad (4)$$

식 (2), (3)의 과정을 ϕ_i ($i=2, \dots, N$) 기저벡터들에 대하여 반복하고 최종적으로 특징벡터 ϕ_1 은 다음의 식 (5)와 같이 갱신된다^[9].

$$\phi_{1, \text{updated}} = \phi_1 + \alpha \sum_{i=2}^N (-r_i \phi_i) \quad (5)$$

여기서 α 는 양의 상수이며 특징벡터 $\phi_{1, \text{updated}}$ 는 분류오차를 감소시키는 방향으로 갱신된다. 이러한 과정을 매 단계에서 $\phi_{1, \text{updated}}$ 을 사용하여 계산된 분류오차들간의 차이가 문턱치(threshold)보다 작을 때까지 반복하여 최종적으로 특징벡터를 추출한다.

$$\phi_{i, \text{updated}}^{(k+1)} = \phi_{i, \text{updated}}^{(k)} + \alpha \sum_{j=1}^M (-r_j^{(k)} \phi_j^{(k)}) \quad (i=1, 2, \dots, M < N) \quad (6)$$

$$\phi_{i, \text{updated}}^{(0)} = \phi_i$$

$$\text{Termination: } |\varepsilon(\phi_i^{(k+1)}) - \varepsilon(\phi_i^{(k)})| < \gamma_{\text{TH}}$$

여기서 식 (6)의 특징벡터 $\phi_{i, \text{updated}}$ 는 다른 기저벡터들과 선형적으로 독립이 아니므로 그람-슈미트(Gram-Schmidt) 방법을 적용하여 전체 기저집합의 직교성을 유지한다^[10]. 그림 4는 순차탐색방법을 도식적으로 보여준다.

식 (2)에서 특징벡터 ϕ_1 을 이동할 때 전체 벡터공간의 차원이 N 차원이므로 $N-1$ 의 자유도(degree of freedom)가 존재하게 된다. 추가로 특징벡터를 구할 경우 이미 구해진 특징벡터 $\phi_{1, \text{updated}}$ 과 함께 최소의 분

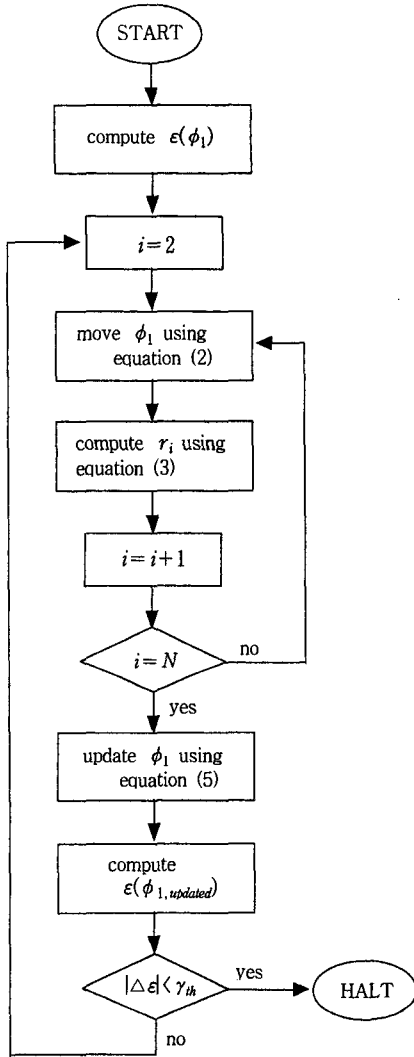


그림 4. 순차탐색의 순서도
Fig. 4. Flowchart of sequential search.

류오차를 얻을 수 있는 특징벡터 $\phi_{2, updated}$ 를 위와 동일한 과정을 통해 구하며, 이 경우 자유도가 하나씩 감소하게 된다. 따라서 순차탐색은 다음에 소개할 전체탐색에 비해 계산속도가 빠른 장점이 있으나 추가로 특징벡터를 구할 경우 이미 구해진 특징벡터를 그대로 사용하기 때문에 이전에 구한 특징벡터가 최적 특징벡터의 일부가 아니라면 결과적으로 볼 때 최적의 해를 기대할 수 없는 단점이 있다.

2. 전체탐색.

앞서 기술한 순차탐색방법은 추가로 특징벡터를 구할 경우 이전에 구해진 특징벡터를 그대로 사용하므로

특징벡터의 수가 증가하는 경우 최적의 해를 기대하기 어려운 단점이 있다. 이러한 문제점을 해결하기 위해 전체탐색방법을 제안한다. 전체탐색방법은 특징 수를 증가시킬 경우 이전에 구해진 특징벡터에 제한받지 않고 임의의 새로운 초기특징벡터 집합을 설정한다. 즉 N 차원 유클리드 벡터공간에서 M 개의 특징벡터를 추출할 경우 먼저 임의의 초기특징벡터 집합 $\Phi_M = \{\phi_1, \phi_2, \dots, \phi_M\}$ ($M < N$)을 설정한다. 그 다음 특징벡터들 사이의 직교성을 유지하며 각각의 특징벡터들을 나머지 벡터 $\phi_{M+1}, \phi_{M+2}, \dots, \phi_N$ 방향으로 조금씩 이동시켜 분류오차의 변화율을 조사한다. 이 경우 $(N-M) \times M$ 의 자유도가 존재하게 된다. 예를 들면 특징벡터 집합 Φ_M 의 벡터 ϕ_j ($j=1, 2, \dots, M$)를 식 (2)와 같이 나머지 특징벡터 ϕ_i 방향으로 이동할 때 $(N-M)$ 개의 새로운 특징벡터집합 $\Phi_{M(i)}^i$ ($i=M+1, \dots, N$)이 존재하게 된다. 여기서 $\Phi_{M(i)}^i$ 는 특징벡터 집합 Φ_M 중에서 특징벡터 ϕ_j 를 ϕ_i 방향으로 이동시켰을 때 생성되는 특징벡터 집합이다. 즉

$$\Phi_{M(i)}^i = \{\phi_1, \phi_2, \dots, \phi_j^i, \dots, \phi_M\}$$

$$\phi_j^i = \phi_j + \delta \phi_i \quad (i=M+1, \dots, N \quad j=1, \dots, M)$$

이다. 새롭게 구해진 특징벡터 집합 $\Phi_{M(i)}^i$ 을 이용하여 분류오차를 계산하고 다음의 식 (7)과 같이 분류오차의 변화율(gradient) r_j^i 을 계산한다.

$$r_j^i = \frac{\Delta \epsilon}{\delta} = \frac{\epsilon(\Phi_{M(i)}^i) - \epsilon(\Phi_{M(i)})}{\delta} \quad (7)$$

여기서 $\epsilon(\Phi_{M(i)}^i)$ 과 $\epsilon(\Phi_{M(i)})$ 는 각각 특징벡터집합 $\Phi_{M(i)}^i$ 과 $\Phi_{M(i)}$ 을 사용하여 패턴분류를 수행하였을 때의 분류오차이다. 이 과정을 $\Phi_{M(i)}^i$ ($i=M+1, \dots, N$)에 대하여 반복하고 오차의 변화율 r_j^i ($i=M+1, \dots, N$)를 계산한다. 최종적으로 특징벡터 ϕ_j 는 식 (8)과 같이 갱신되며 동일한 과정을 벡터 ϕ_j ($j=1, \dots, M$)에 대해 적용하여 특징벡터 집합 $\Phi_{M, updated}$ 을 구한다.

$$\phi_{j, updated} = \phi_j + \alpha \sum_{i=M+1}^N (-r_j^i \phi_i)$$

$$\Phi_{M, updated} = \left[\sum_{j=1}^M \left\{ \phi_j + \alpha \sum_{i=M+1}^N (-r_j^i \phi_i) \right\} \right] \quad (8)$$

이러한 과정을 순차탐색방법과 마찬가지로 분류오차의 변화가 문턱치(threshold)보다 작을 때까지 반복하며 매 단계마다 그람-슈미트(Gram-Schmidt)방법을 적용하여 N 차원 벡터공간에서의 직교기저집합 ϕ_N 의 직교성을 유지한다. 예를 들어 3차원 벡터공간에서 두 개의 특징벡터를 추출한다고 가정하면 다음 식과 같이 벡터집합 $\phi_{2(1)}^3, \phi_{2(2)}^3$ 각각에 대하여 식 (7)를 이용하여 분류오차의 변화율을 계산하고 식 (8)을 이용하여 특징벡터 집합 $\phi_{2, updated}$ 를 구한다.

$$\phi_{2(1)}^3 = \{\phi_1 + \alpha \phi_3, \phi_2\}$$

$$\phi_{2(2)}^3 = \{\phi_1, \phi_2 + \alpha \phi_3\}$$

$$\phi_{2, updated} = \{\phi_{1, updated}, \phi_{2, updated}\}$$

표 1. 8클래스에 관한 정보

Table 1. Information of 8 classes.

No. class	Species	Date	No. of sample
1	OATS	780602	259
2	GRAIN SORGHUM	770308	279
3	WINTER WHEAT	780921	292
4	WINTER WHEAT	770920	292
5	WINTER WHEAT	781026	393
6	SUMMER FALLOW	760928	411
7	SPRING WHEAT	781026	441
8	SPRING WHEAT	780709	454

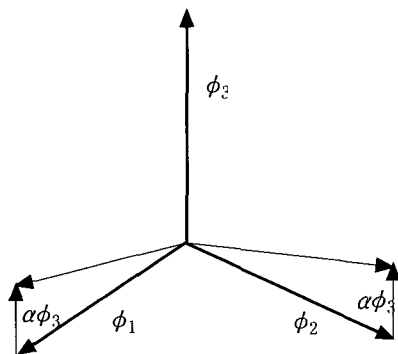


그림 5. $N=3, M=2$ 일때의 전체탐색의 예
Fig. 5. Example of global search ($N=3, M=2$).

그림 5는 위의 과정을 보여준다. 순차탐색 알고리즘에 비해 전체탐색 알고리즘의 장점은 초기벡터집합의 설정이 자유롭다. 즉, 특징 수를 증가시킬 경우 이미 구해진 특징벡터에 제한받지 않고 초기 벡터 집합을 다르게 설정함으로써 지역극소에 도달할 가능성을 줄일

수 있다. 그러나 계산량은 순차탐색에 비해 현저하게 증가하는 단점이 있다.

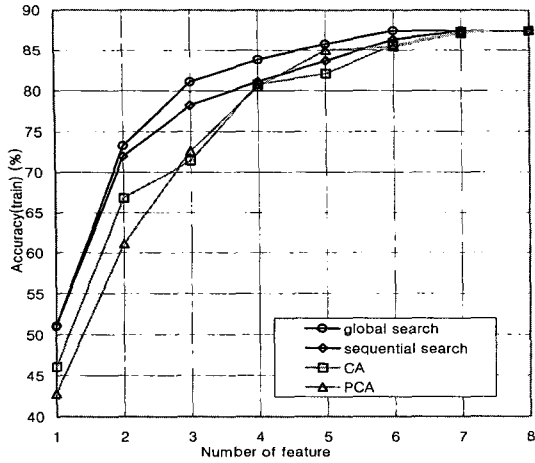
IV. 실험결과 및 고찰

본 논문에서 제안한 특징추출 알고리즘의 성능을 평가하기 위하여 가우시안 최대우도 분류기(Gaussian ML classifier)를 이용하여 분류정확도를 측정하였다. 제안된 알고리즘과 성능비교를 위해 사용된 기존의 특징추출 알고리즘은 canonical analysis 방법과 principal component analysis 방법이며, 실험에 사용된 데이터는 원격 탐사된 실제데이터로^[12] 60개의 스펙트럴 밴드로 구성되었다.

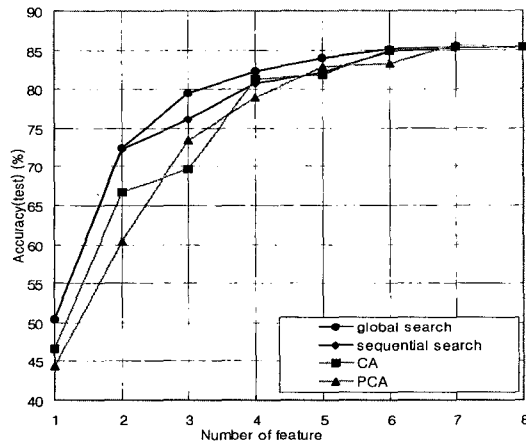
첫 번째 실험에서는 FSS 데이터에서 8개의 클래스를 선택하고 클래스 데이터의 인접 밴드들을 더하여 차원 수를 8로 조정하였다. 선택된 클래스에 관한 정보는 표 1과 같다. 각각의 클래스에서 임의로 추출된 200개의 샘플을 훈련 데이터로 사용하였으며 나머지 데이터를 시험 데이터로 사용하였다^[13]. 실험에 사용된 δ, α 의 값은 각각 0.02와 0.1이고 문턱치는 10^{-4} 이다. 특징벡터를 추출하는 과정에서 초기 특징벡터로 다음과 같은 단위 벡터를 사용하였다.

