

9장 비지도학습 2부

주요 내용

- 군집/군집화
- k-평균
- DBSCAN
- 가우스 혼합

9.4. DBSCAN

DBSCAN 알고리즘

- 각 샘플에 대해 ε -이웃(반경이 ε 인 원)에 자신을 포함하여 몇 개의 샘플이 `min_samples` 개 이상인 **핵심 샘플**_{core instance}인지 여부 확인
- 핵심 샘플의 ε -이웃, 핵심 샘플의 ε -이웃에 포함된 다른 핵심 샘플의 ε -이웃도 동일 군집으로 구분.
- 핵심 샘플이 아니면서 동시에 ε -이웃에 자신 이외의 다른 샘플이 없다면 그런 샘플은 이상치로 간주.

사이킷런의 DBSCAN 모델

- 두 개의 하이퍼파라미터만 사용
- `eps`: ε -이웃 영역의 반경 지정.
- `min_samples`: 핵심 샘플 지정을 위해 ε -이웃에 포함되어야 하는 최소 샘플 수

예제: 초승달 데이터 군집화

- 2개의 군집으로 구분되어야 하는 초승달 데이터셋에 대한 군집화

```
from sklearn.cluster import DBSCAN
from sklearn.datasets import make_moons

X, y = make_moons(n_samples=1000, noise=0.05)
dbscan = DBSCAN(eps=0.05, min_samples=5)
dbscan.fit(X)
```

군집 라벨

- `labels_` 속성: DBSCAN 모델이 찾은 군집 정보는 0, 1, 2, ... 저장
- `-1`은 이상치를 가리킴.

```
>>> dbscan.labels_[:10]
array([ 0,  2, -1, -1,  1,  0,  0,  0,  2,  5])
```

핵심 샘플

- `core_sample_indices_` 속성: 핵심 샘플들의 인덱스 저장

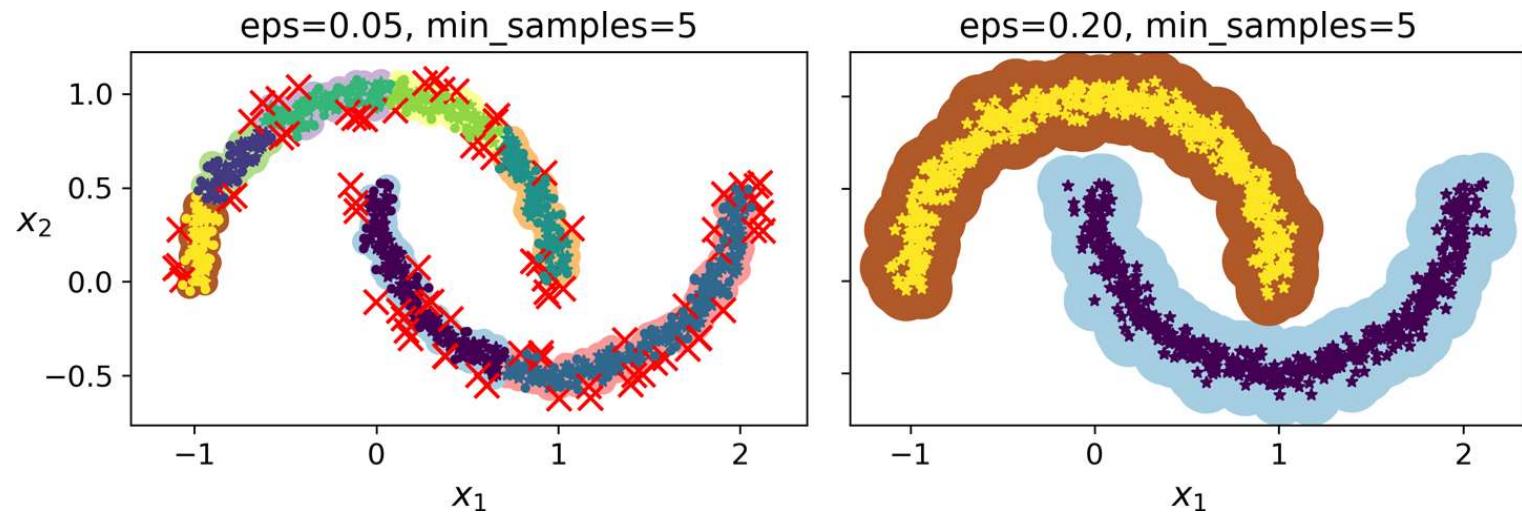
```
>>> dbSCAN.core_sample_indices_[:10]
array([ 0,  4,  5,  6,  7,  8, 10, 11, 12, 13])
```

