# 기계학습_231114오늘

9.0 비지도 학습

비지도 학습의 큰 잠재력
대부분의 머신러닝 애플리케이션이 지도 학습 기반이지만, 사용할 수 있는 데이터는 대부분 레이블이 없음.
얀 르쿤의 유명한 말 “비지도 학습은 케이크의 빵, 지도 학습은 크림, 강화 학습은 체리”
군집(clustering)
비슷한 샘플을 클러스터(cluster)로 모음.
데이터 분석, 고객 분류, 추천 시스템, 검색 엔진, 이미지 분할, 준지도 학습, 차원 축소 등에 사용하는 훌륭한 도구다.
이상치 탐치(outlier detection)
‘정상’ 데이터가 어떻게 보이는지를 학습하고, 비정상 샘플을 감지하는 데 사용.
밀도 추정(density estimation)
데이터셋 생성 확률 과정(random process)의 확률 밀도 함수(probability density function, PDF)를 추정함.
밀도 추정은 이상치 탐지에 널리 사용됨.
밀도가 매우 낮은 영역에 놓인 샘플이 이상치일 가능성이 높음.

9.1 군집

군집(clustering)
각 샘플은 하나의 그룹에 할당
비슷한 샘플을 구별해 하나의 클러스터 또는 비슷한 샘플의 그룹으로 할당하는 작업
군집을 사용하는 다양한 어플리케이션
고객 분류
추천 시스템
데이터 분석
차원 축소 기법
샘플의 친화성 측정
이상치 탐지
제조 분야에서 결함 감지
부정 거래 감지에 활용
준지도 학습
검색 엔진
이미지 분할

9.1.1 k-평균

K-평균 (로이드-포지 알고리즘)
반복 몇 번으로 레이블이 없는 데이터셋을 빠르고 효율적으로 클러스터로 묶는 간단한 알고리즘.
예제
샘플 덩어리 다섯 개로 이루어진 데이터셋
(사이킷런) K-평균 알고리즘 적용
각 군집의 중심을 찾고 가장 가까운 군집에 샘플 할당
군집수(n_clusters) 지정해야 함

9.1.1(1) k-평균

결정경계
결과: 보로노이 다이어그램
평면을 특정 점까지의 거리가 가장 가까운 점의 집합으로 분할한 그림
경계 부분의 일부 샘플을 제외하고 기본적으로 군집이 잘 구성됨.
K-평균 알고리즘의 단점
군집의 크기가 서로 많이 다르면 잘 작동하지 않음.
샘플과 센트로이드까지의 거리만 고려되기 때문
하드 군집과 소프트 군집
하드 군집 – 각 샘플에 대해 가장 가까운 클러스터를 선택.
소프트 군집 – 클러스터마다 샘플에 점수를 부여함. 샘플별로 각 군집 센트로이드와의 거리를 측정

9.1.1(2) k-평균

K-평균 알고리즘
처음에는 센트로이드를 랜덤하게 선정
수렴할 때까지 다음 과정 반복
각 샘플을 가장 가까운 센트로이드에 할당
군집별로 샘플의 평균을 계산하여 새로운 센트로이드 지정

9.1.1(3) k-평균 – 센트로이드 초기화 방법

관성(inertia, 이너셔)
k-mean 모델 평가 방법
정의: 샘플과 가장 가까운 센트로이드와의 거리의 제곱의 합
각 군집이 센트로이드에 얼마나 가까이 모여있는가를 측정
score() 메서드가 측정. (음수 기준)
좋은 모델 선택법
다양한 초기화 과정을 실험한 후에 가장 좋은 것 선택

n_init = 10이 기본값으로 사용됨.
10번 학습 후 가장 낮은 관성을 갖는 모델 선택.
K-평균++
센트로이드를 무작위로 초기화하는 대신 특정 확률분포를 이용하여 선택
센트로이드들 사이의 거리를 크게할 가능성이 높아짐.
KMeans 모델의 기본값으로 사용됨

9.1.1(4) k-평균 (K-평균 속도 개선과 미니배치 k-평균)

elkan 알고리즘
algorithm=elkan: 불필요한 거리 계산을 많이 피함으로, 학습 속도 향상됨.
삼각 부등식을 사용함. algorithm=full:
미니배치 K-평균
전체 데이터셋 대신 각 반복마다 미니배치를 사용해 센트로이드를 조금씩 이동함.
미니배치를 지원하는 K-평균 알고리즘: MiniBatchMeans
memmap 활용
대용량 훈련 세트 활용하고자 할 경우
점진적 PCA 에서 사용했던 기법과 동일
memmap 활용이 불가능할 정도로 큰 데이터셋을 다뤄야 하는 경우
MiniBatchKMeans의 partial_fit() 메서드 활용
미니배치로 쪼개어 학습
미니배치 K-평균의 특징
K-평균 알고리즘 보다 훨씬 빠름.
이너셔는 일반적으로 조금 더 나쁨.
군집수가 증가할 때 이너셔는 더 나쁨