$$\phi = [\phi_1, \phi_2, \dots, \phi_N] = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \\ & & & 1 \end{bmatrix}$$

그림 6은 각 특징추출기법의 성능을 보여주며, 제안된 순차탐색 및 전체탐색 기법이 기존의 방법에 비하여 우수한 성능을 보여주고 있음을 알 수 있다. 특히 3개의 특징을 사용한 경우 제안된 전체탐색 기법은 훈련 데이터의 경우 81.06%의 분류정확도를 보인 반면, canonical analysis는 71.4%, principal component analysis는 72.6%의 성능을 보여주고 있다. 이는 다중 클래스 문제의 경우 기존의 특징추출 방법이 찾지 못하는 우수한 특징들이 존재함을 암시한다. 이 문제에서 canonical analysis는 상대적으로 저조한 성능을 보여주고 있는데 이는 클래스들의 평균벡터가 유사한 데 그 원인을 찾을 수 있다. 그림 7은 차원에 따른 8개 클래스의 평균벡터 성분값을 보여주고 있는데, 일부 클래스들의 평균벡터 성분값이 비교적 유사함을 볼 수 있다. 즉 canonical analysis는 평균벡터에 중점을 둔 기법으로 평균벡터가 유사한 경우 성능이 저하된다^[5]. 다음



(a)



(b)

그림 6. 성능 비교(8클래스 문제)

(a) 훈련데이터 (b) 시험데이터

Fig. 6. Performance comparison(8classes problem).

(a) training data. (b) test data.

실험에서는 FSS 데이터에서 10개, 12개의 클래스를 선택하여 같은 실험을 수행하였다. 표 2-3은 각 클래스에 관한 정보를 보여주며, 10개 클래스 문제의 경우 200개의 훈련 데이터를, 12개 클래스 문제는 300개의 훈련 데이터를 사용하여 평균벡터와 공분산행렬을 추정하였다. 그림 8-9는 각각의 경우에 추정된 클래스들의 평균 벡터들을 3차원 공간에 도시한 그림으로 각 클래스의 평균벡터들간의 유사성은 비교적 적다. 그림 10-11은 각각 10개, 12개 클래스 문제에 대한 성능비교를 보여 주고 있으며 제안된 기법은 기존의 방법들에 비하여 우수한 성능을 나타내고 있다. 특히 전체탐색 기법은 기존 방법들과 비교하여 월등한 성능을 보여주고 있다.

그러나 그림 6, 10, 11에서 클래스 수가 증가하는 경우

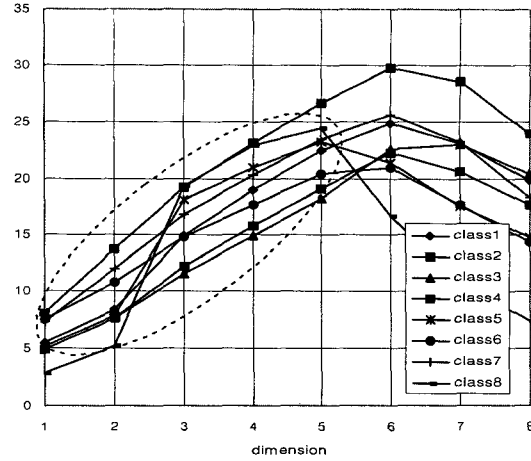


그림 7. 8개 클래스의 평균 벡터

Fig. 7. Mean vectors of 8 classes.

표 2. 10클래스에 관한 정보

Table 2. Information of 10 classes.

No. class	Species	Date	No. of sample
1	WINTER WHEAT	780515	223
2	OATS	780602	259
3	GRAIN SORGHUM	770308	279
4	WINTER WHEAT	780921	292
5	WINTER WHEAT	770920	292
6	SPRING WHEAT	771018	313
7	WINTER WHEAT	781026	393
8	SUMMER FALLOW	760928	411
9	SPRING WHEAT	781026	441
10	SPRING WHEAT	780709	454

표 3. 12클래스에 관한 정보

Table 3. Information of 12 classes.

No. class	Species	Date	No. of sample
1	WINTER WHEAT	770308	691
2	WINTER WHEAT	770626	677
3	WINTER WHEAT	771018	660
4	WINTER WHEAT	770503	657
5	SUMMER FALLOW	770626	643
6	SPRING WHEAT	780726	515
7	SPRING WHEAT	780602	515
8	SPRING WHEAT	780515	474
9	SPRING WHEAT	780921	469
10	SPRING WHEAT	780816	464
11	SPRING WHEAT	780709	454
12	SPRING WHEAT	781026	441

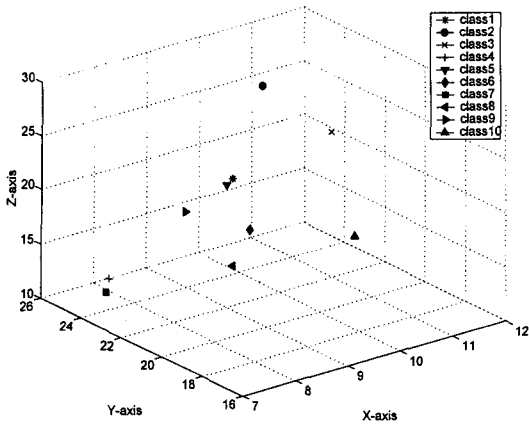


그림 8. 3차원 공간에서의 10개 클래스의 평균벡터
Fig. 8. Distribution of mean vectors of 10 classes in 3 dimensional space.

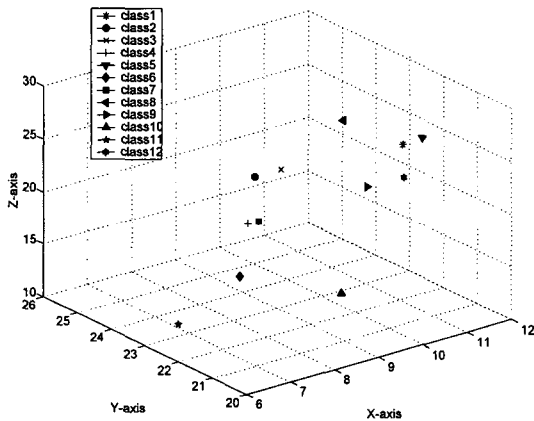
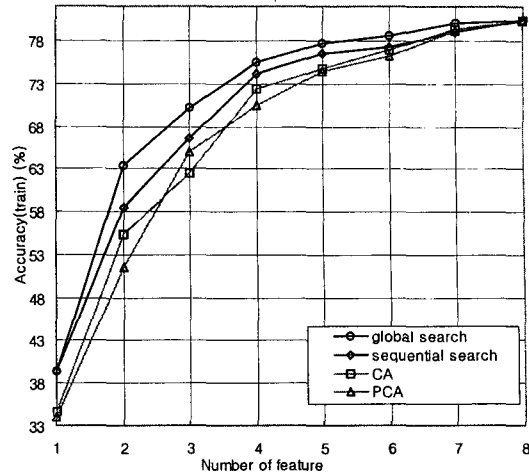
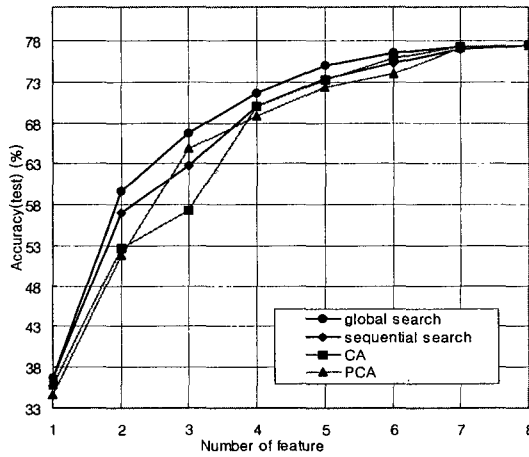


그림 9. 3차원 공간에서의 12개 클래스의 평균벡터
Fig. 9. Distribution of mean vectors of 12 classes in 3 dimensional space.

기존 특징 추출 기법들에 대한 본 논문에서 제안한 특징 추출 기법의 성능 향상 폭은 상대적으로 감소하는



(a)



(b)

그림 10. 성능 비교(10클래스 문제)

(a) 훈련데이터 (b) 시험데이터

Fig. 10. Performance comparison(10classes problem).
(a) training data. (b) test data.