- `components_` 속성: 핵심 샘플로 구성된 데이터셋 저장

```
>>> dbSCAN.components_
array([[ -0.02137124,   0.40618608],
       [ -0.84192557,   0.53058695],
       [  0.58930337,  -0.32137599],
       ...,
       [  1.66258462,  -0.3079193 ],
       [ -0.94355873,   0.3278936 ],
       [  0.79419406,   0.60777171]])
```

군집화 결과

- ε -이웃의 반경을 0.05로 할 때(왼쪽)와 0.2로 할 때(오른쪽)의 차이



DBSCAN과 예측

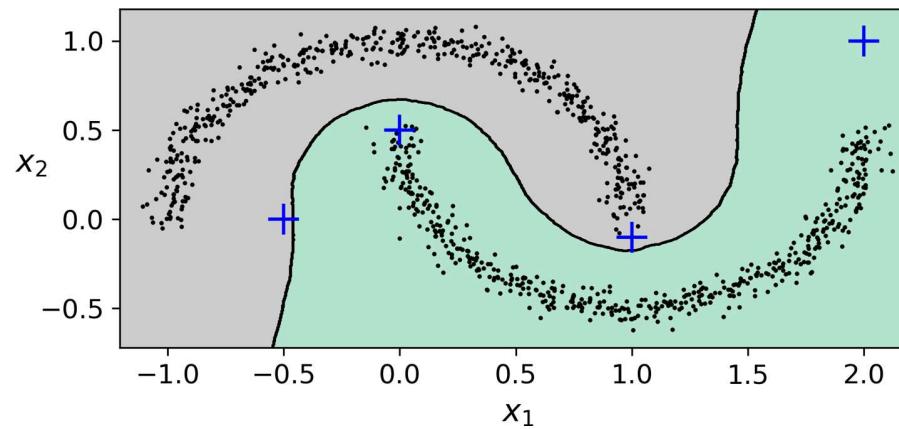
- `fit_predict()` 메서드 지원.
- 반면에 `predict()` 메서드 지원하지 않음.
- 이유: `KNeighborsClassifier` 등 보다 좋은 성능의 분류 알고리즘 활용 가능.
- 아래 코드
 - 핵심 샘플만을 대상으로 `KNeighborsClassifier` 모델 훈련.
 - `dbscan`은 `eps=0.2`로 훈련된 모델을 가리킴.

```
from sklearn.neighbors import KNeighborsClassifier  
  
knn = KNeighborsClassifier(n_neighbors=50)  
knn.fit(dbSCAN.components_, dbSCAN.labels_[dbSCAN.core_sample_indices_])
```

결정 경계

- 아래 그림은 새로운 4개의 샘플에 대한 예측을 보여줌.

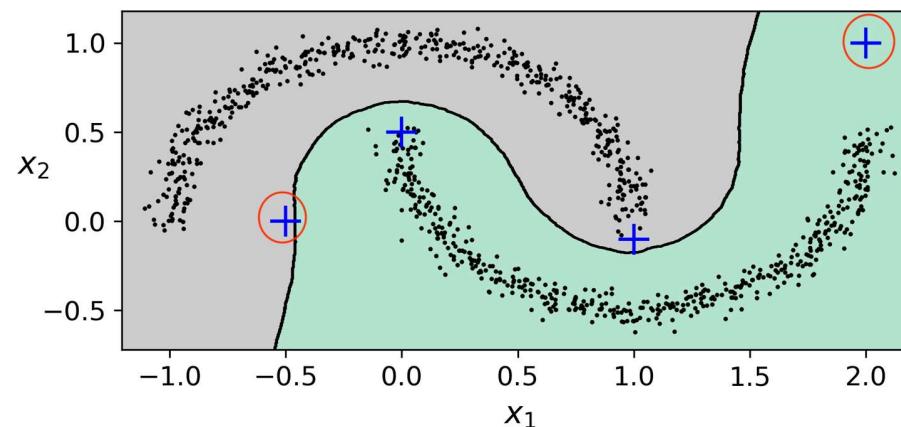
```
>>> X_new = np.array([[-0.5, 0], [0, 0.5], [1, -0.1], [2, 1]])
>>> knn.predict(X_new)
array([1, 0, 1, 0])
>>> knn.predict_proba(X_new)
array([[0.18, 0.82],
       [1., 0.],
       [0.12, 0.88],
       [1., 0.]])
```



이상치 판단

- 두 군집으로부터 일정거리 이상 떨어진 샘플을 이상치로 간주 가능.
- 예를 들어, 양편 끝쪽에 위치한 두 개의 샘플이 이상치로 간주될 수 있음.

```
>>> y_dist, y_pred_idx = knn.kneighbors(X_new, n_neighbors=1)
>>> y_pred = dbscan.labels_[dbscan.core_sample_indices_][y_pred_idx]
>>> y_pred[y_dist > 0.2] = -1
>>> y_pred.ravel()
array([-1,  0,  1, -1])
```



DBSCAN의 장점

- 단 2개의 하이퍼파라미터만을 사용. 하지만 매우 강력한 알고리즘.
- 군집의 모양과 개수에 상관없이 일반적으로 잘 작동.
- 이상치가 있어도 군집 잘 생성.

DBSCAN의 단점

- 군집들의 밀도가 서로 크게 다르거나 두 군집 사이의 영역의 밀도가 충분히 낮지 않으면 서로 다른 군집을 제대로 분리하지 못할 수 있음.
- 알고리즘의 시간복잡도가 $O(m^2n)$ 이기에 대용량 훈련셋을 이용한 훈련은 어려움.

기타 군집 알고리즘

HDBSCAN 모델

- 군집의 밀도가 서로 다른 경우 계층 DBSCAN 군집화를 지원하며 DBSCAN 보다 잘 작동.

사이킷런 제공

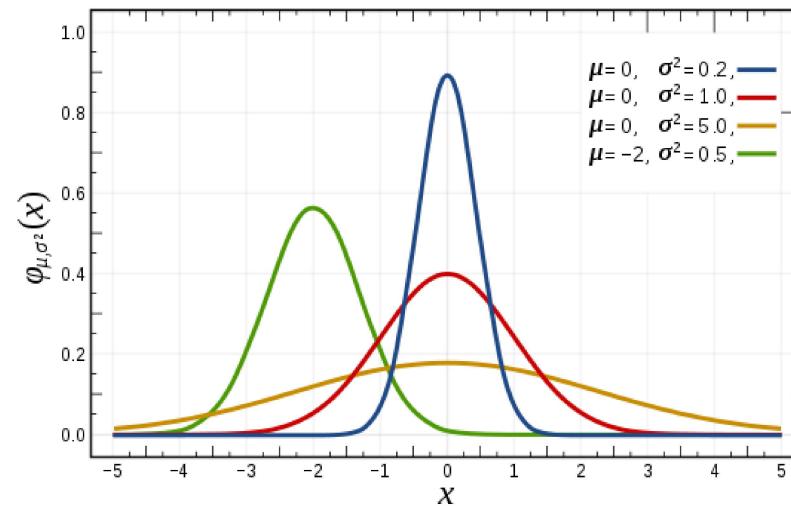
- [Agglomerative clustering](#)
- [BIRCH](#)
- [Mean-shift](#)
- [Affinity propagation](#)
- [Spectral clustering](#)

