

Factorial Code 표현법을 이용한 얼굴 인식

이오영^o 최승진

충북대학교 전기공학부

{loy250, schoi}@engine.chungbuk.ac.kr

Face Recognition Using Factorial Code Representation

Oyoung LEE^o Seungjin CHOI

Dept. of Electrical Engineering, Chungbuk National University

요 약

얼굴인식에서 정보 이론적 접근방법은 얼굴 영상을 작은 기저 영상의 집합으로 분해하는 것을 기초로 한다. 가장 많이 쓰이고 있는 방법은 PCA를 기반으로 하는 eigenface 방법이다. PCA를 기반으로 하는 방법은 데이터의 2차 통계적 구조만을 고려하므로 화소 사이의 고차 통계적 의존성은 고려되지 않는다. Factorial code 표현법은 효과적인 정보 표현의 좋은 방법으로 알려져 있고 이것은 ICA와 밀접한 관련이 있다. Factorial code 표현법은 eigenface 방법과 비교할 때 중요한 정보가 포함되어 있는 데이터의 고차 통계적 구조도 고려되어 더욱 효과적인 정보 표현을 기대할 수 있다. 이 논문에서는 PCA를 이용하여 차원을 줄이고 찾아낸 특징 공간에 Factorial code 표현법을 적용했다. 그리고 얼굴 인식에 있어서 Factorial code 표현법이 eigenface 방법보다 성능이 우수함을 보였다. 제안한 방법의 우수한 성능을 모의실험을 통하여 입증했다.

1. 서론

얼굴인식은 사람마다 다른 얼굴 영상들의 데이터 베이스를 가지고 입력되는 얼굴 영상을 비교하여 누구인지 식별해내는 것이다. 얼굴인식에서 가장 널리 사용되었던 방법은 eigenfaces [7], Fisherfaces [3] 등 PCA를 기반으로 하는 방법이다.

PCA는 데이터의 분산이 최대화 되는 직교 방향을 찾아내는 것이다. 이 방법은 데이터의 2차 구조만을 고려하기 때문에 화소 사이의 고차 통계적 의존성은 고려되지 않았다.

Factorial code 표현법은 효과적인 정보 표현의 하나의 주요한 방법이고 이것은 독립 요소 분석(ICA)과 관계가 깊다. Factorial code 표현법은 데이터의 확률 분포가 가장 비슷한 선형 데이터 표현을 찾는 것이 목적이다. 그러므로 고차의 통계적 구조도 고려된다.

이 논문에서는 기존의 PCA를 기반으로 하는 eigenfaces 방법을 이용하여 얻어진 특징 공간에 Factorial code 표현법을 적용하여 인식 성능을 비교하고, 제안한 방법의 성능이 우수함을 모의실험을 통하여 증명 할 것이다.

2. Eigenfaces

패턴 인식에서 샘플 영상의 화소값을 고차원 공간(즉, 영상 공간)에서의 좌표로 다룬다. N 샘플 얼굴 영상을 고려하면, $\{x_1, \dots, x_N\}$, 각 영상은 m 차의 영상 공간에 속한다. 영상 공간은 매우 고차의 공간

이므로 차원을 줄여야 할 필요가 있고, PCA는 차원을 줄이는데 효과적인 방법으로 널리 알려져 있다. m 차의 영상 공간에서 n 차의 특징 공간으로의 선형 변환을 생각해 보자($n < m$). 특징 벡터 $\{z_i\}$ 는 아래 식처럼 정의된다.

$$z_i = Ux_i, \quad (1)$$

여기서 $U \in \mathbb{R}^{m \times n}$ 는 행이 정규직교하는 선형 변환 행렬이다. PCA에서 행렬 U 는 복원 오차가 최소화되게 선택되고, 열 벡터는 샘플 공분산 행렬 $\{x_i\}$ 의 고유 벡터이다.

평균 영상 μ 는 아래와 같이 정의된다.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i. \quad (2)$$

그리고, 샘플 공분산 행렬 C_x 는

$$C_x = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \\ = \frac{1}{N} \Phi \Phi^T, \quad (3)$$

여기서 Φ 는

$$\Phi = [x_1 - \mu, \dots, x_N - \mu]. \quad (4)$$

행렬 U 의 열 벡터들을 eigenfaces라 부르고, 이것은 저 차원 표현을 위한 기저 영상으로 사용된다.

Snap-shot 방법

위의 방법은 샘플 영상의 크기가 클 경우 공분산 행렬의 계산이 불가능하다. 즉, 영상 공간의 차원이 샘플 영상의 수보다 클 경우, ($N < m$), C_x 의 m 차 고유벡터는 $\Phi^T \Phi$ 행렬의 고유벡터를 계산하여 얻을 수 있다.

$$\Phi^T \Phi e_i = \lambda_i e_i. \quad (5)$$

양변에 Φ 를 곱하면

$$\Phi \Phi^T \Phi e_i = \lambda_i \Phi e_i. \quad (6)$$

공분산 행렬을 구할 수 있다. 이 방법으로 계산량을 줄일 수 있다. 그러나 이 방법은 얻어진 고유 벡터들이 서로 직교하지 않으므로 직교화 방법이 필요하다.