데, 일반적으로 클래스 수가 증가하는 경우 분류 정확

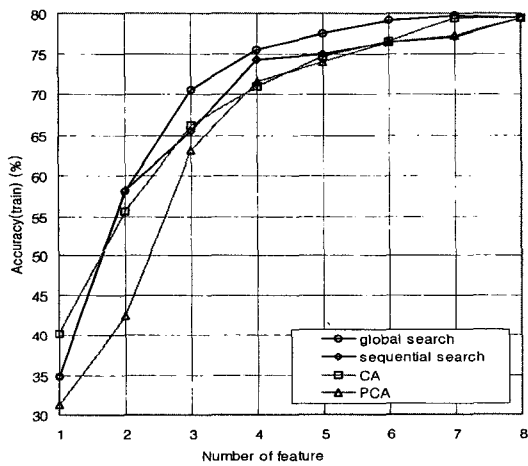
표 4. 클래스 조합(클래스 5개) 40개에 대한 초기 벡터 설정에 따른 성능분석(실제 데이터)

Table 4. Performance analysis of initial vectors(average classification accuracies of 40 sets of 5 classes chosen from real remotely sensed data).

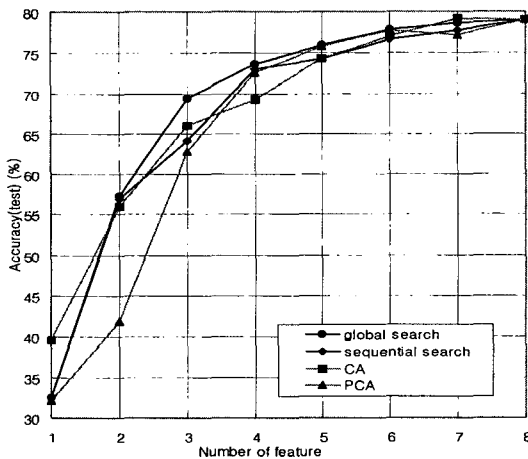
차 원	훈 련 데 이 터					시 험 데 이 터				
	1	2	3	4	5	1	2	3	4	5
전체탐색(단위벡터)	63.28	78.96	84.20	86.08	87.44	62.49	78.86	84.05	85.56	87.17
전체탐색(CA)	66.73	80.46	85.5	87.13	87.44	66.56	80.02	84.59	86.62	87.17
Canonical analysis	61.23	76.49	82.81	85.25	87.44	61.81	76.54	82.38	84.81	87.17
Principal Component analysis	54.11	67.97	79.71	84.21	87.44	55.76	69.87	80.86	83.79	87.17

표 5. 클래스 조합(클래스 5개) 40개에 대한 초기 벡터 설정에 따른 성능분석(합성 데이터)
 Table 5. Performance analysis of initial vectors(average classification accuracies of 40 sets of 5 classes generated with normal distribution).

차 원	훈련 데이터					시험 데이터				
	1	2	3	4	5	1	2	3	4	5
전체탐색(단위벡터)	61.98	79.87	86.61	89.71	91.13	62.01	79.53	85.67	89.07	91.04
전체탐색(CA)	64.63	80.36	87.27	90.05	91.13	64.67	79.73	86.53	89.33	91.04
Canonical analysis	60.46	76.89	84.56	88.34	91.13	60.67	77.02	84.05	87.85	91.04
Principal Component analysis	53.65	70.33	83.62	88.47	91.13	54.37	70.09	83.34	88.02	91.04



(a)



(b)

그림 11. 성능 비교(12클래스 문제)

(a) 훈련데이터 (b) 시험데이터

Fig. 11. Performance comparison(12classes problem).

(a) training data. (b) test data.

용할 경우 전체 특징을 사용하였을 때와 유사한 성능을 얻기 어렵다. 따라서 특징 추출 기법들간의 성능차이는 상대적으로 적게 나타날 수 있다. 또한 그림 10에서 시험 데이터의 경우 이전에 구해진 특징벡터를 그대로 사용하는 순차탐색 기법은 전체 탐색에 비하여 상대적으로 성능향상이 저조한데 이는 본 논문에서 제안한 방법이 초기 특징벡터에 따라 지역극소에 빠질 수 있음을 보여준다.