9.1.1(5) k-평균 (최적의 클러스터 개수 찾기)

최적의 클러스터 개수 설정 방법은?
최적의 군집수를 사용하지 않으면 적절하지 못한 모델을 학습할 수 있음.
관성과 클러스터
클러스터 개수 k가 증가할 수록 관성(inertia)이 작아지므로, 좋은 성능 지표가 아님.
관성만으로 모델을 평가할 수 없음.
관성이 더 이상 획기적으로 줄어들지 않는 지점의 클러스터 개수 선택( k=4 선택 가능)

9.1.1(6) k-평균 (최적의 클러스터 개수 찾기)

실루엣 점수와 클러스터 개수
실루엣 점수
모든 샘플에 대한 실루엣 계수의 평균
실루엣 계수 : -1과 +1사이의 값
+1에 가까운 값: 자신의 클러스터 안에 포함되고, 다른 클러스터와는 멀리 떨어짐.
0에 가까운 값: 클러스터 경계에 위치
-1에 가까운 값: 샘플이 잘못된 클러스터에 할당됨
[그림9-9] 에서 볼때 k=4가 좋은 선택이지만, k=5도 좋은 선택이 될 수 있음.

9.1.1(7) k-평균 (최적의 클러스터 개수 찾기)

실루엣 다이어그램과 클러스터 개수
실루엣 다이어그램
클러스터별 실루엣 계수 모음. 칼 모양의 그래프
칼 두께 : 클러스터에 포함된 샘플의 개수
칼 길이: 클러스터에 포함된 샘플의 실루엣 계수 (길 수록 좋음)
빨간 파선: 클러스트 계수에 해당하는 실루엣 점수
대부분의 칼이 빨간 파선보다 길어야 함 (낮으면 다른 클러스터랑 너무 가까움)
칼의 두께가 서로 비슷해야, 즉, 클러스터별 크기가 비슷해야 좋은 모델임.
실루엣 다이어그램 상에서 k=5가 보다 좋은 모델임.

9.1.2 k-평균의 한계

k-평균은 속도가 빠르고 확장이 용이하다는 장점이 있지만, 완벽한 것은 아님.
K- 평균의 한계
최적이 아닌 솔루션을 피하려면 알고리즘을 여러 번 실행해야 함.
클러스터 개수를 미리 지정해야 함.
클러스터의 크기나 밀집도가 다르거나, 원형이 아닐 경우 잘 작동하지 않음.
데이터에 따라서 잘 수행할 수 있는 클러스터 알고리즘이 다름.

9.1.3 군집을 사용한 이미지 분할

이미지 분할 (image segmentation)
이미지를 세그먼트 여러 개로 분할하는 작업
다양한 클러스터 개수로 k-평균을 사용해 만든 이미지 분할
시맨틱 분할 (semantic segmentation)
동일한 종류의 물체에 속한 모든 픽셀은 같은 세그먼트에 할당
자율주행: 보행자들을 모두 하나의 영역, 또는 각각의 영역으로 할당 가능
색상 분할(color segmentation)
K-평균을 이용하여 색상분할 실행
인공위성 사진 분석 : 한 지역의 전체 산림 면적

K- 평균이 비슷한 크기의 클러스터를 만드는 경향 – 무당 벌레를 하나의 클러스터로 만들지 못함.

9.1.4 군집을 사용한 전처리

미니 MNIST 데이터셋 전처리
MNIST와 비슷한 숫자 데이터셋
8x8 크기의 흑백 사진 1,797개.
전처리 없이 로지스틱회귀 학습시키면 96.89% 정확도임.
K-평균을 전처리 단계로 사용한 후 로지스틱회귀 학습
훈련 세트를 50개의 클러스터로 모음.
군집에서 데이터셋의 차원은 64->50 으로 감소.
정확도: 97.78%로 증가
클러스터 개수 k를 임의로 정하지 않고, GridSearchCV를 사용해 최적의 클러스터 개수를 찾음.
최적 군집수: 99
모델 정확도: 98.22%

9.1.5 군집을 사용한 준지도 학습

군집을 사용한 준지도 학습
레이블이 없는 샘플이 많고 레이블이 있는 샘플이 적을 때 사용
예제: 미니 Mnist
50개 샘플을 대상으로 학습한 모델의 성능: 83.33% 정도
대표 이미지
먼저 훈련 세트를 50개의 클러스터로 모음
그 다음 각 클러스터에서 센트로이드에 가장 가까운 이미지를 찾음.
50개의 이미지를 보고 수동으로 레이블 할당
50개 샘플을 레이블 할당하여 학습된 모델 성능 : 92.2% 정도
무작위 샘플 대신 대표 샘플에 레이블을 할당하는 것이 좋은 방법임.

