논문 2012-49CI-4-6

특징 추출 알고리즘과 Adaboost를 이용한 이진분류기

(Binary classification by the combination of Adaboost and feature extraction methods)

함 승 록*, 곽 노 준**

(Seaung-Lok Ham and Nojun Kwak)

요 약

패턴 인식과 기계 학습 분야에서 분류는 가장 기본적으로 해결해야 하는 문제의 유형이다. Adaboost 알고리즘은 Boosting 알고리즘의 아이디어를 실제 데이터분석에 이용할 수 있도록 개량한 방법으로써, 단계를 반복하여 나온 여러 개의 약한 분류기와 가중치 값들의 조합으로 강한 분류기를 생성하는 두 개의 클래스를 분류하는 분류기이다. 주성분 분석법과 선형 판별 분석법은 높은 차원의 특징 벡터를 낮은 차원의 특징 벡터로 축소하는 특징 벡터의 차원 감소와 데이터의 특징 추출에도 유용하게 사용되는 방법들이다. 본 논문에서는, 주성분 분석법과 선형 판별 분석법을 이용하여 추출한 특징을 Adaboost 알고리즘의 약 분류기로 사용함으로써, 특징 추출과 분류를 동시에 하고, 인식률을 높이는 효율적인 Boosted-PCA와 Boosted-LDA 알고리즘을 제안한다. 마지막 장에서는, 제안하는 알고리즘으로 UCI Data-Set 중 2 Class-Data와 FRGC Data의 남자와 여자 영상에 대해서 분류 실험을 진행하였다. 실험의 결과로 제안한 Boosted-PCA와 Boosted-LDA 알고리즘이 기존의 특징 추출 알고리즘과 최근접 이웃 분류기, SVM 을 이용한 분류기 방법과 비교하여 인식률이 향상됨을 보인다.

Abstract

In pattern recognition and machine learning society, classification has been a classical problem and the most widely researched area. Adaptive boosting also known as Adaboost has been successfully applied to binary classification problems. It is a kind of boosting algorithm capable of constructing a strong classifier through a weighted combination of weak classifiers. On the other hand, the PCA and LDA algorithms are the most popular linear feature extraction methods used mainly for dimensionality reduction. In this paper, the combination of Adaboost and feature extraction methods is proposed for efficient classification of two class data. Conventionally, in classification problems, the roles of feature extraction and classification have been distinct, i.e., a feature extraction method and a classifier are applied sequentially to classify input variable into several categories. In this paper, these two steps are combined into one resulting in a good classification performance. More specifically, each projection vector is treated as a weak classifier in Adaboost algorithm to constitute a strong classifier for binary classification problems. The proposed algorithm is applied to UCI dataset and FRGC dataset and showed better recognition rates than sequential application of feature extraction and classification methods.

Keywords : Adaboost, PCA, LDA, Boosted-PCA, Boosted-LDA

Ⅰ. 서 론

패턴 인식과 기계 학습 분야에서 분류(Classification)는 가장 기본적으로 해결해야 하는 문제의 유형이다. 패턴인식의 대부분을 이루는 분류 작업

* 학생회원, ** 정회원, 아주대학교 전자공학과
(Department of Electrical Engineering, Ajou University)
※ 이 연구는 한국연구재단 지원 (KPF-2012-0003793)에 의해 이루어졌음.
접수일자: 2011년12월15일, 수정완료일: 2012년7월4일

은 분류기를 이용하여 데이터를 특정 클래스에 할당한다. 가장 널리 알려진 분류기는 Support Vector Machine(SVM)[1], K-Nearest Neighbors(K-NN)[2] 그리고 Neural Networks[3]가 있다. 최근 들어서는, 다양한 분류기의 조합이 관심을 받고 있다. 그 중에서 다수의 규칙들을 조합하여 정확도를 향상시키는 Boosting 알고리즘이 관심 분야 중의 하나이다.

1999년에 Freund & Schapire에 의해 도입된 아다부스트 알고리즘(이하 Adaboost 알고리즘)[4]은 Adaptive Boosting으로써 두 개의 클래스를 잘 분류할 수 있는 분류기이다. Adaboost 알고리즘은 Boosting 알고리즘의 아이디어를 실제 데이터분석에 이용할 수 있도록 개량한 방법으로써, 단계를 반복하여 나온 여러 개의 약한 분류기(week classifier)와 가중치 값들의 조합으로 강한 분류기(strong classifier)를 생성하는 알고리즘이다[4~6]. Adaboost 알고리즘은 학습 초기에 데이터의 가중치는 동일한 상태에서 시작하지만, 잘못 분류된 데이터는 가중치를 증가시키고, 분류가 잘된 데이터는 가중치를 감소시키는 단계를 반복하면서 샘플 데이터의 가중치를 재조정한다. 각 단계에서 가중 에러(weighted error)값이 가장 낮은 분류기가 하나의 약한 분류기이다. Viola-Jones 알고리즘[7]은 약한 분류기로 사각 특징을 사용함으로써, Adaboost 알고리즘을 이용하는 효과적인 실시간 물체 검출 알고리즘으로 알려져 있다.

본 논문에서는, 특징 추출 알고리즘인 주성분 분석법(Principle Component Analysis, PCA)[8~9] 및 선형 판별 분석법(Linear Discriminant Analysis, LDA)[10~11]을 이용하여 추출한 특징을 약한 분류기로 사용하는 Adaboost 구조를 제안하였다. 주성분 분석법과 선형 판별 분석법은 높은 차원의 특징 벡터를 낮은 차원의 특징 벡터로 축소하는 특징 벡터의 차원 감소와 데이터의 특징 추출에도 유용하게 사용되는 방법들이다. 클래스를 분류해 낼 수 있는 정보를 유지한 채 특징 벡터의 차원을 감소하게 되면, 적은 양의 특징 벡터만으로도 충분히 정확한 데이터의 분포를 표현하는 효과를 얻으며, 차원의 저주(The curse of dimensinality)와 같은 문제를 해결하는 효과를 얻을 수 있다.

주성분 분석법은 데이터의 주성분에 해당하는 주축을 평균과 분산을 이용한 통계적인 방법으로 구하여, 특징 벡터를 주성분 방향으로 사영(projection) 시키면 차원을 줄일 수 있다. 이는 데이터의 가장 중요한 축들을 찾아 효율적으로 차원을 줄일 수 있는 장점을 가질 수 있다.

선형 판별 분석법은 각 클래스 간 떨어진 정도를 최대화하는 것이 목적으로써, 주성분 분석법과 마찬가지로 특징 벡터의 차원을 감소시키는 기법 중의 하나이다. 클래스간 분산(between-class catter)과 클래스내 분산(within-class scatter)의 비율을 최대화 하는 방법을 이용하여 데이터에 대한 특징 벡터의 차원을 감소한다. 본 논문에서는, 주성분 분석법과 선형 판별 분석법을 이용하여 추출한 특징을 Adaboost 알고리즘의 약분류기로 사용함으로써, 특징 추출과 분류를 동시에 하는 Boosted-PCA와 Boosted-LDA알고리즘을 제안한다.