다음 실험에서는 표 3의 12개 클래스에서 5개의 클래스로 구성된 클래스 조합 40개를 임의로 선택하여 초기 특징벡터 설정에 따른 성능을 분석하였다. 초기 특징벡터로는 앞의 실험에서 사용한 단위벡터와 canonical analysis방법으로 구해진 특징벡터를 사용하였으며 각각의 클래스에 대하여 훈련데이터는 200개이다. 표 4는 40개의 클래스 조합들에 대하여 특징벡터의 수가 증가함에 따른 성능을 측정하고 평균을 구한 결과이다. 표 4에서 특징벡터 1~3개를 사용하는 경우 초기 벡터가 canonical analysis방법으로 구해진 특징벡터 일 때 단위벡터를 사용한 경우보다 약 3.5%~4%의 성능이 향상되었으며 특징벡터를 4개 이상 사용하는 경우 약 1%정도의 성능이 향상되었음을 볼 수 있다. 표 5는 40개 클래스 조합들에 대하여 각각의 클래스의 통계정보(평균벡터, 공분산행렬)를 이용하여 가우시안 특성을 갖는 데이터를 발생시킨 후 실험한 결과이다. 샘플 수는 1000개이고 훈련데이터의 수는 200개이다. 표 5에서 볼 수 있듯이 특징벡터 1~3개를 사용하였을 경우 약 2%~3%, 특징벡터가 4개 이상인 경우 약 1%정도의 성능향상을 나타냈다. 그림 12는 표 4의 초기 벡터 설정에 따른 성능향상 실험 결과를 그래프로 도시한 것이다.

도는 완만하게 증가하며 주어진 몇 개의 특징만을 사

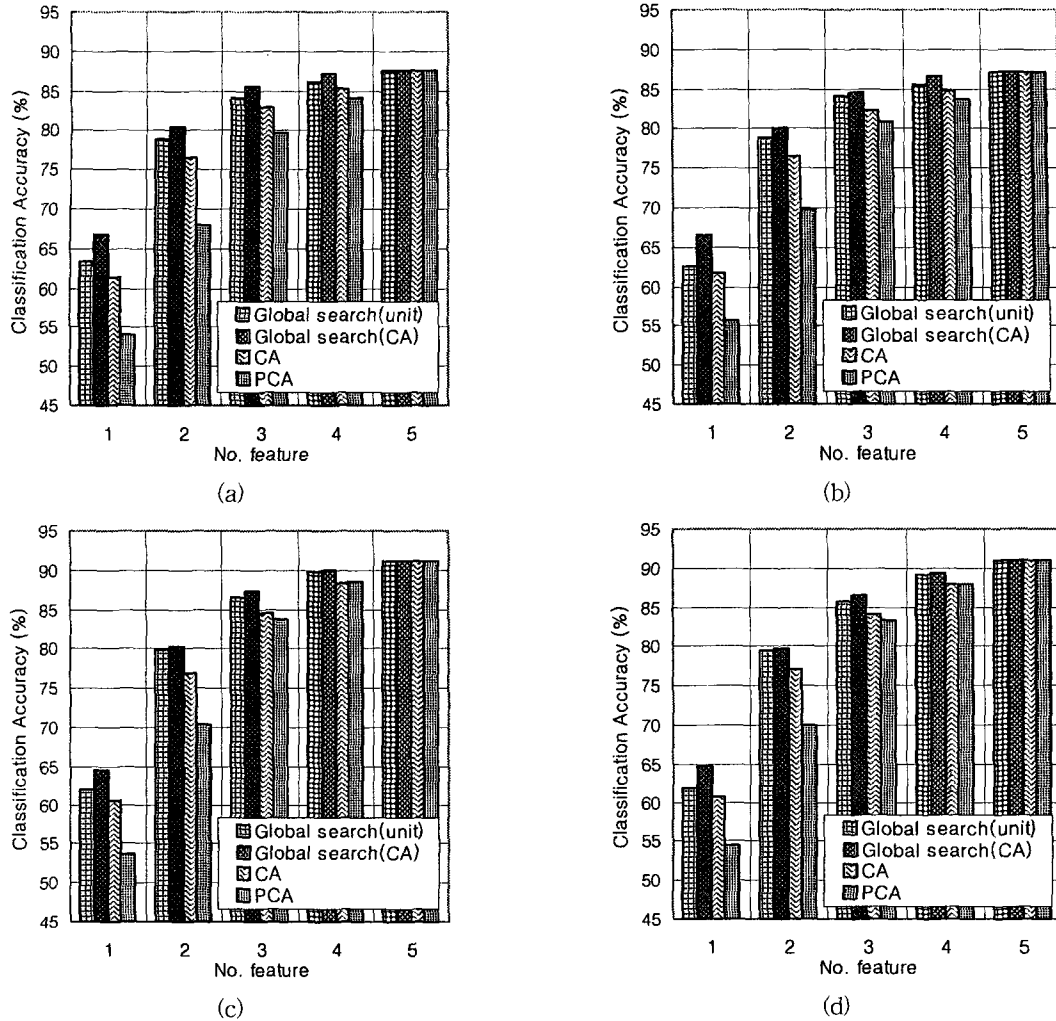


그림 12. 초기벡터 설정에 따른 성능향상

(a) 실제 데이터(훈련) (b) 실제 데이터(시험) (c) 합성 데이터(훈련) (d) 합성 데이터(시험)

Fig. 12. Performance comparison of different initial vectors.