9.5 가우스 혼합 모델

- 데이터셋이 여러 개의 혼합된 가우스 분포를 따르는 샘플들로 구성되었다고 가정.
- 가우스 분포 = 정규분포

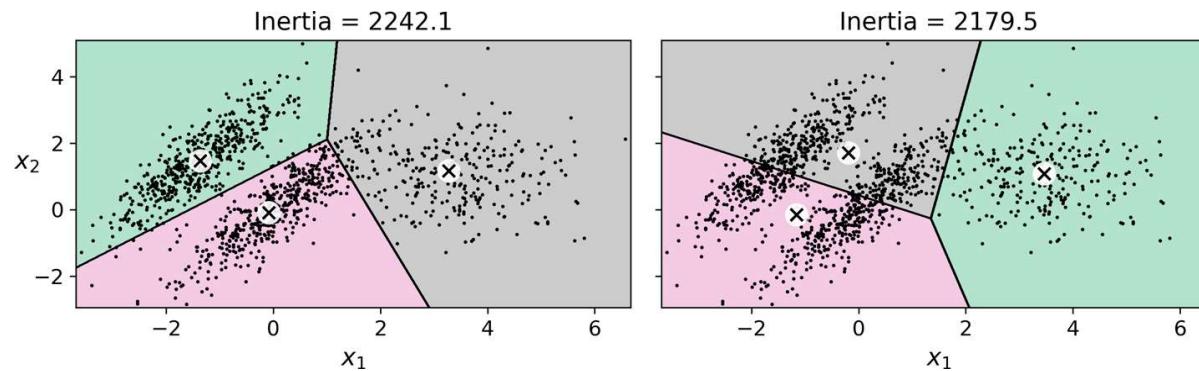
가우스 분포

- 종 모양의 확률밀도함수를 갖는 확률분포



예제

- 아래 그림에 데이터셋: 서로 다른 세 개의 가우스 분포를 따르는 세 개의 데이터 군집들의 혼합으로 구성됨
- 세 개의 가우스 분포의 평균값, 표준편차, 샘플의 개수가 모두 다름
- 각각의 군집을 나타내는 타원의 모양, 위치, 타원 내의 데이터 밀도 등이 모두 다름.

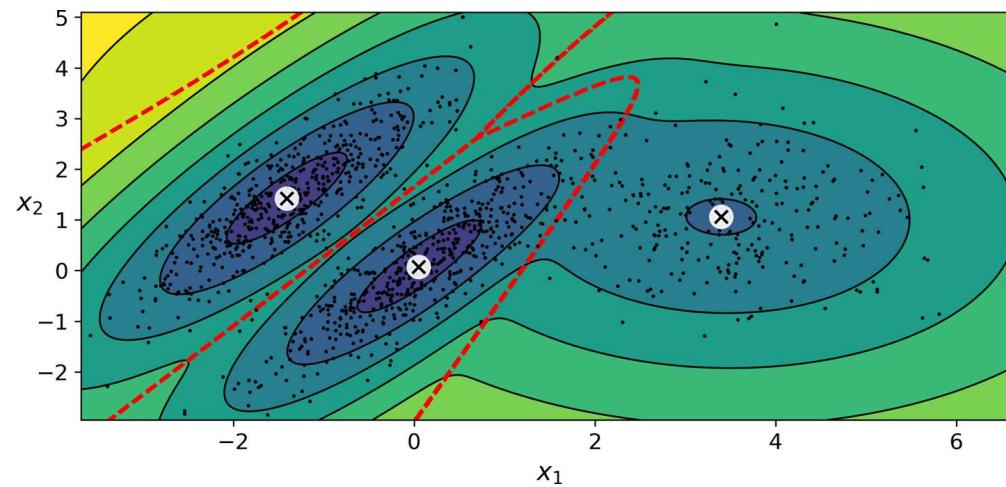


GMM 활용

- 위 데이터셋에 GaussianMixture 모델 적용
- n_components: 군집수 지정
- n_init: 모델 학습 반복 횟수.
 - 파라미터(평균값, 공분산 등)를 무작위로 추정한 후 수렴할 때까지 학습시킴.