논문의 각 장의 내용은 다음과 같다. 제 Ⅱ장에서는 기존의 특징 추출 알고리즘인 주성분 분석법, 선형 판별 분석법과 Adaboost 알고리즘의 방법과 수식에 대하여 설명하고, 제 Ⅲ장에서는 제안하는 Boosted-PCA 알고리즘과 Boosted-LDA 알고리즘의 방법과 수식에 대해 설명한다. 제 Ⅳ장에서는 UCI 데이터와 FRGC 얼굴 데이터에 대한 실험을 통해 기존의 알고리즘과 제안한 알고리즘을 비교 분석한다. 그리고 마지막 Ⅴ장에서는 결론을 제시한다.

Ⅱ. 특징 추출 알고리즘과 Adaboost 알고리즘

1. 주성분 분석법

주성분 분석법은 다변량 분석(multivariate analysis) 방법 중 하나로써 대표적인 차원 축소 방법이다. 샘플 데이터 분포에 대한 정보는 그대로 유지하며, 입력의 차원을 감소시켜 고차원 특징 벡터를 단순화 시키며 내적 구조를 분석한다. Karhunen-Loeve 변환이라고도 불리며, 주성분 분석은 상관(correlated)이 있는 데이터들의 분산을 줄이는 차원에서 상관이 없는 데이터의 집합으로 기준 축을 선형 변환하여, 특징 벡터를 재배치하는 것을 말한다. 데이터 집합의 공분산 행렬에 고유 값 분해를 취해 얻으며, 선형 변환의 결과로 데이터가 투영되는 축은 샘플 데이터 집합의 분산이 최대가 되는 방향이다. 주성분 분석법은 샘플 데이터의 차원감소와 특징 추출의 목적으로 사용되며, 고차원 특징 벡터를 저차원 특징 벡터로 축소하여, 시각화가 어려운 고차원의 데이터의 시각화 목적으로도 사용한다.

주성분 분석법 알고리즘 과정은 그림 1과 같다. 먼

0. 학습 샘플 데이터 구성

$x_1, x_2, \cdots, x_N \in R^d$

x_i들의 재구성 에러를 최소화 하는 d'차원의 부분 공간(subspace)을 찾는 것이 목적이다.

1. 학습 샘플 데이터의 평균과 공분산 행렬 S을 구한다.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad S = \sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T$$

2. 공분산 행렬의 고유 값을 계산한 후 주성분을 선택한다.

$$Sv_i = \lambda_i v_i,\ i = 1 \ldots d',\ \lambda_1 > \lambda_2 > \cdots > \lambda_{d'}$$
$$V = [v_1, v_2, \ldots, v_{d'}] \in R^{d \times d'}$$

d개의 고유벡터 λ_i에서 큰 고유 값을 갖는 d'개만을 선택한다. 고유 값이 크면 클수록 주요 특징을 더 많이 표현하게 된다. 변환행렬 V는 고유 값이 큰 d'개의 고유벡터로 구성된다.

3. d차원의 입력 데이터 x, d'차원의 출력 데이터 y는 다음의 변환 식으로 변환한다.

$$y = V^T(x-\mu) \in R^{d'}$$

그림 1. 주성분 분석에 의한 선형변환 알고리즘
Fig. 1. The PCA algorithm

저, d차원의 공간을 가진 N개의 샘플 데이터 $x_1, x_2, \cdots, x_N \in R^d$가 있다.

주성분 분석법은 고차원 d의 데이터를 저차원 d'으로 투영시키는 선형 변환 행렬 V을 구하는 것이 목적이다. 선형 변환 행렬 V을 구하기 위해서 각각의 샘플 데이터들과 샘플 데이터의 평균의 차 벡터를 계산한다. 차 벡터로 공분산 행렬(covariance matrix)을 계산한다. 공분산 행렬에 대해 고유분석을 행하여 d'개의 가장 큰 고유 값 $\lambda_1, \ldots, \lambda_{d'}$을 선택한다. 여기서 고유 값은 평균에 대한 분산의 정도를 나타낸다.

선택한 고유 값과 관련된 고유 벡터를 구하고 변환 행렬 $V = [v_1, v_2, \ldots, v_{d'}] \in R^{d \times d'}$을 만든다. 주성분 분석법에서 특징 추출은 선형 변환 행렬 V와 평균 μ을 이용한다. 다음 식 (1)으로 d차원의 입력 데이터 x가 d' 차원의 출력 특징 y으로 변환한다.

$$y = V^T(x-\mu) \tag{1}$$

주성분 분석법은 정보 손실을 최소로 하며 차원을 줄이는 것이 목적이다. 하지만 클래스가 다른 샘플 데이터들을 분류하는 것을 목적으로 하지 않기 때문에 특징 벡터의 클래스 정보는 고려하지 않아, 데이터의 분포에 따라 분류 성능이 낮을 가능성이 있다.

2. 선형 판별 분석법

선형 판별 분석법은 주성분 분석법과 같이 대표적인 특징 벡터의 차원 축소 방법이다. 주성분 분석법이 샘플 데이터들의 분산이 최대가 되는 축으로 투영되는 선형 변환 행렬을 찾는 것이 목적이었다면, 선형 판별 분석법은 샘플 데이터들을 가장 분류가 잘되는 축으로 투영하는 선형 변환 행렬을 찾는 것이 목적이다. 선형 판별 분석법은 피셔의 선형 판별의 개념을 도입한 것으로, 클래스간 분산과 클래스내 분산을 정의하여, 같은 클래스에 속하는 샘플 데이터들은 가깝게 분포하게 하고 다른 클래스에 속하는 샘플 데이터들은 멀리 분포하게 하는 방법으로 특징 공간상에서 클래스 분리를 최대화하는 주축으로 사영시켜 특징 벡터의 차원을 감소하는 방법이다. 선형 판별 분석법 알고리즘 과정은 그림 2와 같다. 먼저, N개의 샘플 데이터 $\{x_1, \ldots, x_N\}$ 그리고, 각 학습 샘플 데이터는 c개의 클래스 $\{C_1, \ldots, C_c\}$에 속해있다. N_i는 i번째 클래스의 샘플데이터 개수이다. 정확한 분류를 위한 가장 좋은 변환행렬을 찾기 위하여 클래스간 분산 S_B과 클래스내 분산 S_W을 구한다. 클래스내 분산 S_W는 식 (2)의 클래스내 평균을 이용하여 계산하면 식 (3)과 같이 된다.

