워크북

교과목명 : 머신 러닝

차시명: 10차시

◈ 담당교수: **장 필 훈**

⊙ 세부목차

- 최소오류공식화
- 적용예
- 피셔선형판별과 비교
- PPCA
- 커널PCA
- autoencoder
- 순차데이터 집합
- 마르코프 모델
- 은닉마르코프모델

학습에 앞서

■ 학습개요

차원축소의 대표적인 방법인 PCA에 관해 계속해서 배운다. PCA를 유도하는 두가지 방법(주성분까지 거리의 최소제곱오류, 최대분산)이 결국 같은 결과를 내는 것을 식을 통해 확인하게 된다. 실제로 적용할때 sklearn라이브러리를 어떻게 이용하는지 예제코드로 확인하고, 피셔선형판별과의 차이점도 배운다.

PCA도 확률적 잠재변수모델을 이용해서 확률적으로 접근할수 있고, EM알고리즘을 적용할 수 있다.

선형변환 뿐 아니라 비선형변환에도 이용할 수 있고, 그 방법인 커널 PCA에 관해 배운다. 그와 연관하여 오토인코더에 대해서도 배운다.

순차데이터에서는 마르코프 모델에 관해 배운다. 기본적인 가정과 hmm을 소개하고, 다음시간에 이어서 hmm에 관해 자세히 배우게 된다.

■ 학습목표

1	최소오류공식화를 통해 얻어낸 '주성분'이 최대분산을 통해 얻어낸 것과 동일함을 식으로 확인한다.
2	PCA를 실제로 수행하는 코드예제를 보고 PCA의 입출력을 더 잘 이해한다.
3	확률적 방법으로 PCA를 수행하는 방법을 식으로 자세히 알아본다.
4	PCA로 비선형성을 획득하는 커널PCA에 대해 자세히 알아본다.
5	autoencoder의 입출력형태, 응용방법을 알아본다.

■ 주요용어

용어	해설
정규직교	여러 벡터가 서로 내적이 0이고, 크기가 1이면 정규직교벡터라고 한다.
퍼셉트론	다수의 입력으로 부터 하나의 출력을 내는 연산을 지칭. 보통 출력에 비선형함수를 추가하여 비선형성을 획득한다. 퍼셉트론을 많이 병렬로 배치하여 하나의 layer로 삼고 이것을 여러층 쌓아서 신경망을 만든다.
autoencoder	다층퍼셉트론 중에 입력과 출력이 동일한 것을 지칭. 입출력만 보면 활용도가 없으나, 중간단계에서 데이터의 추상화/압축이 일어나기 때문에 encoder/decoder를 분리하여 응용할 수 있다.
i.i.d.	independent and identically distributed. 여러개의 확률변수가 상호 독립, 동일 분포이면 i.i.d.조건을 만족한다고 한다.

	학습하기

최소오류공식화가 최대분산의 경우와 같은 결과를 얻음을 확인해보겠습니다.

x_n벡터가 u_i벡터들의 합으로 투영된다고 가정하고 식을 다음과 같이 쓸 수 있습니다.

$$\mathbf{x}_n = \sum_{i=1}^{D} \alpha_{ni}\mathbf{u}_i = \sum_{i=1}^{D} (\mathbf{x}_n^{\mathrm{T}}\mathbf{u}_i)\mathbf{u}_i$$

두번째 식은 차원감소(D차원에서 M차원으로) 이전에 차원분리된 상태를 나타낸 것입니다.

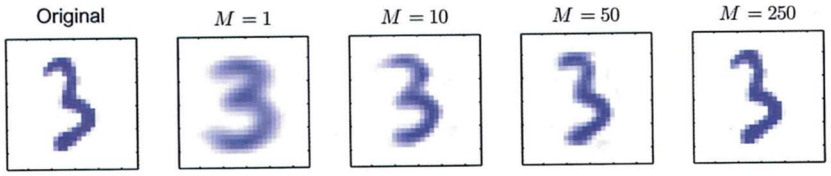
$$\hat{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni}\mathbf{u}_i + \sum_{M+1}^{D} b_i\mathbf{u}_i$$

제곱오류를 사용하기로 하고, 각각의 변수(u, z, b)에 대해 미분하고 0으로 둡니다. 정규직교 조건과 함께 이용해서 제곱오류식에 대입하면 다음을 얻습니다.