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

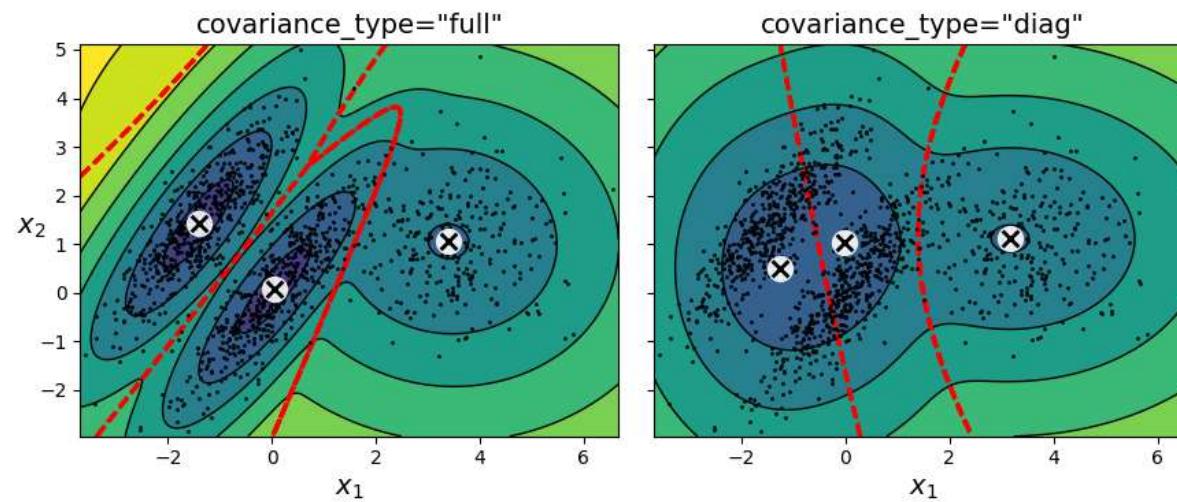
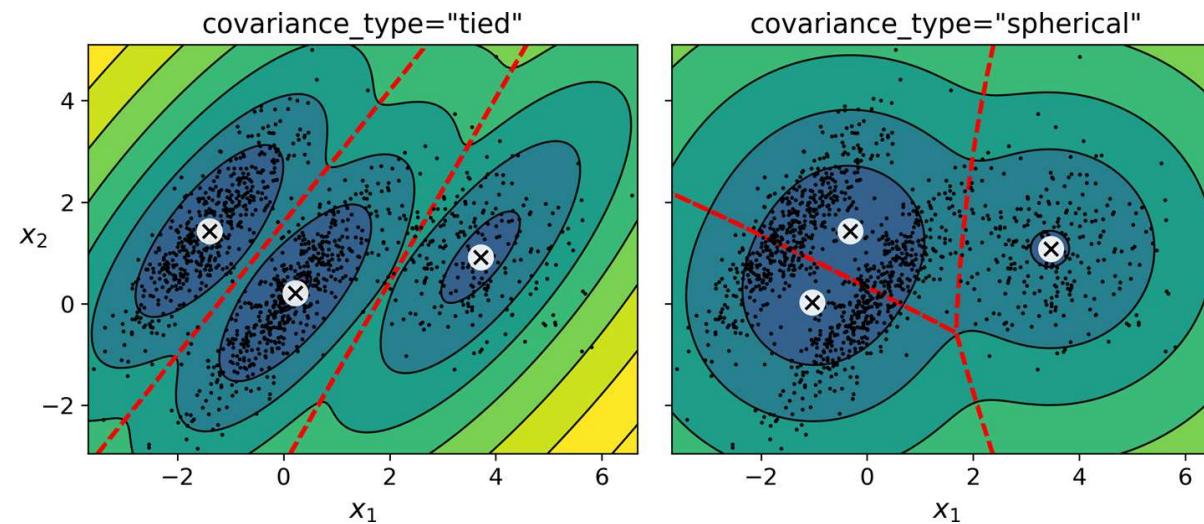
- 아래 그림은 학습된 모델을 보여줌.
 - 군집 평균: **X** 표시
 - 결정 경계: 빨강 파선
 - 밀도 등고선: 진한 파랑색에 가까울 수록 데이터 밀도 높음



GMM 모델 규제

- 특성수가 크거나, 군집수가 많거나, 샘플이 적은 경우 최적 모델 학습 어려움.
- covariance_type 하이퍼파라미터를 이용하여 공분산(covariance)에 규제를 가해서 학습을 도와줄 수 있음.

```
gm_full = GaussianMixture(n_components=3, n_init=10,  
                           covariance_type="full", random_state=42)  
gm_tied = GaussianMixture(n_components=3, n_init=10,  
                           covariance_type="tied", random_state=42)  
gm_spherical = GaussianMixture(n_components=3, n_init=10,  
                               covariance_type="spherical", random_state=42)  
gm_diag = GaussianMixture(n_components=3, n_init=10,  
                           covariance_type="diag", random_state=42)
```



GMM 알고리즘 시간 복잡도

`GaussianMixture` 모델의 훈련 시간은 데이터셋의 크기 m , 차원(특성 수) n , 군집 수 k , 그리고 공분산 규제 방식에 의존한다.