$$\mu_i = \frac{1}{N}\sum_{x \in C_i} x \tag{2}$$

$$S_W = \sum_{i=1}^{c}\sum_{x \in C_i}(x-\mu_i)(x-\mu_i)^T \tag{3}$$

클래스간 분산 S_B은 총 평균 μ와 총 분산행렬(total

scatter matrix) S_T을 통하여 정의된다. 총 평균 μ와 총 분산행렬 S_T은 아래의 식들과 같이 정의한다.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{i=1}^{c} N_i \mu_i \tag{4}$$

$$S_T = \sum_{x \in C_i} (x-\mu)(x-\mu)^T \tag{5}$$

$$\begin{aligned} S_T &= \sum_{i=1}^{c}\sum_{x \in C_i} (x-\mu_i+\mu_i-\mu)(x-\mu_i+\mu_i-\mu)^T \\ &= \sum_{i=1}^{c}(x-\mu_i)(x-\mu_i)^T + \sum_{i=1}^{c}\sum_{x \in C_i}(\mu_i-\mu)(\mu_i-\mu)^T \\ &= S_W + \sum_{i=1}^{c} N_i(\mu_i-\mu)(\mu_i-\mu)^T \end{aligned} \tag{6}$$

총 분산행렬을 식 (6)과 같이 변형하면 클래스간 분산 S_B을 구할 수 있다. 따라서 총 분산행렬은 클래스내 분산과 클래스간 분산의 합이다. 선형 판별 분석법은 샘플 데이터들의 분류가 가장 잘 되는 축으로 투영하는 선형 변환 행렬 w을 찾는 것이 목표이다. 클래스내 분산을 작게 하고, 클래스간 분산을 크게 만드는 최적화된 변환 행렬 w는 $S_W^{-1}S_B$의 가장 큰 고유 값들이 모인 고유벡터이다.

3. Adaboost 알고리즘

Adaboost 알고리즘은 구별 기능이 약한 분류기들을 선별하여 성능이 향상된 강한 분류기를 만들기 위해 사용된다. Adaboost 알고리즘은 검출하고자 하는 물체의 영상(positive sample)과 그 외의 영상(negative sample)을 가장 잘 분류할 수 있는 약한 분류기(week classifier)의 조합을 찾아 강한 분류기(strong classifier)를 만들어 내는 과정이다. 약한 분류기를 이용하여 정확히 인식된 샘플의 가중치를 감소시키고, 잘못 인식된 샘플의 가중치를 증가시켜 다음 약한 분류기에 반영한다. 최종적인 강한 분류기는 약한 분류기의 조합으로 구성하며, weighted error가 작은 약한 분류기의 수가 증가할수록 오류율이 감소한다.

약한 분류기 $g_j(x)$는 임의의 어떤 샘플 x에 대한 특징 값(feature) f_j, 임계 값(threshold) θ_j과 부등호의 방향(parity)을 표시하는 p_j를 통해 식 (7)와 같이 수식으로 표현되며 분류 결과를 1(positive sample) 또는 0(negative sample)으로 나타낸다. 임계 값 θ_j는 약한

0. 학습 샘플 데이터 구성

$(x_1, y_1), \ldots, (x_N, y_N),\ x_i \in R^d,\ y_i \in C_1, \ldots, C_c$

1. 클래스간 분산 S_B과 클래스내 분산 S_W을 구한다.

$$S_B = \sum_{i=1}^{c} N_i(\mu_i-\mu)(\mu_i-\mu)^T$$

$$S_W = \sum_{i=1}^{c}\sum_{x \in C_i}(x-\mu_i)(x-\mu_i)^T$$

$$\mu_i = \frac{1}{N}\sum_{x \in C_i} x,\quad \mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{i=1}^{c} N_i\mu_i$$

2. 클래스간 분산 S_B과 클래스내 분산 S_W의 비를 최대화하는 변환행렬을 구한다. 이를 목적함수로 나타내면 다음과 같다.

$$W_{opt} = \arg\max\left\{\frac{w^T S_B w}{w^T S_W w}\right\} = [W_1, \ldots, W_m]$$

3. 최적화된 변환행렬 W_{opt}은 일반화된 고유값 문제를 통하여 가장 큰 m개의 고유 값을 갖는 클래스간 분산 S_B과 클래스내 분산 S_W의 집합이다.

$$S_B w_i = \lambda_i S_W w_i (i = 1, \ldots, m)$$

4. d차원의 입력 데이터 x, d'차원의 출력 데이터 y는 다음의 변환식으로 변환한다.

$$y = W^T(x-\mu) \in R^{d'}$$

그림 2. 선형 판별 분석에 의한 선형변환 알고리즘
Fig. 2. The LDA algorithm.

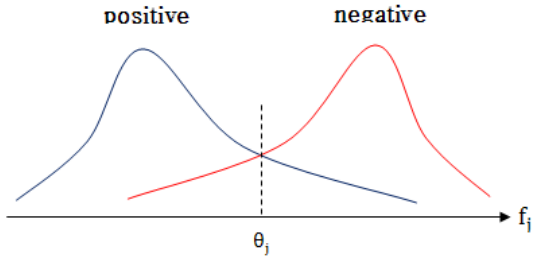

그림 3. 약한 분류기의 예
Fig. 3. The example of weak classifier.

분류기의 분류 성능이 최대가 되도록 하는 값으로 설정한다. 또한 그림 3과 같이 positive sample에 해당하는 특징 값들이 negative sample에 해당하는 특징 값들보다 평균적으로 작은 값을 가진다면 부등호의 방향을 표시하는 p_j는 1, 반대의 경우라면 -1로 결정된다.

$$g_j(x)=\begin{cases}1 & \text{if } p_j f_j(x) < p_j\theta_j \\ 0 & otherwise\end{cases} \tag{7}$$

분류 성능이 좋은 약한 분류기를 선택하고 각각의 중요도를 측정하여 가중치를 부여하고 조합하여 강한 분류기를 만드는 Adaboost 알고리즘의 과정은 그림 4와 같다.

0. 주어진 학습 데이터는 아래와 같다.

$(x_1, y_1),...,(x_N, y_N),\ x_i \in R^d, y_i \in Y \in \{0,1\}$

1. 가중치를 초기화: i번째 샘플에 대한 초기 가중치는 아래의 식과 같다. m과 n은 0(Negative sample)의 총 개수과 1(Positive sample)의 총 개수이다.

$$w_{1,i}=\begin{cases}\frac{1}{2m} & \text{if } y_i=0 \\ \frac{1}{2n} & \text{if } y_i=1\end{cases}$$

2. 약한 분류기의 개수가 T개라면, $t=1,...,T$까지 반복하여 다음 연산을 수행한다.

1) 각각의 약한 분류기 $\{g_j\}_{j=1...k}$에 대하여 오차 $\epsilon=\sum_{i=1}^{N} w_{t,j}|h_j(x_i)-y_i|$를 구한다. 이 때 가장 작은 오차 ϵ_t를 가지는 약한 분류기 g_j가 단계의 약한 분류기 h_t가 된다.

2) 계수 $\alpha_t=\log(\frac{1-\epsilon_t}{\epsilon_t})$를 계산하고, 가중치 $w_{t+1,i}=w_{t,i}\times e^{\alpha_t|h_t-y_i|}$와 같이 변경하고, $w_{t+1,i}\leftarrow\frac{w_{t+1,i}}{\sum_{j=1}^{u} w_{t+1,i}}$와 같이 정규화 한다.

