



Сравнение kmeans и EM алгоритма

докладчик: Касерес Гутьеррес Леонард

08.04.2024, поступление

Содержание

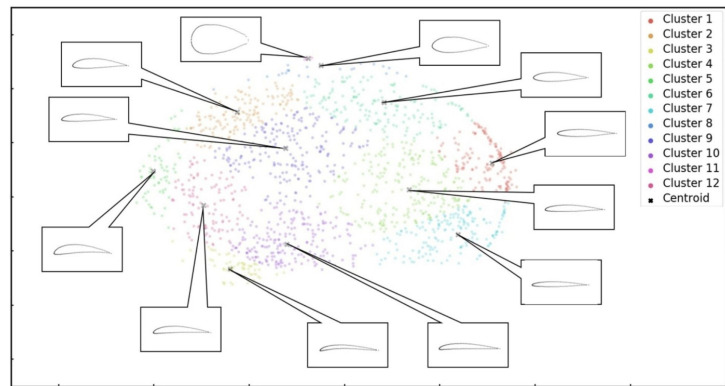
1. Описание задачи(введение)
 - a. Введение в задачу кластеризации
 - b. Примеры применения, актуальность
 - c. Алгоритмы кластеризации
 - d. Проблема?
2. Два решения задачи кластеризации
 - a. Kmeans
 - i. Описание
 - ii. Тестирование на 1-ом датасете
 - b. EM-алгоритм
 - i. Описание
 - ii. Реализация
 - iii. Тестирование на 1-ом датасете
3. Сравнение Kmeans и EM-алгоритма
4. Модификация EM-алгоритма
5. Анализ
 - a. Сильные и слабые стороны работы
 - b. Недостатки обоих методов и границы применимости
 - c. Повторяемость результатов(git = qr)



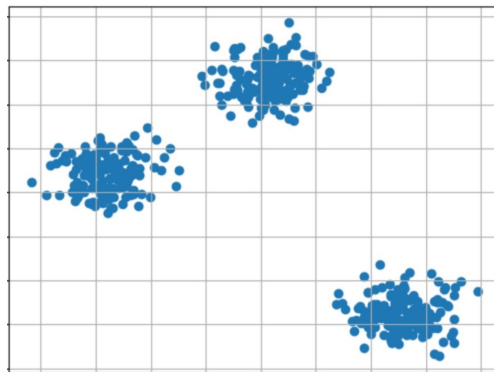
jupyter ноутбук

Задача кластеризации

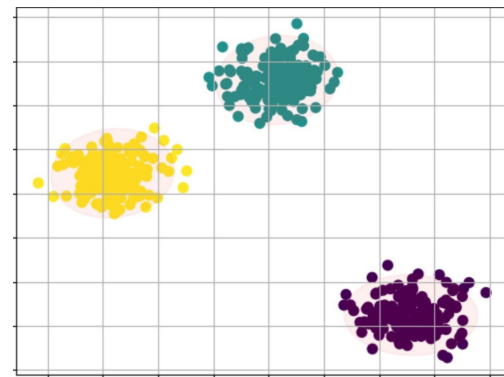
Кластеризация — это задача разбиения множества объектов на похожие группы, называемые кластерами.



Нетривиальный пример кластеризации



Пример данных

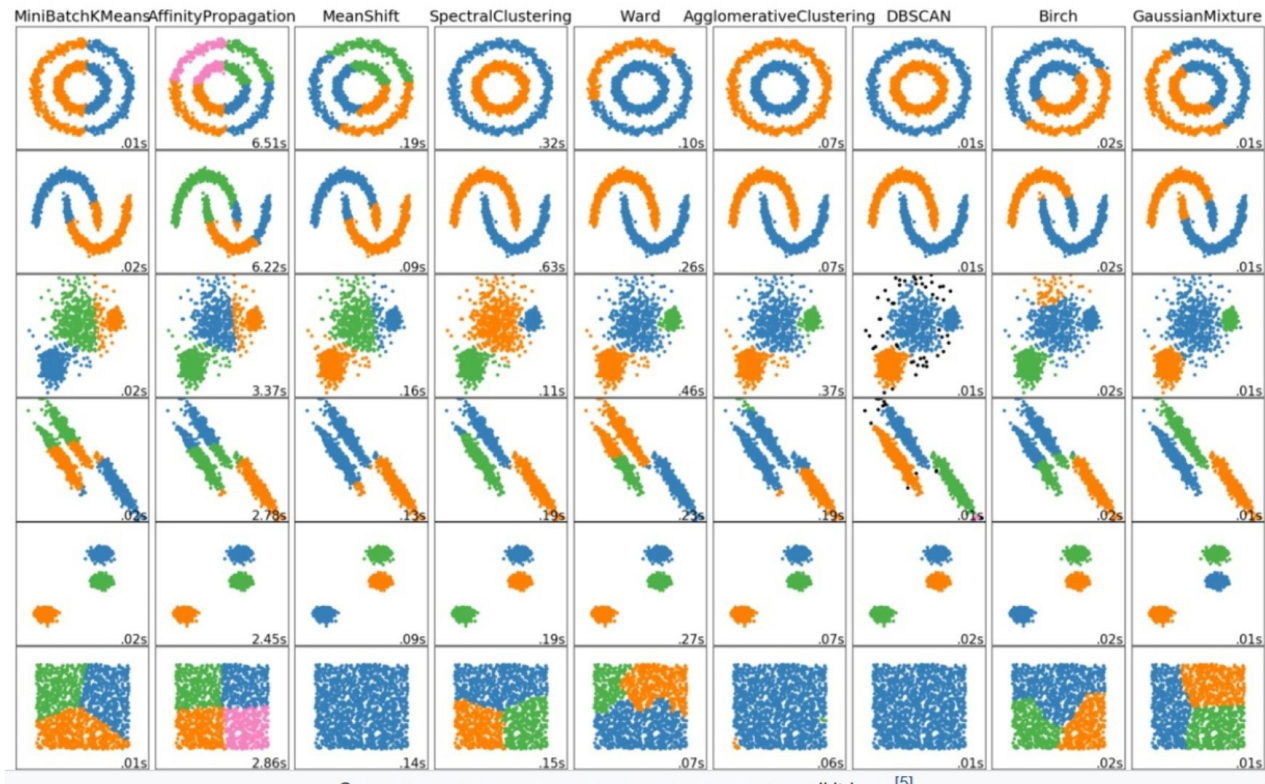


Результат работы KMeans

Алгоритмы кластеризации используются в следующих задачах:

1. Информатика
2. Медицина
3. Социальные науки
4. Бизнес
5. и т.д.

Алгоритмы кластеризации

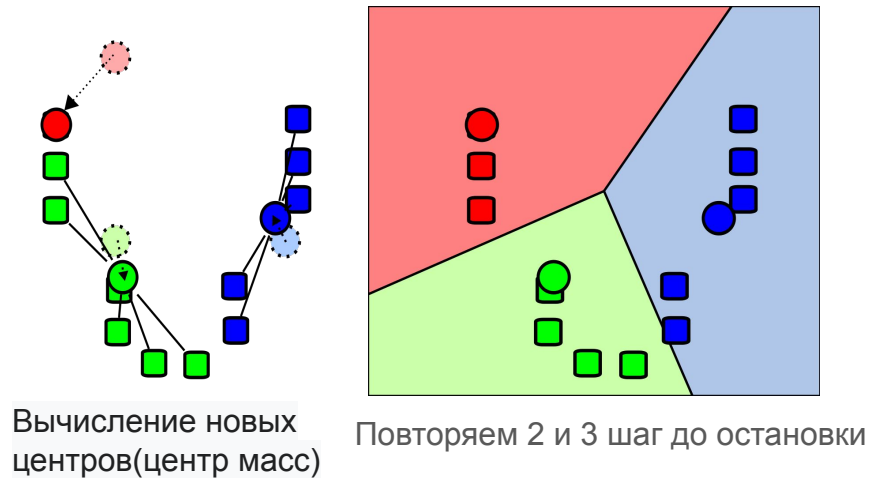
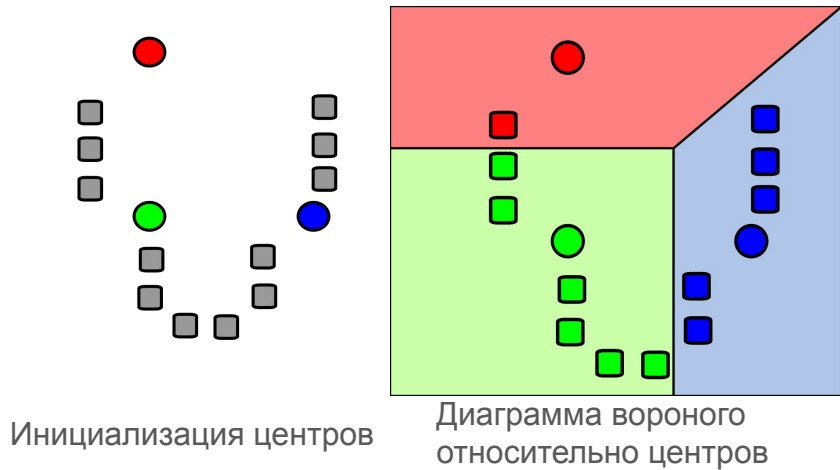


Проблемы:

1. Универсальность
2. Не существует правильного ответа

Разные алгоритмы кластеризации на разных данных

KMeans

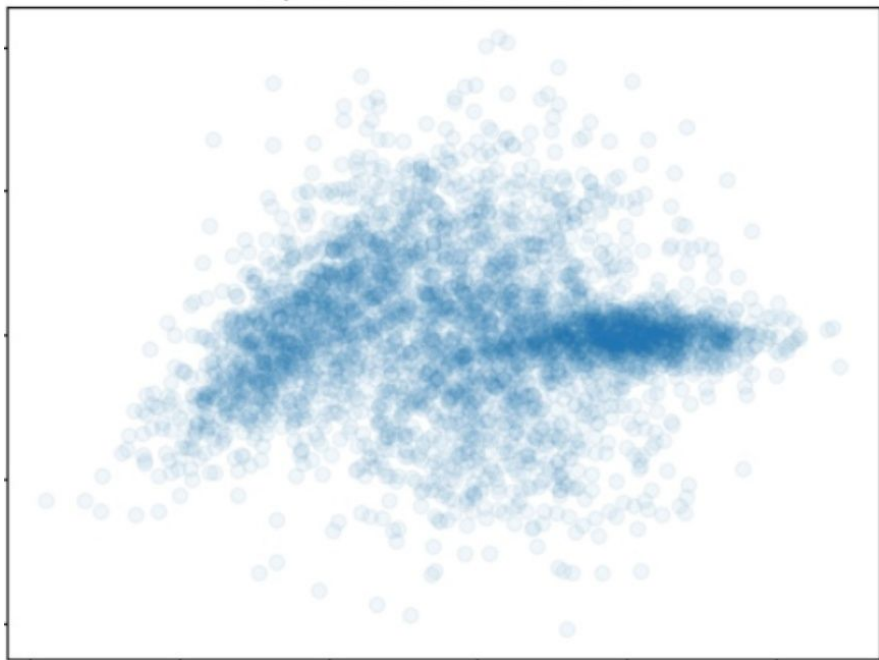


Наиболее популярный алгоритм кластеризации

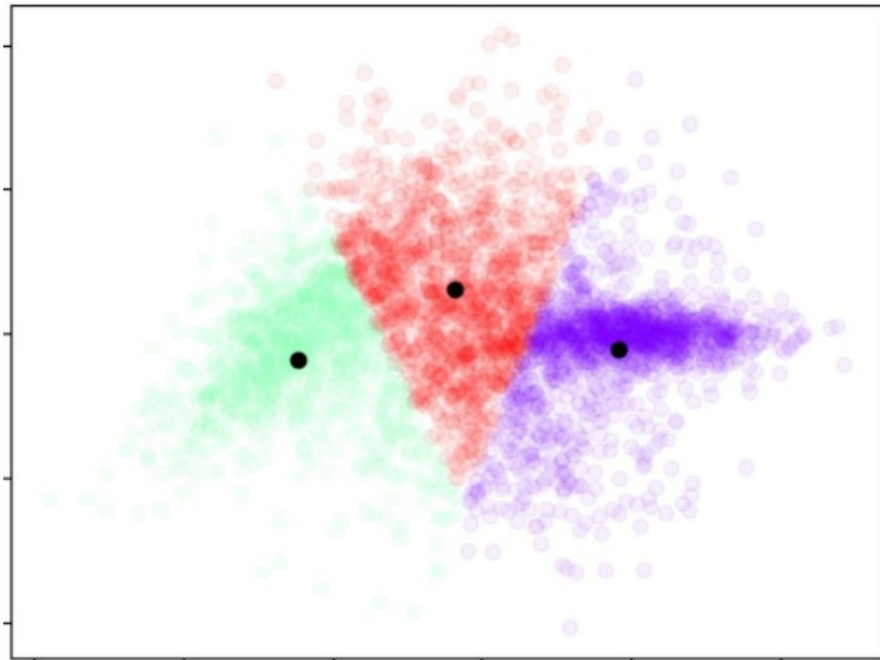
минимизирующий следующую функцию:

$$V = \sum_{i=1}^k \sum_{x \in S_i} (x - \mu_i)^2$$

Предложенный первый датасет

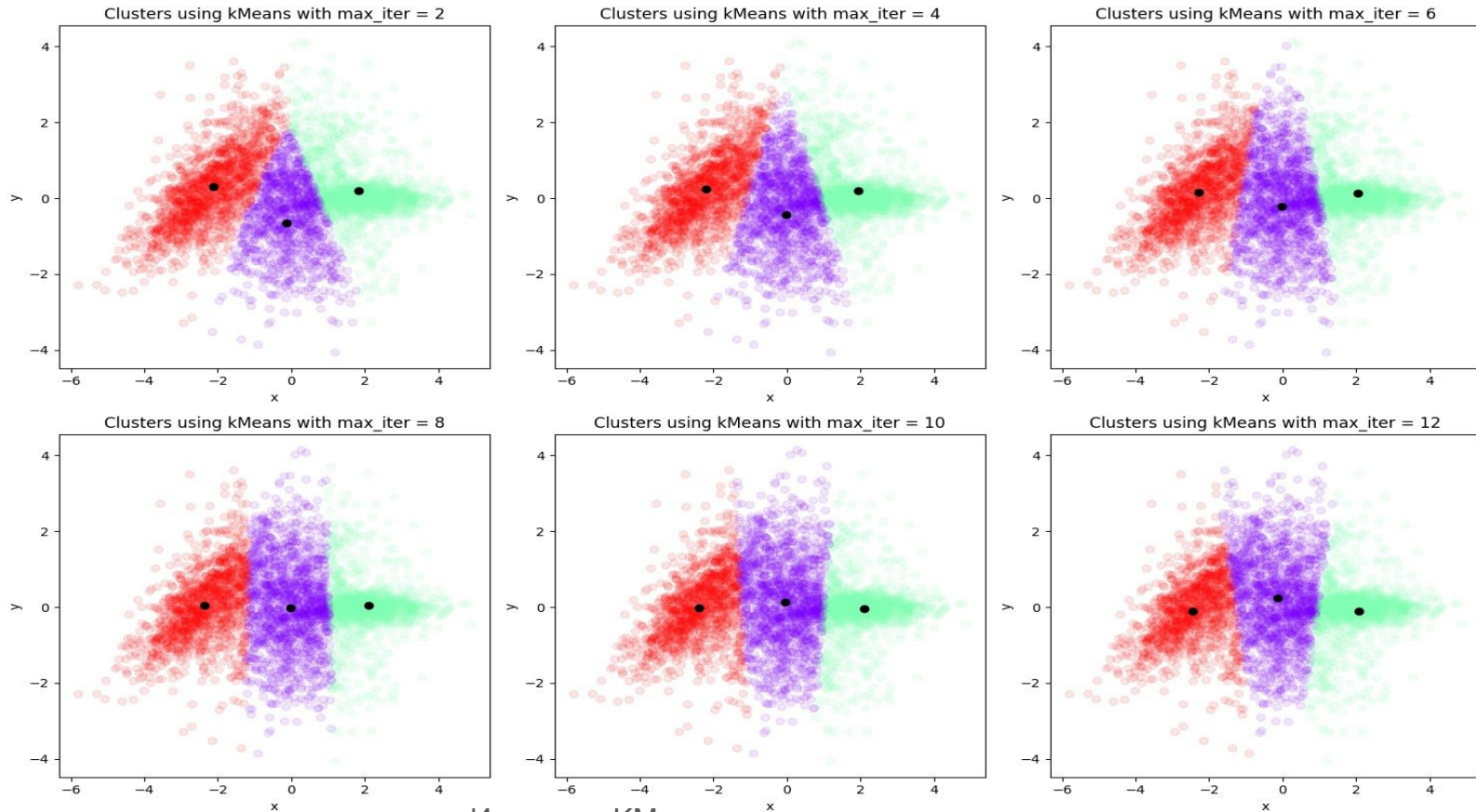


первый датасет



KMeans кластеризация данных

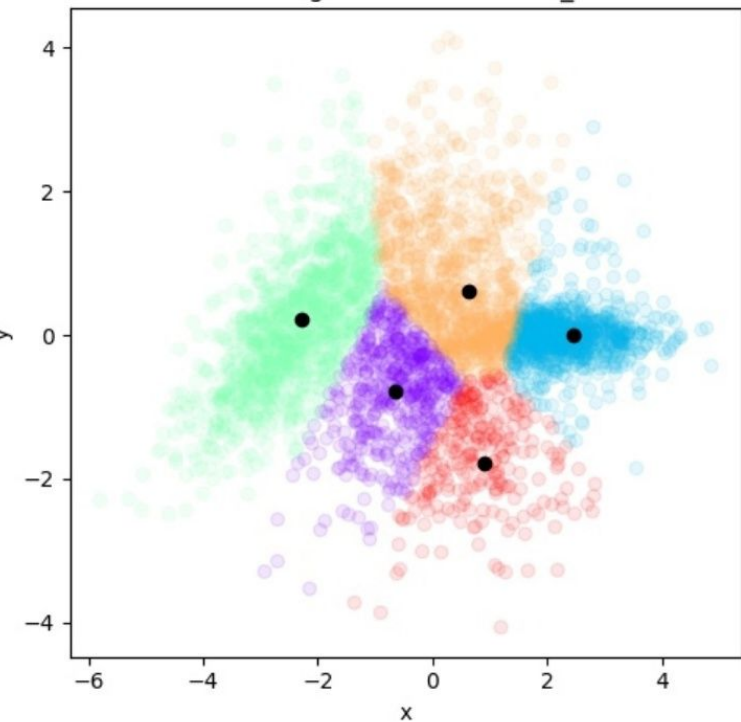
Динамика KMeans



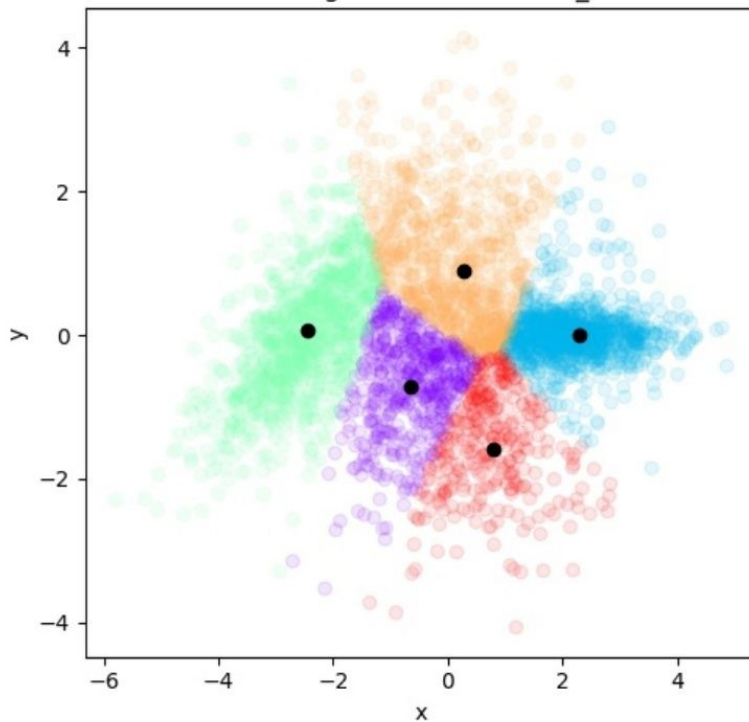
Итерации KMeans заданного датасета

Границы KMeans

Clusters using kMeans with max_iter = 2



Clusters using kMeans with max_iter = 4



Ярко
выраженная
граница

ЕМ-алгоритм

ЕМ-алгоритм заключается в повторении двух шагов:

- Е-шага (expectation-step), шага выведения функции $l(x, z|\theta)$.
- М-шага (maximization-step), шага ее максимизации.

$$l(x|\theta) \quad \ggg \quad l(x, z|\theta)$$

z - некая латентная, ненаблюдаемая переменная.