제곱오류J : $$J = \frac{1}{N}\sum_{n=1}^{N}\sum_{n=1}^{N}\|\mathbf{x}_n^T\mathbf{u}_i - \bar{\mathbf{x}}_n^T\mathbf{u}_i\|^2 = \sum_{M+1}^{D}\mathbf{u}_i^T\mathbf{S}\mathbf{u}_i$$

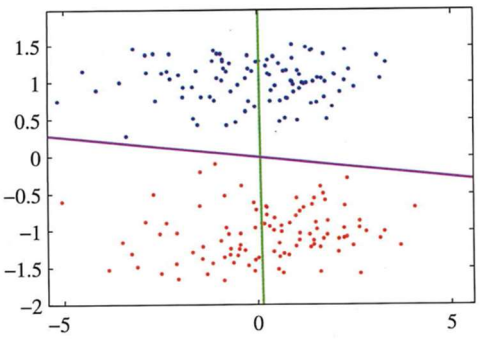
정규직교조건과 함께 라그랑주 승수법을이용하면 임의의 M에 대해 J의 최소화 일반해는 최대분산의 경우와 같은 해를 얻습니다.

PCA는 여러군데에 이용되는데, 차원감소가 주 목적이므로 아래와 같이 이미지에도 사용될 수 있습니다.



Bishop.Fig.12.5

M은 재구성에 사용된 차원을 말합니다. M=10일때를 보면, 단 10차원만 사용해도 숫자모양을 충분히 알아볼 수 있음을 알 수 있습니다. (10차원의 basis방향이 아주 잘 잡혀있기 때문입니다. 최소제곱법과 분산최대법은 바로 이런 '좋은' basis를 선택하는 방법입니다.)



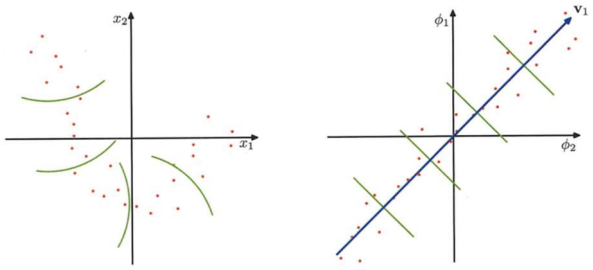
피셔선형판별과 비교해보면, 둘다 선형변환이지만 PCA는 비지도학습이라는 것이 가장 큰 차이입니다. 피셔의 경우 클래스라벨의 정보가 있습니다. 하지만 PCA는 분류가 목적이 아니기 때문에 라벨이 없습니다. 왼쪽그림(Bishop, Fig. 12.7)의 녹색이 피셔선형변환 결과(축)이고 자주색이 최대분산방향입니다.

<PPCA>

확률적 잠재변수모델의 최대가능도 해로 PCA를 표현하는것이 P(probabilistic)PCA입니다. EM알고리즘 응용이 가능하고 missing value를 다룰 수 있다는 장점이 있습니다. 많은 확률모델이 생성모델이기도 하므로 missing value를 다룰 수 있다는 점은 PPCA만의 장점은 아닙니다. PPCA는 난이도에 비해 중요도가 다소간 떨어집니다. 강의를 참고해 주세요.

<커널 PCA>

커널방법이 다 그렇듯이, 선형변환방법으로 비선형변환까지 다룰 수 있게 해줍니다. 원 데이터공간에서는 주성분이 비선형이지만, 특징공간에서 PCA를 수행하는 것입니다. 특징공간상 공분산행렬을 계산하고, 고유벡터를 구하는 등 방법은 본질적으로 같습니다.



Bishop.Fig.2.16

위의 그림이 커널 PCA의 개념적 도식을 잘 나타내주고 있습니다. 왼쪽이 원 데이터 공간, 오른쪽이 특징공간입니다. 공분산 행렬 $C = \frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}}$ 를 고유벡터식에 대입하면 다음과 같습니다.