3. 강한 분류기는 아래와 같다.

$$H(x)=\begin{cases}1 & \sum_{t=1}^{T}\alpha_t h_t(x) > 0.5 \\ 0 & otherwise\end{cases}$$

그림 4. Adaboost 알고리즘
Fig. 4. The Adaboost algorithm.

Step 0. 먼저 학습할 샘플 데이터를 구성한다. 학습할 샘플 데이터의 수는 N개 이며, x_i는 각각의 샘플 데이터의 값을, y_i는 각 샘플에 해당하는 클래스 정보에 따라 positive sample이면 1, negative sample이면 0의 값을 갖는다.

Step 1. 모든 샘플 데이터에 대하여 동일한 가중치를 줘서 초기화 한다. m과 n은 0(Negative sample)의 총 개수와 1(Positive sample)의 총 개수이다. 가중치란 각각의 학습 샘플들이 약한 분류기의 분류 성능을 결정하는데 얼마나 큰 영향을 미치는가에 대한 척도이며, 오분류되는 학습 샘플들의 가중치의 총합이 최소인 약한 분류기가 선택된다. 즉, 다시 말해서 가중치가 높은 샘플들을 가장 정확하게 분류하는 특징이 약한 분류기로 선택되는 것이다.

Step 2. Adaboost 알고리즘에서 가장 중요한 부분이다. T개의 약한 분류기를 선택하기 위하여 1)~2)까지의 과정을 T번 반복하여 수행한다.

1)에서는 약한 분류기를 결정하는 단계이다. 약한 분류기는 수많은 특징들 중에서 임의의 어떤 j번째 분류기 g_j를 학습시키고 가중치를 고려한 오차 ϵ를 계산한다. 각각의 분류기의 오차 ϵ를 비교하여 가장 최소값을 ϵ_t로 결정하고 최적의 약한 분류기 h_t로 결정한다.

2)에서는 가중치 갱신과 정규화 하는 과정이다. 가중치를 갱신하기 위하여 오차 ϵ_t를 이용하여 계산한다. α_t는 약한 분류기 h_t의 가중치를 나타내는 중요한 값이다. 만약 오차 ϵ_t가 0.5보다 같거나 작은 값을 가진다면 α_t는 양수의 값을 가진다. 오차 ϵ_t가 작아질수록 α_t의 값은 증가한다. 임의의 i번째 학습 샘플인 x_i가 약한 분류기 h_t에 의해 정확하게 분류되었다면 α_t값이 증가하여 가중치를 낮추어 적용한다. 잘못 분류되었으면 가중치를 높여서 적용하여 다음 반복에서 잘못 분류했던 데이터를 잘 분류하는 약한 분류기를 선택하도록 한다. 가중치를 갱신한 후 각각의 학습 샘플들의 가중치의 총합이 1이 되도록 정규화 한다.

Step 3. 강한 분류기 H는 Step 2의 1)과 2)를 T번 반복 수행하여 얻은 T개의 약한 분류기 h_t와 α_t를 조합

하여 구성한다. α_t는 각 단계에서 선택된 약한 분류기에 대한 가중치이다. 즉, t번째 약한 분류기 h_t의 가중치는 오차 ϵ_t에 반비례하기 때문에 분류 성능이 좋은 약한 분류기일수록 큰 가중치를 갖게 하여 약한 분류기들의 조합으로 만들어지는 최종적인 강한 분류기의 성능을 더욱 향상시키는 역할을 한다.

Ⅲ. 제안하는 Boosted-PCA와 Boosted-LDA 알고리즘

1. Boosted-PCA 알고리즘

제안하는 Boosted-PCA 알고리즘[12]은 주성분 분석법 알고리즘과 Adaboost 알고리즘을 결합한 알고리즘으로써, 주성분 분석법 알고리즘을 이용하여 추출한 특징을 Adaboost 알고리즘의 약한 분류기로 사용하고자 한다. 이를 위해서는 기존의 주성분 분석법 알고리즘에 가중치 $w_{t,i}$를 고려한 계산 과정이 들어가야 한다. 이 때 가중치 $w_{t,i}$는 Adaboost 알고리즘 중 t 단계의 i번째 샘플 데이터의 가중치를 말한다. 가중치를 고려하기 위하여 평균을 구할 때 산술평균 대신 식(8)을 이용하여 가중평균(weighted mean)을 계산한다.

$$\mu_t = \sum_{i=1}^{N} w_{t,i} x_i \tag{8}$$

또한 가중평균을 이용하여 가중 공분산 행렬(weighted scatter matrix) S_t을 식 (9)과 같이 계산한다.

$$S_t = \sum_{i=1}^{N} w_{t,i}(x_i - \mu_t)(x_i - \mu_t)^T \tag{9}$$

가중 공분산 행렬 S_t의 고유값 분석(eigen analysis)을 통하여 고유 값이 큰 k개의 고유 벡터(eigenvector) $\{v_j\}_{j=1 \cdots k}$를 구한다. 제안하는 Boosted-PCA 알고리즘에서는 고유 벡터 v_j를 약한 분류기로 사용함으로써 샘플 데이터 x를 아래 식 (10)을 이용하여 나타낸다.

$$g_j(x) = \begin{cases} 0 & \text{if } v_j^T x \le p_j \tau_j \\ 1 & \text{if } v_j^T x > p_j \tau_j \end{cases} \tag{10}$$

약한 분류기 $g_j(x)$는 샘플 데이터 x와 고유 벡터 v_j를 이용한 변환한 특징 벡터 $v_j^T x$, 임계 값 τ_j과 부등호의 방향을 표시하는 p_j를 통해 표현되며 분류 결과를 1(positive sample) 또는 0(negative sample)으로 나타낸다. 임계 값 τ_j은 약한 분류기 $g_j(x)$의 오차율이 가장 작은 값으로 선택한다.

$$(p_j, \tau_j) = \underset{(p_j, \tau_j)}{\arg min} \sum_{i=1}^{N} w_{t,i} |g_j(x_i) - y_i| \tag{11}$$

분류 성능이 좋은 약한 분류기 $\{g_j\}_{j=1 \cdots k}$를 선택하고 각각의 중요도를 측정하여 가중치를 부여하고 조합하여 강한 분류기를 만드는 Boosted-PCA 알고리즘의 과정은 그림 5와 같다.

