Clase 098 — Gaussian Mixture Models

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 70 min.

Clase 098 — Gaussian Mixture Models

Parte: 1 — Machine Learning Clásico · Fuente: Géron, cap. 9. Duración estimada: 70 min.

Objetivo

Entender los Gaussian Mixture Models (GMM) como modelo probabilístico de soft clustering, ajustarlos con el algoritmo EM en scikit-learn, elegir el número de componentes con BIC/AIC, y conocer las variantes (covariance_type, BayesianGaussianMixture) para aplicarlas en clustering, densidad y detección de anomalías.

Resultados de aprendizaje

Al finalizar la clase vas a poder:

1. Explicar qué es un GMM y cómo se diferencia de K-Means (asignación dura vs. probabilística).
2. Ajustar un GaussianMixture con scikit-learn y obtener predict, predict_proba y score_samples.
3. Seleccionar el número óptimo de componentes comparando BIC y AIC en una grilla.
4. Elegir el covariance_type apropiado (full, tied, diag, spherical) según supuestos y tamaño del dataset.
5. Usar BayesianGaussianMixture para que el modelo descarte componentes innecesarios automáticamente.

Temas

- Modelos de mezcla: intuición y fórmula.
- Algoritmo Expectation-Maximization (EM): paso E (responsabilidades) + paso M (actualizar medias, covarianzas, pesos).
- API de sklearn.mixture.GaussianMixture: n_components, covariance_type, n_init, tol.
- Métodos: predict_proba (soft), score_samples (log-densidad), sample (generar datos).
- Selección de modelo con BIC y AIC.
- Variantes de covarianza y su impacto en parámetros / sesgo / varianza.
- Bayesian GMM con prior de Dirichlet: aprende cuántos componentes hacen falta.
- Detección de anomalías por umbral sobre densidad.

Definiciones y características

- Gaussian Mixture Model (GMM): modelo generativo que asume que los datos provienen de una mezcla ponderada de K distribuciones gaussianas multivariadas. Cada punto tiene probabilidad de pertenecer a cada componente.
- Mixture (mezcla): combinación convexa $p(x) = \Sigma\ \pi k \cdot N(x \mid \mu k, \Sigma k)$ con pesos πk que suman 1.
- EM algorithm (Expectation-Maximization): procedimiento iterativo que alterna entre estimar las responsabilidades de cada componente sobre cada punto (E) y reestimar parámetros maximizando la verosimilitud esperada (M). Converge a un óptimo local.
- Soft clustering: asignación probabilística — cada punto recibe un vector de probabilidades de pertenencia, no una etiqueta única. Útil cuando los clusters se solapan.
- covariance_type: controla la forma de las matrices de covarianza.

- full: cada componente tiene su propia matriz completa (más flexible, más parámetros).
- tied: todos comparten la misma matriz.
- diag: matrices diagonales (ejes alineados).
- spherical: una sola varianza escalar por componente (clusters esféricos).
- BIC (Bayesian Information Criterion): $-2 \cdot \log L + k \cdot \log n$. Penaliza fuerte la complejidad; preferí el modelo con BIC mínimo. Tiende a elegir modelos más parsimoniosos.
- AIC (Akaike Information Criterion): $-2 \cdot \log L + 2 \cdot k$. Penaliza menos que BIC; suele elegir modelos algo más complejos.
- BayesianGaussianMixture: variante con prior de Dirichlet sobre los pesos. Si fijás n_components alto, el modelo "apaga" los componentes sobrantes (pesos ≈ 0), evitando elegir K manualmente.

Dataset / recursos

- sklearn.datasets.make_blobs con clusters de varianzas distintas (para ver el aporte vs. K-Means).
- sklearn.datasets.load_iris para validar agrupamiento contra etiquetas reales.
- Opcional: dataset Old Faithful (geyser) — clásico ejemplo de mezcla bimodal.

Ejercicios

1. Ajuste básico: generá make_blobs con 3 centros y cluster_std variable. Ajustá GaussianMixture(n_components=3) y compará predict con las etiquetas reales (ARI).
2. Soft vs. hard: sobre el mismo dataset, mostrá predict_proba de 5 puntos cerca de la frontera. Compará con la asignación dura de K-Means.
3. Selección de K con BIC/AIC: ajustá GMMs con n_components de 1 a 10. Graficá BIC y AIC vs. K e identificá el mínimo.
4. covariance_type: repetí el ajuste con los 4 tipos sobre un dataset con clusters elípticos rotados. Compará BIC y visualizá las elipses de covarianza.
5. Bayesian GMM: ajustá BayesianGaussianMixture(n_components=10, weight_concentration_prior=0.01) sobre datos con 3 clusters reales y mostrá que los pesos efectivos son ≈ 3.

Homework verificable

Sobre load_iris (sin usar la etiqueta para entrenar):

1. Estandarizá las features.
2. Ajustá GMMs con n_components de 1 a 8 y covariance_type en ['full', 'tied', 'diag', 'spherical'].
3. Elegí el (n_components, covariance_type) con BIC mínimo.
4. Calculá el Adjusted Rand Index entre predict del mejor modelo y y real.

Criterio: ARI ≥ 0.85 y el mejor modelo elegido por BIC tiene n_components ∈ {2, 3}.

Errores comunes

1. No estandarizar las features. GMM con covariance_type='spherical' o 'diag' es muy sensible a la escala.
2. Usar n_init=1 (default). EM converge a óptimos locales; subí a n_init=10 para resultados estables.
3. Elegir K mirando solo la log-verosimilitud. Siempre aumenta con K; usá BIC/AIC que penalizan complejidad.

4. covariance_type='full' con pocos datos y alta dimensión: explota los parámetros (K · d · (d+1)/2) y sobreajusta. Bajá a diag o tied.
5. Asumir que predict_proba da incertidumbre calibrada. Da responsabilidades dentro del modelo; si el modelo está mal especificado, las probas pueden ser engañosas.

Preguntas frecuentes

1. ¿GMM o K-Means? K-Means es más rápido y simple, pero asume clusters esféricos de igual tamaño y asigna duro. GMM permite clusters elípticos, tamaños distintos, solapamiento y da probabilidades. Si tus clusters son claramente esféricos y no se solapan, K-Means alcanza.
2. ¿BIC o AIC? Si querés el modelo más parsimonioso y tenés muchos datos, usá BIC. Si priorizás capacidad predictiva y no te molesta un modelo algo más complejo, usá AIC. En la práctica, mostrá los dos y mirá si coinciden.
3. ¿Cómo detecto anomalías con GMM? Calculá score_samples(X) (log-densidad) y marcá como anomalía los puntos con score bajo un percentil (ej. 4%).
4. ¿Sirve para datos de alta dimensión? Con covariance_type='full' no escala bien (muchos parámetros). Reducí dimensión con PCA o usá diag/tied.
5. ¿Qué hace BayesianGaussianMixture que no haga GMM? Aprende cuántos componentes son necesarios: si fijás n_components alto, los sobrantes quedan con peso ≈ 0. Evita la búsqueda en grilla de K.

Referencias

- Géron, Hands-On ML (3ª ed.), cap. 9 § "Gaussian Mixtures".
- scikit-learn — Gaussian Mixture Models.
- scikit-learn — GaussianMixture y BayesianGaussianMixture.
- Bishop, Pattern Recognition and Machine Learning, cap. 9 (EM y mixtures).

Siguiente clase

Clase 099 — Detección de anomalías: Isolation Forest, LOF, One-Class SVM

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb