Machine Learning

Instructors: Mohammadreza A. Dehaqani, Babak n. Arabi, Mostafa Tavassolipour

Amirhossein Bonakdar, Amirreza Zare

Fall 2025

Additional Homework

Clustering, EM, Mixture Models

Question 1: Convergence of K-Means

Consider the standard K-Means objective function for a dataset $X = \{x_1, \ldots, x_m\}$ and a set of k centers $\mu = \{\mu_1, \ldots, \mu_k\}$:

$$F(\mu, C) = \sum_{j=1}^{m} \|x_j - \mu_{C(j)}\|_2^2$$

a) **Assignment Step:** In this step, $C(j) = \arg\min_l \|x_j - \mu_l\|_2^2$.

i. Construct a counter-example or provide a formal argument showing that if ties are broken non-deterministically, the algorithm is not guaranteed to terminate.

ii. Define a specific, deterministic tie-breaking rule. Prove that under your rule, the objective function $F(\mu, C)$ is strictly non-increasing, thereby guaranteeing termination.

b) **Update Step:** During this step, let $S_l = \{x_j \mid C(j) = l\}$ be the set of points assigned to cluster l.

i. What would happen to the standard update $\mu_l = \frac{1}{|S_l|} \sum_{x \in S_l} x$ when $S_l = \emptyset$.

ii. Propose an algorithmic policy to handle the case where $S_l = \emptyset$.

iii. Prove whether your policy preserves the monotonicity property $F(\mu^{(t+1)}, C^{(t+1)}) \leq F(\mu^{(t)}, C^{(t)})$.

Question 2: Regularized K-Means and MAP Estimation

Consider the regularized K-Means objective function with a penalty parameter $\lambda > 0$:

$$J(\mu, C) = \sum_{j=1}^{m} \|x_j - \mu_{C(j)}\|_2^2 + \lambda \sum_{k=1}^{K} \|\mu_k\|_2^2$$

a) **Update Step:** Let the assignments C be fixed.

i. Derive the closed-form expression for the optimal centroid μ_k^* in terms of the cluster size n_k, the empirical mean $\bar{x}_k$, and λ.

ii. Interpret the result: Specifically, describe the behavior of μ_k^* as the ratio λ / n_k increases.

b) **Probabilistic Mapping:** Assume the negative log-likelihood of the data is scaled as $\frac{1}{2\sigma^2} \sum_{j=1}^{m} \|x_j - \mu_{C(j)}\|^2$.

i. Prove that minimizing $J(\mu, C)$ is formally equivalent to Maximum A Posteriori (MAP) estimation.

ii. Identify the specific family and parameters (mean and variance) of the prior distribution $p(\mu_k)$ implied by the regularization term.

iii. Express the variance of this prior explicitly in terms of λ and σ^2.

Question 3: The K-Means Objective Identity

Recall the k-means clustering problem with data $x_1, \ldots, x_m$. Our goal was to find k clusters denoted by $C_1, \ldots, C_k \subseteq [m]$ such that $\cup_{i=1}^{k} C_i = [m]$ (cover all data points) and for all $i \neq j, C_i \cap C_j = \phi$ (and are disjoint).

To measure the "goodness" of the cluster we defined

$$Z(C_1, \ldots, C_k) = \sum_{l=1}^{k} \frac{1}{2|C_l|} \sum_{i,j \in C_l} \|x_i - x_j\|_2^2.$$

Show that we can equivalently express Z as

$$Z(C_1, \ldots, C_k) = \sum_{l=1}^{k} \sum_{i \in C_l} \|x_i - \mu_l\|_2^2$$

where $\mu_l = \frac{1}{|C_l|} \sum_{i \in C_l} x_i$ is the centroid of cluster l.

Question 4: Gaussian Mixture Models (GMMs)

A. Two datasets ("MOONS" and "STRIPES") were each clustered by 3 different methods: K-means clustering, Gaussian-Mixture-Models and Hierarchical Clustering (single link). The results are shown in Figure 1.