Expectation-step

Инициализация: задать начальные условия на θ_{old} .

Е.1-шаг: найти условное распределение латентных переменных $p(Z|X, \theta_{old})$.

Е.2-шаг: построить функцию $Q(\theta, \theta_{old}) = E_{Z|X, \theta}(\ell(x, z|\theta)|x, \theta_{old})$.

$$p(z|x, \theta_{old}) = \frac{p(z, x|\theta_{old})}{p(x|\theta_{old})} \qquad p(z_i = 1, x_i|\theta_{old}) = p_1 \cdot p(x_i|z_i = 1, \theta_{old}),$$

$$P(z_i = 1|x_i, \theta_{old}) = \frac{f(x_i|z_i = 1, \theta_{old})p_1}{p_1 f(x_i|z_i = 1, \theta_{old}) + p_2 f(x_i|z_i = 2, \theta_{old}) + (1 - p_1 - p_2) f(x_i|z_i = 3, \theta_{old})}$$

Expectation-step

$$Q(\theta, \theta_{old}) = \mathbb{E} [\ln p(x, z|\theta) | x, \theta_{old}]$$

$$\begin{aligned} Q(\theta|\theta_{old}) = & \sum_i P(z_i = 1|x, \theta_{old})[\ln f(x_i|\theta) + \ln p_1] + P(z_i = 2|x, \theta_{old})[\ln f(x_i|\theta) + \ln(p_2)] \\ & + (1 - P(z_i = 1|x, \theta_{old}) - P(z_i = 2|x, \theta_{old}))[\ln f(x_i|\theta) + \ln(1 - p_1 - p_2)] \end{aligned}$$

$$\ln f(x) = -\frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x - \mu)^2}{2\sigma^2}$$

Maximization-step

$$Q'_{\mu_1} = \sum_i P(z_i = 1|x, \theta_{old}) \frac{(x_i - \mu_1)}{\sigma_1^2}$$

$$\mu_1^{new} = \frac{\sum_i P(z_i = 1|x, \theta_{old}) x_i}{\sum_i P(z_i = 1|x, \theta_{old})}$$

Maximization-step - все параметры

$$\mu_{j,new} = \frac{\sum_{i=1}^n x_i P(Y_i = j \mid x_i, \theta_{old})}{\sum_{i=1}^n P(Y_i = j \mid x_i, \theta_{old})}$$

$$\sigma_{j,new}^2 = \frac{\sum_{i=1}^n (x_i - \mu_j)^2 P(Y_i = j \mid x_i, \theta_{old})}{\sum_{i=1}^n P(Y_i = j \mid x_i, \theta_{old})}$$

$$p_{j,new} = \frac{1}{n} \sum_{i=1}^n P(Y_i = j \mid x_i, \theta_{old}).$$

Реализация ЕМ-алгоритма(без модификации)

```
def em_clustering(data, n_clusters = 3, max_iter=100, tol=1e-4):
    np.random.seed(42)
    n_samples, n_features = data.shape

    means = np.random.rand(n_clusters, n_features)
    covariances = np.array([np.eye(n_features) for _ in range(n_clusters)])
    weights = np.ones(n_clusters) / n_clusters
    for _ in range(max_iter):
        # E-step
        likelihoods = np.array([multivariate_normal_pdf(data, mean=means[i],
                                                         cov=covariances[i]) for i in range(n_clusters)]).T

        weighted_likelihoods = likelihoods * weights
        cluster_probs = weighted_likelihoods / weighted_likelihoods.sum(axis=1)[:, np.newaxis]

        # M-step
        new_means = np.dot(cluster_probs.T, data) / cluster_probs.sum(axis=0)[:, np.newaxis]
        new_covariances = np.array([np.dot((data - new_means[i]).T,
                                           np.dot(np.diag(cluster_probs[:, i]),
                                                  (data - new_means[i]))) / cluster_probs[:, i].sum() for i in range(n_clusters)])

        new_weights = cluster_probs.sum(axis=0) / n_samples

        if np.linalg.norm(new_means - means) < tol:
            break

        means = new_means
        covariances = new_covariances
        weights = new_weights

    return means, covariances, weights, np.array([multivariate_normal_pdf(data, mean=means[i], cov=covariances[i]) for i in range(n_clusters)]).T
```



Обоснование EM-алгоритма

$$\begin{aligned}\ell(x|\theta) &= \sum_i \ln f(x_i|\theta) = \sum_i \sum_j P(Z = j) \ln f(x_i|\theta) = \sum_i \sum_j P(Z = j) \ln \frac{f(x_i, Z = j|\theta)}{P(Z = j|x_i, \theta)} = \\ &= \sum_i \sum_j P(Z = j) \ln \frac{f(x_i, Z = j|\theta)P(Z = j)}{P(Z = j|x_i, \theta)P(Z = j)} = \\ &= \sum_i \sum_j P(Z = j) \ln \frac{f(x_i, Z = j|\theta)}{P(Z = j)} + \sum_i \sum_j P(Z = j) \ln \frac{P(Z = j)}{P(Z = j|x_i, \theta)} = \\ &= M(P(Z = j), \theta) + D_{KL}[P(Z = j) || P(Z = j|x_i, \theta)].\end{aligned}$$

Обоснование EM-алгоритма

Е-шаг.

Максимизируем $M(P(Z = j), \theta)$ по $P(Z = j)$.

Так как $\ell(x|\theta)$ не зависит от $P(Z = j)$, то максимум $M(P(Z = j), \theta)$ по $P(Z = j)$ будет достигнут, когда D_{KL} минимальна.

Минимальная $D_{KL}(A||B)$ равна 0, и это достигается, когда $A||B$. Из этого делаем вывод, что на Е-шаге мы устанавливаем

$$P(Z = j) := P(Z = j|x_i, \theta^{old}).$$

Обоснование EM-алгоритма

М-шаг.

Максимизируем $M(P(Z = j), \theta)$ по θ . Распишем M ещё раз:

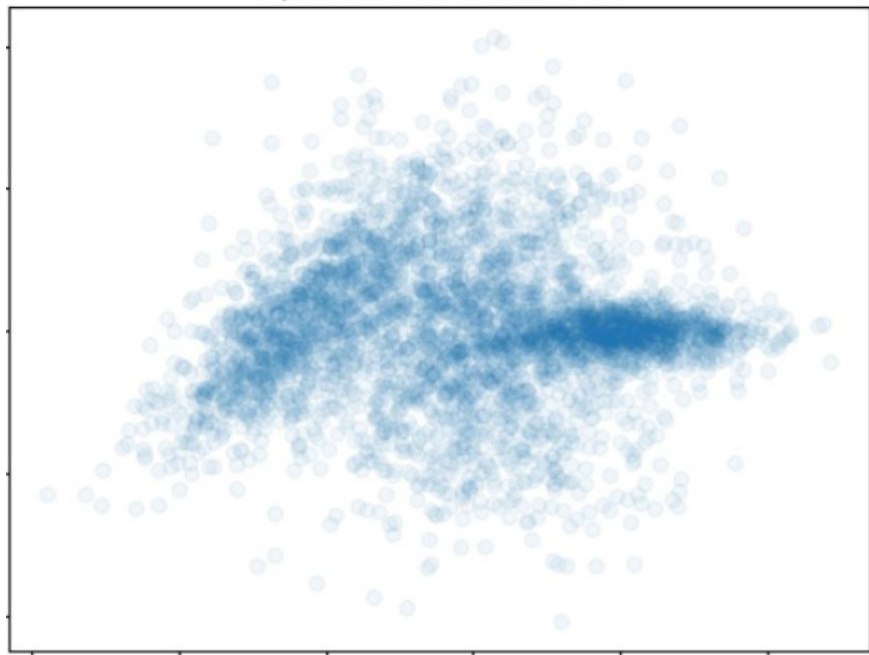
$$M = \sum_i \sum_j P(Z = j) \ln \frac{f(x_i, Z = j|\theta)}{P(Z = j)}$$

Заметим, что знаменатель подлогарифмического выражения не зависит от θ . Выбросим его и заменим $P(Z = j)$ на результат, полученный нами на Е-шаге:

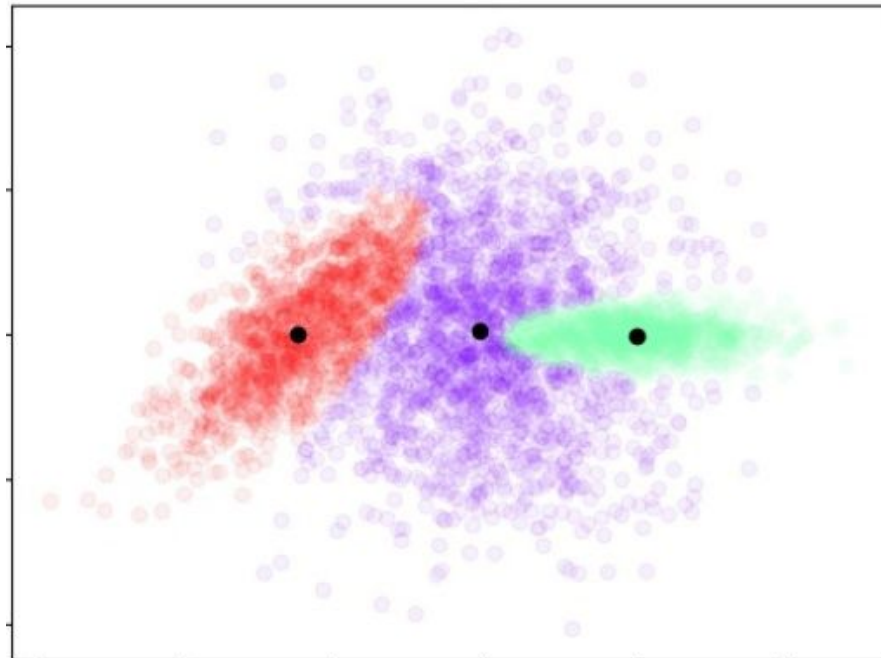
$$M = \sum_i \sum_j P(Z = j|x_i, \theta^{old}) \ln f(x_i, Z = j|\theta) = \sum_i E_{Z|x_i, \theta_{old}}(\ln f(x_i, Z = j|\theta)) := Q(\theta|\theta^{old}).$$

Далее мы максимизируем Q по θ , обновляем θ на аргмаксимум Q , и возвращаемся к Е-шагу.

Предложенный первый датасет

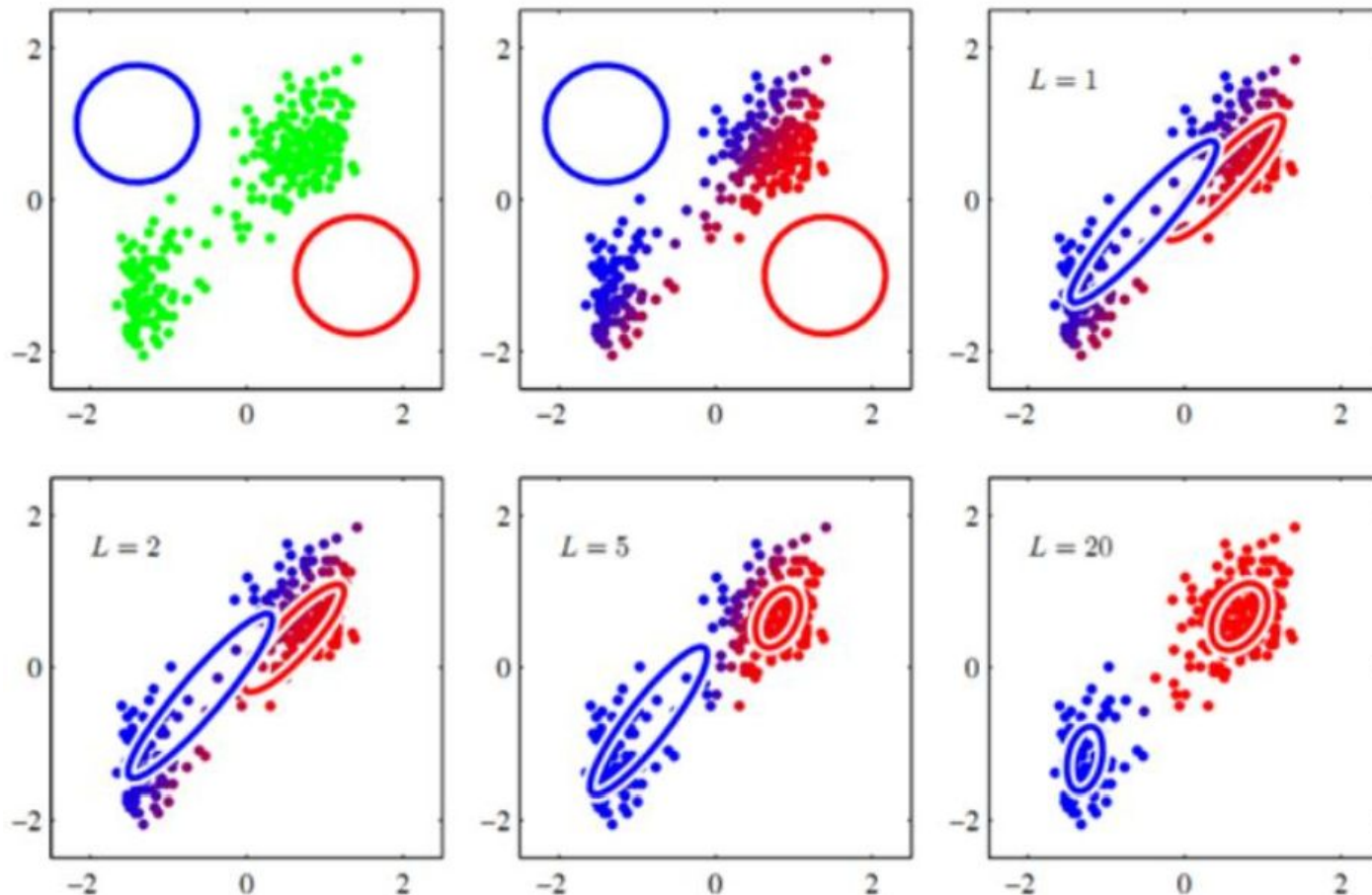


первый датасет



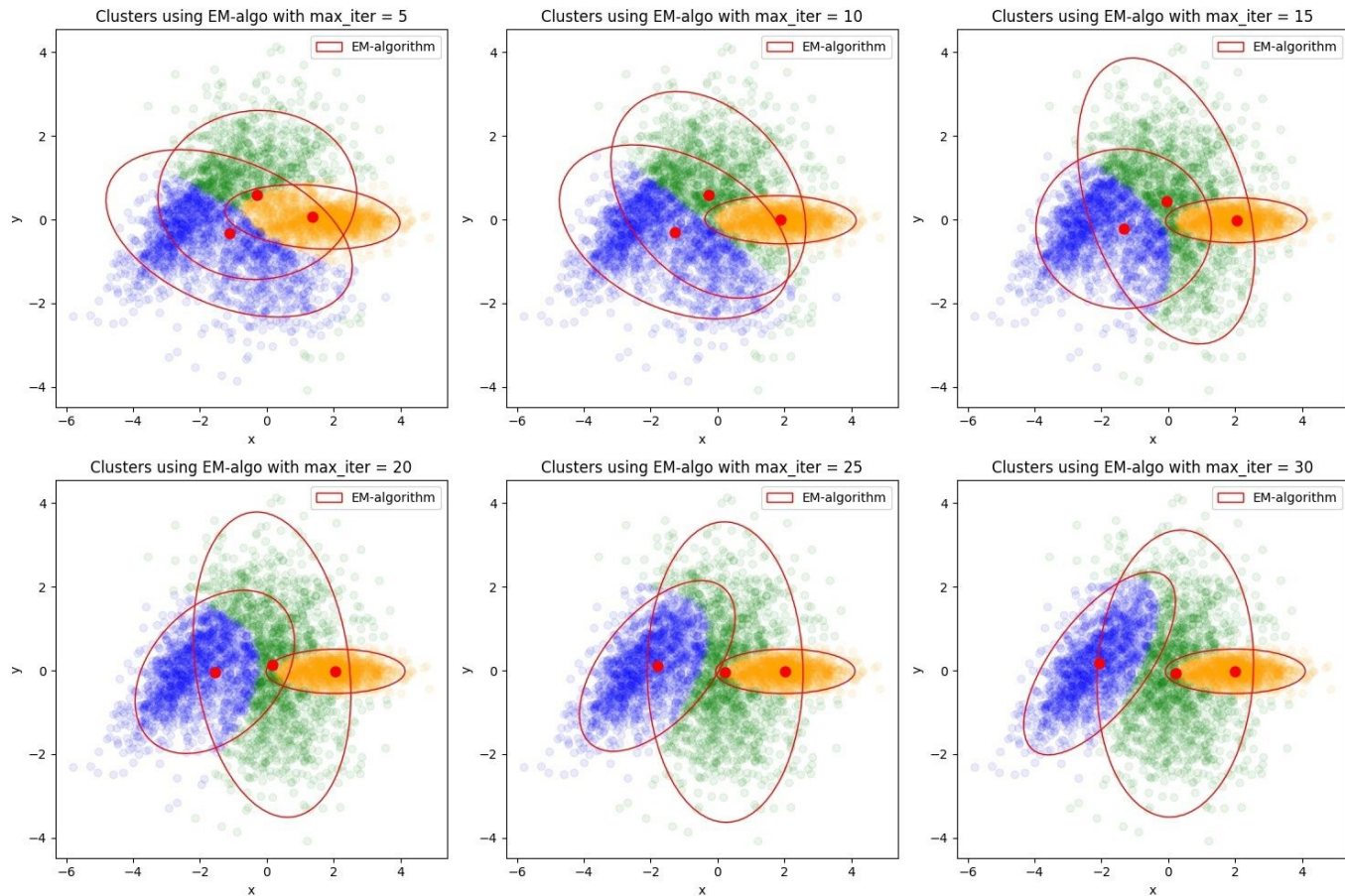
ЕМ-алгоритм кластеризация данных

Динамика ЕМ-алгоритма



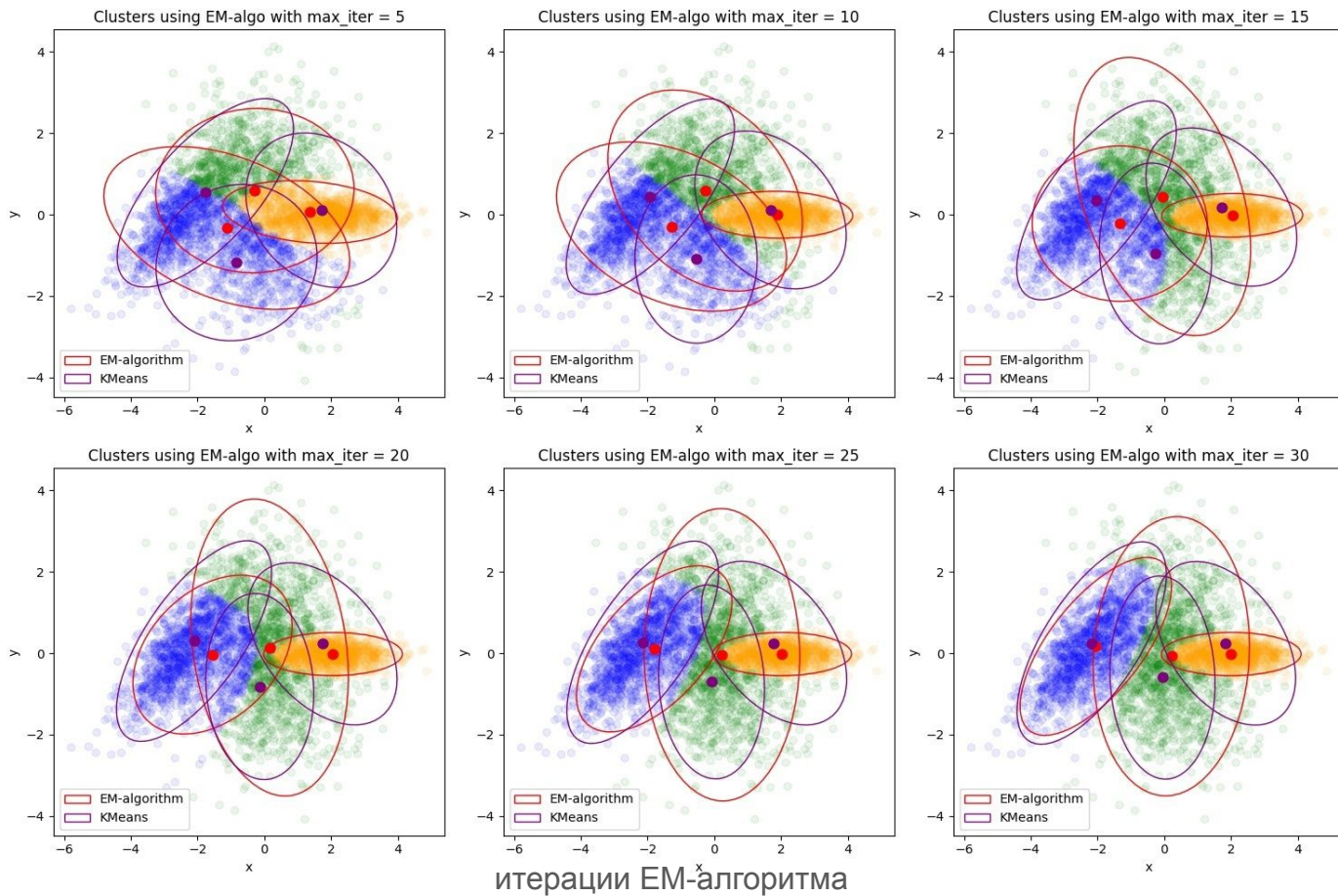
Итерации
алгоритма

Динамика ЕМ-алгоритма



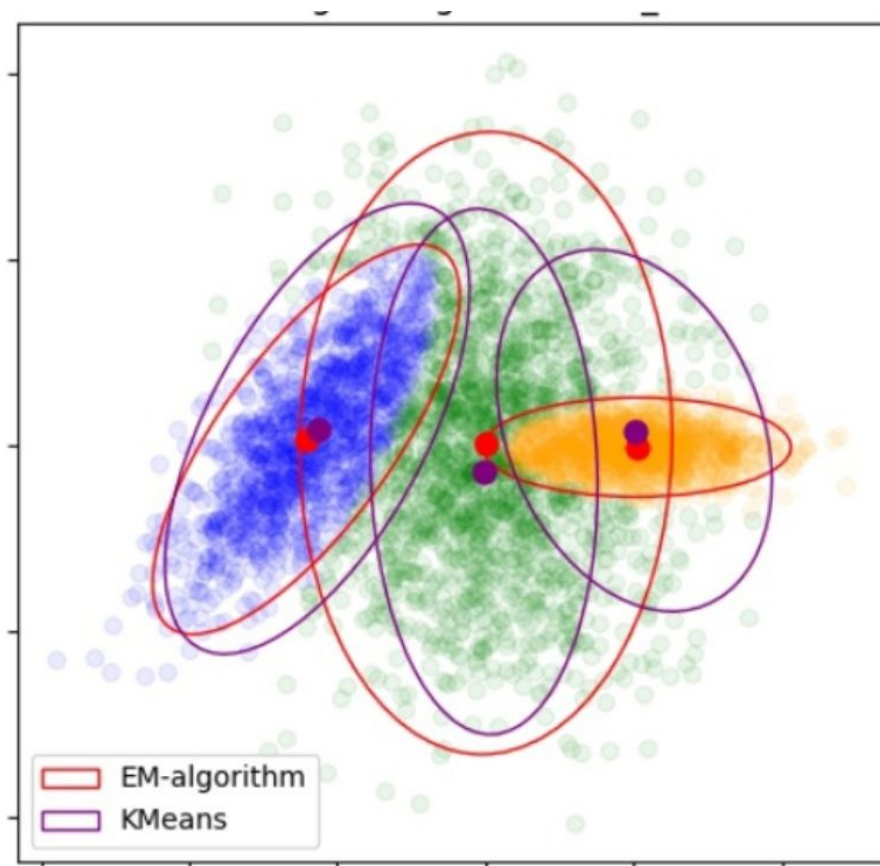
Итерации ЕМ-алгоритма

Сравнение Kmeans и EM-алгоритма

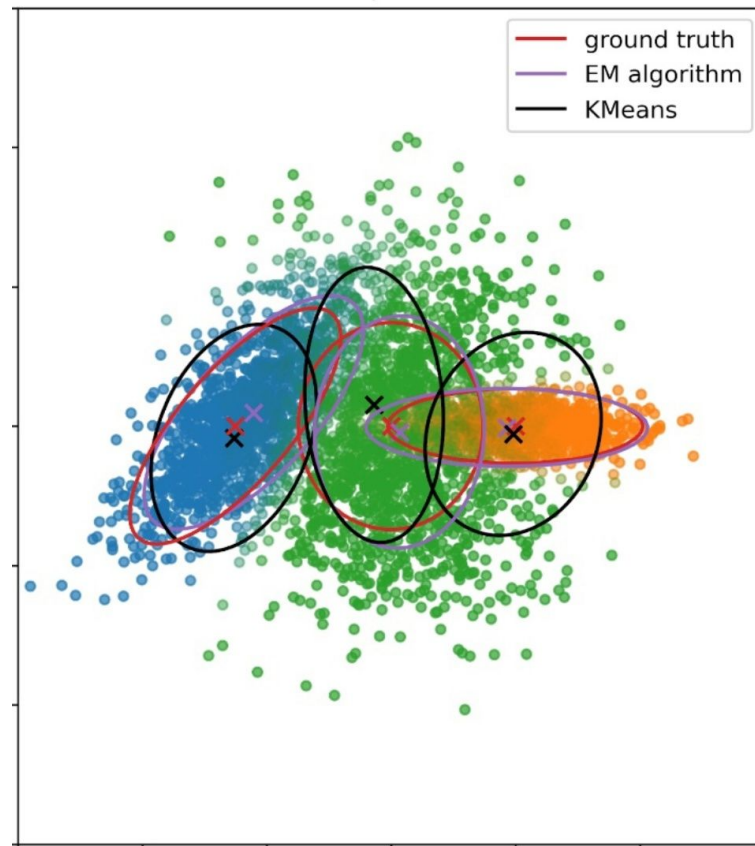


Kmeans плохо
определяет
распределения

Сравнение Kmeans и EM-алгоритма



Реализация и сравнение двух алгоритмов



Верная кластеризация

Модификация алгоритма - распределение Лапласа

Expectation-step:

$$f(x) = \frac{\alpha}{2} e^{-\alpha|x-\beta|}$$

Плотность вероятности

$$Q(\theta|\theta_{old}) = \sum_i P(z_i = 1|x, \theta_{old})[\ln f(x_i|\theta) + \ln p_1] + P(z_i = 2|x, \theta_{old})[\ln f(x_i|\theta) + \ln(p_2)] \\ + (1 - P(z_i = 1|x, \theta_{old}) - P(z_i = 2|x, \theta_{old}))[\ln f_L(x_i|\theta) + \ln(1 - p_1 - p_2)]$$

Q практически не меняется

Maximization-step - все параметры

$$\frac{1}{a^{new}} = \frac{\sum_i P(z_i = 1|x, \theta_{old}) |x_i - b|}{\sum_i P(z_i = 1|x, \theta_{old})}$$

$$b^{new} = \frac{\sum_i P(z_i = 1|x, \theta_{old}) x_i}{\sum_i P(z_i = 1|x, \theta_{old})}$$

Формулы получаются простым дифференцированием Функции Q с 2 нормальными и 1 лапласовским распределением

