점진적인 주성분분석기법을 이용한 고차원 자료의 특징 추출

김병주*

Feature Extraction on High Dimensional Data Using Incremental PCA

Byung-Joo Kim*

요 약

고차원 자료를 효율적으로 처리하기 위해서는 특징 추출 기법이 필요하다. 주성분분석 방법은 대표적인 특징추출 방법 이지만 학습 자료의 차원이 큰 경우에는 고유공간을 계산하기 위해 많은 기억공간과 계산량을 필요로 한다. 본 논문에서는 고차원 자료의 특징 추출을 위해 점진적인 주성분분석 방법을 사용한다. 제안한 방법에 대해 신경망에서 점진적인 주성분분석을 하는 대표적인 방법인 APEX 모델과 실험을 통해 비교해 본 결과 제안된 방법이 APEX 모델 보다 성능이 우수함을 나타내었다.

ABSTRACT

High dimensional data requires efficient feature extraction techniques. Though PCA(Principal Component Analysis) is a famous feature extraction method it requires huge memory space and computational cost is high. In this paper we use incremental PCA for feature extraction on high dimensional data. Through experiment we show that proposed method is superior to APEX model.

키워드

점진적 주성분분석, 고유값, 고유벡터, 고유공간

I. 서 론

고차원 데이터를 처리하는 문제가 점점 증가되고 있다. 하지만 대부분의 고차원 데이터는 데이터의 분포 형태를 알기가 어렵다. 또한 고차원 데이터는 이를 처리하기 위해 많은 양의 메모리와 계산량을 필요로 한다. 따라서 고차원 데이터를 처리하는 경우에는 자료를 저차원으로 축소하는 과정이 필요한데 이때 원래 자료의 성질을 잃어버리지 않으면서 자료의 성격을 잘 반영할 수 있는 특징을 추출하는 것이 필요하다. 주성분분석(Principal Component Analysis:PCA) 방법은 학습 자료의 특징추출을 위해 사용하는 대표적인 기법중의 하나이다[1]. 하지만 PCA 방법은 다음과 같은 단점이 있다. 첫째 PCA는 일괄처리(batch) 방식으로 동작한다. 이는 새로운 학습 자료가 추가 되면 고유공간(eigenspace)을 다시 계산 하여야 하는 단점이 있다. 또한 고차원 데이터를 처리하기 위해서는 많은 양의 메모리를 필요로 한다. 이러한 문제점을 해결하기 위한 기존의 연구를 살펴보면 크게 두가지로 나눌수 있다. 첫번째는 일괄처리 방식의 문제점을 해결하기 위해 학습 자료를 순차적으로 받아들이며 이전의 고유공간과 새로이 추가된 학습 자료에 의해 새로운 고유공간을 계산하는 방식이다. 이러한 점진적인 방식은 다시 크게 두 가지로 나눌 수 있다. 하나는 새로운 고유공간을 계산하는데 있

*영산대학교 네트워크정보공학부

접수일자 : 2004. 05. 21

어 평균의 갱신을 허용하지 않는 방법이며[2][3] 또 다른 접근 방법은 Hall[4]에 의해 제안된 방법으로 평균의 갱신을 허용하는 방법이다. Hall은 그의 논문에서 평균을 갱신하는 방법이 고정하는 방법에 비해 분류문제에서 우수한 성능을 나타냄을 실험을 통해 보였다. 하지만 기존의 방법에서는 학습시 고유공간의 차원 유지에 관한 명확한 규칙이 제안되지 않았다. 새로운 학습 자료를 표현하기 위해 고유공간의 차원을 증가시키면 학습 자료는 잘 표현할 수 있으나 차원의 증가로 인해 더 많은 기억공간을 필요로 한다. 반면에 고유공간의 차원을 일정 크기로 고정하면 이로 인해 학습자료에 대해 일정량의 정보 손실을 감수해야 한다. 특징 추출을 위한 두 번째 접근방법으로 신경망의 헤비안 학습규칙을 기반으로 한 APEX 모델이 있으며 이는 Kung과 Diamantaras에 의해 제안되었다[5]. 본 논문에서는 Hall이 제안한 방법에서 고유공간의 차원 유지에 관한 규칙을 추가한 수정된 점진적인 주성분분석 방법을 제안한다. 제안한 방법에 대해 신경망에서 주성분분석을 하는 대표적인 방법인 APEX 모델과 실험을 통해 비교해본 결과 제안된 방법이 APEX 모델 방법보다 성능이 우수함을 나타내었다. 논문의 구성은 다음과 같다. 2장에서는 점진적인 PCA(Incremental PCA:IPCA) 기법 및 고유공간의 차원 유지를 위한 방법을 제안한다. 3장에서는 신경망 기반의 대표적인 특징 추출 기법인 APEX에 대하여 설명한다. 4장에서는 두 방법의 성능을 검정하기 위해 실험에 의해 나타난 결과를 보인다. 마지막 5장에서는 결론 및 향후 연구에 대해서 이야기 한다.

II. 점진적 PCA

Hall에 의해 제안된 IPCA 방법을 간략히 설명하기 전에 먼저 수식에서 사용되는 벡터 및 행렬은 다음과 같이 정의한다. 벡터는 열벡터(column vector)를 의미하며 소문자로 나타내고 행렬은 대문자로 표시한다. 그리고 행렬의 크기는 아랫첨자로 나타낸다. 예를 들어 A_{mn} 는 $m \times n$ 크기의 행렬을 의미한다. 행렬에서 열의 확장(column extension)은 대괄호(square brackets)로 나타낸다. 따라서 $[A_{mn}b]$ 는 $m \times (n+1)$ 크기의 행렬을 나타내며 벡터 b는 행렬 A의 마지막 열에 추가 된다.

동적인 고유공간 갱신 기법을 설명하기 위해 다음을 정의한다. $U_{nk} = [u_j], j = 1 \cdots k$ 는 현재까지 학습 자료 $x_i, i = 1 \cdots N$ 에서 구한 고유벡터 집합을 나타내며, $\lambda = diag(\Lambda_{nn})$ 는 고유치 행렬(eigenvalue matrix) Λ_{nn} 의 대각요소를 내림차순으로 정렬한 것을 나타내며 $\bar{x}$ 는 x 의 평균을 의미한다. 온라인 PCA 방법은 학습자료 x_{N+1} 이 추가 되었을 때 이전 학습 자료의 저장 없이 갱신하는 것이다. 먼저 새로운 학습 자료가 추가 되었을 때 갱신된 평균은 식 (1)과 같이 구한다.

$$\bar{x}' = \frac{1}{N+1}(N\bar{x} + x_{N+1}) \quad (1)$$

식 (1)에 의해 새로운 평균이 구해지면 추가된 학습 자료에 의해 갱신된 고유 벡터 집합을 구할 수 있다.

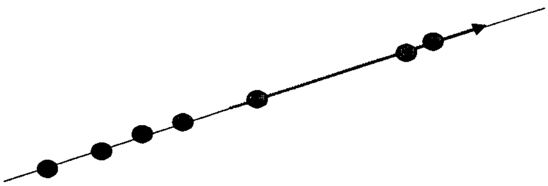

그림 1. 이차원 공간에서 하나의 고유벡터에 의해 모든 자료가 표현된 상태
Fig. 1 A set of points in two dimensional space, all data points can be described by a single base vector.

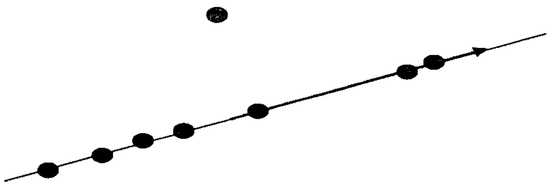

그림 2. 자료집합에 새로운 자료가 추가된 상태
Fig. 2 A new point is added into the data set.

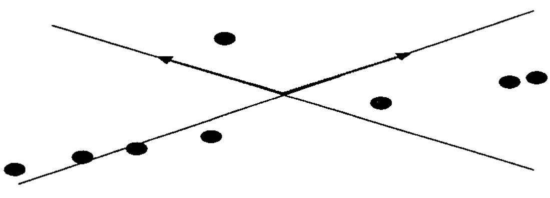

그림 3. 새로운 자료의 추가에 대해 고유벡터를 추가하여 모든 학습 자료를 표현
Fig. 3 We increased the dimensionality in order for all the points to be fully described by the two eigenvectors.

그림 1, 2, 3은 이러한 고유공간 갱신 기법을 개념적으로 설명하고 있다. 갱신된 고유 벡터 집합을 구하기 위해서는 이전의 고유벡터를 회전행렬(rotational matrix)에 적용하여야 하는데 이를 위해 먼저 직교잔차벡터(orthogonal residual vector)를 구해야 한다. 직교잔차벡터는 식 (2) 와 같이 계산된다.

$$\hat{h} = (U_{nk}a_{N+1} + \bar{x}) - x_{N+1} \tag{2}$$

식 (2) 에서 구한 $\hat{h}$을 정규화한 것은 식 (3)과 같이 표시된다.

$$h_{N+1} = \frac{h_{N+1}}{\|h_{N+1}\|_2} \quad \text{for} \quad |h_{N+1}|_2 > 0$$
$$\text{and } h_{N+1} = 0 \text{ otherwise} \tag{3}$$

여기서 $\|h_{N+1}\|_2$은 노름(norm)을 의미한다. $q = k+1$라 정의하면 새로운 고유벡터 U'_{nq}은 식 (3)에 의해 구해진 h_{N+1}과 이전의 고유벡터 U_{nk}에 의해 생성된 행렬을 회전행렬 $R_{(k+1)(k+1)}$에 적용하여 구할 수 있으며 식 (4)와 같이 구한다.

$$U'_{nq} = [U_{nk}\ h'_{N+1}]R_{(k+1)(k+1)} \tag{4}$$

여기서 $R_{(k+1)(k+1)}$ 은 회전행렬 이며 식 (5)의 고유공간의 해이다.

$$D_{(k+1)(k+1)}R_{(k+1)(k+1)} = R_{(k+1)(k+1)}\Lambda'_{(k+1)(k+1)} \tag{5}$$

여기서 $\Lambda'_{(k+1)(k+1)}$ 은 새로운 고유치들의 대각행렬이다. 행렬 $D_{(k+1)(k+1)}$는 다음과 같이 구성할 수 있다.

$$D_{(k+1)(k+1)} = \frac{N}{N+1}\begin{bmatrix}\Lambda_{kk} & 0\\ 0^T & 0\end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix}aa^T & \gamma a\\ \gamma a^T & \gamma^2\end{bmatrix} \tag{6}$$

여기서 $\gamma = h^T_{N+1}(x_{N+1} - \bar{x})$, $a = U^T(x_{N+1} - \bar{x})$와 같이 구하며 0은 k차원의 영벡터(zero vector)이다. 행렬 $D_{(k+1)(k+1)}$를 구성하는 몇 가지 방법이 제안 되었는데 Hall이 제안한 방법만이 평균을 갱신할 수 있도록 제공하는데 이 기법은 평균의 갱신을 허용하지 않는 기법에 비해 성능이 우수한 것으로 알려져 있다[4]. 하지만 Hall의 방법에서는 학습시 고유공간의 차원 유지에 관한 명확한 규칙이 제안되지 않았다. 새로운 학습 자료를 표현하기 위해 고유공간의 차원을 증가시키면 학습 자료는 잘 표현할 수 있으나 차원의 증가로 인해 더 많은 기억 공간을 필요로 한다. 반면에 고유공간의 차원을 일정 크기로 고정하면 이로 인해 학습자료에 대해 일정량의 정보 손실을 감수해야한다.일반적으로 고유공간의 차원을 유지하기 위한 방법에는 다음과 같은 것이 있다.

(1) 잔차벡터가 일정 역치(threshold)를 초과하면 고유벡터를 추가.
(2) 최근에 구해진 고유치(eigenvalue)가 이전에 구한 고유치의 일정 비율을 초과할 경우에 고유벡터를 추가.
(3) 구해진 고유치의 값이 첫 번째 고유치의 일정 비율보다 작은 경우 고유벡터를 제거(차원을 줄임).
(4) 고유공간의 차원을 학습시 일정하게 고정.

본 논문에서는 (2) 방법을 채택하며 새로 구한 고유치의 값이 이전에 구한 고유치의 70% 이상을 초과하면 고유공간의 차원을 증가하는 방법을 사용한다.