(a) real data(training) (b) real data(test) (c) generated(training) (d) generated data(test)

V. 결론

본 논문에서는 다중 클래스 패턴분류 문제에서 분류 오차를 최소화하는 특징 추출 최적화 알고리즘을 제안하였다. 제안된 방법은 순차탐색 및 전체탐색 기법으로서 직접적으로 분류오차를 최소화하여 패턴분류 시 최적의 해를 기대할 수 있다. 실험결과 제안된 알고리즘이 기존의 알고리즘들보다 우수한 성능을 보였으며, 다중 클래스 패턴분류 문제의 경우 기존의 방법이 찾지 못하는 우수한 특징들이 존재함을 알 수 있었다. 전체 탐색방법은 여러 개의 특징을 추출하는 경우 순차탐색

방법보다 분류정확도 측면에서 향상된 성능을 나타냈으며 또한, 기존 알고리즘으로 구해진 특징벡터를 초기 벡터로 설정하여 성능을 향상시킬 수 있었다. 본 논문에서는 제안된 알고리즘을 가우시안 최대우도 분류기에 적용하였지만, 기타 다른 분류기에도 적용이 가능하다.

참고 문헌

[1] Y. Mallet, D. Coomans, J. Kautsky and Oliver De Vel "Classification Using Adaptive

- Wavelets for Feature Extraction," *IEEE Trans. Computer*, vol. 19, no. 10, pp. 1058-1066, 1997.
- [2] J. A. Richards, *Remote Sensing Digital Image Analysis*. Springer-Verlag, 1993.
- [3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.
- [4] C. Lee and D.A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, pp. 388-400, 1993.
- [5] C. Lee and J. Hong, "Optimizing Feature Extraction for Multiclass cases," *IEEE Intl. Conf. on Systems Man and Cybernetics*, pp. 2545-2548, 1997.
- [6] D. H. Foley and J. W. Sammon, "An Optimal Set of Discriminant Vectors," *IEEE Trans. Computer*, vol. C-24, no. 3, pp. 281-289, 1975.
- [7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic Press, 1990.
- [8] X. Guorang, C. Peiqi, W. Minhui, "Bhattacharyys distance Feature Selection," *IEEE Proc. ICPR '96*, pp. 195-199, 1996.
- [9] H. P. Schwefel, *Evolution and Optimum Seeking*. John Wiley-Sons, Inc, 1994.
- [10] C .G. Cullen, *Matrices and Linear Transformation*. Addison_wesley Publishing Company, 1972.
- [11] L. L. Biel and et. al., "A Crops and Soils Data Base For Scene Radiation Research," *Proc. Machine Process. of Remotely Sensed Data Symp., West Lafayette, Indiana*, 1982.
- [12] C. Lee and D.A. Landgrebe, "Analyzing High Dimensional Multispectral Imagery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 31, no, 4, pp. 792-800, 1993.
- [13] S. Raudys, V. Pikelis, "On Dimensionality, Sample Size, Classification Error, and Complexity of Classification Algorithm in Pattern Recognition," *IEEE Trans. on pattern anal. & machine intel*, vol. 2, no. 3, 1990.

 저 자 소 개



崔 義 善 (正會員)

1998년 2월 연세대학교 전자공학과 졸업(공학사). 1998년 3월~현재 연세대학교 전기·컴퓨터공학과 석사과정 주관심 분야는 패턴인식, 영상 신호처리



李 哲 熙 (正會員)

1980년 3월~1984년 2월 서울대학교 전자공학과 졸업(공학사) 1984년 3월~1986년 2월 서울대학교 대학원 전자공학과(공학석사). 1986년 9월~1987년 3월 Technical University of Denmark (Researcher). 1987년 8월~1992년 12월 Purdue University Electrical Engineering(Ph. D.). 1993년 7월~1996년 8월 National Institutes of Health, Maryland, USA. (Visiting fellow). 1996년 9월~1999년 8월: 연세대학교 기계전자공학부 조교수. 1996년 9월~현재 연세대학교 기계전자공학부 부교수. 주관심분야는 신호처리, 영상처리, 패턴 인식, 신경망