

C. eigenface와 영상인식 응용

PCA가 영상인식에 활용되는 대표적인 예는 **얼굴인식**(face recognition)이다. 그리고 이와 관련된 개념 혹은 용어로서 **eigenface**(아이겐페이스)라는게 있다.

다음과 같은 20개의 45x40 얼굴 이미지들이 있다고 하자.



<그림 4> 45x40 얼굴 이미지 20장

이미지에서 픽셀 밝기값을 일렬로 연결하여 벡터로 만들면 이들 각각의 얼굴 이미지는 $45 \times 40 = 1,800$ 차원의 벡터로 생각할 수 있다 (즉, 각각의 이미지는 1,800 차원 공간에서 한 점(좌표)에 대응).

↑
X 이미지를 computer science 측면에서
분석할 때는 해당 이미지를 벡터
(픽셀 값을 원소로 차지는 벡터)로 변환한다.



이제 이 20개의 1,800차원 점 데이터들을 가지고 PCA를 수행하면 데이터의 차원 수와 동일한 개수의 주성분 벡터들을 얻을 수 있다. 이렇게 얻어진 주성분 벡터들을 다시 이미지로 해석한 것이 **eigenface**이다 (얼굴 이미지를 가지고 얻은 벡터이기에 eigenface라 부른다). 실제 위 이미지에 대해 얻어진 1,800개의 eigenface들 중 분산이 큰 순서대로 처음 20개를 나열하면 아래 그림과 같다.



<그림 5> 처음 20개의 eigenface들

위 그림에서 볼 수 있듯이 ^① 앞부분 eigenface들은 데이터들에 공통된 요소(얼굴의 전반적인 형태)를 나타내고 ^② 뒤로 갈수록 세부적인 차이 정보를 나타낸다. ^③ 그리고 더 뒤로 가면 거의 노이즈(noise)성 정보를 나타낸다.

앞서 PCA를 통해 얻어진 주성분 벡터들은 서로 수직인 관계에 있다고 말한 바 있다. 이 말은 주성분 벡터들이 n 차원 공간을 생성하는 기저(basis) 역할을 할 수 있음을 의미한다. 즉, PCA로 얻은 주성분 벡터들을 e_1, e_2, \dots, e_n 라하면 임의의 n 차원 데이터 x 는 $x = c_1e_1 + c_2e_2 + \dots + c_n e_n$ 과 같이 e_i 들의 일차결합으로 표현될 수 있다 (이 때, 상수계수 c_i 들은 x 와 e_i 의 내적 즉, $c_i = x \cdot e_i$ 로 계산할 수 있으며 이와 같이 어떤 데이터 집합의 데이터들을 그들의 주성분 벡터들의 일차결합으로 표현하는 것을 Karhunen-Loève transform (KLT) 또는 Hotelling transform이라 부른다).

그런데, 뒷부분의 주성분 벡터들은 데이터 분포에 포함된 노이즈(noise)성 정보를 나타내기 때문에 뒷부분은 버리고 전반부 k개의 주성분 벡터들만을 가지고 원래 데이터를 표현하면 노이즈가 제거된 데이터를 얻을 수 있다. 즉, 원래의 x 가 $x = c_1e_1 + c_2e_2 + \dots + c_nen$ 일 때 $x_k = c_1e_1 + \dots + c_k e_k$ 로 x 를 근사하는 것이다. 위 얼굴 이미지들에 대해 전반부의 일부($k = 20, 10, 5, 2$) eigenface들만을 이용한 근사 이미지들은 아래 그림과 같다 (클릭시 확대 이미지).



<그림 6> k개의 eigenface만을 이용한 데이터 복원(reconstruction)

그림에서 볼 수 있듯이 많은 수의 eigenface를 이용하면 원본 얼굴과 거의 유사한 근사(복원) 결과를 볼 수 있지만 k가 작아질수록 개인 고유의 얼굴 특성은 사라지고 공통의 얼굴 특성이 남게 된다 ($k=20$ 인 경우 원래 얼굴이 그대로 살아나지만 $k=2$ 인 경우 개인 간의 구분이 거의 사라짐을 볼 수 있다).

☞ k개의 주성분 벡터만을 이용하여 원래 데이터를 표현하는 것은 통상적으로 근사라는 용어보다는 복원(reconstruction)이라는 용어가 주로 사용된다.

☞ 노이즈(noise)에 대해 좀더 생각해 보면, 앞서 말했듯이 PCA는 개별 데이터에 대한 분석이 아니라 전체 데이터에 대한 집합적 분석 도구이다. 만일 강아지 100마리에 대한 PCA 분석 결과와 고양이 100마리에 대한 PCA 분석 결과가 있다고 하자. 이 때, 강아지 데이터에서 얻어진 eigenface들 중 앞의 것들은 (고양이와 구분되는) 강아지 고유의 형태 정보를 나타내고 뒤로 갈수록 강아지들 내부에서 강아지들 사이의 차이점을 표현할 수 있는 정보를 나타낸다. 그리고 더 뒤로 나가면 노이즈성 정보를 표현한다. 마찬가지로 고양이 데이터에 대한 eigenface들은 주요한 성분일수록 고양이 공통의 성분, 뒤로 갈수록 고양이 개체 사이의 차이를 가르는 요소를 나타낸다. 그런데, 어디서 어디까지가 데이터 공통 성분이고 어디까지가 데이터의 차이인지, 그리고 어디부터 노이즈 성분인지 그 구분은 명확하지 않다. 그 경계를 이론적으로 계산하는 방법론 등도 있긴 하지만 대부분은 응용에 따라서, 그리고 데이터에 따라서 주관적으로 또는 실험적으로 결정하는 것이 통상적이다.

Eigenface (PCA응용)

PCA를 하여 나온 eigenvector를 이용하여 얼굴을 인식하는 방법인데, 현재 많이 쓰이는 방법은 아니다.

아래와 같은 순서로 진행한다.

1. 사람 얼굴을 데이터로 모은다. 이 때, 얼굴 데이터는 같은 구조, 같은 밝기이며, 눈과 입이 비슷한 위치에 있어야 한다.

2. 사진들을 벡터화한다. 흑백의 100x100이미지라면 사진당 10000차원 벡터 하나가 나온다.

3. 각 사진을 점으로 하여 PCA를 진행한다. 이 과정에서 k 개의 Eigenvector를 선택할 것이다. 이들 Eigenvector는 사진과 같은 차원을 가질테니 시각화도 가능하다.

4. 새로운 사진이 들어오면 이걸 Eigenvector들로 사영(projection)한다.

만약 새로운 사진이 얼굴 사진이 아니라면 분산이 제대로 보존이 안 될 것이다. 왜냐하면, 우리가 얻은 Eigenvector는 얼굴사진의 분산만을 보존하는 것이기 때문이다. 또한, 얼굴 사진이 아니라면 reconstruction error 또한 클 것이다. 이 성질을 이용하여 얼굴 사진인지 아닌지 판별해내는 것이 eigenface이다.

↑ 보통 적은 개수의 eigenvector를 선택해서, 이를 선형결합한다. 선형결합의 결과는 한의 보정된 얼굴 모습을 나타내는 벡터이다.