Boosted-PCA 알고리즘은 주성분 분석법 알고리즘을 이용하여 추출한 특징을 Adaboost 알고리즘의 약한 분류기로 사용한다. 반복 과정에서 오차율이 가장 작은 특징을 모아 T개의 약한 분류기를 구성한다. 강한 분류기 H는 T개의 약한 분류기 h_t와 α_t를 조합하여 구성한다. α_t는 각 단계에서 선택된 약한 분류기에 대한 가중치이다. 또한, 반복 과정에서 얻은 약한 분류기로 사용하는 특징 벡터 v_j, 임계 값 τ_j, 가중평균 μ_t 그리고 부등호의 방향을 표시하는 p_j을 이용하여 새로운 테스트 데이터를 1(positive sample) 또는 0(negative sample)으로 분류한다. 새로운 테스트 데이터가 들어오면 가중평균과의 차이를 구한 후 약한 분류기로 사용되는 특징 벡터 v_j로 사영한다. 마지막으로 임계값과 부등호의 방향 값 그리고 가중치를 이용하여 사영된 테스트 데이터를 1(positive sample) 또는 0(negative sample)으로 분류한다.

2. Boosted-LDA 알고리즘

제안하는 Boosted-LDA 알고리즘은 선형 판별 분석법 알고리즘과 Adaboost 알고리즘을 결합한 알고리즘이다. Adaboost 알고리즘의 약한 분류기로 선형 판별 분석법을 통하여 추출한 특징을 사용한다. 이를 위해서는 기존의 선형 판별 분석법 알고리즘에 가중치 $w_{t,i}$계산 과정이 있어야 한다. 가중치를 고려하기 위하여 평균을 구할 때 산술평균 대신 가중평균(weighted mean)을 계산한다. 선형 판별 분석법은 클래스내 평균과 전체 평균이 필요한데, 가중평균으로 계산하면 식 (12), 식 (13)으로 나타낼 수 있으며, 샘플 데이터에 대한 가중치의 합은 1이다.

0. 주어진 학습 데이터는 아래와 같다.

$(x_1, y_1), ..., (x_N, y_N),\ x_i \in R^d, y_i \in Y \in \{0,1\}$

1. 가중치를 초기화: i번째 샘플에 대한 초기 가중치는 아래의 식과 같다. m과 n은 0(negative sample)의 총 개수과 1(positive sample)의 총 개수이다.

$$w_{1,i} = \begin{cases} \frac{1}{2m} & \text{if } y_i = 0 \\ \frac{1}{2n} & \text{if } y_i = 1 \end{cases}$$

2. $t = 1, ..., T$까지 반복하여 다음 연산을 수행한다.

1) Weighted PCA:

$$S_t = \sum_{i=1}^{N} w_{t,i}(x_i - \mu_t)(x_i - \mu_t)^T,\ \mu_t = \sum_{i=1}^{N} w_{t,i} x_i$$

가중 공분산 행렬 S_t의 고유값 분석을 통하여 고유 값이 큰 k개의 고유 벡터 $\{v_j\}_{j=1...k}$를 구한다.

2) 약한 분류기를 구성한다.

$$g_j(x) = \begin{cases} 0 & \text{if } v_j^T x \le p_j \tau_j \\ 1 & \text{if } v_j^T x > p_j \tau_j \end{cases}$$

3) 약한 분류기를 선택한다. 각각의 약한 분류기 $\{g_j\}_{j=1...k}$를 통해 오차율을 아래와 같이 계산한다.

$$\epsilon = \sum_{i=1}^{N} w_{t,j} |g_j(x_i) - y_i|$$

가장 작은 오차율을 가지는 약한 분류기g_j를 선택하여, Adaboost의 t번째 특징 h_t이 된다.

4) 계수 $\alpha_t = \log(\frac{1-\epsilon_t}{\epsilon_t})$를 계산하고, 가중치 $w_{t+1,i} = w_{t,i} \times e^{\alpha_t |h_t - y_i|}$와 같이 변경하고,

$w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{u} w_{t+1,i}}$ 와 같이 정규화 한다.

3. 강한 분류기는 아래와 같다.

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > 0.5 \\ 0 & otherwise \end{cases}$$

그림 5. Boosted-PCA 알고리즘
Fig. 5. The Boost-PCA algorithm.

$$\mu_k = \frac{1}{\sum_{x \in C_k} w_{t,i}} \sum_{x \in C_k} w_{t,i} x \tag{12}$$

$$\mu = \sum w_{t,i} x \tag{13}$$

계산한 가중평균들로 클래스간 분산 S_B과 클래스내 분산 S_W을 구하면 식 (14)과 같다.

$$S_B = \sum_{k=1}^{c} \left(\sum_{x \in C_k} w_{t,i} \right) (\mu_k - \mu)(\mu_k - \mu)^T$$

$$S_W = \sum_{k=1}^{c} \sum_{x \in C_k} w_{t,i} (x - \mu_k)(x - \mu_k)^T \tag{14}$$

클래스간 분산 S_B과 클래스내 분산 S_W의 비를 최대화하는 변환행렬 W_j을 구한다. 제안하는 Boosted-LDA 알고리즘에서는 변환행렬 W_j를 약한 분류기로 사용함으로써 샘플 데이터 x를 아래 식 (15)을 이용하여 나타낸다.

$$g_j(x) = \begin{cases} 0 & \text{if } v_j^T x \le p_j \tau_j \\ 1 & \text{if } v_j^T x > p_j \tau_j \end{cases} \tag{15}$$

약한 분류기 $g_j(x)$는 샘플 데이터 x와 고유 벡터 v_j를 이용한 변환한 특징 벡터 v_j^T, 임계 값 τ_j과 부등호의 방향을 표시하는 p_j를 통해 표현되며 분류 결과를 1(positive sample) 또는 0(negative sample)으로 나타내고, 임계 값 τ_j은 약한 분류기 $g_j(x)$의 오차율이 가장 작은 값으로 선택한다.

$$(p_j, \tau_j) = \underset{(p_j, \tau_j)}{\arg min} \sum_{i=1}^{N} w_{t,i} |g_j(x_i) - y_i| \tag{16}$$

식 (16)과 같이 분류 성능이 좋은 약한 분류기 $\{g_j\}_{j=1...k}$를 선택하고 각각의 중요도를 측정하여 가중치를 부여하고 조합하여 강한 분류기를 만드는 Boosted-LDA 알고리즘의 과정은 그림 6과 같다.

Boosted-LDA 알고리즘은 샘플 데이터의 클래스 분리를 최대화하는 변환행렬을 각 단계의 약한 분류기로 사용하여 강한 분류기를 구성한다. 계산 과정에서 나온 임계값과 부등호의 방향 값 그리고 가중치를 이용하여 사영된 테스트 데이터를 1(positive sample) 또는 0(negative sample)으로 분류한다.

0. 주어진 학습 데이터는 아래와 같다.

$(x_1, y_1), ..., (x_N, y_N),\ x_i \in R^d, y_i \in Y \in \{0,1\}$

1. 가중치를 초기화: i번째 샘플에 대한 초기 가중치는 아래의 식과 같다. m과 n은 0(negative sample)의 총 개수과 1(positive sample)의 총 개수이다.

$$w_{1,i} = \begin{cases} \frac{1}{2m} & \text{if } y_i = 0 \\ \frac{1}{2n} & \text{if } y_i = 1 \end{cases}$$

2. $t = 1, ..., T$까지 반복하여 다음 연산을 수행한다.