9.1.5 군집을 사용한 준지도 학습(1)

레이블 전파
레이블을 동일한 클러스터에 있는 모든 샘플로 전파
정확도 : 93.3%
군집에 속한 전체 샘플 보다 센트로이드에 가까운 20% 정도에게만 레이블 전파 후 학습
센트로이드에 가깝기 때문에 레이블의 정확도가 매우 높음.
정확도 : 94.0%
전파된 레이블이 실제로 매우 좋기 때문에 결과가 좋음.

9.1.6 DBSCAN

밀집된 연속적 지역을 클러스터로 정의
사이킷런의 DBSCAN 모델
두 개의 하이퍼파라미터 사용
eps : ε-이웃 범위 ( 주어진 기준값 ε 반경 내에 위치한 샘플)
min_samples: ε 반경 내에 위치하는 이웃의 수
핵심샘플과 군집
핵심샘플: $\varepsilon$ 반경 내에 자신을 포함해서 min-samples개의 이웃을 갖는 샘플
군집: 핵심샘플로 이루어진 이웃들로 구성된 그룹
이상치
핵심샘플이 아니면서 동시에 핵심샘플의 이웃도 아닌 샘플.
반달모양 데이터 활용

9.1.6 DBSCAN (1)

DBSCAN과 예측
predict() 메서드 지원하지 않음. fit_predict() 메서드 제공함.
이유: KNeighborsClassifier 등 보다 좋은 성능의 분류 알고리즘 활용 가능.
아래 코드: 핵심샘플 대상 훈련.
새로운 샘플에 대한 예측 가능
새로운 4개의 샘플에 대한 예측을 보여줌.
이상치 판단
두 군집으로부터 일정거리 이상 떨어진
샘플을 이상치로 간주 가능.
양편 끝쪽에 위치한 두 개의 샘플이 이상치로
간주될 수 있음.

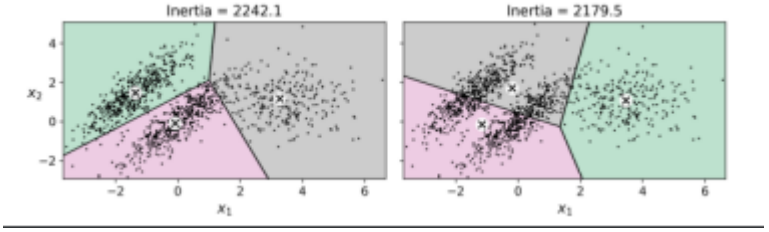
9.1.6 DBSCAN (2)

DBSCAN의 장단점
매우 간단하면서 매우 강력한 알고리즘.
하이퍼파라미터: 단 2개
군집의 모양과 개수에 상관없음.
이상치에 안정적임. 군집 간의 밀집도가 크게 다르면 모든 군집 파악 불가능.
계산복잡도
시간복잡도: 약 $O(m \log m)$. ($m$은 샘플 수)
샘플 개수에 대해 거의 선형적으로 증가
공간복잡도: 사이킷런의 DBSCAN 모델은 의 메모리 요구

---

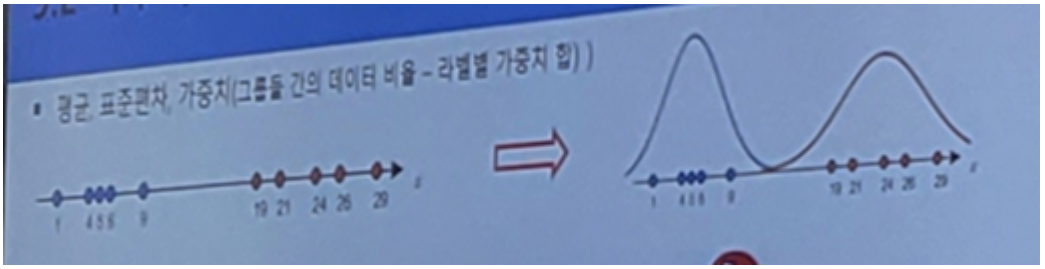
9.2 가우시안 혼합

*가우시안 혼합 모델( GMM(Gaussian Mixture Model))
- 샘플이 '파라미터가 알려지지 않은 여러 개의 혼합된 가우시안 분포'에서 생성되었다고 가정하는 확률 모델
- 가우시안 분포 = 정규분포 : 종 모양의 확률밀도함수를 갖는 확률분포
*클러스터
- 하나의 가우시안 분포에서 생생된 모든 샘플들의 그룹
- 타원형 모양.
- 그림처럼 일반적으로 모양, 크기, 밀집도, 방향 다름.
- 핵심 : 각 샘플이 어떤 정규분포를 따르는지를 파악