III. APEX

Kung과 Diamantaras는 식 (7)과 같은 입출력 관계를 가지는 신경망기반의 주성분기법을 개발하였다.

$$z(t) = W^T(t)x(t) \text{ and } y(t) = z(t) + H^T(t)y(t) \tag{7}$$

여기서 $x(t) \in R^p$는 입력 벡터이며 $y(t) \in R^m$ ($m \le p$)는 출력벡터를 나타낸다. 그리고 $W(t)$는 입력노드와 출력노드간의 직접 연결된 $p \times m$ 차원의 연결가중치 행렬을 나타내며 $H(t)$는 $m \times m$ 차원의 출력노드들 끼리 측면 연결된 연결가중치 행렬을 나타내며 상삼각행렬의 형태를 띤다. APEX 학습 모델의 구조는 그림 1에 나타나 있다.

행렬 W와 H의 열벡터는 다음과 같이 나타낸다 $W=[w_1\ w_2 \cdots\ w_m]$, $H=[0\ h_2\ \cdots\ h_m]$. APEX 모델의 연결가중치 W에 대한 학습 규칙은 식 (8)과 같으며

$$W(t+1)=W(t)+ \eta[X(t)\overline{Y}(t)-W(t)\overline{Y}^2(t)] \quad (8)$$

H 에 대한 학습규칙은 식 (9)와 같다.

$$H(t+1)=H(t)-\eta SUT[Y(t)\overline{Y}(t)]-\eta H(t)\overline{Y}^2(t) \quad (9)$$

여기서 η 는 학습률을 나타내며 X 는 $p\times m$ 차원의 행렬, Y 와 $\overline{Y}$ 는 $m\times m$ 크기의 행렬이며 식 (10)과 같이 정의된다.

$$X=[x_1\ x_2\ \cdots\ x_m],\quad Y=[y_1\ y_2\ \cdots\ y_m],\quad \overline{Y}=diag(y_1, y_2, \cdots, y_m) \quad (10)$$

연산자 SUT는 행렬 [ ]안의 요소 중 상삼각 부분을 되돌려준다. Kung과 Diamantaras 는 APEX 모델의 수렴성을 증명하여 제안된 모델의 안정성을 보였다[5].

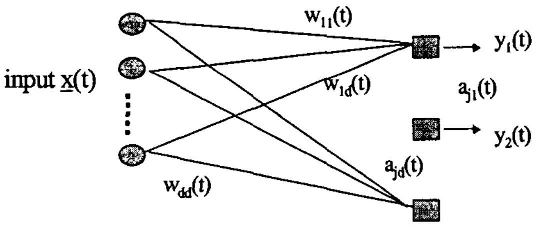

그림 4. APEX 학습모델의 구조
Fig. 4 APEX learning architecture

IV. 실 험

제안된 방법의 성능을 비교해 보기 위해 Scholkopf의 논문[6]에서 사용한 toy data자료와 대용량 데이터에 대한 성능을 평가하기 위해 NIST handwritten digit set[7]에 대해 적용하였다.

IV-1. Toy Data

제안된 방법에 의한 특징추출 성능과 APEX 모델의 성능을 비교하기 위해 식 (11)에 의해 생성된 41개의 자료에 대해 특징추출 능력을 비교하였다.

$$y=x^2+0.2\varepsilon:\quad \varepsilon\sim N(0,1),\quad x=[-1,1] \quad (11)$$

제안된방법에 대해 식 (11)에서 생성된 자료에 대해 적용해본 결과 최종적으로 2개의 고유벡터(eigenvector)를 얻었다. 동일한 조건하에서 성능을 비교하기 위해 APEX의 출력노드는 2개로 한다. 표 1은 학습률을 다양하게 변화시키면서 APEX 모델을 학습한 결과를 나타내고 있다. 일반적으로 신경망 기반의 학습모델은 학습에 필요한 매개변수를 설정하는데 있어 많은 어려움이 존재한다. 최적의 매개변수를 설정하는 것은 일반적으로 어려운 문제이며 풀고자하는 문제에 따라 경험적으로 설정하는 것이 일반적인 방법이다. 본 실험에서도 학습률을 다양하게 변화시켜 학습을 하였다. 학습 결과는 표 1에 나타나 있다. 표1에서 $\cos\Theta$ 는 PCA기법에 의해 구해진 고유벡터와 제안된방법에 의해 구해진 고유벡터, APEX에 의해 구해진 고유벡터와 이루는 각을 의미한다. 따라서 $\cos\Theta$ 는 제안된방법과 APEX모델의 성능을 비교할 수 있는 주요한 요소가 된다. 표1에서 제안된 방법이 APEX 모델보다 특징 추출 성능이 우수함을 잘 보여주고 있다.

