БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет прикладной математики и информатики
Кафедра дискретной математики и алгоритмики

КУРСОВОЙ ПРОЕКТ

Тема: «Исследование и практическая реализация EM-алгоритма для задачи кластеризации на основе моделей смеси гауссовых распределений (Gaussian Mixture Models)»

Студент 3 курса 3 группы
Санчук Сергей Александрович

Руководитель:
Буславский Александр Андреевич

Минск 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Актуальность темы. Задачи кластерного анализа занимают центральное место в современной теории машинного обучения и интеллектуального анализа данных (Data Mining). Необходимость выявления скрытой структуры в неразмеченных данных возникает в самых разных областях: от сегментации изображений и сжатия информации до биоинформатики и финансового скоринга.

Наиболее распространенным методом кластеризации является алгоритм K-means (метод k-средних). Несмотря на вычислительную эффективность, данный подход обладает существенными ограничениями: он предполагает сферическую форму кластеров и использует «жесткое» (hard) разбиение, при котором каждый объект однозначно приписывается одному кластеру. Однако реальные данные часто имеют сложную геометрическую структуру, могут обладать анизотропией (вытянутостью вдоль определенных направлений) и существенно перекрываться. В таких условиях использование метрических алгоритмов приводит к значительным ошибкам первого и второго рода.

Альтернативой выступает вероятностное моделирование, в частности использование моделей смеси гауссовых распределений (Gaussian Mixture Models, GMM). Данный подход позволяет аппроксимировать произвольную плотность распределения данных и реализует принцип «мягкой» (soft) кластеризации, возвращая вероятность принадлежности объекта к каждому из кластеров. Ключевой проблемой при использовании GMM является оценка параметров модели в условиях неполной информации (отсутствия меток классов). Стандартный метод максимального правдоподобия (MLE) в данном случае не имеет аналитического решения, что обуславливает необходимость применения итеративных методов оптимизации, основным из которых является EM-алгоритм (Expectation-Maximization).

Изучение математического аппарата EM-алгоритма и его программная реализация «с нуля» позволяют глубоко понять принципы работы с латентными переменными, проблемы численной стабильности и методы оптимизации функций правдоподобия, что является необходимым базисом для специалиста в области прикладной математики.

Объект исследования — вероятностные модели смеси распределений,

используемые в задачах машинного обучения без учителя.

Предмет исследования — EM-алгоритм как итерационный метод нахождения оценок максимального правдоподобия в вероятностных моделях со скрытыми переменными.

Цель работы — теоретическое исследование математического обоснования EM-алгоритма и разработка его программной реализации для задачи кластеризации на основе смеси гауссовых распределений с ковариационной матрицей общего вида.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести теоретический анализ EM-алгоритма: изучить его вывод через вариационную нижнюю оценку (ELBO), обосновать шаги E (Expectation) и M (Maximization), а также рассмотреть условия сходимости последовательности оценок правдоподобия.

2. Исследовать свойства модели GMM: проанализировать геометрическую интерпретацию параметров многомерного нормального распределения (вектор математического ожидания, ковариационная матрица) и их влияние на форму и ориентацию кластеров.

3. Выполнить сравнительный анализ GMM и алгоритма K-means, выявив теоретические взаимосвязи (K-means как частный случай GMM) и различия в гибкости моделирования данных.

4. Разработать программную реализацию алгоритма на языке Python с использованием библиотеки NumPy. Реализация должна поддерживать работу с ковариационными матрицами общего вида (Full Covariance) и включать механизмы обеспечения численной стабильности (Log-Sum-Exp trick).

5. Провести экспериментальное исследование разработанного алгоритма на синтетических данных (включая анизотропные кластеры и данные различной плотности) и реальных наборах данных (Iris Fisher), оценив качество кластеризации с помощью метрики ARI (Adjusted Rand Index) и визуального анализа.

Методы исследования. В работе использованы методы теории вероятностей, математической статистики (метод максимального правдоподобия, байесовский вывод), линейной алгебры (матричное дифференцирование, спектральное разложение) и методы оптимизации.

Практическая значимость. Разработанное программное обеспечение представляет собой гибкий инструмент для вероятностного анализа данных, способный, в отличие от стандартных метрических алгоритмов, корректно выделять кластеры сложной эллиптической формы и оценивать неопределенность принадлежности объектов к классам.

ГЛАВА 1
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ МОДЕЛИРОВАНИЯ СМЕСЕЙ РАСПРЕДЕЛЕНИЙ

В данной главе рассматривается математическая модель смеси гауссовых распределений (Gaussian Mixture Model, GMM), вводятся скрытые переменные для описания генеративной природы данных и формулируется задача оценки параметров методом максимального правдоподобия.

1.1 Математическая модель GMM

Модель смеси гауссовых распределений основывается на предположении, что наблюдаемые данные $X = \{x_1, \ldots, x_N\}$, где $x_n \in \mathbb{R}^D$, порождены взвешенной суммой K компонент, каждая из которых имеет нормальное распределение.

Функция плотности вероятности для произвольного вектора x в модели GMM определяется следующим образом:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k), \tag{1}$$

где:

- K — количество компонент (кластеров) смеси;
- π_k — априорный вес (коэффициент смешивания) k-й компоненты, удо-

влетворяющий условиям нормировки:

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^{K} \pi_k = 1; \tag{2}$$

- $\mathcal{N}(x|\mu_k, \Sigma_k)$ — плотность D-мерного нормального распределения k-й компоненты;
- $\theta = \{\pi_1 \dots \pi_K, \mu_1 \dots \mu_K, \Sigma_1 \dots \Sigma_K\}$ — полный набор параметров модели.

Многомерное нормальное распределение для k-й компоненты задается формулой:

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right), \tag{3}$$

где $\mu_k \in \mathbb{R}^D$ — вектор математического ожидания (центр кластера), а $\Sigma_k \in \mathbb{R}^{D \times D}$ — ковариационная матрица.

1.1.1 Геометрическая интерпретация параметров

Геометрия кластера в модели GMM полностью определяется параметрами μ_k и Σ_k. Линии уровня плотности вероятности (изолинии) многомерного гауссова распределения представляют собой эллипсоиды (в двумерном случае — эллипсы).

1. Вектор средних μ_k определяет положение центра симметрии эллипсоида в пространстве признаков.
2. Ковариационная матрица Σ_k определяет форму, размер и ориентацию эллипсоида.

В рамках данной работы рассматривается наиболее общий случай — полная ковариационная матрица (Full Covariance). Матрица Σ_k является симметричной и положительно определенной. Ее свойства можно проанализировать через спектральное разложение (eigendecomposition):

$$\Sigma_k = U_k \Lambda_k U_k^T = \sum_{j=1}^{D} \lambda_{kj} u_{kj} u_{kj}^T, \tag{4}$$

где λ_{kj} и u_{kj} — собственные числа и соответствующие им ортонормированные собственные векторы матрицы Σ_k.

- Собственные векторы u_{kj} задают направления главных осей эллипсоида рассеяния. Наличие ненулевых недиагональных элементов в Σ_k означает наличие корреляции между признаками, что геометрически выражается в повороте эллипсоида относительно осей координат.
- Корни из собственных чисел $\sqrt{\lambda_{kj}}$ пропорциональны длине полуосей эллипсоида вдоль соответствующих направлений.

Использование полной ковариационной матрицы позволяет моделировать данные с анизотропной структурой, где кластеры могут быть вытянуты и наклонены под произвольным углом.