9.2 가우시안 혼합-EM 알고리즘

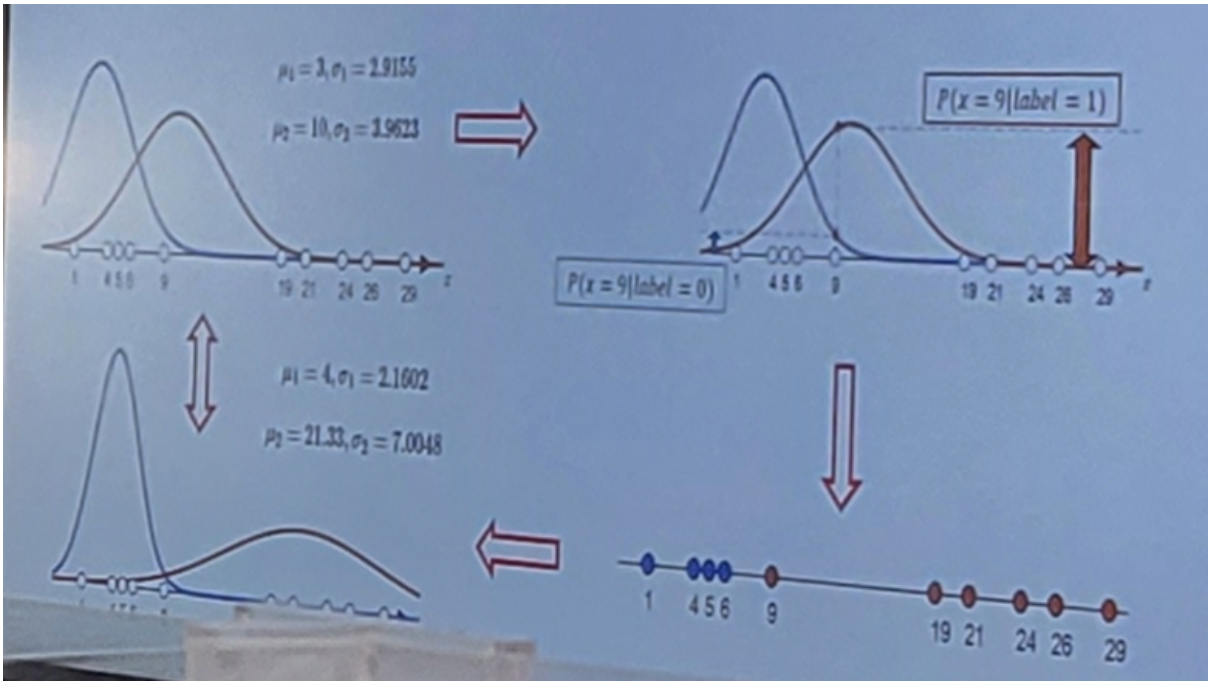
*라벨 있으면 : 분포예측가능



*라벨없으면 : 분포예측 불가

*라벨얻기위해 분포필요 분포얻기위해 라벨 필요함.

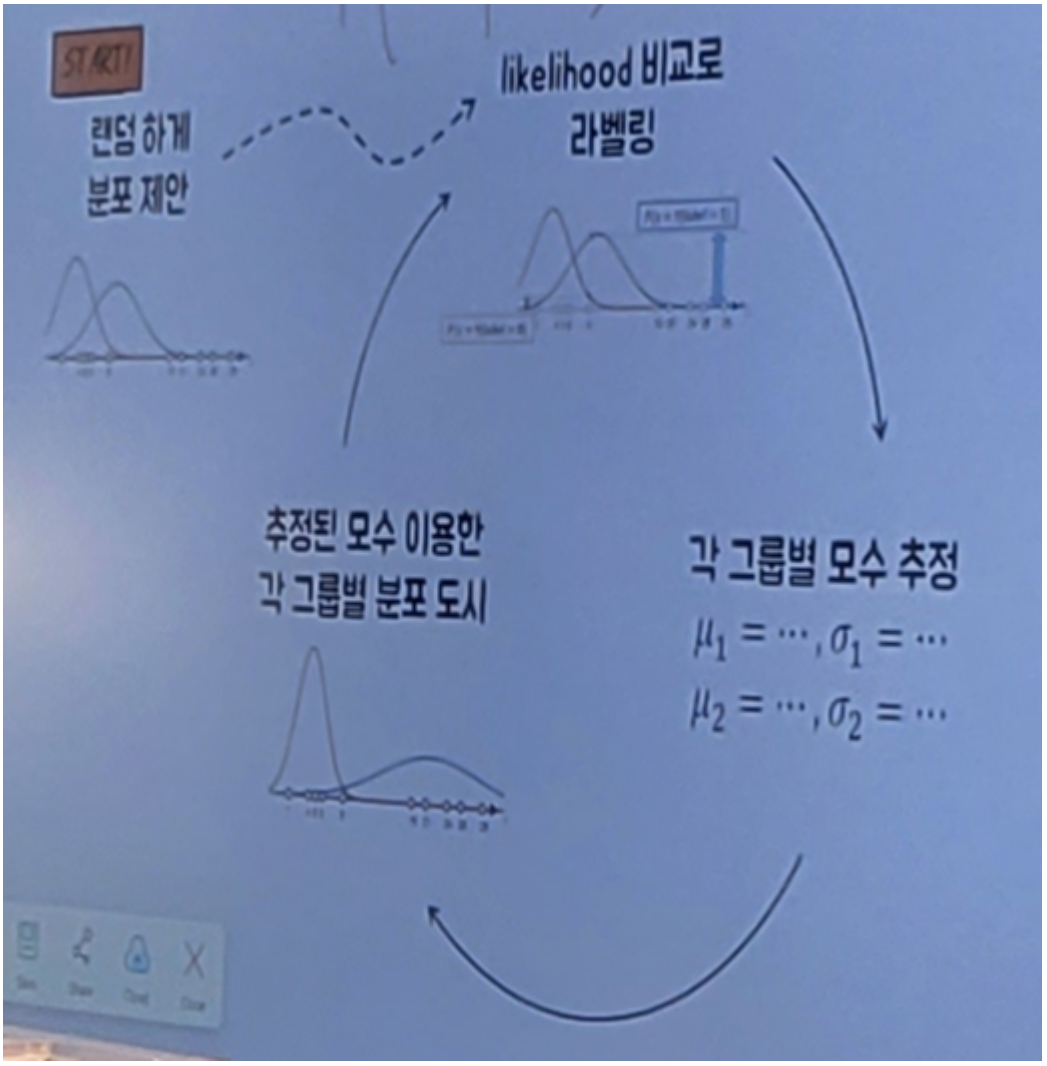
-랜덤하게 분포 생성→ 아래를 0 위를1로 정함→ 라벨링 → 분포도 그림