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\} = \lambda_i\mathbf{v}_i$$

벡터 v_i는 phi(x_n)의 선형결합이므로 다음과 같이 적을 수 있습니다.

$$\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

이를 위에 대입하면,

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{T}\sum_{m=1}^{N}a_{im}\phi(\mathbf{x}_M) = \lambda_i\sum_{n=1}^{N}a_{in}\phi(\mathbf{x}_n)$$

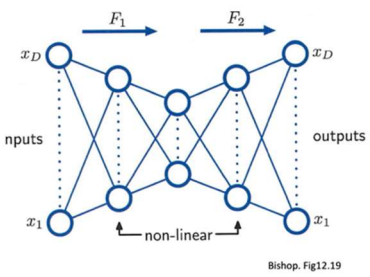
양변에 phi(x_l)^T를 곱해서 커널함수로 나타냅니다.

$$\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x}_l,\mathbf{x}_n)\sum_{m=1}^{N}a_{im}k(\mathbf{x}_n,\mathbf{x}_m) = \lambda_i\sum_{n=1}^{N}a_{in}k(\mathbf{x}_l,\mathbf{x}_n), \quad k(\mathbf{x}_n,\mathbf{x}_m) = \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m)$$

projection도 커널만을 사용해서 가능합니다.(그렇지 않다면 커널방법을 쓰는 의미가 없겠습니다)

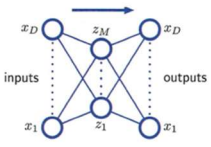
<autoencoder>

오토인코더는 최근들어 더 많이 쓰이고 있는 방법입니다. 입출력 크기가 같은 다층 퍼셉트론을 오토인코더라고 하는데 구조와 오류함수는 다음과 같습니다.



Bishop. Fig12.19

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\|y(\mathbf{x}_n,\mathbf{w}) - \mathbf{x}_n\|^2$$



Bishop. Fig12.18

오류함수를 보면 알수 있듯이 입력과 출력의 차이점을 오류로 씁니다. 입력이 그대로 출력으로 나온다면 가장 좋은 오토인코더겠죠. 그런데 이런걸 왜 쓸까요. 입력을 바로 출력으로 주면 되지 않을까요. 오토인코더는 중간단계에 차원감소되는 부분을 나중에 쓰게 됩니다. 그러니까 훈련단계의 출력이 나중에 inference단계의 출력은 아닌 것입니다. 차원이 감소하므로 정보량이 감소하게 되고, 감소한 정보에

서 다시 원 정보를 추출해내는 정도(입력과 출력의 일치 정도)를 척도로 훈련하므로 적은 차원으로 가장 많은 정보를 담도록 네트웍이 훈련되게 됩니다. 이경우, 우리가 따로 계산할것이 없으므로 구현도 쉽고, 적용도 쉽습니다. 이론적으로는 표준PCA가 이 네트웍의 특별케이스임이 알려져 있습니다.

<순차데이터집합>

이제 순차데이터에 관해 배워보겠습니다. 순차데이터는 결국 hmm을 배우기 위한 전단계입니다.

우리는 보통 데이터포인트들간 의존이 없고 모두 동일한 분포에서 추출되었다고 가정합니다(두번째 가정은 보통 자명해 보입니다) 하지만 실제로는 이런 가정이 깨질때가 있습니다. 특히 순차데이터의 경우 이전의 데이터에 다음의 데이터가 의존하는 경우가 많습니다. 금융데이터가 대표적입니다(최근 관측값이 아주 오래전 관측값보다 더 많은 정보를 가지고 있을 것이라고 생각하는게 상식(?)적입니다)

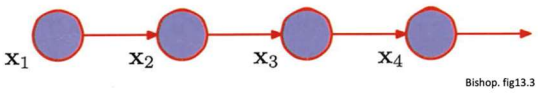
데이터가 생성되는 원 분포가 시간이 지나도 일정하면 stationary하다고 합니다.