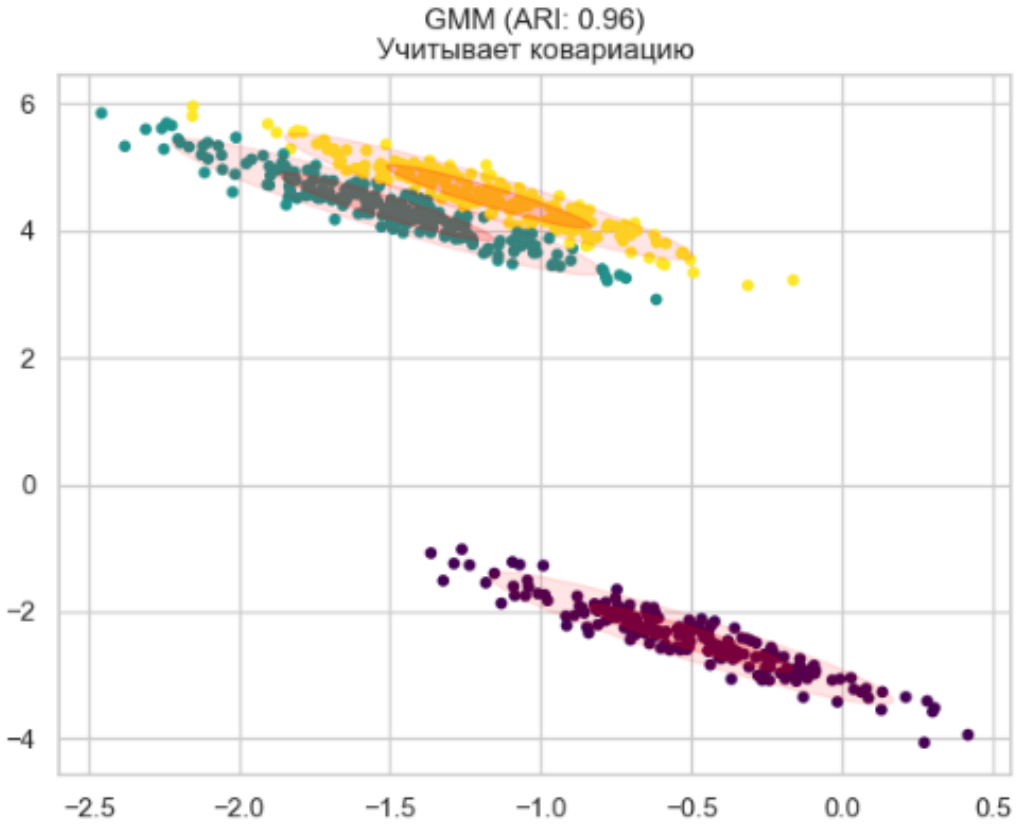

Рис. 1 – Схематичное изображение GMM. Показаны несколько эллипсов разной ориентации, определяемые ковариационными матрицами.

1.2 Байесовский подход к скрытым переменным

Для формализации задачи кластеризации удобно рассматривать процесс генерации данных через введение латентных (скрытых) переменных.

Пусть z — дискретная случайная величина, представляющая собой K-мерный бинарный вектор, в котором только один элемент равен 1, а остальные — 0 (one-hot encoding). Если $z_k = 1$, это означает, что объект порожден k-й компонентой смеси.

Маргинальное распределение z задается априорными весами:

$$p(z_k = 1) = \pi_k.$$

Условное распределение наблюдаемого вектора x при известном z является нормальным:

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k).$$

Совместное распределение наблюдаемых и скрытых переменных имеет вид $p(x,z) = p(x|z)p(z)$. Тогда маргинальное распределение $p(x)$, полученное суммированием по всем возможным состояниям z, возвращает нас к исходной формуле смеси:

$$p(x) = \sum_z p(x|z)p(z) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k). \tag{5}$$

Одной из ключевых задач кластеризации является определение апостериорной вероятности того, что наблюдаемый объект x_n принадлежит кластеру k. Эту величину называют «ответственностью» (responsibility) k-й компоненты за n-й объект и обозначают $\gamma(z_{nk})$. Согласно теореме Байеса:

$$\gamma(z_{nk}) \equiv p(z_k = 1|x_n) = \frac{p(z_k = 1)p(x_n|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x_n|z_j = 1)} = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}. \tag{6}$$

Величина $\gamma(z_{nk})$ принимает значения в интервале $[0, 1]$ и реализует принцип «мягкой» кластеризации: вместо жесткого присвоения метки алгоритм оценивает степень уверенности в принадлежности объекта к каждому из кластеров.

1.3 Проблема оценки параметров

Для настройки параметров модели θ используется метод максимального правдоподобия (Maximum Likelihood Estimation, MLE). Функция правдоподобия для выборки X определяется как произведение плотностей вероят-

ностей для всех независимых наблюдений:

$$L(\theta|X) = \prod_{n=1}^{N} p(x_n|\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k). \tag{7}$$

На практике максимизируют логарифм функции правдоподобия (Log-Likelihood), так как логарифм является монотонно возрастающей функцией и упрощает работу с экспоненциальным семейством распределений:

$$\ln L(\theta|X) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right) \to \max_{\theta}. \tag{8}$$

Невозможность прямого аналитического решения

Прямая максимизация данного выражения сталкивается с серьезными трудностями. Если попытаться найти экстремум, приравняв производные по параметрам к нулю, то наличие суммы под знаком логарифма ($\ln \sum \ldots$) не позволяет получить аналитическое выражение для параметров в замкнутом виде. Например, уравнение для μ_k будет зависеть от апостериорных вероятностей $\gamma(z_{nk})$, которые, в свою очередь, сложным образом зависят от всех параметров μ, Σ и π. Это приводит к системе нелинейных уравнений, не имеющей явного решения.

Данная проблема решается путем введения скрытых переменных Z и перехода к итерационной процедуре оптимизации, известной как EM-алгоритм, математическое обоснование которого приводится в следующей главе.

ГЛАВА 2
МАТЕМАТИЧЕСКОЕ ОБОСНОВАНИЕ EM-АЛГОРИТМА

В данной главе приводится формальный вывод алгоритма Expectation-Maximization (EM) как метода итеративной максимизации правдоподобия для вероятностных моделей со скрытыми переменными. Обосновывается введение вариационной нижней границы (ELBO) и детально рассматриваются шаги алгоритма для случая смеси гауссовых распределений (GMM).

2.1 Вариационная нижняя граница (ELBO) и декомпозиция правдоподобия

Основная сложность максимизации логарифма правдоподобия $\ln p(X|\theta)$ заключается в наличии скрытых переменных Z. Для решения этой проблемы преобразуем целевую функцию, введя произвольное распределение $q(Z)$ на множестве скрытых переменных.

Проведем формальный вывод соотношения, связывающего правдоподобие, вариационную нижнюю границу (ELBO) и дивергенцию Кульбака-Лейблера.

Запишем логарифм правдоподобия, умножив его на единицу в виде суммы вероятностей $q(Z)$ (условие нормировки $\sum_Z q(Z) = 1$):

$$\ln p(X|\theta) = \ln p(X|\theta) \cdot \sum_Z q(Z) = \sum_Z q(Z) \ln p(X|\theta). \tag{9}$$

Воспользуемся формулой условной вероятности $p(X,Z|\theta) = p(Z|X,\theta)p(X|\theta)$, откуда следует $p(X|\theta) = \frac{p(X,Z|\theta)}{p(Z|X,\theta)}$. Подставим это выражение под знак логарифма:

$$\ln p(X|\theta) = \sum_Z q(Z) \ln \left(\frac{p(X,Z|\theta)}{p(Z|X,\theta)} \right). \tag{10}$$

Умножим и разделим выражение под логарифмом на $q(Z)$:

$$\ln p(X|\theta) = \sum_Z q(Z) \ln \left(\frac{p(X,Z|\theta)}{q(Z)} \cdot \frac{q(Z)}{p(Z|X,\theta)} \right). \tag{11}$$

Используя свойство логарифма произведения ($\ln(ab) = \ln a + \ln b$), разобьем сумму на два слагаемых:

$$\ln p(X|\theta) = \underbrace{\sum_Z q(Z) \ln \left(\frac{p(X,Z|\theta)}{q(Z)} \right)}_{\mathscr{L}(q,\theta)} + \underbrace{\sum_Z q(Z) \ln \left(\frac{q(Z)}{p(Z|X,\theta)} \right)}_{KL(q||p)}. \tag{12}$$

Второе слагаемое с обратным знаком представляет собой определение дивергенции Кульбака-Лейблера (KL-дивергенции) между распределением

$q(Z)$ и истинным апостериорным распределением $p(Z|X,\theta)$:

$$KL(q||p) = -\sum_Z q(Z)\ln\left(\frac{p(Z|X,\theta)}{q(Z)}\right). \tag{13}$$

Таким образом, мы получили фундаментальное тождество декомпозиции правдоподобия:

$$\ln p(X|\theta) = \mathscr{L}(q,\theta) + KL(q||p). \tag{14}$$

Где:

- $\mathscr{L}(q,\theta)$ (ELBO — Evidence Lower Bound) — вариационная нижняя граница. Она зависит как от параметров модели θ, так и от выбранного распределения $q(Z)$.
- $KL(q||p)$ — мера расхождения между приближенным распределением $q(Z)$ и истинным апостериорным распределением скрытых переменных. Согласно неравенству Гиббса, $KL(q||p) \geq 0$, причем равенство достигается тогда и только тогда, когда $q(Z) = p(Z|X,\theta)$.

Из неотрицательности KL-дивергенции следует, что $\mathscr{L}(q,\theta)$ является нижней оценкой логарифма правдоподобия:

$$\ln p(X|\theta) \geq \mathscr{L}(q,\theta). \tag{15}$$

Этот вывод обосновывает стратегию EM-алгоритма: вместо сложной прямой максимизации $\ln p(X|\theta)$ мы итеративно максимизируем нижнюю границу $\mathscr{L}(q,\theta)$, последовательно приближая ее к истинному правдоподобию.

2.2 Итерационный процесс оптимизации

Алгоритм состоит из двух шагов, повторяющихся до сходимости: E-шаг (ожидание) и M-шаг (максимизация).

2.2.1 E-шаг (Expectation)

На этом шаге параметры модели θ фиксируются (используются значения $\theta^{(t)}$ с предыдущей итерации). Задача состоит в максимизации нижней границы $\mathscr{L}(q,\theta^{(t)})$ относительно распределения $q(Z)$.

Из уравнения декомпозиции видно, что $\ln p(X|\theta^{(t)})$ не зависит от $q(Z)$. Следовательно, максимизация ELBO эквивалентна минимизации $KL(q||p)$. Минимум достигается при:

$$q^{(t+1)}(Z) = p(Z|X, \theta^{(t)}). \tag{16}$$

Для модели GMM это означает вычисление апостериорных вероятностей («ответственностей») для каждого объекта x_n и каждой компоненты k:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}. \tag{17}$$

2.2.2 M-шаг (Maximization)

На этом шаге фиксируется распределение $q(Z)$ (то есть значения γ_{nk}) и производится максимизация ожидания полного правдоподобия по параметрам θ:

$$Q(\theta, \theta^{(t)}) = E_Z[\ln p(X, Z|\theta)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \ln(\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)) \to \max_{\theta}. \tag{18}$$

Раскроем логарифм плотности нормального распределения:

$$Q(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_{nk} \left(\ln \pi_k - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k) \right) + C. \tag{19}$$

1. Обновление средних μ_k

Продифференцируем Q по μ_k и приравняем к нулю:

$$\frac{\partial Q}{\partial \mu_k} = \sum_{n=1}^{N} \gamma_{nk} \Sigma_k^{-1} (x_n - \mu_k) = 0. \tag{20}$$

Умножая на Σ_k, получаем:

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} x_n, \tag{21}$$

где $N_k = \sum_{n=1}^{N} \gamma_{nk}$ — эффективное число объектов в кластере k.

2. Обновление ковариационной матрицы Σ_k

Для вывода формулы обновления Σ_k воспользуемся свойствами следа матрицы (Tr). Скалярное произведение в квадратичной форме можно записать через след:

$$(x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k) = Tr\left((x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k)\right) = Tr\left(\Sigma_k^{-1}(x_n-\mu_k)(x_n-\mu_k)^T\right). \tag{22}$$

Тогда часть функции Q, зависящая от Σ_k, примет вид:

$$Q_{\Sigma_k} = -\frac{1}{2}\sum_{n=1}^{N}\gamma_{nk}\left(\ln|\Sigma_k| + Tr\left(\Sigma_k^{-1}(x_n-\mu_k)(x_n-\mu_k)^T\right)\right). \tag{23}$$

Введем матрицу разброса $S_k = \sum_{n=1}^{N}\gamma_{nk}(x_n-\mu_k)(x_n-\mu_k)^T$. Учитывая, что $\ln|\Sigma_k| = -\ln|\Sigma_k^{-1}|$, перепишем выражение относительно обратной матрицы $\Lambda_k = \Sigma_k^{-1}$:

$$Q_{\Lambda_k} = \frac{1}{2}N_k\ln|\Lambda_k| - \frac{1}{2}Tr(\Lambda_k S_k). \tag{24}$$

Воспользуемся формулами матричного дифференцирования:

$$\frac{\partial\ln|\Lambda|}{\partial\Lambda} = \Lambda^{-T} = \Sigma^T = \Sigma \quad \text{(в силу симметрии)}, \tag{25}$$

$$\frac{\partial Tr(\Lambda S)}{\partial\Lambda} = S^T = S. \tag{26}$$

Дифференцируем Q по Λ_k:

$$\frac{\partial Q}{\partial\Lambda_k} = \frac{1}{2}N_k\Sigma_k - \frac{1}{2}S_k = 0. \tag{27}$$

Отсюда получаем итоговую формулу для оценки полной ковариационной матрицы:

$$\Sigma_k^{new} = \frac{S_k}{N_k} = \frac{\sum_{n=1}^{N}\gamma_{nk}(x_n-\mu_k^{new})(x_n-\mu_k^{new})^T}{N_k}. \tag{28}$$

3. Обновление весов π_k

Для максимизации по π_k необходимо учитывать ограничение $\sum\pi_k = 1$.

Используем метод множителей Лагранжа:

$$L(\pi, \lambda) = \sum_{k=1}^{K} N_k \ln \pi_k + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right). \tag{29}$$

Дифференцируя и приравнивая к нулю, получаем классический результат:

$$\pi_k^{new} = \frac{N_k}{N}. \tag{30}$$

2.3 Сходимость и статистические свойства

Важнейшим свойством EM-алгоритма является монотонное неубывание правдоподобия на каждой итерации. Пусть $\theta^{(t)}$ — параметры на шаге t, а $\theta^{(t+1)}$ — параметры, полученные после M-шага. Тогда справедливо неравенство:

$$\ln p(X|\theta^{(t+1)}) \geq \ln p(X|\theta^{(t)}). \tag{31}$$

Доказательство (схема):

1. На E-шаге мы выбираем $q^{(t+1)}$ так, чтобы $KL(q||p) = 0$, следовательно $\ln p(X|\theta^{(t)}) = \mathscr{L}(q^{(t+1)}, \theta^{(t)})$.