Реализация EM-алгоритма(с модификацией)

```
def em_algorithm_mody(X, num_components, num_iterations):
    n, d = X.shape
    # Initialize parameters
    np.random.seed(42)
    means = np.random.rand(num_components, d)
    covariances = [np.eye(d) for _ in range(num_components)]
    weights = np.ones(num_components) / num_components
    scales = np.ones(num_components) # Initialize scales

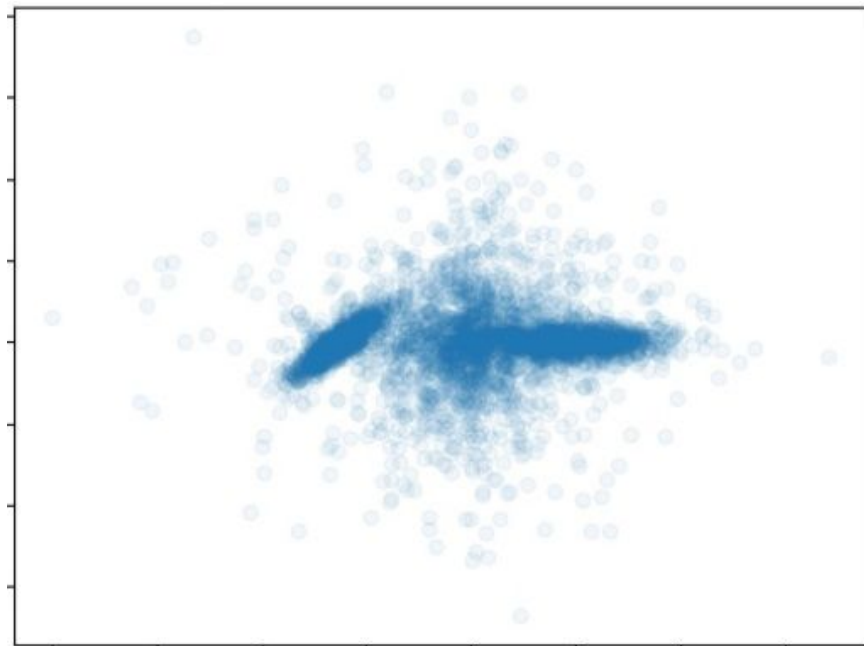
    for _ in range(num_iterations):
        # E-step
        probs = np.zeros((n, num_components))
        for i in range(num_components):
            if i < num_components - 1:
                probs[:, i] = weights[i] * (
                    multivariate_normal.pdf(X, mean=means[i], cov=covariances[i])
                )
            else:
                probs[:, i] = weights[i] * laplace_pdf(X[:, 0], X[:, 1], means[i][0], means[i][1], scales[i])
        probs_2 = probs / probs.sum(axis=1)[:, np.newaxis]
        probs = probs_2

        # M-step
        for i in range(num_components):
            weights[i] = np.sum(probs[:, i]) / n
            means[i] = np.sum(X * probs[:, i][:, np.newaxis], axis=0) / np.sum(probs[:, i])
            diff = X - means[i]
            covariances[i] = np.dot(diff.T, diff * probs[:, i][:, np.newaxis]) / np.sum(probs[:, i])
            if i == num_components - 1:
                scales[i] = np.sum(np.abs((X - means[i])) * probs[:, i][:, np.newaxis]) / np.sum(probs[:, i])

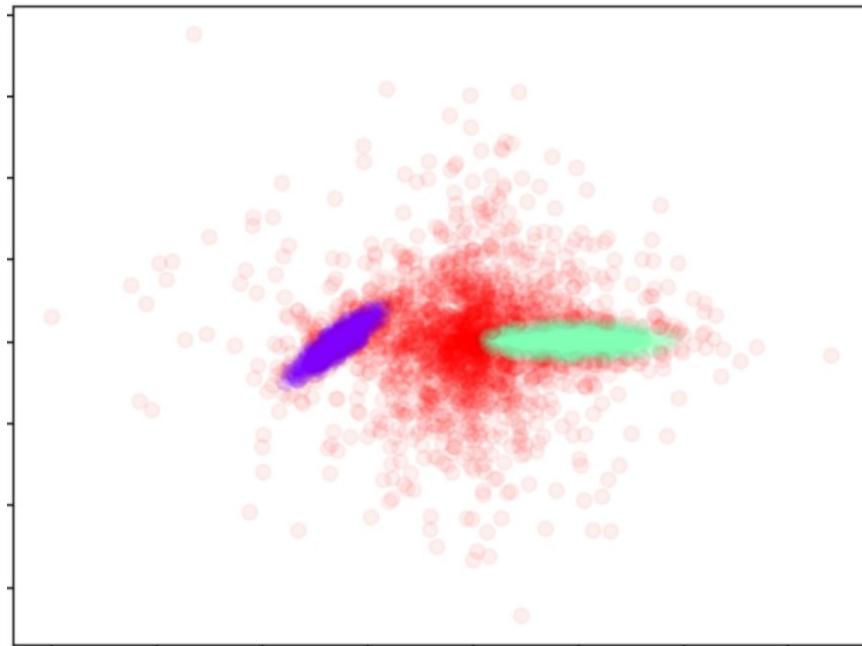
    return means, covariances, weights, scales, probs
```



Ем-алгоритм с модификацией

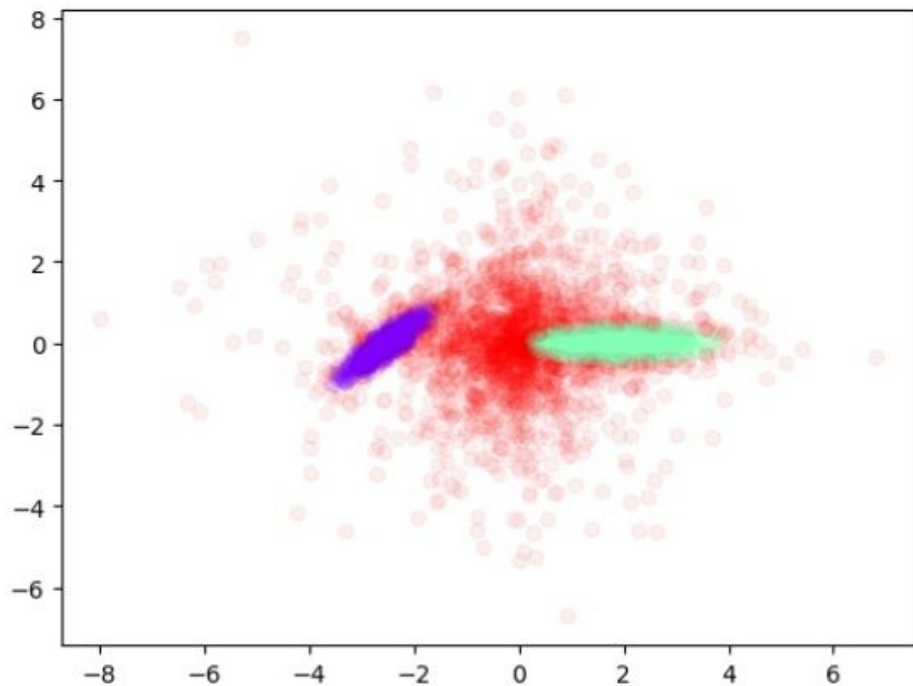


второй датасет

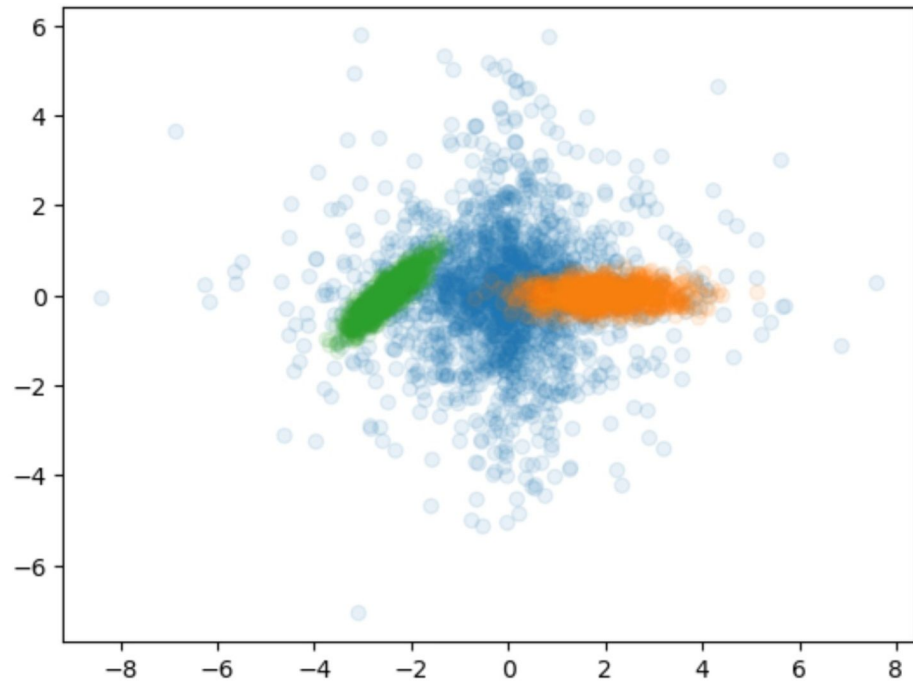


ЕМ-алгоритм с модификацией

Em-алгоритм с модификацией

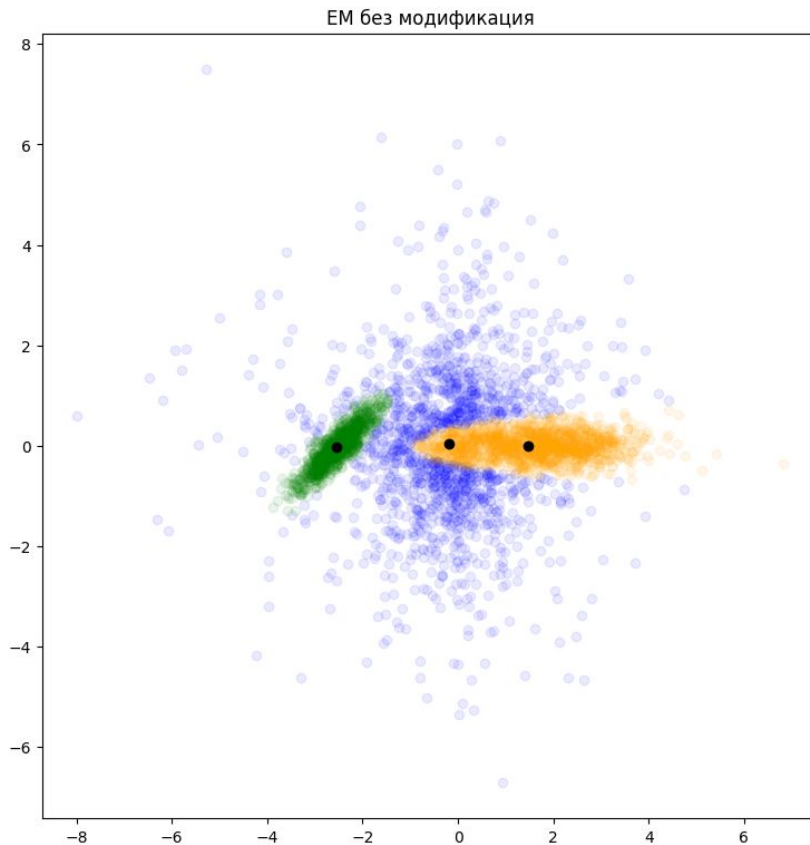
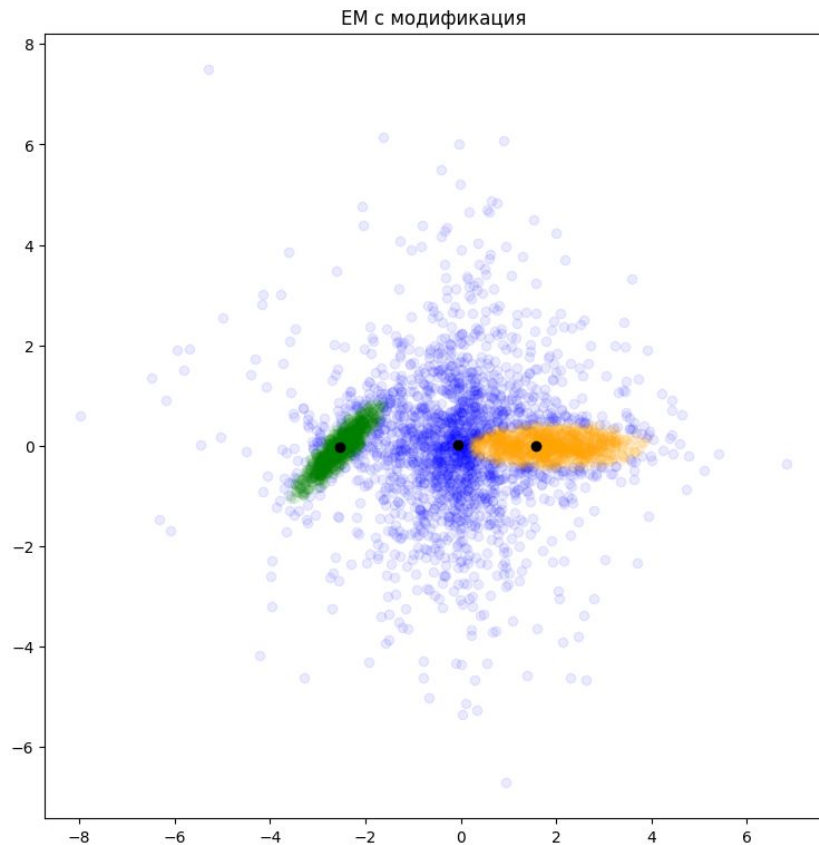


Реализация модификации

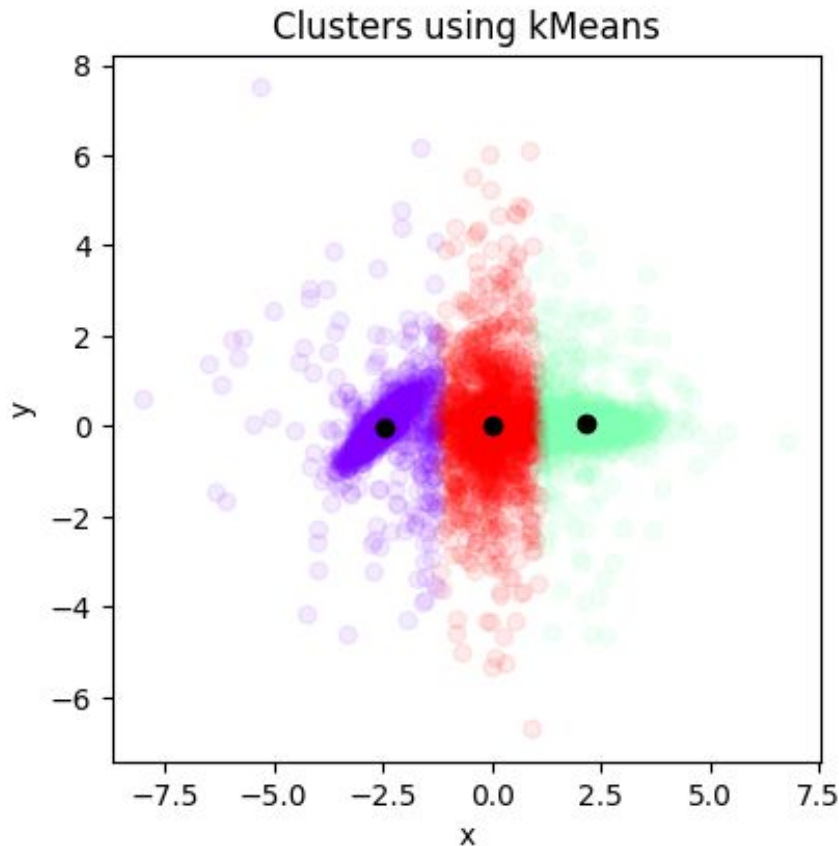
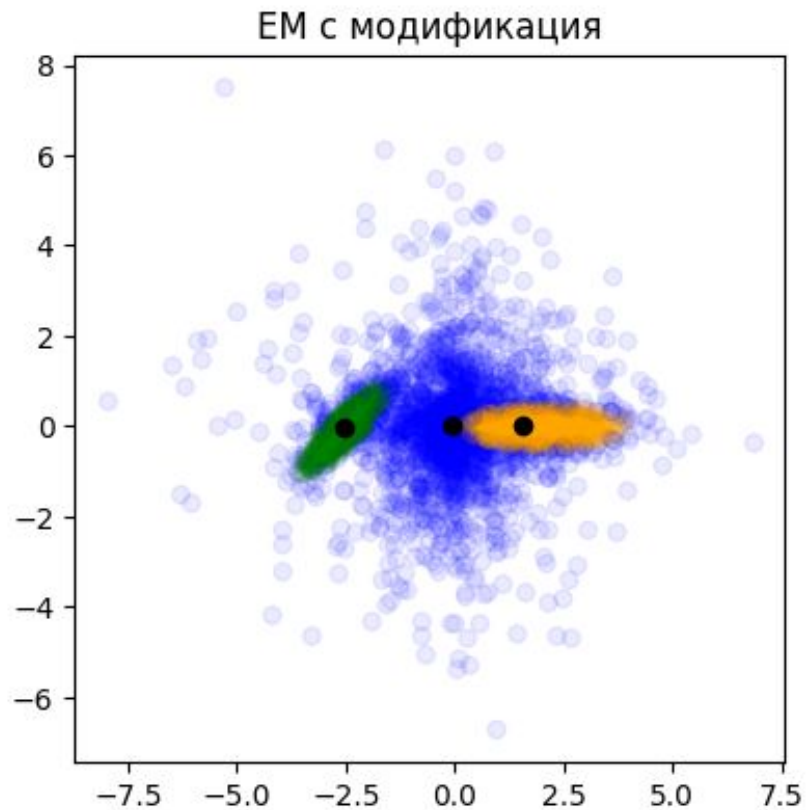