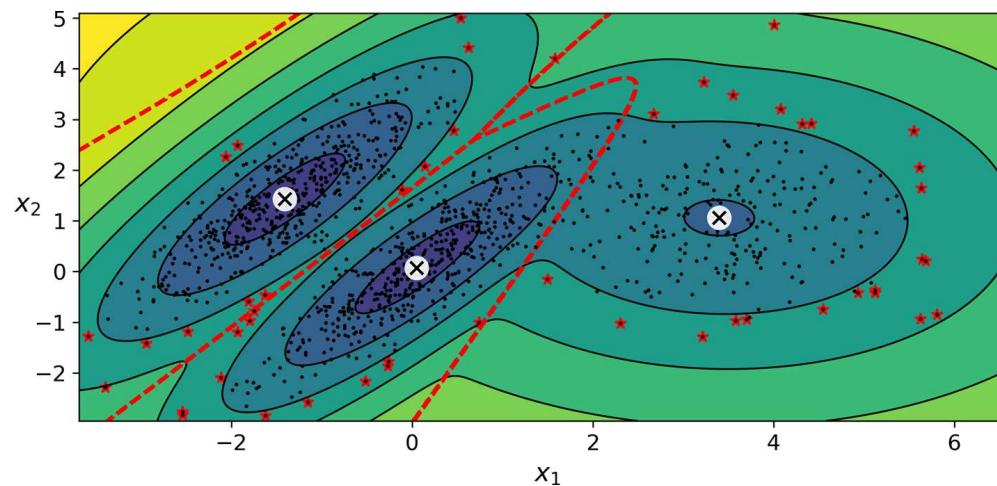
- '`spherical`' 또는 '`diag`' 방식: $O(kmn)$
- '`tied`' 또는 '`full`' 방식: $O(kmn^2 + kn^3)$

이상치 탐지

- score_samples() 메서드는 임의의 위치에서의 확률 밀도의 로그값을 측정.

```
>>> gm.score_samples(X).round(2)
array([-2.61, -3.57, -3.33, ..., -3.51, -4.4 , -3.81])
```

- 밀도가 임곗값보다 낮은 지역에 있는 샘플을 이상치로 간주 가능.



군집수 지정

- k-평균에서 사용했던 관성 또는 실루엣 점수 사용 불가.
 - 군집이 타원형일 때 값이 일정하지 않기 때문.
- 대신에 **이론적 정보 기준** 을 최소화 하는 모델 선택 가능.

이론적 정보 기준

- m : 샘플 수
- p : 모델이 학습해야 할 파라미터 수
- \hat{L} : 모델의 가능성 함수의 최댓값
- 학습해야 할 파라미터가 많을 수록 벌칙이 가해짐.
- 데이터에 잘 학습하는 모델일 수록 보상을 더해줌.
- BIC: Bayesian information criterion

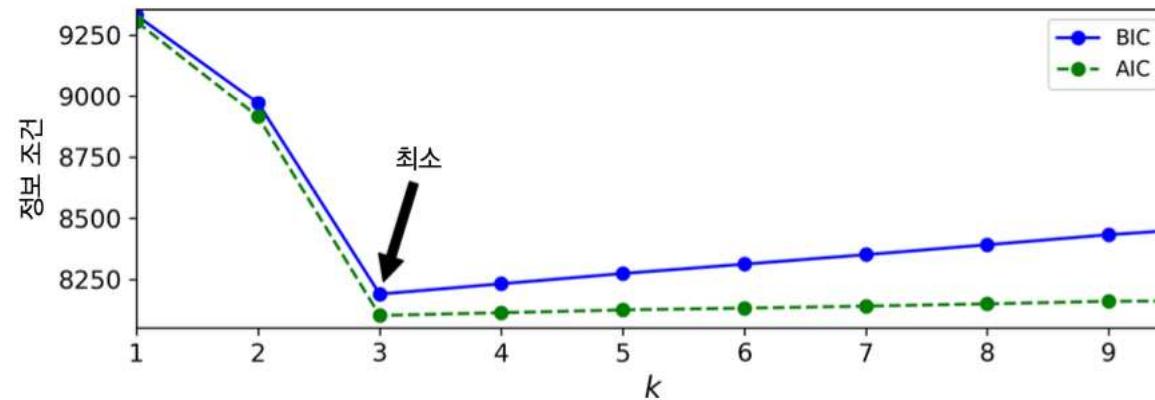
$$\log(m) p - 2 \log(\hat{L})$$

- AIC: Akaike information criterion

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

군집수와 이론적 정보 기준

- 아래 그림은 군집수 k 와 AIC, BIC의 관계를 보여줌.
- $k = 3$ 이 최적으로 보임.