2. На M-шаге мы максимизируем $\mathscr{L}$ по θ, поэтому $\mathscr{L}(q^{(t+1)}, \theta^{(t+1)}) \geq \mathscr{L}(q^{(t+1)}, \theta^{(t)})$.

3. Для новых параметров $KL(q^{(t+1)}||p_{\theta^{(t+1)}}) \geq 0$.

4. Суммируя, получаем: $\ln p(X|\theta^{(t+1)}) = \mathscr{L}(\dots) + KL(\dots) \geq \ln p(X|\theta^{(t)})$.

Поскольку функция правдоподобия ограничена сверху (для корректно определенных моделей), последовательность значений сходится к локальному максимуму (или седловой точке).

Полученные оценки являются состоятельными, но смещенными (для конечных выборок). В рамках курсовой работы важно отметить, что EM-алгоритм не гарантирует нахождение глобального максимума, и результат сильно зависит от начальной инициализации параметров, что обуславливает необходимость использования стратегии мультистарта или инициализации методом K-means.

ГЛАВА 3
СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ

В задачах обучения без учителя выбор алгоритма критически зависит от структуры данных. В данной главе проводится теоретическое сравнение метода K-means (k-средних) и EM-алгоритма для GMM. Показано, что K-means является частным, предельным случаем GMM, накладывающим жесткие ограничения на модель данных.

3.1 K-means как предельный случай GMM

Алгоритм K-means минимизирует сумму квадратов внутрикластерных расстояний (Inertia):

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2, \tag{32}$$

где $r_{nk} \in \{0,1\}$ — бинарный индикатор принадлежности.

Рассмотрим модель GMM с двумя ограничивающими допущениями:

1. Изотропность и равенство дисперсий. Ковариационные матрицы всех компонент равны и пропорциональны единичной матрице с малым параметром ε:
$$\Sigma_k = \varepsilon I, \quad \forall k.$$

2. Равенство априорных вероятностей. Веса компонент фиксированы: $\pi_k = \frac{1}{K}$.

В этом случае плотность вероятности принимает вид:

$$p(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi\varepsilon)^{D/2}} \exp\left(-\frac{1}{2\varepsilon}\|x - \mu_k\|^2\right). \tag{33}$$

Рассмотрим E-шаг EM-алгоритма, вычисляющий апостериорную вероятность («ответственность»):

$$\gamma_{nk} = \frac{\pi_k \exp\left(-\frac{1}{2\varepsilon}\|x_n - \mu_k\|^2\right)}{\sum_j \pi_j \exp\left(-\frac{1}{2\varepsilon}\|x_n - \mu_j\|^2\right)} = \frac{\exp\left(-\frac{1}{2\varepsilon}\|x_n - \mu_k\|^2\right)}{\sum_j \exp\left(-\frac{1}{2\varepsilon}\|x_n - \mu_j\|^2\right)}. \tag{34}$$

Исследуем поведение γ_{nk} при стремлении дисперсии к нулю ($\varepsilon \to 0$). В сумме экспонент доминирующим будет слагаемое, для которого показатель степени (квадрат евклидова расстояния $\|x_n - \mu_j\|^2$) минимален. Пусть $k^* = \arg\min_j \|x_n - \mu_j\|^2$ — индекс ближайшего к точке x_n центроида. Тогда:

$$\lim_{\varepsilon \to 0} \gamma_{nk} = \begin{cases} 1, & \text{если } k = k^*, \\ 0, & \text{если } k \neq k^*. \end{cases} \tag{35}$$

Таким образом, при исчезающе малой дисперсии «мягкое» вероятностное распределение вырождается в «жесткое» (hard) присвоение, эквивалентное шагу назначения меток в K-means. Соответственно, максимизация правдоподобия становится эквивалентной минимизации евклидовых расстояний.

3.2 Геометрические и структурные различия

Несмотря на родственную связь, в общем случае (при использовании полной ковариационной матрицы в GMM) алгоритмы демонстрируют принципиально разное поведение.

1. Гибкость формы кластеров (Сферичность vs Эллиптичность)

   - K-means неявно предполагает, что кластеры имеют сферическую форму. Границы разделения между кластерами всегда линейны (являются срединными перпендикулярами к отрезкам, соединяющим центроиды). Это приводит к ошибкам при работе с анизотропными данными (вытянутыми вдоль определенного направления). Алгоритм «разрезает» вытянутые структуры, пытаясь вписать их в сферы.
   - GMM (Full Covariance) моделирует кластеры как эллипсоиды. Благодаря недиагональным элементам ковариационной матрицы Σ_k, модель способна учитывать корреляцию между признаками, поворачивая оси эллипсоида вдоль направления наибольшей дисперсии данных. Это позволяет корректно выделять наклонные и вытянутые кластеры.

2. Учет плотности и разброса данных

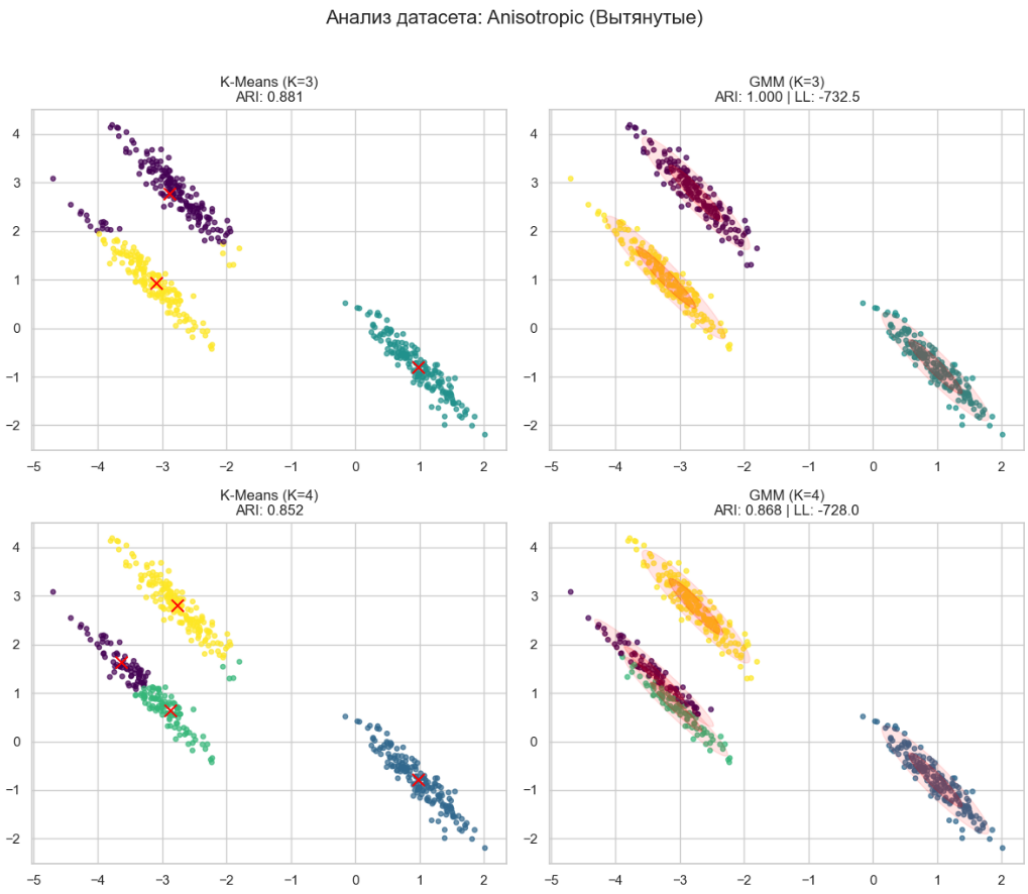

Рис. 2 – Схематичное сравнение границ разделения. Слева: K-means (линейные границы). Справа: GMM (квадратичные границы, эллипсы).