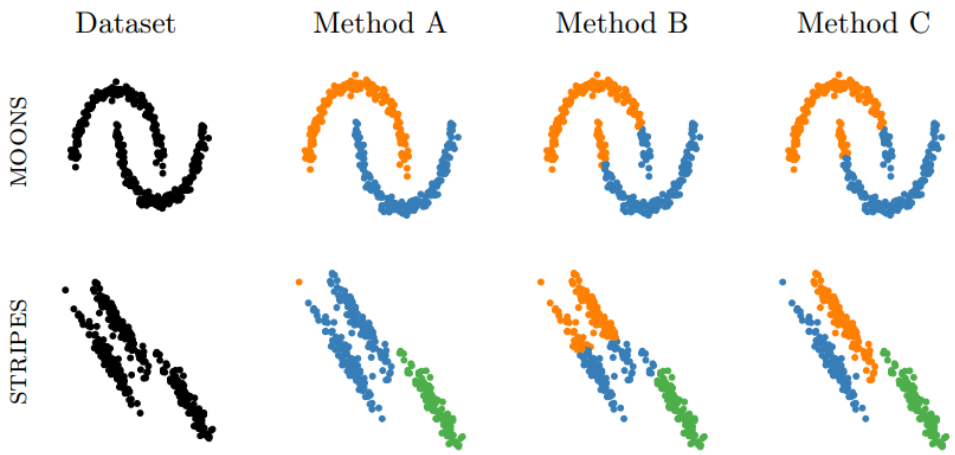

Figure 1: Different Clustering Methods

Decide which method corresponds to A, B and C. Explain your decision.

B. Given the data from Table 1, and the initial configuration $\pi_1, \pi_2 = \frac{1}{2}$, $\mu_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$, $\Sigma_1, \Sigma_2 = I$, perform 1 iteration of the (soft partition) EM algorithm to fit a GMM. Which cluster would $x^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ belong to according the initial/final parameters?

x_1	x_2
0	0
0	1
-1	2
2	0
3	0
4	-1

Table 1: Dataset for EM algorithm

Question 5: EM

Consider a Bernoulli mixture model with mixing coefficients $\boldsymbol{\pi}$ and parameter vectors $\boldsymbol{\mu}_k$. We introduce a prior distribution $p(\boldsymbol{\mu}_k|a_k, b_k)$ over the parameters $\boldsymbol{\mu}_k$, defined by the Beta distribution:

$$\text{Beta}(\mu|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$$

Additionally, we introduce a Dirichlet prior $p(\boldsymbol{\pi}|\boldsymbol{\alpha})$ over the mixing coefficients $\boldsymbol{\pi}$:

$$\text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\hat{\alpha})}{\Gamma(\alpha_1)\dots\Gamma(\alpha_K)}\prod_{k=1}^{K}\pi_k^{\alpha_k-1}$$

Derive the EM algorithm for maximizing the posterior probability $p(\boldsymbol{\mu}, \boldsymbol{\pi}|\mathbf{X})$ by performing the following steps:

(a) Formulate the expected complete-data log-posterior function.

(b) Derive the E-step (Expectation step).

(c) Derive the M-step (Maximization step) for the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$.

Question 6: EM & GMM (simulation)

Part 1: The Geometry of Clustering Objective

1. Generate a synthetic dataset using `make_blobs`:

```
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=1500, centers=3,
                  cluster_std=1.0, random_state=170)
```

2. Apply this specific linear transformation to stretch the data:

```
transformation = [[0.6, -0.6], [-0.4, 0.8]]
X_aniso = np.dot(X, transformation)
```

3. Fit a standard `KMeans` model with `n_clusters=3` and `random_state=170`.
4. Fit a `GaussianMixture` (GMM) with `n_components=3`, `covariance_type='full'`, and `random_state=170`.
5. Plot the cluster assignments for both to compare how they handle the shapes.

Analysis Question
Why does the K-Means algorithm (mathematically) fail to capture the elongated shape, while GMM succeeds?

