학습목표

- 타깃이 없는 데이터를 사용하는 비지도 학습과 대표적인 알고리즘을 소개합니다.
- 대표적인 군집 알고리즘인 k-평균을 배웁니다.
- 대표적인 차원 축소 알고리즘인 주성분 분석(PCA)을 배웁니다.

Chapter 06

비지도 학습

비슷한 과일끼리 모으자!

06-1 군집 알고리즘

핵심 키워드

비지도 학습 히스토그램 군집

흑백 사진을 분류하기 위해 여러 가지 아이디어를 내면서 비지도 학습과 군집 알고리즘에 대해 이해합니다.

출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

최적의 k 찾기

k-평균 알고리즘의 단점 중 하나는 클러스터 개수를 사전에 지정해야 한다는 것입니다. 실전에서는 몇 개의 클러스터가 있는지 알 수 없습니다. 어떻게 하면 적절한 k 값을 찾을 수 있는지 알아보겠습니다.

사실 군집 알고리즘에서 적절한 k 값을 찾기 위한 완벽한 방법은 없습니다. 몇 가지 도구가 있지만 저마다 장단점이 있습니다. 여기서는 적절한 클러스터 개수를 찾기 위한 대표적인 방법인 **엘보우**elbow 방법에 대해 알아보겠습니다.

1. Elbow Method가 필요한 이유?

- K-means는 비지도 학습이기 때문에 정답 레이블이 없음.
- 따라서 적절한 군집 수 K를 스스로 결정해야 함.
- 너무 적은 K → 군집이 너무 뭉뚱그려짐 (과소분류)
- 너무 많은 K → 군집이 너무 세분화됨 (과적합, 해석력 저하)

정의: Elbow Method는 K값을 증가시키며 성능 변화를 관찰하여 최적의 K를 찾는 시각적 기법

2. 핵심 개념: WCSS (Within-Cluster Sum of Squares), inertia

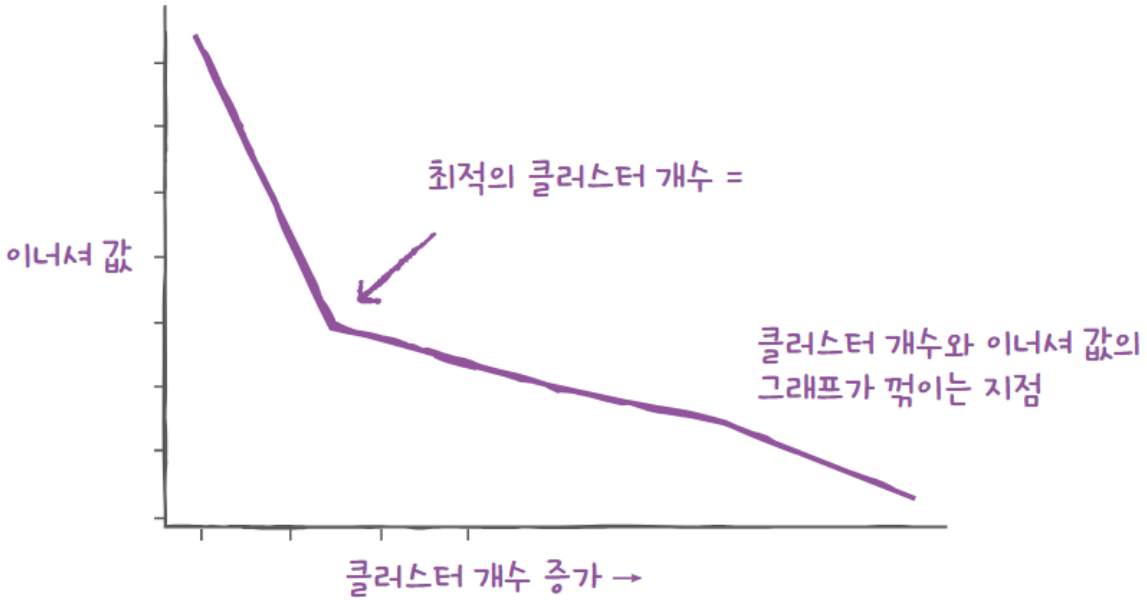
- WCSS는 **클러스터 내의 거리 제곱합**, 즉 각 데이터 포인트가 **자기 클러스터 중심점에 얼마나 가까운가**를 측정한 지표

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

- *K*가 증가할수록 WCSS는 항상 감소함 → 더 많은 클러스터가 더 잘 설명하기 때문
- 하지만 무작정 *K*를 늘리는 것은 좋지 않음

3. Elbow Method의 작동 원리

- 여러 *K* 값에 대해 K-means 클러스터링을 수행하고, WCSS 값을 계산함
- WCSS를 *K*에 따라 그래프로 그리면 급격히 감소하다가 완만해지는 지점이 생김
- 이 "팔꿈치(elbow)" 지점이 최적의 K값이 되는 것



출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

Advanced. Silhouette Score란?

- Silhouette Score(실루엣 점수)는 각 데이터가 **잘 맞는 클러스터에 속했는지**를 측정하는 **군집 품질 지표**
- 클러스터 간 응집도(cohesion)와 분리도(separation)를 동시에 고려

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

기호	의미
$a(i)$	샘플 i와 **자기 클러스터 내** 다른 샘플들과의 평균 거리 (작을수록 좋음)
$b(i)$	샘플 i와 **가장 가까운 다른 클러스터**의 샘플들과의 평균 거리 (클수록 좋음)
$s(i)$	실루엣 점수 (값 범위: -1 ~ 1)

출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

▶ 키워드로 끝내는 핵심 포인트

- **k-평균** 알고리즘은 처음에 랜덤하게 클러스터 중심을 정하고 클러스터를 만듭니다. 그다음 클러스터의 중심을 이동하고 다시 클러스터를 만드는 식으로 반복해서 최적의 클러스터를 구성하는 알고리즘입니다.

- **클러스터 중심**은 k-평균 알고리즘이 만든 클러스터에 속한 샘플의 특성 평균값입니다. 센트로이드centroid라고도 부릅니다. 가장 가까운 클러스터 중심을 샘플의 또 다른 특성으로 사용하거나 새로운 샘플에 대한 예측으로 활용할 수 있습니다.

- **엘보우 방법**은 최적의 클러스터 개수를 정하는 방법 중 하나입니다. 이너셔는 클러스터 중심과 샘플 사이 거리의 제곱 합입니다. 클러스터 개수에 따라 이너셔 감소가 꺾이는 지점이 적절한 클러스터 개수 k가 될 수 있습니다. 이 그래프의 모양을 따서 엘보우 방법이라고 부릅니다.

▶ 핵심 패키지와 함수

scikit-learn

- KMeans는 k-평균 알고리즘 클래스입니다.

  n_clusters에는 클러스터 개수를 지정합니다. 기본값은 8입니다.

  처음에 랜덤하게 센트로이드를 초기화하기 때문에 여러 번 반복하여 이너셔를 기준으로 가장 좋은 결과를 선택합니다. n_init는 이 반복 횟수를 지정합니다. 기본값은 10이었으나, 사이킷런 버전 1.4에서는 'auto'로 변경될 예정입니다.

  max_iter는 k-평균 알고리즘의 한 번 실행에서 최적의 센트로이드를 찾기 위해 반복할 수 있는 최대 횟수입니다. 기본값은 200입니다.

06-3 주성분 분석

핵심 키워드

차원 축소 | 주성분 분석 | 설명된 분산

차원 축소에 대해 이해하고 대표적인 차원 축소 알고리즘 중 하나인 PCA(주성분 분석) 모델을 만들어 봅니다.