- K-means чувствителен к различиям в диаметрах кластеров. Поскольку алгоритм использует только Евклидово расстояние, точки, находящиеся на периферии «широкого» (разреженного) кластера, могут быть ошибочно отнесены к соседнему «узкому» (плотному) кластеру, если его центр геометрически ближе.
- GMM оценивает собственные ковариационные матрицы для каждого кластера. Это позволяет модели корректно обрабатывать ситуацию, когда один кластер компактен (малая $|\Sigma|$), а другой имеет большой разброс (большая $|\Sigma|$). Расстояние Махаланобиса, неявно используемое в показателе экспоненты, нормирует удаленность точки на дисперсию соответствующего кластера.

3. Характер неопределенности

   K-means возвращает однозначный ответ, что может быть критично для граничных объектов. GMM возвращает вероятностный вектор. Это позволяет выявлять объекты со «смешанной» природой (например, $\gamma_{n1} \approx$

$0.49, \gamma_{n2} \approx 0.51$) и интерпретировать их как шумовые или переходные, что повышает интерпретируемость результатов анализа.

Вывод: Использование GMM с полной матрицей ковариации теоретически обосновано для данных сложной геометрической структуры, тогда как область применимости K-means ограничена компактными, хорошо разделимыми группами сферической формы.

ГЛАВА 4
ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА

В рамках курсового проекта была разработана собственная реализация EM-алгоритма для моделей смеси гауссовых распределений. Программный комплекс написан на языке Python 3.x с использованием библиотеки линейной алгебры NumPy. Реализация выполнена в парадигме объектно-ориентированного программирования (класс `GMM`), что обеспечивает модульность кода и удобство проведения экспериментов.

4.1 Архитектура и векторные вычисления

Ключевым требованием к реализации являлась эффективность вычислений при обработке больших массивов данных. В языке Python использование явных циклов (например, `for` по объектам выборки) приводит к существенному замедлению работы интерпретатора. Для решения этой проблемы была применена векторизация вычислений.

Входные данные представляются в виде матрицы X размерности $(N \times D)$, где N — число объектов, D — размерность пространства признаков. Все параметры модели хранятся в виде многомерных массивов (тензоров):

- Векторы средних `means`: матрица $(K \times D)$.
- Ковариационные матрицы `covs`: тензор $(K \times D \times D)$, что соответствует использованию полной ковариационной матрицы.
- Веса компонент `weights`: вектор длины K.

Особое внимание было уделено M-шагу, где происходит обновление параметров. Использование механизма транслирования (broadcasting) библио-

теки NumPy позволило реализовать вычисление взвешенных ковариационных матриц без циклов по объектам, используя тензорные операции. Фрагмент реализации приведен в Листинге 1.

Листинг 1 – Векторизованное обновление ковариационных матриц

```
def _update_covs(self, X: np.ndarray, gamma: np.ndarray) ->
    np.ndarray:
    """Обновление ковариационных матриц с использованием
       broadcasting."""
    N, D = X.shape
    N_k = np.sum(gamma, axis=0)

    # Создание тензора разностей (K, N, D)
    diff = X - self.means[:, None, :]

    # Векторизованное вычисление взвешенной ковариации
    # gamma.T[:, :, None] имеет размерность (K, N, 1)
    numerator_part = gamma.T[:, :, None] * diff # (K, N, D)
    numerator_part = np.transpose(numerator_part, axes=(0, 2,
       1)) # (K, D, N)

    # Матричное умножение (Batch matrix multiplication)
    numerator = numerator_part @ diff # Результат (K, D, D)

    covs = numerator / (N_k[:, None, None] + 1e-10)

    # Регуляризация (см. п. 4.2.2)
    covs[:, np.arange(D), np.arange(D)] += self.r

    return covs
```

4.2 Обеспечение численной стабильности

При прямой реализации формул EM-алгоритма возникают проблемы арифметического переполнения (overflow) или исчезновения порядков (underflow), связанные с ограниченной точностью чисел с плавающей точкой.

4.2.1 Log-Sum-Exp Trick

Функция плотности гауссова распределения содержит экспоненту. При высокой размерности D значения вероятностей могут быть экстремально ма-

лыми, что интерпретируется компьютером как машинный ноль.

Для предотвращения этого все вычисления на E-шаге производятся в логарифмическом масштабе. Вместо вероятностей $p(x)$ вычисляются их логарифмы. Для вычисления логарифма суммы экспонент (необходимого для знаменателя в формуле Байеса) применяется тождество Log-Sum-Exp:

$$\ln \sum_i \exp(x_i) = a + \ln \sum_i \exp(x_i - a), \quad \text{где } a = \max_i(x_i). \tag{36}$$

Реализация E-шага с защитой от переполнения представлена в Листинге 2.

Листинг 2 – Реализация E-шага (Log-Sum-Exp)

```
def _e_step(self, X: np.ndarray) -> Tuple[np.ndarray, float]:
    """E-шаг: Вычисление ответственности (Gamma) через
       Log-Sum-Exp."""

    # 1. Логарифм числителя: ln(pi_k) + ln(N(x|...))
    log_pdf = self._calc_log_pdf(X)
    log_weighted_pdf = log_pdf + np.log(self.weights + 1e-300) #
       (N, K)

    # 2. Log-Sum-Exp Trick для знаменателя
    # Находим максимум для каждого объекта (axis=1) для стабилиза
       ции
    max_log_weighted = np.max(log_weighted_pdf, axis=1,
       keepdims=True)

    # Вычисляем ln(sum(exp(x - max))) + max
    log_sum_exp = max_log_weighted + np.log(
        np.sum(np.exp(log_weighted_pdf - max_log_weighted),
           axis=1, keepdims=True)
    )

    # 3. Вычисление ln(gamma) = ln(числитель) - ln(знаменатель)
    log_gamma = log_weighted_pdf - log_sum_exp
    gamma = np.exp(log_gamma)

    # Возвращаем gamma и текущее значение Log-Likelihood
    return gamma, np.sum(log_sum_exp)
```

4.2.2 Регуляризация ковариационной матрицы

В процессе обучения ковариационная матрица Σ_k может стать вырожденной (сингулярной), что делает невозможным её обращение. Для борьбы с этим применяется Тихоновская регуляризация: к главной диагонали матрицы ковариации на каждой итерации добавляется малая положительная константа ε (параметр `r` в коде).

В Листинге 1 данная операция выполняется строкой:

```
covs[:, np.arange(D), np.arange(D)] += self.r
```

Это гарантирует положительную определенность матрицы и геометрически предотвращает «схлопывание» эллипсоида кластера в плоскость.

4.3 Стратегии инициализации

Поскольку EM-алгоритм является локальным методом оптимизации, качество найденного решения критически зависит от начального приближения. В программе реализован гибридный подход: инициализация параметров с помощью алгоритма K-Means. Это позволяет начать оптимизацию GMM не со случайной точки, а с уже сформированных грубых кластеров.