1) Weighted LDA:

$$S_B = \sum_{k=1}^{c} \left(\sum_{x \in C_k} w_{t,i} \right) (\mu_k - \mu)(\mu_k - \mu)^T$$

$$S_W = \sum_{k=1}^{c} \sum_{x \in C_k} w_{t,i} (x - \mu_k)(x - \mu_k)^T$$

$$\mu_k = \frac{1}{\sum_{x \in C_k} w_{t,i}} \sum_{x \in C_k} w_{t,i} x,\ \mu = \sum w_{t,i} x$$

클래스간 분산 S_B과 클래스내 분산 S_W의 비를 최대화하는 변환행렬 W_j을 구한다.

2) 약한 분류기를 구성한다.

$$g_j(x) = \begin{cases} 0 & \text{if } v_j^T x \le p_j \tau_j \\ 1 & \text{if } v_j^T x > p_j \tau_j \end{cases}$$

3) 약한 분류기를 선택한다. 각각의 약한 분류기 $\{g_j\}_{j=1 \dots k}$를 통해 오차율을 아래와 같이 계산한다.

$$\epsilon = \sum_{i=1}^{N} w_{t,j} |g_j(x_i) - y_i|$$

가장 작은 오차율을 가지는 약한 분류기g_j를 선택하여, Adaboost의 t번째 특징 h_t이 된다.

4) 계수 $\alpha_t = \log(\frac{1-\epsilon_t}{\epsilon_t})$를 계산하고, 가중치 $w_{t+1,i} = w_{t,i} \times e^{\alpha_t |h_t - y_i|}$와 같이 변경하고,

$w_{t+1,i} \leftarrow \frac{w_{t+1,i}}{\sum_{j=1}^{u} w_{t+1,i}}$ 와 같이 정규화 한다.

3. 강한 분류기는 아래와 같다.

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) > 0.5 \\ 0 & otherwise \end{cases}$$

그림 6. Boosted-LDA 알고리즘
Fig. 6. The Boost-LDA algorithm.

Ⅳ. 실 험

1. UCI dataset[13~14]

실험에서 사용한 UCI dataset은 패턴인식과 기계 학습에서 가장 널리 사용되는 실험 데이터이다. UCI (University of Califonia, Irvine) machine learning repository 사이트에서 제공되는 표준 데이터들 중에서 두 개의 클래스를 가진 데이터들로 실험을 수행 하였다. 두 개의 클래스를 가지는 Sonar, Pima, Heart disease, Liver, Breast cancer, Australian 총 6개의 데이터를 이용하여 실험을 하였고, 데이터의 구성은 표 1과 같다. Sonar의 경우 208개의 샘플 데이터의 개수를 가지고 있으며, 각 샘플 데이터마다 60개의 변수를 가지고 있는 데이터의 집합이다.

가. 기존의 특징 추출 알고리즘만을 이용한 실험 결과

표 1에서 두 개의 클래스를 가지는 Sonar, Pima, Heart disease, Liver, Breast cancer, Australian 데이터에 대하여 기존의 특징 추출 알고리즘과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능을 분석하고자 한다.

UCI dataset에서 90%를 학습 샘플 데이터로 사용하고, 나머지 10%를 실험 샘플 데이터로 구성하였다. 총 208개의 샘플 데이터의 수를 가진 Sonar 데이터의 경우 187개를 학습 샘플 데이터로 나머지 21개를 실험 샘플 데이터로 구성하였다. 표 2의 인식률 계산은 총 10번의 실험을 통하여 얻은 인식률 값을 평균과 표준편차로 나타낸 것으로 교차검증(cross validation)의 방법으로 얻은 결과이다. 주성분 분석법은 6개의 최소 변수 개수를 가진 Liver 데이터로 인하여 다른 데이터들 모두 5차원의 고유 벡터를 사용하여 변환행렬을 구성하였다.

표 1. UCI Dataset 구성도
Table 1. Summary of UCI Dataset.

	# of variables	# of class	# of instances
Sonar	60	2	208
Pima	8	2	768
Heart disease	13	2	297
Liver	6	2	345
Breast cancer	9	2	683
Australian	14	2	690

표 2. 테스트 실험 결과
Table 2. The result of experiment.

	PCA+1NN	PCA+SVM	LDA+1NN	LDA+SVM
Sonar	57.14±4.48	52.85±1.50	**78.57±8.17**	77.61±10.05
Pima	64.28±1.10	64.93	66.85±5.05	**75.19±4.03**
Heart disease	53.33±1.05	73.66±13.46	68.00±9.83	**86.67±7.20**
Liver	58.52±2.17	58.82	58.52±6.85	**68.82±5.22**
Breast cancer	76.32±3.13	76.61±4.29	95.58±1.96	**95.73±2.63**
Australian	55.07±3.05	66.23±3.13	75.07±5.37	**82.17±3.20**

표 2를 살펴보면 주성분 분석법과 선형 판별 분석법을 비교하였을 때 선형 판별 분석법의 성능이 더 좋은 결과를 보였다. 이 실험결과로 보아 클래스 정보를 고려하는 것이 분류 성능을 높이는 효과를 가지는 것을 알 수 있다. 그리고 SVM을 이용한 분류기 방법이 최근접 이웃 분류기 방법보다 더 효과적임을 알 수 있다. 특히, Pima 데이터의 경우 75.19%를 보이며 다른 방법들에 비해 약 10% 정도 향상된 인식률을 보인다.

나. 제안하는 Boosted-PCA, Boosted-LDA 이용한 실험 결과

제안하는 Boosted-PCA 알고리즘 또한 5차원의 고유벡터를 사용하여 변환행렬을 구성하였고, 실험 데이터도 동일한 학습 샘플 데이터와 실험 샘플 데이터를 사용하였다. 인식률 계산도 동일하게 총 10번의 교차검증으로 결과 값을 계산하였다. Boosted-Random, Boosted-PCA 알고리즘과 Boosted-LDA 알고리즘은 총 30 단계를 거쳐 강한 분류기를 만들었다.

Boosted-Random알고리즘은 Gaussian Random함수를 이용하여 생성한 1000개의 벡터를 Adaboost알고리즘의 약한 분류기로 사용하여, Boosted-PCA와 Boosted-LDA의 반복 과정과 동일하게 강한 분류기를 구성하는 알고리즘이다.

표 3. 테스트 실험 결과
Table 3. The result of experiment.

	Boosted-Random	Boosted-PCA	Boosted-LDA
Sonar	**80.47±7.59**	72.38±5.40	80.00±9.73
Pima	69.22±4.74	69.22±4.58	**75.45±2.96**
Heart disease	75.00±8.49	74.33±13.05	**85.00±5.27**
Liver	69.41±6.07	61.76±5.54	**69.70±3.93**
Breast cancer	96.91±1.08	**97.35±1.35**	96.17±1.86
Australian	69.85±3.66	69.56±5.11	**82.75±3.37**

다. 기존의 특징 추출 알고리즘만을 이용한 실험 결과와 제안하는 Boosted-PCA, Boosted-LDA 이용한 실험 결과의 비교

그림 7의 테스트 결과 비교를 살펴보면 기존의 특징 추출 알고리즘과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 제안한 Boosted-PCA와 Boosted-LDA 알고리즘의 성능이 우수하다는 것을 알 수 있다.