출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

차원과 차원 축소

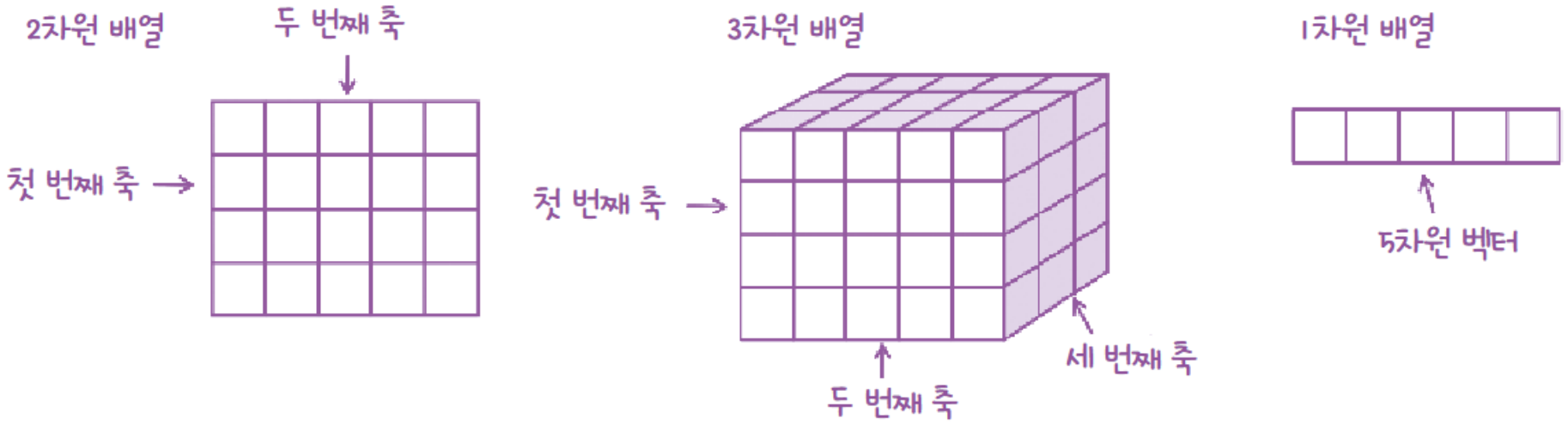
지금까지 우리는 데이터가 가진 속성을 특성이라 불렀습니다. 과일 사진의 경우 10,000개의 픽셀이 있기 때문에 10,000개의 특성이 있는 셈이죠. 머신러닝에서는 이런 특성을 **차원**dimension이라고도 부릅니다. 10,000개의 특성은 결국 10,000개의 차원이라는 건데 이 차원을 줄일 수 있다면 저장 공간을 크게 절약할 수 있을 것입니다.

이를 위해 비지도 학습 작업 중 하나인 **차원 축소**dimensionality reduction 알고리즘을 다루어 보겠습니다. 3장에서 특성이 많으면 선형 모델의 성능이 높아지고 훈련 데이터에 쉽게 과대적합된다는 것을 배웠습니다. 차원 축소는 데이터를 가장 잘 나타내는 일부 특성을 선택하여 데이터 크기를 줄이고 지도 학습 모델의 성능을 향상시킬 수 있는 방법입니다.

또한 줄어든 차원에서 다시 원본 차원(예를 들어 과일 사진의 경우 10,000개의 차원)으로 손실을 최대한 줄이면서 복원할 수도 있습니다. 이 절에서는 대표적인 차원 축소 알고리즘인 **주성분 분석**principal component analysis을 배우겠습니다. 주성분 분석을 간단히 **PCA**라고도 부릅니다.

✚ 여기서 잠깐 2차원 배열과 1차원 배열의 차원은 다른 건가요?

2차원 배열과 1차원 배열(벡터)에서 차원이란 용어는 조금 다르게 사용합니다. 다차원 배열에서 차원은 배열의 축 개수가 됩니다. 가령 2차원 배열일 때는 행과 열이 차원이 되죠. 하지만 1차원 배열, 즉 벡터일 경우에는 원소의 개수를 말합니다. 다음 그림을 참고하세요.



이 절에서는 혼돈을 피하고자 가능하면 차원 대신 특성을 사용합니다. 하지만 차원이란 단어를 완전히 배제하기는 어렵습니다. 이 책이나 다른 책을 볼 때 참고하세요.

벡터의 개념 및 연산

1. 벡터란?

- 벡터는 크기(magnitude)와 방향(direction)을 가진 수학적 객체
- 컴퓨터에서는 숫자들의 집합으로 표현
- 벡터는 위치가 아니라 이동 방향과 크기를 나타냄
- (0,0)에서 시작해 (x,y)로 가는 화살표
- 내적은 각도를, 정규화는 방향만 남기기, 크기는 에너지/거리로 해석

벡터의 개념 및 연산

2. 벡터의 연산

- 덧셈 / 뺄셈 $\begin{bmatrix}1\\2\end{bmatrix} + \begin{bmatrix}3\\4\end{bmatrix} = \begin{bmatrix}1+3\\2+4\end{bmatrix} = \begin{bmatrix}4\\6\end{bmatrix}$

- 스칼라 곱 (Scalar multiplication) $2 \cdot \begin{bmatrix}3\\4\end{bmatrix} = \begin{bmatrix}6\\8\end{bmatrix}$

- 벡터 크기 (Norm, 길이) $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$ 예: $\mathbf{v} = (3,4) \rightarrow \|\mathbf{v}\| = \sqrt{9+16} = 5$

벡터의 개념 및 연산

2. 벡터의 연산

- **내적 (Dot Product)** $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ 예: $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 1 \cdot 3 + 2 \cdot 4 = 11$

  기하학적으로 두 벡터 사이의 각도 정보를 담고 있음 $\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos\theta$

- **정규화 (Normalization)** $\hat{\mathbf{v}} = \dfrac{\mathbf{v}}{\|\mathbf{v}\|}$ 예: $\mathbf{v} = (3, 4) \Rightarrow \hat{\mathbf{v}} = \left(\frac{3}{5}, \frac{4}{5}\right)$

  벡터의 길이를 1로 함

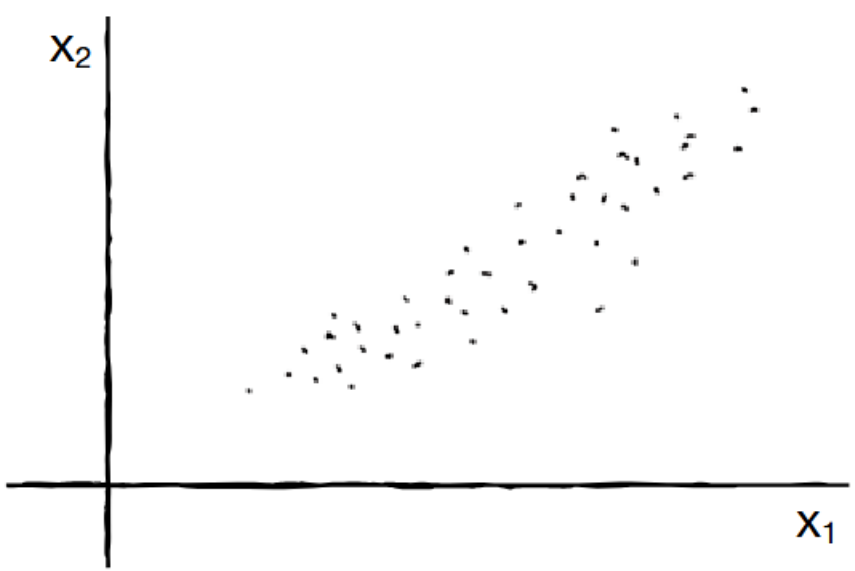
벡터의 개념 및 연산

3. 머신러닝에서 벡터의 응용

분야	예
입력 데이터	각 샘플이 벡터 형태 (예: 10개의 특성 → 10차원 벡터)
가중치 벡터	선형 모델의 weight = 벡터
PCA	주성분도 벡터, 데이터도 벡터
신경망	레이어 사이의 연산도 벡터와 행렬의 곱