Gram-Schmidt 직교화

선형독립인 C_i 의 열 벡터, $\{s_1, \dots, s_N\}$, 정규직교 벡터, $\{v_1, \dots, v_N\}$,라 하면

$$y_i = s_i - \sum_{j=1}^{i-1} (s_i^T v_j) v_j. \quad (7)$$

여기서 $v_i = y_i / \|y_i\|$ 이다.

3. Factorial code 표현

PCA로 구한 특징 벡터 $\{z_i\}$ 의 Factorial code 표현을 찾아 보자. n 차원 벡터 z 의 수학적 표현은,

$$z = As, \quad (8)$$

여기서 $A \in \mathbb{R}^{n \times n}$ 의 열 벡터는 기저 벡터이고, $s \in \mathbb{R}^n$ 의 요소는 기저 계수라 부른다.

Factorial code 표현법은 관찰되어진 데이터의 확률 분포와 (8)식 모델의 확률 분포가 가장 잘 조화를 이루는 A 와 s 를 찾는 것이 목적이다. 기저 계수는 통계적 독립이라 가정한다. 즉,

$$p(s) = \prod_{i=1}^n p_i(s_i). \quad (9)$$

관찰되어진 확률 밀도를 $p^o(z)$, 모델의 확률 밀도를 $p(z)$ 라 표시하자. 가장 잘 조화를 이루는 $p^o(z)$ 와 $p(z)$, 즉 행렬 A 를 찾기 위하여 이 두개 사이의 Kullback-Leibler divergence를 생각하여 보자. 일함수 R 은 아래와 같이 주어진다.

$$R = \int p^o(z) \log \frac{p^o(z)}{p(z)} dz. \quad (10)$$

(8)식에서 기저 계수 벡터 s 에서 특징 벡터 z 로의 변환은 선형이기 때문에 변환의 Jacobian은 간단히 행렬식의 절대값이 되므로 아래 식을 얻을 수 있다.

$$\log p(z) = -\log |\det A| + \sum_{i=1}^n \log p_i(s_i). \quad (11)$$

손실함수 L 은 아래 식과 같고, $\log p^o(z)$ 는 A 와 무관하므로 무시를 하였다.

$$L = \log |\det A| - \sum_{i=1}^n \log p_i(s_i), \quad (12)$$

기울기 감소 (gradient descent)법을 적용한 학습 알고리즘을 살펴보면,

$$\begin{aligned} \Delta A &= -\eta_i \frac{\partial L}{\partial A} \\ &= -\eta_i A^{-1} \{I - \varphi(s) s^T\}, \end{aligned} \quad (13)$$

위의 식에서 $\eta_i > 0$ 는 학습율이고, $\varphi(s)$ 는 i 번째 요소가 $\varphi_i(s_i)$ 인 비선형 함수로 아래 식처럼 주어진다.

$$\varphi_i(s_i) = -\frac{d \log p_i(s_i)}{ds_i}. \quad (14)$$

여기에 자연 기울기 (natural gradient) 방법을 적용하여 보면 (13)식은 다음 식처럼 바뀌게 된다. [1]

$$\begin{aligned} \Delta A &= -\eta_i A A^T \frac{\partial L}{\partial A} \\ &= -\eta_i A \{I - \varphi(s) s^T\}. \end{aligned} \quad (15)$$

학습이 한번 끝나면 구해진 A 를 이용하여 $s = A^{-1}z$ 를 계산하여 기저 계수 벡터 s 를 계산하고, 다시 (15)식을 이용하여 A 를 구하는 반복 계산을 A 가 수렴할 때 까지 반복을 한다.

A 를 학습시키는 대신에 A^{-1} 를 학습시키는 것도 가능하다. A^{-1} 는 ICA 필터와 같다. [4,6] $W = A^{-1}$ 로 정의하면, 자연 기울기 방법을 적용한 W 에 대한 학습 알고리즘은 아래 식과 같고, 이것은 잘 알려진 ICA 알고리즘이다.

$$\Delta W = \eta_i \{I - \varphi(s) s^T\} W. \quad (16)$$

4. 모의실험 결과

모의실험에서 AR 얼굴영상 데이터베이스 [5]를 사용하였다. 이 데이터베이스 중에서 40명의 표정이 다르고 조명조건이 다른 얼굴 정면 영상 10개씩을 사용하였고, 또한 10개의 영상은 5개씩 2주의 시간 차이를 두고 촬영한 영상을 사용하였다.

모두 400개의 얼굴 영상 중에 200장은 학습영상으로 사용하였고 나머지 200장은 테스트 영상으로 사용하였다. 한 개의 얼굴영상의 크기는 46×50 이고, 그림 1과 그림 2는 학습영상과 테스트 영상의 샘플이다.