순차데이터가 이전의 데이터에 의존한다고 했을 때, 이전의 '모든'데이터에 대한 종속성을 고려하는 것은 비현실적입니다. 직전 몇개에 의존한다고 가정하는 것이 더 자연스러운데요, 직전 1개의 데이터에만 의존한다고 가정하는 것이 우리가 전시간에 배운 마르코프모델입니다.

만일 하나가 아니라 더 많은 데이터에 의존한다면 어떨까요? 모든 데이터에 대한 의존성을 따지는 방법도 있겠지만, 다른 방법도 있습니다. 잠재변수를 가정하고, 잠재변수에 따라 데이터를 추정하는 것입니다.(이것을 상태공간모델state space model이라고 합니다) 이 잠재변수가 이산(discrete)인 경우가 은닉 마르코프모델입니다.

일차 마르코프연쇄는 다음과 같습니다.

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^{N} p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$$

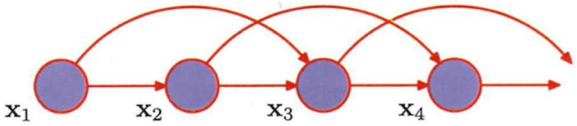


Bishop. fig13.3

$$p(\mathbf{x}_n | \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1})$$

2차 마르코프 연쇄는 다음과 같습니다.

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = p(\mathbf{x}_1) p(\mathbf{x}_2 | \mathbf{x}_1) \prod_{n=3}^{N} p(\mathbf{x}_{n-1} | \mathbf{x}_{n-2})$$



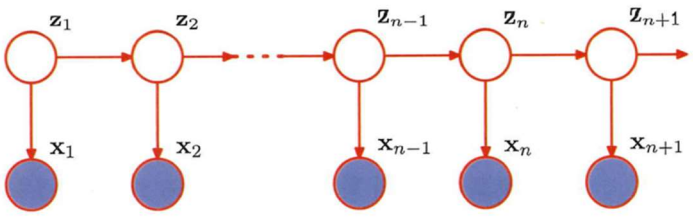
Bishop. fig13.4

이전의 두개 데이터에 의존하는 것을 관찰할 수 있습니다. 이렇게 n차까지 무한히 확장할 수 있으나, 확장할수록 계산량은 기하급수적으로 증가하게 됩니다.

비슷한 것으로 자기회귀모델이 있습니다. 과거 변수값들의 선형조합으로 변수 예측가능한 모델을 말

합니다. 모델로는 무엇이나 쓸 수 있습니다. 선형일수도 있고, 뉴럴넷을 도입할 수도 있습니다.

상태공간모델은 다음과 같이 나타낼 수 있습니다.

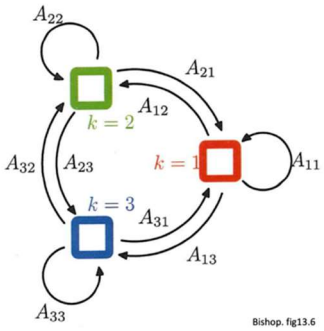


Bishop. fig13.5

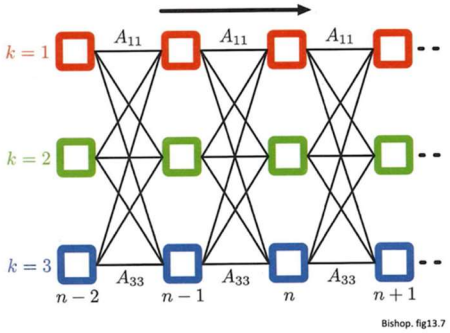
$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{z}_1, \ldots, \mathbf{z}_N) = \mathrm{p}(\mathbf{z}_1)\left[\prod_{n=2}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1})\right]\prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{z}_n)$$

x_n데이터가 z_n에 의존하고 z_n이 매개변수를 가진 순차데이터가 됩니다. 이로써 모든 x_n은 서로 의존입니다.(독립이 아닙니다.)

은닉 마르코프 모델은 상태공간모델의 일종입니다. 상태별로 전이확률(행렬)이 있습니다.



Bishop. fig13.6



Bishop. fig13.7

왼쪽의 모델을 시간순으로 펼친것이 오른쪽입니다. 상태의 변환을 시간의 변화와 함께 관찰할 수 있습니다.

다음시간에는 hmm에 대해 더 자세히 알아보겠습니다.

연습문제

1. 정규직교조건이란, 벡터들의 내적이 0이라는 뜻이다.
    a. X
    b. 내적이 0이면서 동시에 크기가 1이어야 한다. '정규'의 뜻.
2. 일반적인PCA를 수행했을때, 주성분벡터의 최대 차원수는 원 데이터의 차원수와 같다.
    a. O
    b. 다른공간으로 사영(projection)한다고 가정했을 때, 차원수를 감소시키지 않으면 원래 데이터의 차원수로 유지된다. 더 큰 공간으로 투영하는 것은 transform vector의 rank까지 증가시키지는 않으므로(기저벡터중 하나가 다른 기저벡터들의 선형결합으로 표현가능 = 추가적인 정보를 가지지 않음) 의미가 없다.
3. 피셔선형판별과는 달리 주성분분석은 비선형변환이다.
    a. X

b. PCA도 선형변환이다.

4. P(probablistic)PCA는 선형 가우시안 방법론의 일종이다.
    a. O
    b. 잠재변수공간에서 잠재변수를 선형변환을 통해 (가우시안) 매개변수 공간으로 변환한다.
5. PPCA의 경우 계산에 효율이 있다.
    a. O
    b. PCA는 닫힌 해를 구할 수 있으므로 PPCA가 필수적인 것은 아니나, PCA와는 달리 공분산 행렬을 모두 다 구할 필요가 없으므로 EM의 계산비용이 더 많이 들더라도 공분산의 일부만 필요할때는 계산상 이득이 있다.
6. autoencoder는 PCA의 일종으로 볼 수 있다.
    a. X
    b. 그 반대이다. autoencoder가 더 일반성을 가지며 비선형변환도 가능하다. PCA는 autoencoder의 일종임이 알려져 있다.
7. 여러개의 확률변수가 서로 독립이고 모두 다른 분포를 따른다면, iid조건을 만족하다고 할 수 있다.
    a. X
    b. 모두 같은 분포를 따라야 한다.(identically)

정리하기

1. 특정 벡터(정규직교제한이 있다)까지의 거리제곱합을 최소화 하는 방식으로 데이터포인트들에 대해 계산하면, 주성분을 얻을 수 있다.
    a. 위와 같이 얻은 해는 최대분산을 목표로 전개한 것과 동일하다.
2. 주성분벡터들을 이용해서 차원감소를 시도할 수 있다.
3. 피셔선형판별은 supervised, PCA는 unsupervised방식이다.
    a. 둘 다 선형변환
4. 잠재변수 공간에서 선형변환으로 데이터공간의 분포를 나타내는 매개변수를 선형으로 모델링할 수 있다.
    a. 확률적PCA라고 함
5. 확률적 PCA 에서는 EM알고리즘을 쓸 수 있다. 각 데이터포인트에 대해 원 분포의 잠재변수를 추정할 수 있기 때문이다.
6. 비선형변환을 이용해서 PCA를 비선형공간에 대해서도 수행할 수 있다.(커널PCA)
    a. 원 데이터공간에서는 비선형이지만, 특징공간에서는 선형.
    b. 계산비용이 많이 든다
7. 입출력크기가 같은 퍼셉트론을 디자인하고, 입출력의 차이로 학습을 하면, 자기 자신을 재현해

내는 네트워크를 얻는다. 이때 중간의 hidden layer를 이용해서(encoder부분) 차원감소등을 수행할 수 있다. 이것을 autoencoder라고 한다.

a. 표준PCA를 포함하는 일반적인 케이스

8. 데이터가 생성되는 원 분포가 시간이 지나도 일정하면 stationary, 변하면 non-stationay라고 한다.

참고하기

Bishop, C. M. "Bishop-Pattern Recognition and Machine Learning-Springer 2006." Antimicrob. Agents Chemother (2014): 03728-14.

다음 차시 예고

- 은닉마르코프모델의 특징
- hmm에서 최대가능도법
- 바움웰치
- 비터비
- hmm의 확장