Part 2: The Optimization Landscape & Initialization Objective

1. Create a "difficult" dataset using `make_blobs` with **exact parameters**:
   - `n_samples=500`
   - `centers=4`
   - `cluster_std=2.5`
   - `random_state=42`
2. Run the GMM algorithm 50 times on this dataset.
3. **IMPORTANT**: For each run, use `n_components=4`, `init_params='random'`, `n_init=1`, and set `random_state=i` (where i is the loop index 0..49) to ensure reproducible random starts.

4. Store the final log-likelihood for each run and plot a histogram.

Analysis Questions

1. The GMM log-likelihood is a **non-convex** function. Explain why this property causes the histogram to show multiple peaks, and why initialization is critical.
2. Propose a practical solution to mitigate this problem.

Part 3: Singularity (The Infinite Likelihood Problem) Objective

1. Create a dataset by taking the FIRST 100 points from Task 1 (`X[:100]`) and appending a single outlier point `[10, 10]`.
2. Initialize a `GaussianMixture` with `n_components=2`, `reg_covar=0` (no regularization), `init_params='random'`, and `random_state=42`.
3. Manually force the starting means to be:

```
gmm_sing.means_init = np.array([X[0], [15, 15]])
```

4. Attempt to fit the model and catch the resulting error.

Analysis Questions

1. Using the Gaussian PDF formula, explain mathematically why the likelihood becomes infinite when a component collapses onto a single point (i.e., $\sigma \to 0$ while $x = \mu$).
2. How does `reg_covar` solve this problem?

Part 4: Covariance Constraints (Bias-Variance Tradeoff) Objective

1. Use the anisotropic (stretched) dataset `X_aniso` from Task 1.
2. Fit 4 separate GMM models, each with `n_components=3` and `random_state=170`.
3. Use a different `covariance_type` for each: `spherical`, `diag`, `tied`, `full`.
4. Plot the resulting cluster assignments and ellipses for each.

Analysis Question
Based on the resulting plot, which covariance type is the most flexible? Why do we not just always use the most flexible one?

Part 5: Generative Modeling (Creating New Data) Objective

1. Load the `digits` dataset (8x8 images).
2. Fit a `GaussianMixture` to the digit data with **exact parameters**:
   - `n_components=30`
   - `covariance_type='full'`
   - `random_state=42`
3. Use the `.sample(10)` method to generate 10 *new* digit samples.
4. Visualize them.

Analysis Questions

1. What is the difference between a **Generative Model** (like GMM) and a **Discriminative Model** (like Logistic Regression) regarding what they learn about the data?
2. Could a standard K-Means algorithm generate new, realistic variations of a digit? Why or why not?

Part 6: Bayesian GMM **Background**: In standard GMM, the mixing weights $\pi = (\pi_1, \ldots, \pi_K)$ are point estimates. In **Bayesian GMM**, we place a **Dirichlet prior** on π.

Gamma Function:

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt, \quad \Gamma(n) = (n-1)! \text{ for } n \in \mathbb{Z}^+$$

Dirichlet Distribution:

$$p(\pi|\alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}, \quad \sum_k \pi_k = 1$$

When $\alpha_1 = \cdots = \alpha_K = \alpha$ (symmetric):

- $\alpha < 1$: Prefers sparse vectors (mass at corners of simplex).
- $\alpha = 1$: Uniform over the simplex.
- $\alpha > 1$: Prefers dense, balanced vectors (mass at center).

Objective

1. Generate data: `make_blobs(n_samples=500, centers=3, random_state=42)`.
2. Fit `BayesianGaussianMixture(n_components=10, weight_concentration_prior=1e-3, random_state=42)`.
3. Plot `bgmm.weights_`.

Analysis Questions

1. We set `weight_concentration_prior=1e-3`, meaning $\alpha = 10^{-3} < 1$. According to the Dirichlet properties above, what type of π vectors does this prior favor?
2. Based on your answer to (1), explain why most of the 10 components have weight ≈ 0.