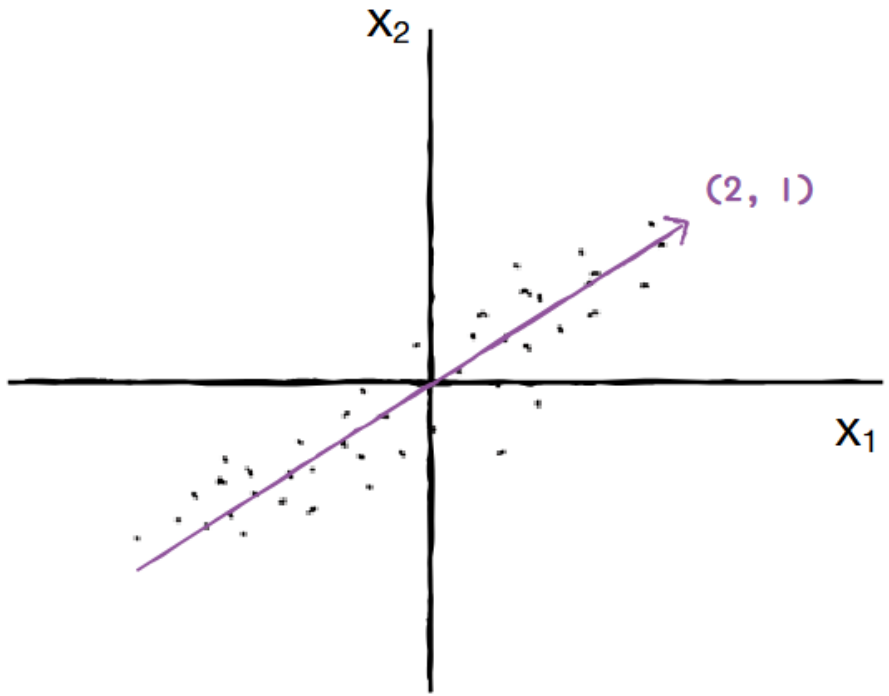
주성분 분석 소개

주성분 분석PCA은 데이터에 있는 분산이 큰 방향을 찾는 것으로 이해할 수 있습니다. 분산은 데이터가 널리 퍼져있는 정도를 말합니다. 분산이 큰 방향이란 데이터를 잘 표현하는 어떤 벡터라고 생각할 수 있습니다. 이해하기 쉽도록 다음과 같은 2차원 데이터를 생각해 보죠.



출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

앞에서 찾은 직선이 원점에서 출발한다면 두 원소로 이루어진 벡터로 쓸 수 있습니다. 예를 들어 다음 그림의(2, 1)처럼 나타낼 수 있겠죠.