9.2 EM 알고리즘을 이용한 가우시안 혼합 과정

*정규분포의 변화 과정

- 사전지식 있는 상태에서 데이터 들어오면 likelihood가최대되는것을 선택 데이터 비교 라벨링




9.2 가우시안 혼합 – GMM 활용

*GMM 활용

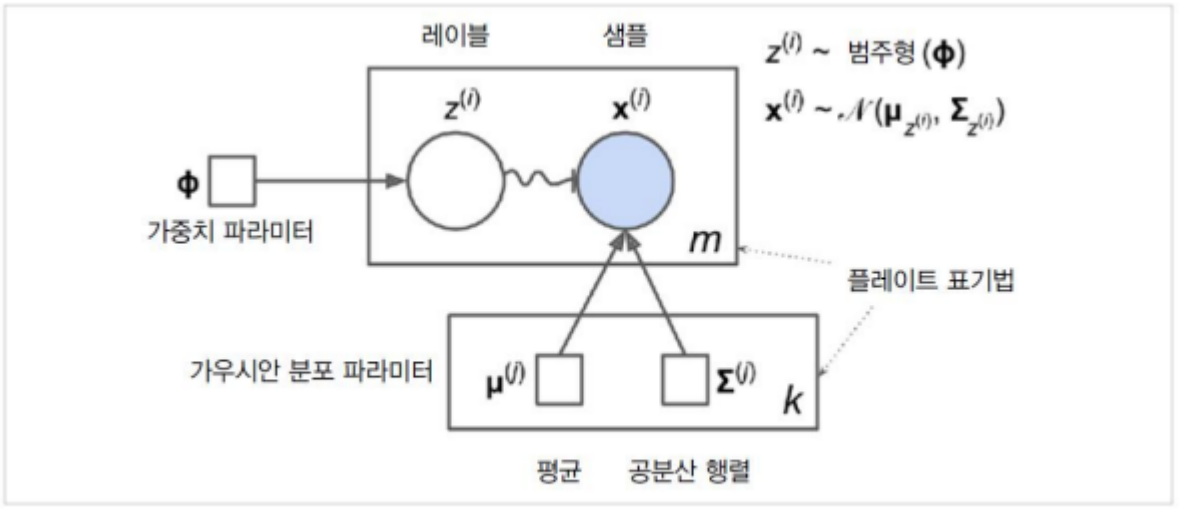



그림 9-16 가우시안 혼합 모델의 그래프 모형. 파라미터(사각형), 확률 변수(원), 조건부 의존성(실선 화살표)

*GMM 훈련과 EM(Expectation-Maximization) 으로 파라미터 추정
- GaussianMixture 모델 적용, n_components: 군집수 지정

```
from sklearn.mixture import GaussianMixture
```

```
gm = GaussianMixture(n_components=3, n_init=10, random_state=42)
gm.fit(X)
```

- n_init: 모델 학습 반복 횟수 : 파라미터(평균값, 공분산 등)를 무작위로 추정한 후 수렴할 때까지 학습시킴.
- EM 알고리즘이 추정한 파라미터를 확인

```
gm.weights_
```

```
array([0.39025715, 0.40007391, 0.20966893])
```

```
gm.means_
```