주성분 분석법과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 제안한 Boosted-PCA 알고리즘의 성능이 우수하다. 특히, Breast cancer데이터의 경우 Boosted-PCA방법이 97.35%로 가장 높은 인식률을 보인다. 또한, 선형 판별 분석법과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 제안한 Boosted-LDA 알고리즘을 이용하면 약 1%이내 인식률 향상 효과를 얻을 수 있다.

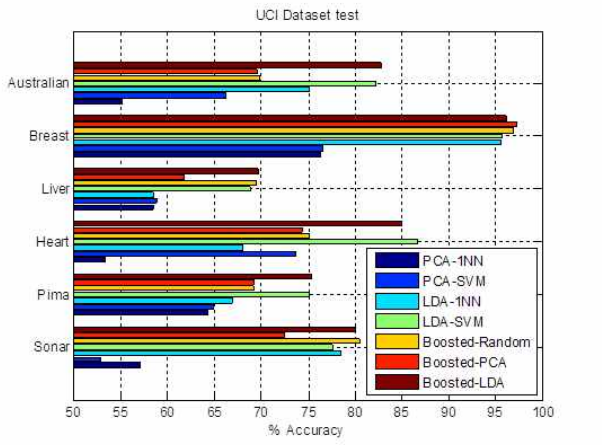

그림 7. Boosted-LDA 알고리즘
Fig. 7. The Boost-LDA algorithm.

2. FRGC dataset[15~16]

FRGC dataset는 사람 얼굴 데이터로서 남자와 여자를 구분하기 위하여 사용한 실험 데이터이다. FRGC

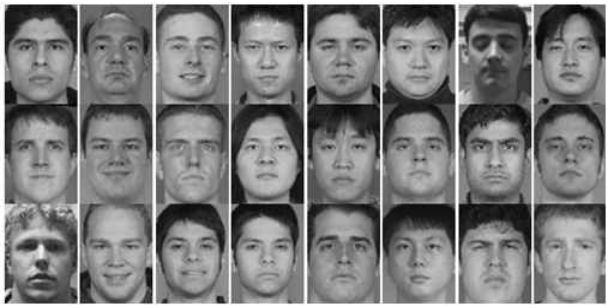

그림 8. 남자 얼굴 영상의 일부 예
Fig. 8. Example of Man images.

그림 9. 여자 얼굴 영상의 일부 예
Fig. 9. Example of Woman images.

data는 총 30,200장으로써 남자 데이터 17,238장, 여자 데이터 13,962장으로 구성 되어 있다. 모든 데이터는 픽셀 단위 크기의 흑백영상이다. 그림 8은 FRGC dataset 중 남자 얼굴 영상, 그림 9는 여자 얼굴 영상의 예이다.

학습 샘플 데이터는 남자 1000장, 여자 1000장으로 총 2000장으로 구성되어 있고, 실험 샘플 데이터는 남자 2000장, 여자 2000장으로 총 4000으로 구성되어 있다.

가. 기존의 특징 추출 알고리즘만을 이용한 실험 결과

총 30,200장인 FRGC dataset에서 랜덤으로 2000장의 학습 샘플 데이터를 구성하였고, 4000장을 실험 샘플 데이터로 구성하였다. 표 4의 결과는 UCI dataset과 마찬가지로 계산은 총 10번의 실험으로 얻은 인식률 값을 평균으로 나타낸 것으로 교차검증의 방법으로 얻었다.

그리고 각 특징 추출 알고리즘마다 사용한 변환 행렬의 차원수는 주성분 분석법의 Eigenface 100차원, 선형 판별 분석법의 Fisherface는 주성분 분석법을 이용해 특징 벡터를 100차원으로 낮추고, 다시 선형 판별 분석법을 이용하여 변환 행렬을 구성하였다. 남자와 여자 분류문제에서 주성분 분석법과 선형 판별 분석법을 비교하였을 때 UCI dataset과 마찬가지로 선형 판별 분석법의 성능이 더 좋은 결과를 보였다. 또한, SVM 분류기가 최근접 이웃 분류기보다 더 나은 성능을 보였다.

표 4. 테스트 실험 결과
Table 4. The result of experiment.

	PCA+1NN	PCA+SVM	LDA+1NN	LDA+SVM
FRGC	57.74±7.13	81.37±3.40	87.33±1.45	**92.29±3.28**

나. 제안하는 Boosted-PCA, Boosted-LDA 이용한 실험 결과

제안하는 Boosted-PCA 알고리즘 또한 100차원의 고유 벡터를 사용하여 변환행렬을 구성하였고, Boosted-LDA 알고리즘도 선형 판별 분석법과 동일하게 주성분 분석법으로 특징 벡터를 100차원으로 낮추고, 다시 선형 판별 분석법을 이용하여 변환 행렬을 구성하였다.

Boosted-Random, Boosted-PCA 알고리즘과 Boosted-LDA 알고리즘은 총 30 단계를 거쳐 강한 분류기를 만들었다. 실험 데이터 또한 동일한 학습 샘플 데이터와 실험 샘플 데이터를 사용하였으며, 인식률 계산도 총 10번의 교차검증으로 결과 값을 계산하였다.

이 실험에서도 Boosted-LDA 알고리즘이 92.67% 성능을 보임으로써 LDA와 SVM분류기를 사용했을 때 보다 0.38%의 인식률 향상 효과를 얻을 수 있다.

표 5. 테스트 실험 결과
Table 5. The result of experiment.

	Boosted-Random	Boosted-PCA	Boosted-LDA
FRGC	82.21±0.65	73.33±0.89	**92.67±0.45**

다. 기존의 특징 추출 알고리즘만을 이용한 실험 결과와 제안하는 Boosted-PCA, Boosted-LDA 이용한 실험 결과의 비교

실험 결과는 그림 10과 같다. FRGC dataset에 대해서도 Boosted-LDA알고리즘의 성능이 가장 우수하다. 선형 판별 분석법과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 제안한 Boosted-LDA 알고리즘의 성능이 우수하다. 본 논문에서 제안한 Boosted-PCA 알고리즘의 경우는 주성분 분석법과 최근접 이웃 분류기를 이용한 방법의 성능보다는 높지만, 주성분 분석법과 SVM을 이용한 분류기 방법의 성능보

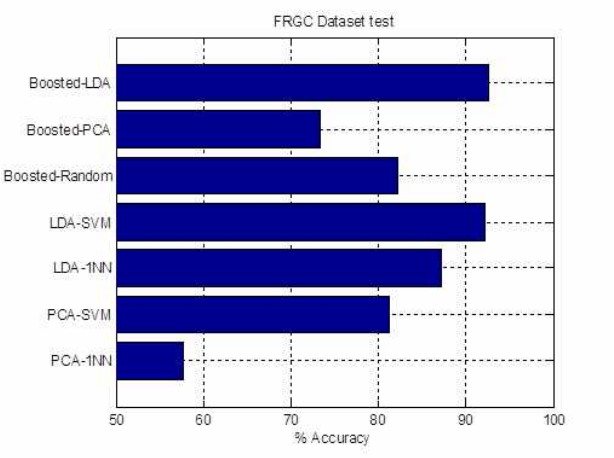

그림 10. 테스트 결과 비교
Fig. 10. Comparison results.

다 낮음을 볼 수 있다. 약한 분류기 계산과정인 식 (10)에서 다른 약한 분류기로 교체하는 방법 등에 대해 향후 더 연구해야 할 것이다.

V. 결 론

본 논문에서는 기존의 특징 추출 알고리즘과 Adaboost 알고리즘의 결합을 통해 두 개의 클래스를 효율적으로 분류하는 이진분류기를 제안하였다. 본 논문에서 제안한 Boosted-PCA 알고리즘은 데이터의 가장 중요한 축들을 찾아 효율적으로 차원을 줄일 수 있는 주성분 분석법으로 얻은 변환 행렬을 Adaboost의 약한 분류기로 사용하였고, 약한 분류기의 결합을 통하여 강한 분류기를 구성하여 두 개의 클래스를 분류하였다. Boosted-LDA 알고리즘은 클래스 분리를 최대화하는 주축으로 사영시켜 특징 벡터의 차원을 감소하는 방법인 선형 판별 분석법으로 얻은 변환 행렬을 Adaboost의 약한 분류기로 사용하였다. 기존의 특징 추출 알고리즘의 분류와 달리 단계를 거치면서 분류에 효과적인 특징 값들을 추출하고, 추출한 특징 값을 약한 분류기로 사용함으로써 인식률의 향상을 보였다.

UCI dataset에 대한 실험결과를 보면, 주성분 분석법과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 Boosted-PCA 알고리즘의 분류율이 평균 5% 정도 향상되었으며, 선형 판별 분석법과 최근접 이웃 분류기, SVM을 이용한 분류기 방법의 성능보다 근소하지만 Boosted-LDA 알고리즘의 분류율이 향상됨을 확인할 수 있다. FRGC dataset에 대한 결과를 보면, Boosted-LDA알고리즘의 성능이 가장 우수하지만, Boosted-PCA 알고리즘의 경우는 기존의 주성분 분석법과 SVM을 이용한 분류보다 분류율이 낮았다. 이를 통해 본 논문에서 제안한 Boosted-PCA, Boosted-LDA 알고리즘은 단계를 거치면서 분류에 효과적인 특징 값들을 추출하고, 추출한 특징 값을 약한 분류기로 사용함으로써 인식률의 향상을 얻을 수 있다는 것을 실험적으로 입증하였다.

향후 과제로는 데이터에 따라 분류율이 낮은 부분에 대해서 보완할 여지가 있으며, 두 개의 클래스가 아닌 다양한 클래스 데이터의 분류 문제에 대해서도 확장 가능하도록 해야 된다. 또한, PCA와 LDA 뿐만 아니라 ICA나 BDA 같은 다른 특징 추출 알고리즘과 Adaboost의 결합을 통하여 인식률 향상을 기대할 수 있다.

참 고 문 헌

[1] Christopher J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, DataMining and Knowledge Discovery ,Vol. 2 , pp. 121-167, 1998.
[2] Cover T. M. and Hart P. E, “Nearest Neighbor Pattern Classification”, IEEE Transactions on Information Theory, Vol .IT-13, no. 1, pp. 21-27, 1967.
[3] Simon Haykin, “Neural networks”, 2nd Edition, PrenticeHall, 1999.
[4] Y. Freund, R. E. Schapire, “A Short Introduction to Boosting”, Journal of Japanese Society for Artificial Intelligence, Vol. 14, no. 5, pp. 771-780, 1999.
[5] Yoav Freund, Robert E. Schapire, “A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting,” In European Conference on Computational Learning Theory, pp. 23-37, 1995.
[6] Robert. E. Schapire and Yoram Singer, “Improved boosting algorithms using confidence-rated predictions,” Machine Learning, Vol. 37, no. 3, pp. 297-336, 1999.
[7] P. viola and M. J. Jones, “Robust Real-time Face Detection”, International Journal of Computer Vision, Vol. 57, No. 2, pp. 137-154, 2004.

[8] Matthew Turk and Alex Pentland, "Eigenface for Recognition," Journal of Cognitive Neuroscience Vol. 3, no. 1, pp.70-86, 1991.
[9] I.T.Joliffe, "Principal Component Analysis," Springer-Verlag, 1986.
[10] Friedman, J. H. "Regularized Discriminant Analysis," Journal of the American Statistical Association (American Statistical Association) 84(405), pp. 165 - 175, 1989.
[11] Martinez, A. M.; Kak, A. C. "PCA versus LDA," IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 23, no. 2, pp. 228 - 233, 2001.
[12] 함승록, 곽노준, "Boosted-PCA를 이용한 이진분류기", 신호처리합동학술대회 논문집, Vol. 24, no. 1, pp. 195-197, 2011
[13] UCI Data Sets, Availabe: http://archive.ics.uci.edu/ml/datasets.html.
[14] P. M. Murphy and D. W. Aha, "UCI repository of machine learning databases," 1994, For more information contact or http://www.cs.toronto.edu/_delve/.
[15] P. Jonathon Phillips, Harry Weschsler, Jeffery Huang, and Patrick J.Rauss, "The FERET database and evaluation procedure for face-recognition algorithms", Imageand Vision Computing, Vol. 16, no. 5, pp. 295-306, 1998.
[16] P. J. Phillips et. al., "Overview of the Face Recognition Grand Challenge," IEEE Conference on Computer Vision and Pattern Recognition 2005.

저 자 소 개

함 승 록(학생회원)
2010년 아주대학교 전자공학부 학사 졸업.
2012년 2월 아주대학교 전자공학부 석사 졸업.
2012년 4월 ~ 현재 LG전자
<주관심분야 : 컴퓨터 비전, 영상 처리>

곽 노 준(정회원)
1997년 서울대학교 전기공학부 졸업.
1999년 서울대학교 전기 컴퓨터 공학부 석사 졸업.
2003년 서울대학교 전기 컴퓨터 공학부 박사 졸업.
2003년 3월 ~ 2006년 8월 삼성전자 책임연구원.
2006년 9월 ~ 2007년 2월 서울대학교 전기컴퓨터 공학부 BK조교수.
2007년 3월 ~ 2011년 2월 아주대학교 전자공학부 조교수.
2011년 3월 ~ 현재 아주대학교 전자공학부 부교수.
<주관심분야 : 패턴인식, 컴퓨터 비전, 영상 처리>