Реализация инициализации приведена в Листинге 3.

Листинг 3 – Инициализация через K-Means

```
def _init_with_kmeans(self, X: np.ndarray) -> None:
    """Инициализация параметров через K-Means."""
    N, D = X.shape
    # Используем стандартную реализацию для быстрого старта
    kmeans = KMeans(n_clusters=self.K, n_init=self.n_init,
       random_state=42)
    labels = kmeans.fit_predict(X)

    self.means = kmeans.cluster_centers_
    self.weights = np.zeros(self.K)
    self.covs = np.zeros((self.K, D, D))

    for k in range(self.K):
        X_k = X[labels == k]
        self.weights[k] = len(X_k) / N

        # Начальная оценка ковариации по результатам K-Means
```

```
            if len(X_k) > 1:
                self.covs[k] = np.cov(X_k, rowvar=False)
            else:
                self.covs[k] = np.eye(D)

            # Превентивная регуляризация
            self.covs[k] += np.eye(D) * self.r
```

Данный подход значительно ускоряет сходимость алгоритма и позволяет избегать субоптимальных локальных максимумов функции правдоподобия, что подтверждено экспериментально в Главе 5.

ГЛАВА 5
ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Целью экспериментальной части работы являлась верификация разработанной программной реализации EM-алгоритма, анализ устойчивости его сходимости, а также сравнение качества кластеризации с классическим методом K-means на наборах данных различной геометрической структуры.

Эксперименты проводились на синтетических и реальных данных. Для оценки качества кластеризации использовалась метрика ARI (Adjusted Rand Index), которая принимает значение 1.0 при полном совпадении с эталонной разметкой и значения около 0 при случайном разбиении.

5.1 Анализ сходимости и устойчивости

Первым этапом тестирования была проверка корректности работы оптимизационной процедуры. Эксперимент проводился на наборе данных `Isotropic Blobs` (сферические кластеры). В ходе работы алгоритма фиксировалось значение логарифма правдоподобия (Log-Likelihood) на каждой итерации. График демонстрирует монотонный рост целевой функции и быстрый выход на плато, что подтверждает корректность реализации E-шага и M-шага.

Также было проведено исследование влияния инициализации. При случайной инициализации параметров (`init_random`) алгоритм в ряде запусков сходился к локальным экстремумам (кластеры «склеивались» или делились некорректно), показывая низкие значения Log-Likelihood (LL). Использование гибридной стратегии с инициализацией через K-means (`init_kmeans`)

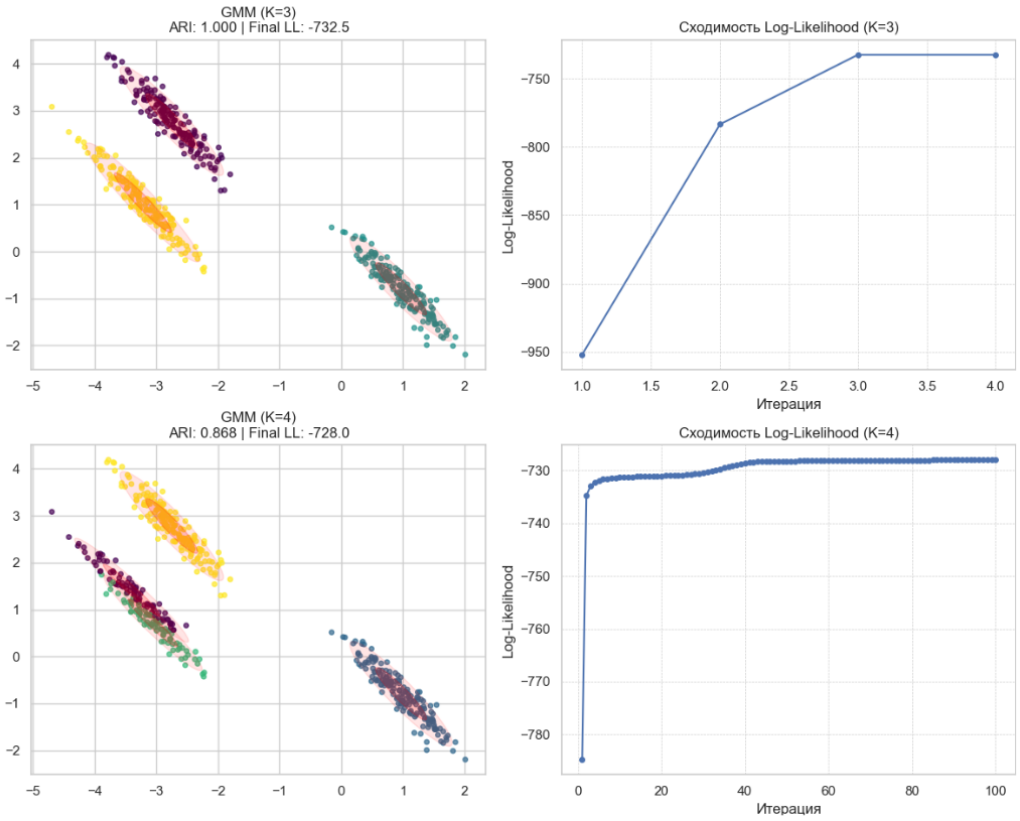

Рис. 3 – Динамика максимизации функции правдоподобия (Log-Likelihood по итерациям).

обеспечило стабильную сходимость к глобальному максимуму во всех тестах.

5.2 Сравнительный анализ на синтетических данных

Ключевой задачей было продемонстрировать преимущества использования полной ковариационной матрицы в модели GMM по сравнению с жесткими ограничениями алгоритма K-means.

5.2.1 Анизотропные данные (Anisotropic Blobs)

Набор данных содержит кластеры, сильно вытянутые вдоль диагонали и расположенные близко друг к другу. Это моделирует наличие сильной линейной корреляции между признаками.

- K-Means (ARI ≈ 0.88): Алгоритм, основываясь на Евклидовом расстоянии, предполагает сферичность кластеров. В результате границы разделения перпендикулярны линии, соединяющей центры, что приводит к некорректному «разрезанию» вытянутых хвостов кластеров.

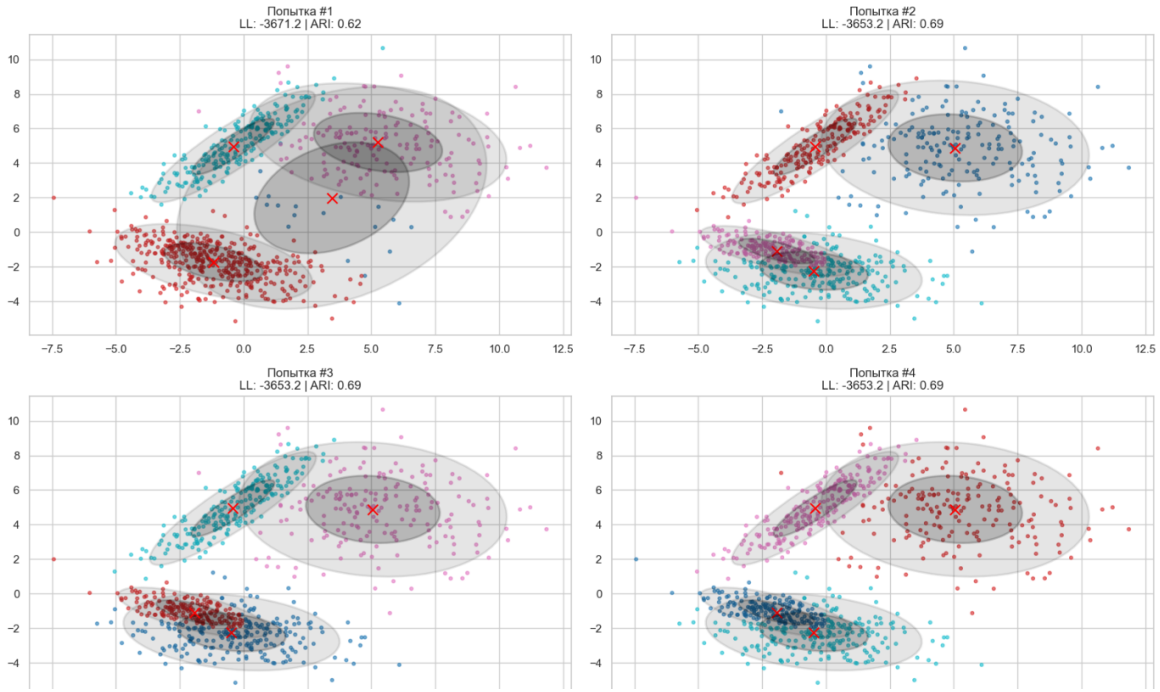

Рис. 4 – Сравнение попыток запуска. Слева/Справа — разные исходы случайной инициализации, показывающие попадание в локальные минимумы.

- GMM (ARI = 1.0): Благодаря обучению полной матрицы ковариации $\boldsymbol{\Sigma}_k$, модель адаптировала форму эллипсов концентрации под структуру данных. Наклон главных осей эллипсов совпал с направлением разброса данных, что обеспечило идеальное разделение.

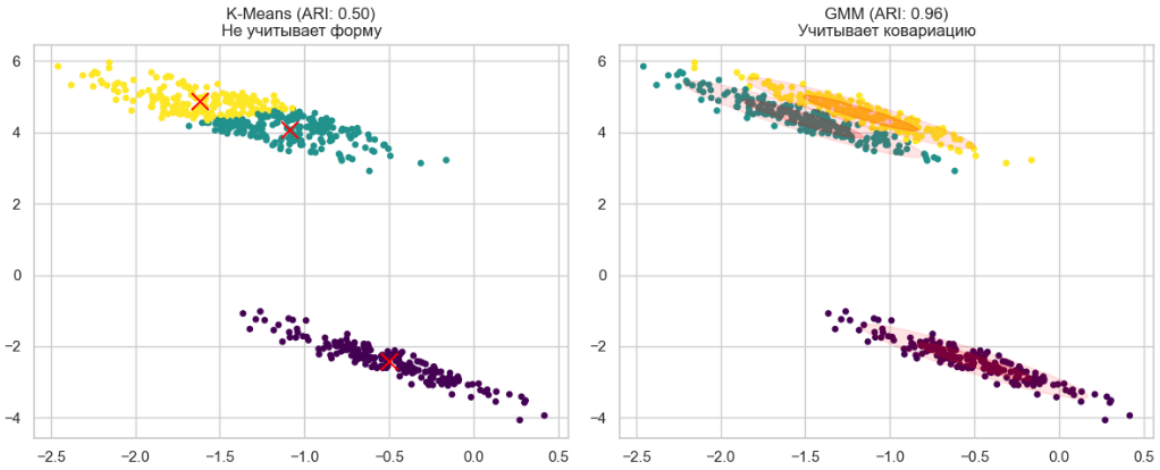

Рис. 5 – Сравнение на датасете Anisotropic. Слева: K-means (ошибки на границах). Справа: GMM (корректные наклонные эллипсы).

5.2.2 Данные с различной дисперсией (Varied Variance)

Эксперимент моделирует ситуацию, когда один кластер очень плотный (малая дисперсия), а соседний — разреженный (большая дисперсия).

- K-Means (ARI ≈ 0.79): Граница проводится посередине между центроидами. Периферийные точки большого кластера, геометрически более близкие к центру малого кластера, ошибочно относятся к последнему.
- GMM (ARI ≈ 0.95): Алгоритм учитывает различную "ширину"распределени Граница решений смещается в сторону более плотного кластера, что соответствует байесовскому правилу минимизации ошибки.

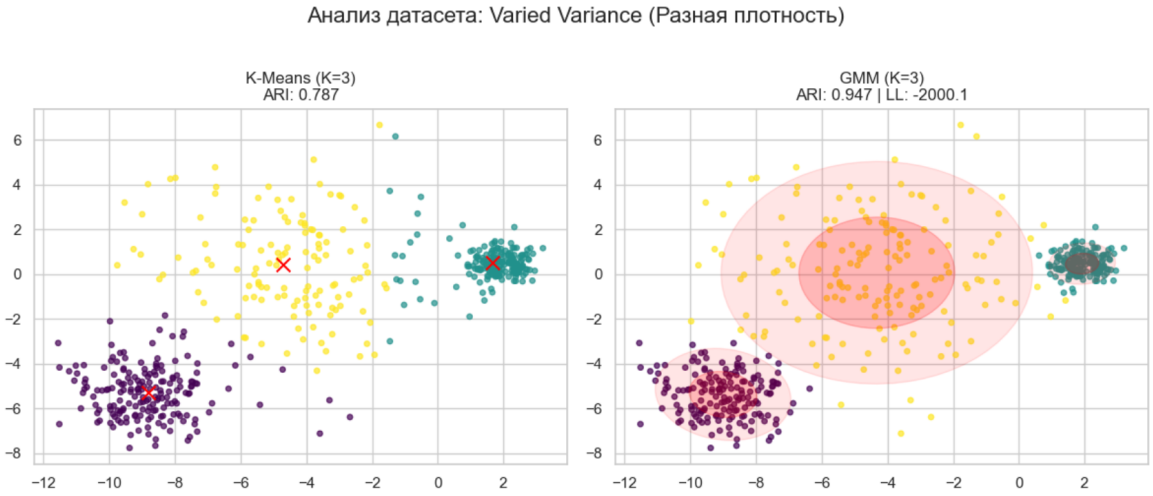

Рис. 6 – Сравнение на датасете Varied Variance. GMM корректно учитывает разный размер кластеров.

5.2.3 Невыпуклые данные (Moons)

Датасет представляет собой два вложенных полумесяца. Это сложная задача для параметрических методов, так как данные не описываются одним нормальным распределением.

- K=2: Оба алгоритма (K-means и GMM) не справились, проведя линейную/квадратичную границу.
- K=10: GMM продемонстрировал способность работать как универсальный аппроксиматор плотности. Сложная форма полумесяца была покрыта «цепочкой» из нескольких эллипсов. Это позволяет использовать

GMM не только для кластеризации, но и для оценки плотности распределения сложной конфигурации.