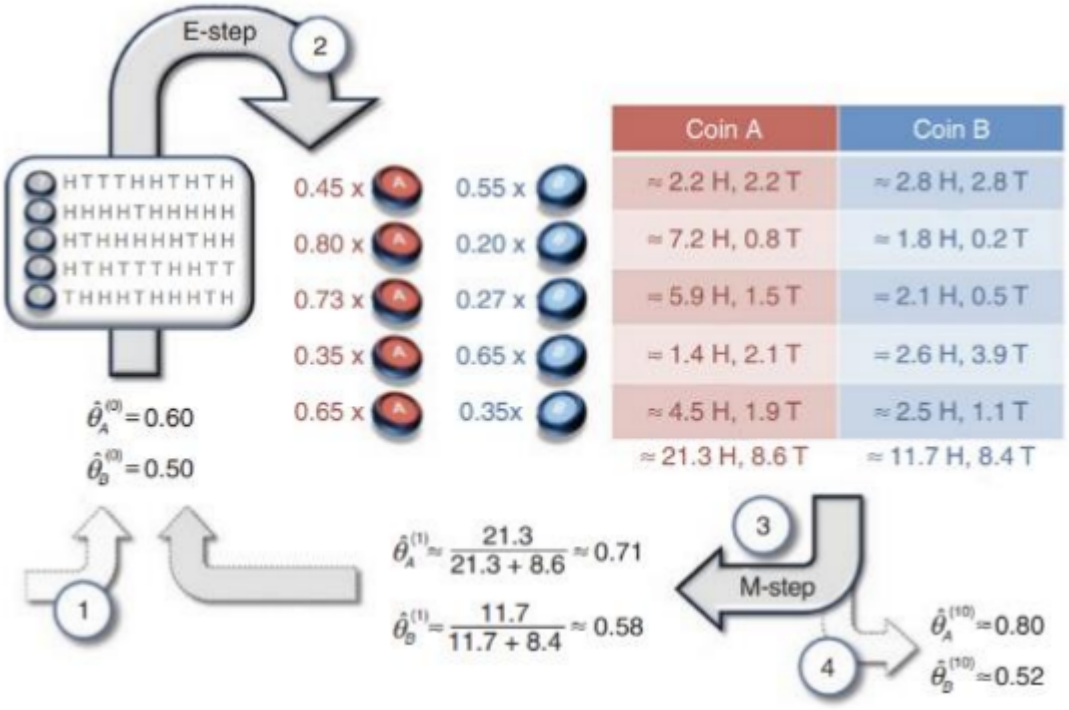
```
array([[ 0.05131611,  0.07521837],
       [-1.40763156,  1.42708225],
       [ 3.39893794,  1.05928897]])
```

```
gm.covariances_
```

```
array([[[ 0.68799922,  0.79606357],
        [ 0.79606357,  1.21236106]],
```

9.2 가우시안 혼합-EM 알고리즘

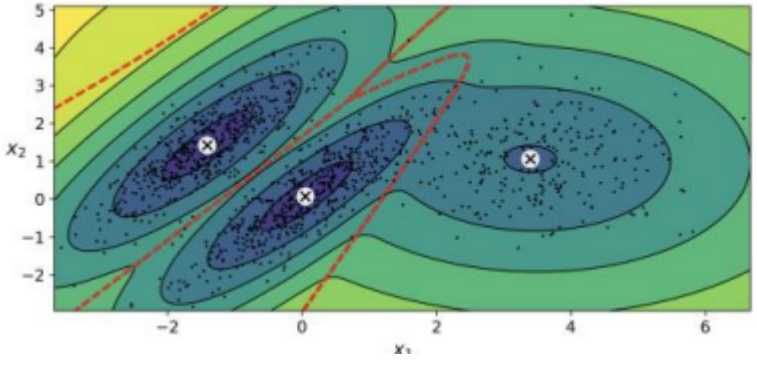
*초기값 : 랜덤하게 확률값을 넣음.
*E-Step : Hidden variable의 resposibility를 계산하는 단계
*M-Step : 추정값 θ를 업데이트 하는 단계




9.2 가우시안 혼합 (3)

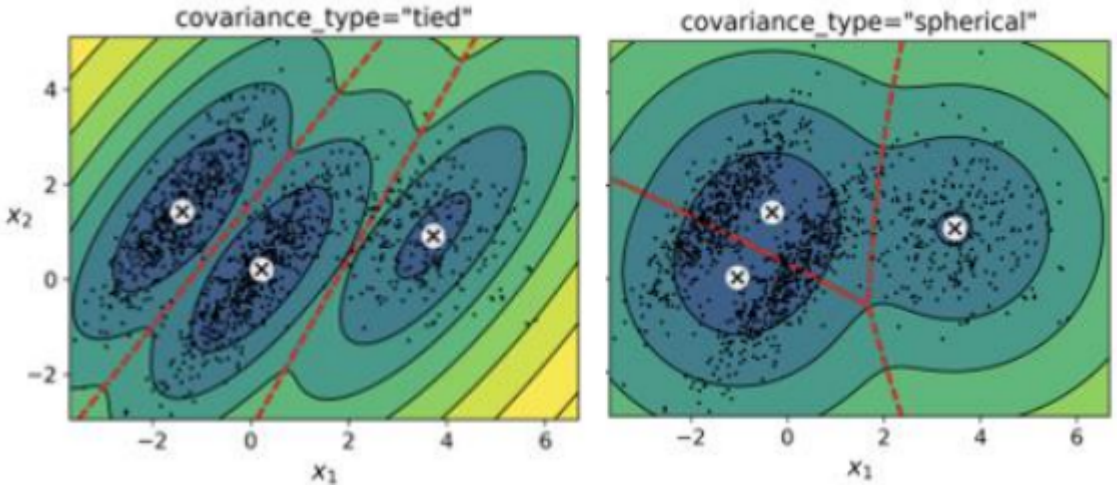
*학습된 모델을 보여줌
- 군집 평균, 결정 경계, 밀도 등고선

→ 중앙(흰색)이 평균, 색깔별로 표준편차




*GMM 모델 규제
- 특성수가 크거나, 군집수가 많거나, 샘플이 적은 경우 최적 모델 학습어려움
- covariance_type 설정 : 공분산(covariance)에 규제를 가해 학습도움
- covariance_type 옵션값
full : 아무런 제한 없음, 기본값, 모든 클러스터가 어떤 크기의 타원도 될수 있음.

spherical  :모든 클러스터가 원형이지만 지름은 다를 수 있음
diag : 어떤 타원형도 가능, 타원의 축이 좌표축과 평행하다고 가정.
tied : 모든 군집의 동일 모양, 동일 크기, 동일 방향을 갖는다고 가정




9.2.1 가우시안 혼합을 사용한 이상치 탐지

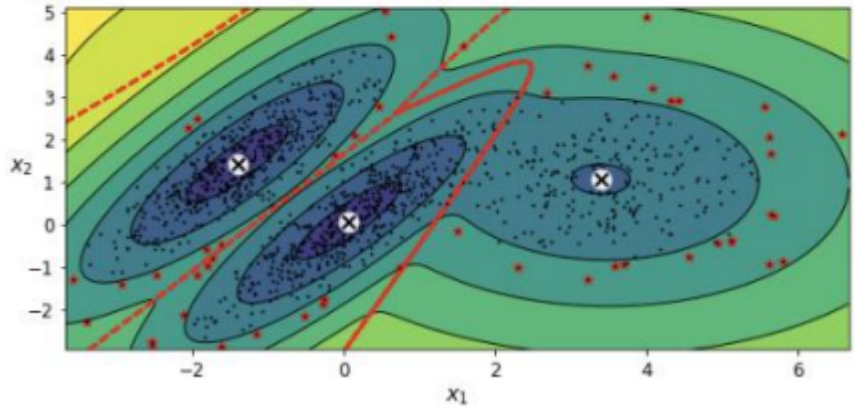
*이상치 탐지 : 보통과 많이 다른 샘플 감지 작업
*가우시안혼합 모델 활용한 이상치 탐지
- 밀도가 임곗값보다 낮은 지역에 있는 샘플 = 이상치
- 정밀도/재현율 트레이드 오프 :

거짓 양성이 너무 많다= 임곗값을 더 낮춤
거짓 음성이 너무 많다= 임곗값을 더 높임.

*코드

```
densities = gm.score_samples(X)
density_threshold = np.percentile(densities, 4)
anomalies = X[densities < density_threshold]
```

그림 저장: mixture_anomaly_detection_plot




9.2.2 클러스터 개수 선택하기

*K-평균에서 사용했던 관성 또는 실루엣 점수 사용 불가 : 군집이 타원형일 때 값이 일정하지 않기 때문

*이론적 정보 기준을 '최소화' 하는 모델 선택 가능

*이론적 정보 기준

- BIC: Bayesian information criterion

$$\log(m)\, p - 2\log(\hat{L})$$

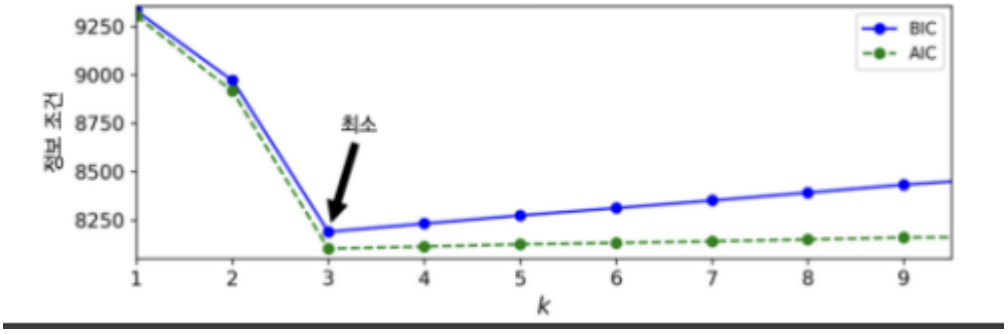
- AIC: Akaike information criterion

$$2\, p - 2\log(\hat{L})$$

- 특징 : 학습할 파라미터 많은 모델 = 벌칙, 데이터에 잘 학습하는 모델 = 보상

*군집수와 정보조건

- 그림: 군집수 와 AIC, BIC의 관계를 보여줌.(K = 3이 최적)