그림 1. 학습 영상의 샘플.



그림 2. 테스트 영상의 샘플.

모의 실험은 3가지 방법으로 실행했다. 이 방법들은 PCA를 적용하여 차원을 축소하는 방법들을 달리 한 것이다.

방법 1. 일반적인 eigenface 방법으로 얼굴영상의 크기가 작은 경우 전체 공분산 행렬을 계산하는 방법.

방법 2. Snap-shot 방법을 사용한 방법으로 얼굴영상의 크기가 큰 경우 전체 공분산 행렬을 계산하는 것이 불가능 할 때나 계산량을 줄이고자 할 때 사용하는 방법.

방법 3. Snap-shot 방법을 사용하여 얻어진 고유 벡터를 직교화 과정을 수행하여 고유 벡터가 서로 정규직교하게 만드는 방법.

3가지 방법 모두 기여도가 큰 순서로 20개의 고유 벡터와 50개의 고유벡터를 선택하여 얼굴 인식에 사용하였고, 이것에 모두 Factorial code 표현법을 적용하여 얼굴 인식의 성능을 평가하였다.

얼굴 인식의 방법은 학습 영상에서 얻어진 특징 벡터와 테스트 영상에서 얻은 특징 벡터사이의 유클리디안 거리를 측정하여 가장 가까운 거리를 가지는 영상을 선택하는 방법으로 인식의 성능을 평가 하였고, 그 결과를 그림 5.에 나타내었다.

5. 결론

이 논문에서는 Factorial code 표현법을 적용하여 가져 영상을 얻었고, PCA를 사용한 eigenface 방법보다 Factorial code 표현법을 사용한 방법의 성능이 우수함을 보였다. 이는 데이터의 2차 통계적 구조만을 고려하는 PCA와는 대조적으로 Factorial code 표현법은 데이터의 더 많은 중요한 정보를 포함하는 고차 통계적 구조까지 고려하기 때문이다. 그리고 이 논문에서는 Factorial code 표현법과 ICA를 기반으로 하는 비선형 정보 최대화 [2]와 같은 것임을 보였다. 또, 자연 기율기 방법을 Factorial code 표현법에 적용해 보았다. 그리고 이것들을 컴퓨터 모의 실험을 통하여 검증하였다.

6. 참고 문헌

- [1] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251-276, Feb. 1998.
- [2] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski. Independent component representations for face recognition. In *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III*, pages 528-539, San Jose, California, Jan. 1998.
- [3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):711-720, Oct. 1997.
- [4] A. Bell and T. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129-1159, 1995.
- [5] A. Martinez and R. Benavente. The AR face database. Technical Report CVC #24, Computer Vision Center, Purdue University, June 1998.
- [6] B. A. Olshausen and D. J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607-609, 1996.
- [7] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71-86, 1991.

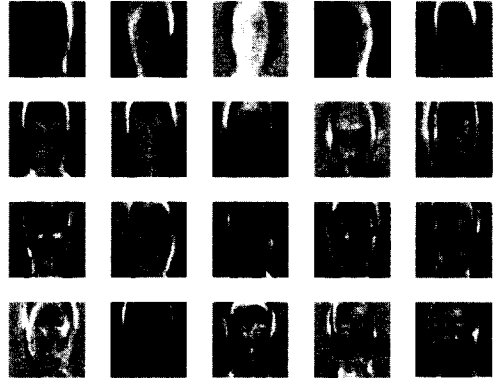


그림 3. 기여도가 큰 20개의 eigenvectors.

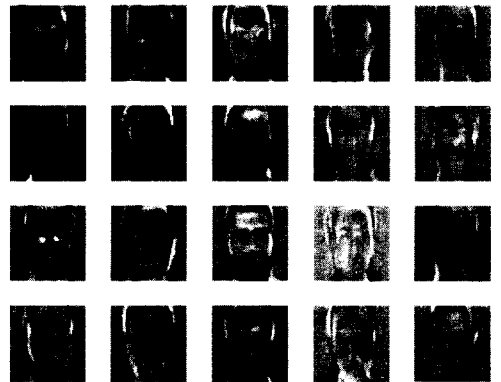


그림 4. 20개의 Factorial code 표현의 기저 벡터.

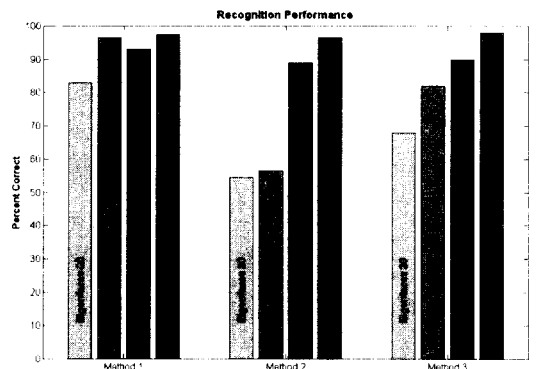


그림 5. 인식 성능 비교.