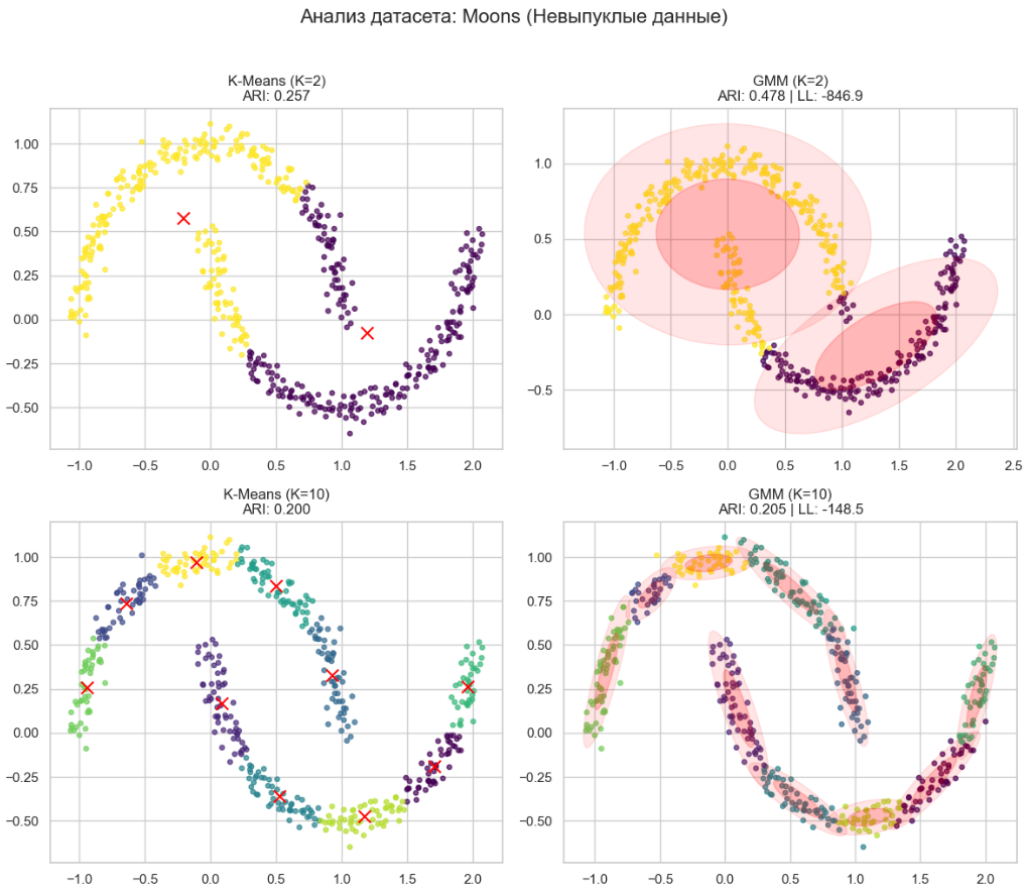

Рис. 7 – Аппроксимация сложной формы датасета Moons с помощью GMM (K=10).

5.3 Апробация на реальных данных (Iris Fisher)

Заключительный эксперимент проводился на классическом наборе данных ирисов Фишера (кластеризация по 4 признакам, визуализация по двум).

На графике видно, что класс Setosa линейно отделим, и оба алгоритма справляются с ним успешно. Однако классы Versicolor и Virginica имеют зону естественного перекрытия. В отличие от K-Means, который проводит жесткую границу, GMM построил мягкую модель: эллипсы ковариации накладываются друг на друга. Для спорных точек в области пересечения алгоритм вернул вероятности принадлежности $\gamma_{nk} \approx 0.5$, что объективно отражает природу данных и является важным преимуществом вероятностного подхода.

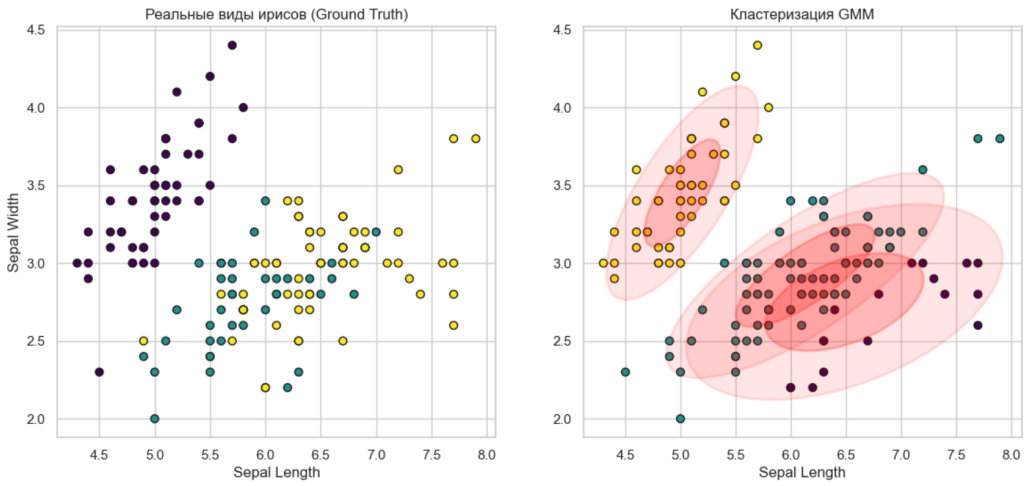

Рис. 8 – Результаты на Iris Dataset. Визуализация показывает мягкое пересечение эллипсов для перекрывающихся классов.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта было проведено теоретическое исследование и практическая реализация алгоритма Expectation-Maximization для задачи кластеризации на основе смеси гауссовых распределений (GMM).

Основные результаты работы:

1. Теоретический базис. Изучена математическая модель GMM и вывод формул EM-алгоритма через максимизацию вариационной нижней границы (ELBO). Показано, что классический алгоритм K-means является частным случаем GMM, возникающим при наложении жестких ограничений на ковариационные матрицы (сферичность) и вероятности (жесткая привязка).

2. Программная реализация. Разработан программный модуль на языке Python (библиотека NumPy), реализующий GMM с полной матрицей ковариации. В реализации применены методы обеспечения численной стабильности:

   - Log-Sum-Exp Trick для предотвращения арифметического переполнения при работе с малыми вероятностями.
   - Векторизация матричных операций для повышения производительности.

- Тихоновская регуляризация для предотвращения сингулярности ковариационных матриц.

3. Экспериментальные выводы. Сравнительный анализ подтвердил теоретические преимущества GMM перед K-means:

   - Модель с полной матрицей ковариации успешно кластеризует анизотропные данные, корректно определяя ориентацию кластеров (correlation aware).
   - Вероятностная природа алгоритма позволяет корректно обрабатывать кластеры различной плотности и объема, а также возвращать меру уверенности в классификации («мягкая» кластеризация).
   - Гибридная инициализация (K-means + EM) является предпочтительной стратегией, минимизирующей риск попадания в локальные экстремумы функции правдоподобия.

Таким образом, цель работы достигнута. Разработанный алгоритм демонстрирует высокую гибкость и может быть рекомендован для анализа данных со сложной геометрической структурой, где предположения метрических алгоритмов (таких как K-means) не выполняются.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Харин Ю. С., Зуев Н. М., Жук Е. Е. Теория вероятностей, математическая и прикладная статистика. — Минск: БГУ, 2011. — (Глава 14: Статистический анализ смесей распределений).

[2] Bishop C. M. Pattern Recognition and Machine Learning. — Springer, 2006. — 738 p. (Chapter 9: Mixture Models and EM).

[3] Murphy K. P. Machine Learning: A Probabilistic Perspective. — MIT Press, 2012. — 1067 p.

[4] Dempster A. P., Laird N. M., Rubin D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm // Journal of the Royal Statistical Society. Series B (Methodological). — 1977. — Vol. 39, No. 1. — P. 1–38.

[5] VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. — O'Reilly Media, 2016. (Section: In Depth - Gaussian Mixture Models).

[6] Документация библиотеки Scikit-learn: Gaussian Mixture Models [Электронный ресурс]. — URL: `https://scikit-learn.org/stable/modules/mixture.html` (дата обращения: 10.12.2025).