이 벡터 (원본 데이터에 있는 어떤 방향)를 주성분 (principle component)라고 부름

주성분 벡터의 원소 개수는 원본 데이터셋에 있는 특성 개수와 같음

벡터의 크기 (norm): $\|\mathbf{v}\| = \sqrt{2^2 + 1^2} = \sqrt{5} \approx 2.24$

PCA가 판단하기에 데이터가 가장 많이 퍼진 방향이 "특성 1의 2배 + 특성2의 1배" 방향이라는 뜻

출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

Q. 왜 "원본 데이터에 있는 어떤 방향"인가?

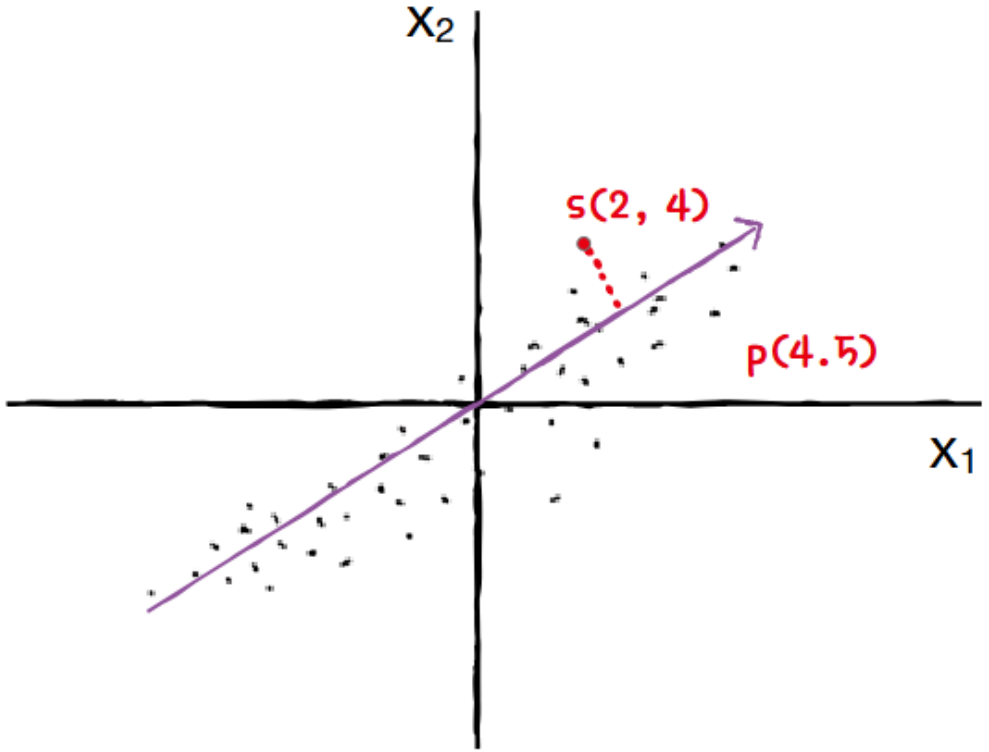
- 주성분 벡터는 원본 특성 공간(feature space) 내에서의 한 방향.
- 즉, 주성분 벡터는 원본 데이터의 각 특성에 대해 얼마나 기여하는지(가중치)를 나타냄
- 예를 들어, 데이터가 3개의 특성(예: 키, 몸무게, 나이)을 가지고 있다면:

PC1 = [0.5, -0.3, 0.8]

> 이는 "키 0.5 + 몸무게 -0.3 + 나이 0.8 방향으로 가장 많이 퍼져 있다"는 뜻

Q. 왜 원소 개수가 특성 수와 같은가?

- 주성분 벡터는 원본 특성 공간에서의 벡터.
- 즉, 각 주성분은 원래 특성 축들로 표현된 방향.
- 따라서 특성이 n개라면, 주성분 벡터도 n-차원 벡터임.



- 원본 데이터는 주성분을 사용해 차원을 줄일 수 있음.
- 예를 들어 샘플 데이터 s(2, 4)를 주성분에 직각으로 투영 (벡터 내적) 하면 1차원 데이터 p(4.24) (스칼라 값)를 만들 수 있음.

주성분은 원본 차원과 같고 주성분으로 바꾼 데이터는 차원이 줄어든다는 점을 꼭 기억하세요. 주성분이 가장 분산이 큰 방향이기 때문에 주성분에 투영하여 바꾼 데이터는 원본이 가지고 있는 특성을 가장 잘 나타내고 있을 것입니다.

출처: 박해선, 『혼자 공부하는 머신러닝+딥러닝』, 한빛미디어(2020)

▶ 키워드로 끝내는 핵심 포인트

- **차원 축소**는 원본 데이터의 특성을 적은 수의 새로운 특성으로 변환하는 비지도 학습의 한 종류입니다. 차원 축소는 저장 공간을 줄이고 시각화하기 쉽습니다. 또한 다른 알고리즘의 성능을 높일 수도 있습니다.

- **주성분 분석**은 차원 축소 알고리즘의 하나로 데이터에서 가장 분산이 큰 방향을 찾는 방법입니다. 이런 방향을 주성분이라고 부릅니다. 원본 데이터를 주성분에 투영하여 새로운 특성을 만들 수 있습니다. 일반적으로 주성분은 원본 데이터에 있는 특성 개수보다 작습니다.