표 1. 고유공간 기법과 신경망 기법의 성능비교
Table. 1 Performance comparison of proposed and APEX method.

방법	Iteration	학습률	$cos\theta_1$	$cos\theta_2$	MSE
APEX	50	0.01	0.9993	0.7084	14.85
APEX	50	0.05	수렴되지 않음		
APEX	500	0.01	0.9995	0.9970	4.44
APEX	500	0.05	0.9861	0.9432	4.63
APEX	1000	0.01	0.9995	0.9970	4.40
APEX	1000	0.05	0.9861	0.9432	4.63
IPCA	100		1	1	0.02

IV-2. NIST 자료

제안된 방법과 APEX 모델의 특징추출 성능을

평가하기 위해 이미지 자료에 대해 두 방법을 적용하였다. 실험 자료는 NIST(National Institute of Standards and Technology)에서 제공하는 필기체 숫자를 사용하였다. 원래의 이미지 자료는 16 X 16 크기로 정규화 하였다. 앞의 실험과 마찬가지로 동일한 조건하에서 성능 평가를 위해 먼저 제안된 방법에 학습자료를 적용한 결과 최종적으로 6개의 고유벡터를 구하였다. 따라서 APEX 모델의 출력 노드의 개수도 6개로 설정하였다. 그림 2는 원래 이미지와 각 방법에 의해 재구성된 이미지를 나타낸다. 그림에서 알수 있듯이 필기체 숫자 자료에 대해서도 제안된 방법이 APEX 모델에 비해 원래 이미지를 더 잘 복원해냄을 볼 수 있다.

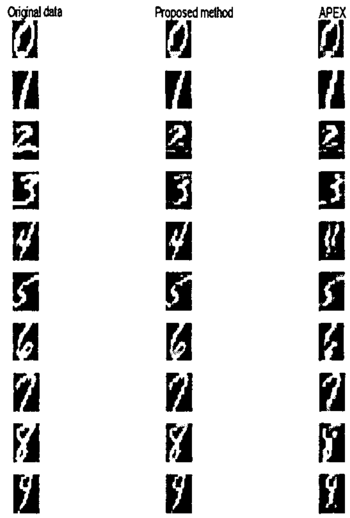

그림 5. NIST 자료에 대해 제안된 방법과 APEX 방법에 의해 복원된 결과
Fig. 5. Reconstruction result from proposed and APEX method on NIST data.

V. 결 론

본 논문은 증가하고 있는 고차원 자료에 대해 효율적으로 특징추출을 하기 위한 점진적인 주성분분석 방법을 제안하였다. 제안한 방법의 효율성을 보이기 위해 toy data와 NIST handwritten 자료에 대해 특징 추출 성능과 복원 성능을 바탕으로 그 성능을 비교하고자 하였다. 실험결과를 통해 제안된 방법이 APEX 방법에 비해 성능이 우수하며 학습시 안정적으로 학습이 이루어 졌다.

참고문헌

[1] Jolliffe, I.T., Principal component analysis, Springer-Verlag, New York, 1986.

[2] Diamantaras, K.I. and Kung, S.Y, Principal Component Neural Networks: Theory and Applications, New York John Wiley&Sons, Inc. 1996.

[3] Winkeler, J. Manjunath, B.S. and Chandrasekaran, S., "Subset selection for active object recognition," In CVPR, IEEE Computer Society Press, vol. 2, pp. 511-516 June 1999.

[4] Chandrasekaran, S., Manjunath, B.S., Wang, Y.F., Winkler, J. and Zhang, H., "An eigenspace update algorithm for image analysis," Graphical Models and Image Processing, Vol.59(5), : pp. 321-332, 1997.

[5] Diamantaras, K.I. and Kung, S.Y, Principal Component Neural Networks: Theory and Applications, New York John Wiley&Sons, Inc. 1996.

[6] Scholkopf, B., Smola, A. and Muller, K.R : Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation 10(5), (1998) 1299-1319

[7] http://yann.lecun.com/exdb/mnist/

저자소개

김병주(Byung-Joo Kim)

경북대학교 이학박사(컴퓨터과학전공)
부산대학교 이학석사(전자계산학)
부산대학교 이학사(전산통계학)
영산대학교 네트워크정보공학부 교수
※관심분야 : 기계학습, 컴퓨터보안