베이즈 가우스 혼합 모델

BayesianGaussianMixture 모델

- 최적의 군집수를 자동으로 찾아줌.
- 단, 최적의 군집수보다 큰 수를 `n_components`에 전달해야 함.
 - 즉, 군집에 대한 최소한의 정보를 알고 있다고 가정.
- 자동으로 불필요한 군집 제거

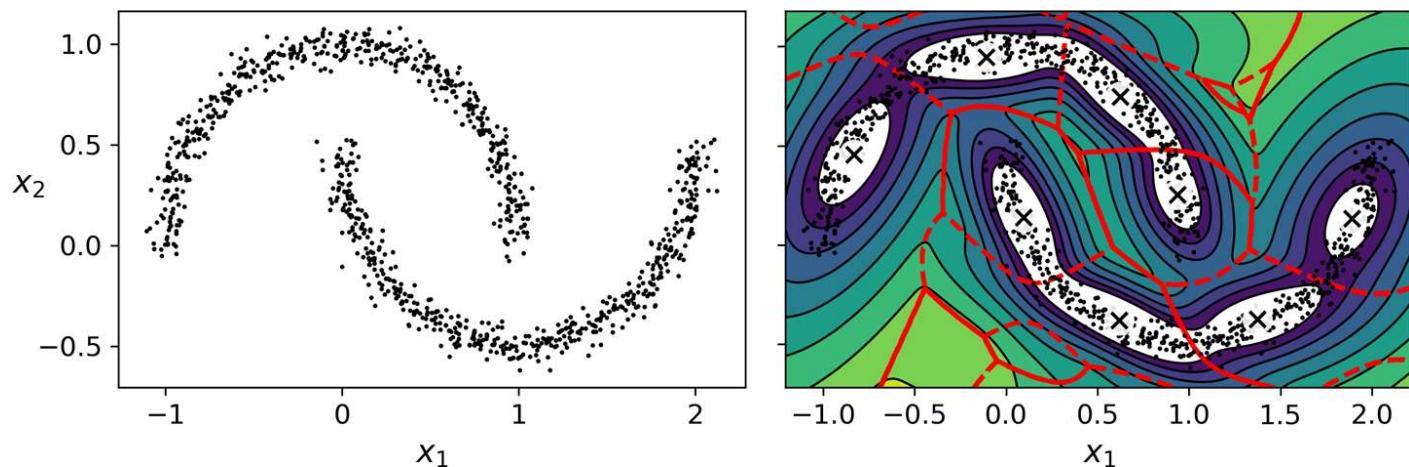
```
from sklearn.mixture import BayesianGaussianMixture  
  
bgm = BayesianGaussianMixture(n_components=10, n_init=10, random_state=42)  
bgm.fit(X)
```

- 결과는 군집수 3개를 사용한 이전 결과와 거의 동일.
- 군집수 확인 가능

```
>>> np.round(bgm.weights_, 2)
array([0.4 , 0.21, 0.4 , 0.  , 0.  , 0.  , 0.  , 0.  , 0.  , 0.  , 0.  ])
```

(베이즈) 가우스 혼합 모델의 장단점

- 타원형 군집에 잘 작동.
- 하지만 다른 모양을 가진 데이터셋에서는 성능 좋지 않음.
- 예제: 달모양 데이터에 적용하는 경우
 - 억지로 타원을 찾으려 시도함.



이상치 탐지와 특이치 탐지를 위한 다른 알고리즘

- Fast-MCD
- Isolation forest
- Local outlier factor (LOF)
- One-class SVM
- `inverse_transform()` 메서드를 지원하는 PCA 등의 차원 축소 알고리즘: 이상치의 경우 재구성 오류가 크다는 성질 이용