9.2.3 베이즈 가우시안 혼합 모델

*BayesianGaussianMixture 모델
- 최적의 군집수를 자동으로 찾아줌.
- 최적의 군집수보다 큰 수를 n_components에 전달해야 함.(군집에 대한 최소한의 정보를 알고 있다고 가정)
- 자동으로 불필요한 군집 제거

```
from sklearn.mixture import BayesianGaussianMixture
```

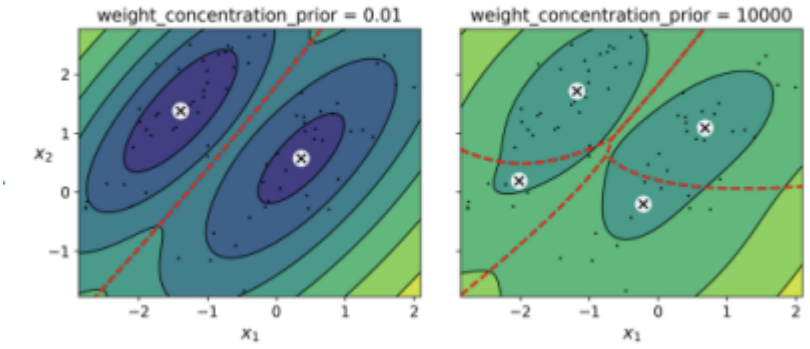
```
bgm = BayesianGaussianMixture(n_components=10, n_init=10, random_state=42)
bgm.fit(X)
```

*베이즈 확률통계론 활용

식 9-2 베이즈 정리

$$p(\mathbf{z}\,|\,\mathbf{X}) = \text{사후 확률} = \frac{\text{가능도} \times \text{사전 확률}}{\text{증거}} = \frac{p(\mathbf{X}\,|\,\mathbf{z})p(\mathbf{z})}{p(\mathbf{X})}$$

*사전 믿음
- 군집수가 어느 정도일까를 나타내는 지수
- weight_concentration_prior 하이퍼파라미터:
n_components에 설정된 군집수에 대한 규제로 사용, 작은 값 : 특정 군집의 가중치를 0에 가깝게 만들어 군집수를 줄이게함,
큰 값:  n_components에 설정된 군집수가 유지되도록 함




9.2.3 베이즈 가우시안 혼합 모델

*가우시안혼합 모델의 장단점
- 장점 – 타원형 클러스터에 잘 작동
- 단점 :

다른 모양을 가진 데이터셋 = 성능안좋음

예제: 달모양 데이터에 적용하는 경우(억지로 타원을 찾으려 시도→2개가 아니라 8개의 클러스터찾음)

### 짝이랑 생각해보기

K최근접 이웃 알고리즘의 장점을 통해 효과적으로 활용할 수 있는 컴퓨터 보안관련 연구주제

이상 탐지를 위한 KNN 기반 시스템: 가능함! KNN은 패턴 인식과 이상 탐지에 유용해서,
네트워크 트래픽이나 사용자 행동에서 비정상적인 패턴을 찾는 데 쓸 수 있음.
네트워크 보안과 사이버 보안 분야에서 큰 도움이 될 것임.

멀웨어 분류와 탐지: 멀웨어 샘플 분석해서 정상 프로그램과 멀웨어 구분하는 KNN 기반 모델을
만들 수 있음. 멀웨어의 변종이나 새로운 형태에 빠르게 대응하는 데 유용할 것임.

사용자 인증 시스템: 생체 인식 데이터(지문, 홍채, 얼굴 인식 등)로 개인 인증하는 KNN 기반
시스템을 만들 수 있음. 다양한 데이터 유형에 적용 가능하니, 더 안전하고 정확한 인증 시스템
구축에 기여할 것임.

피싱 이메일과 웹사이트 탐지: 이메일이나 웹사이트의 특성을 분석해 피싱 시도를 탐지하는
KNN 모델을 개발할 수 있음. 이는 사용자의 개인 정보와 데이터 보호에 중요한 역할을 할 것임.

네트워크 침입 탐지 시스템(NIDS) 개선: 네트워크 트래픽 데이터를 분석해 침입을 탐지하는
KNN 알고리즘을 사용할 수 있음. 네트워크 보안 강화와 비정상 활동 신속 탐지에 도움이 될 것임.