Верная кластеризация

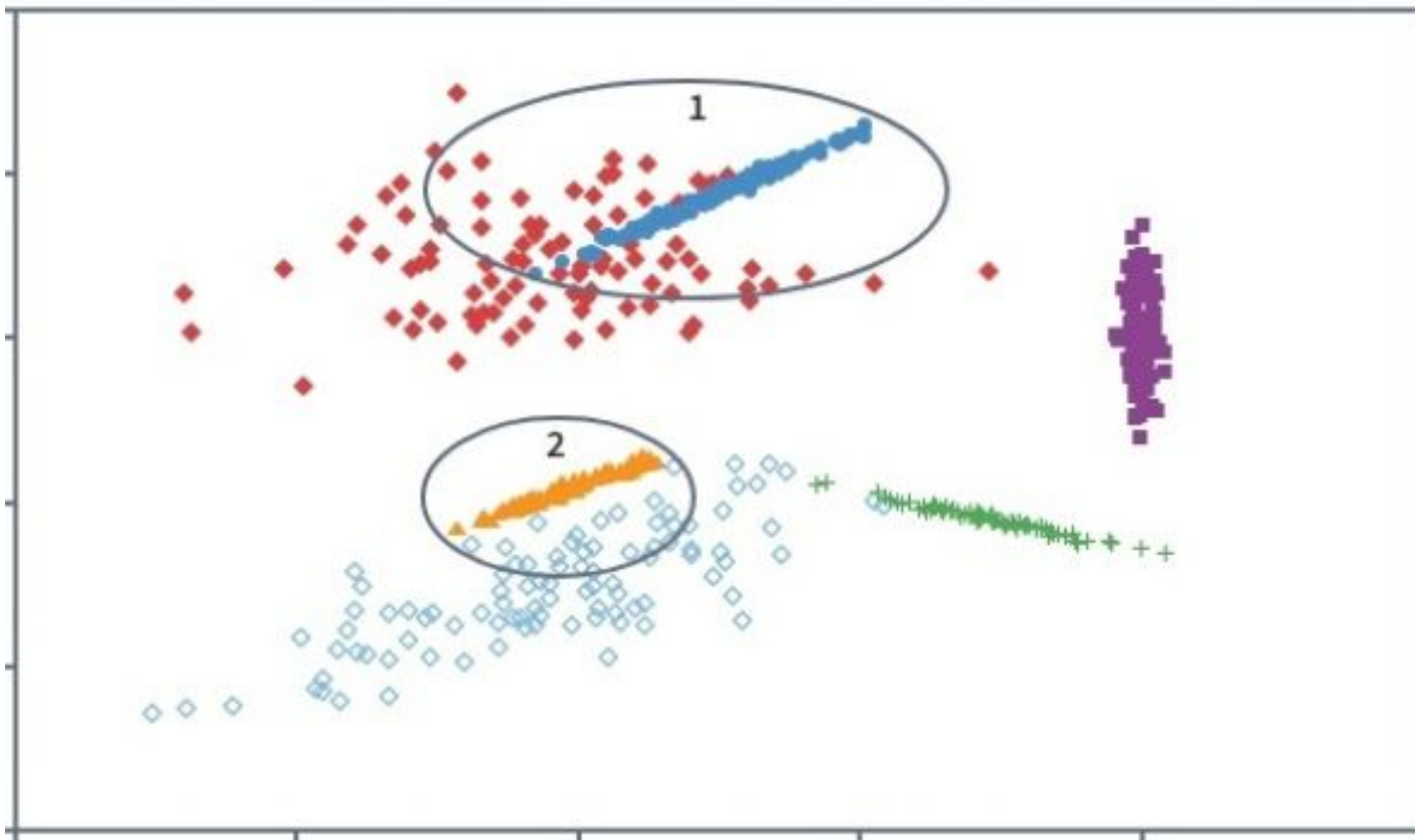
ЕМ-алгоритм с модификацией и без



Сравнение EM-алгоритма с модификацией и KMeans

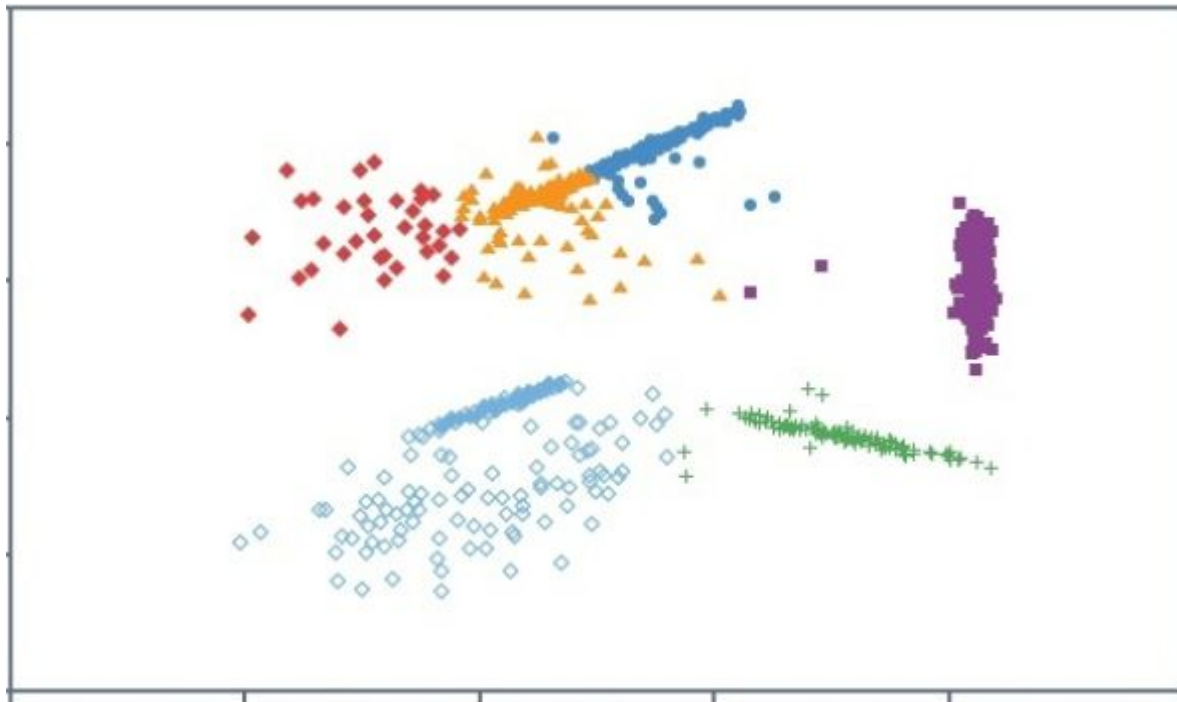


Сильные и слабые стороны алгоритмов



Правильная кластеризация

Сильные и слабые стороны алгоритмов

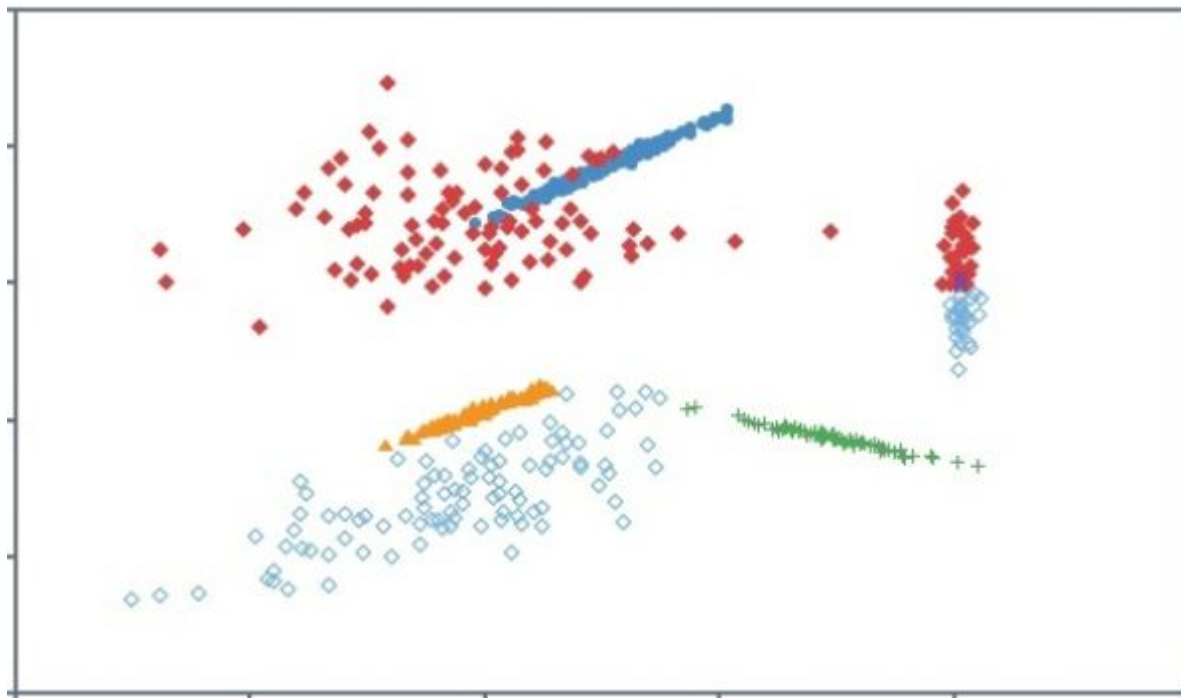


Результат работы KMeans

KMeans:

1. Хорошо работает с обособленными кластерами(фиол, зел)
2. Плохо работает с перекрывающимися кластерами

Сильные и слабые стороны алгоритмов

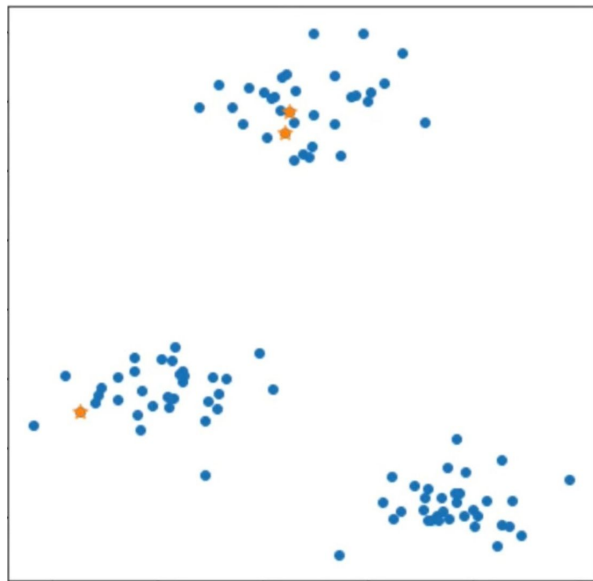


Результат работы EM-алгоритма

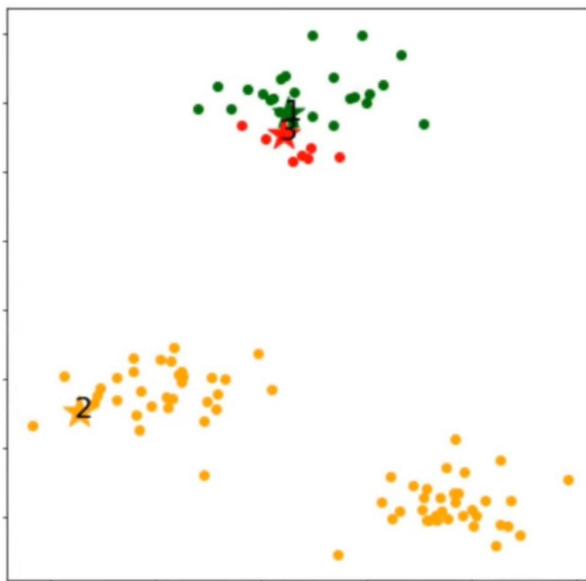
ЕМ-алгоритм:

1. Хорошо работает с распределениями и перекрывающимися кластерами
2. Иногда плохо распознает обособленный кластер

Недостатки обоих методов



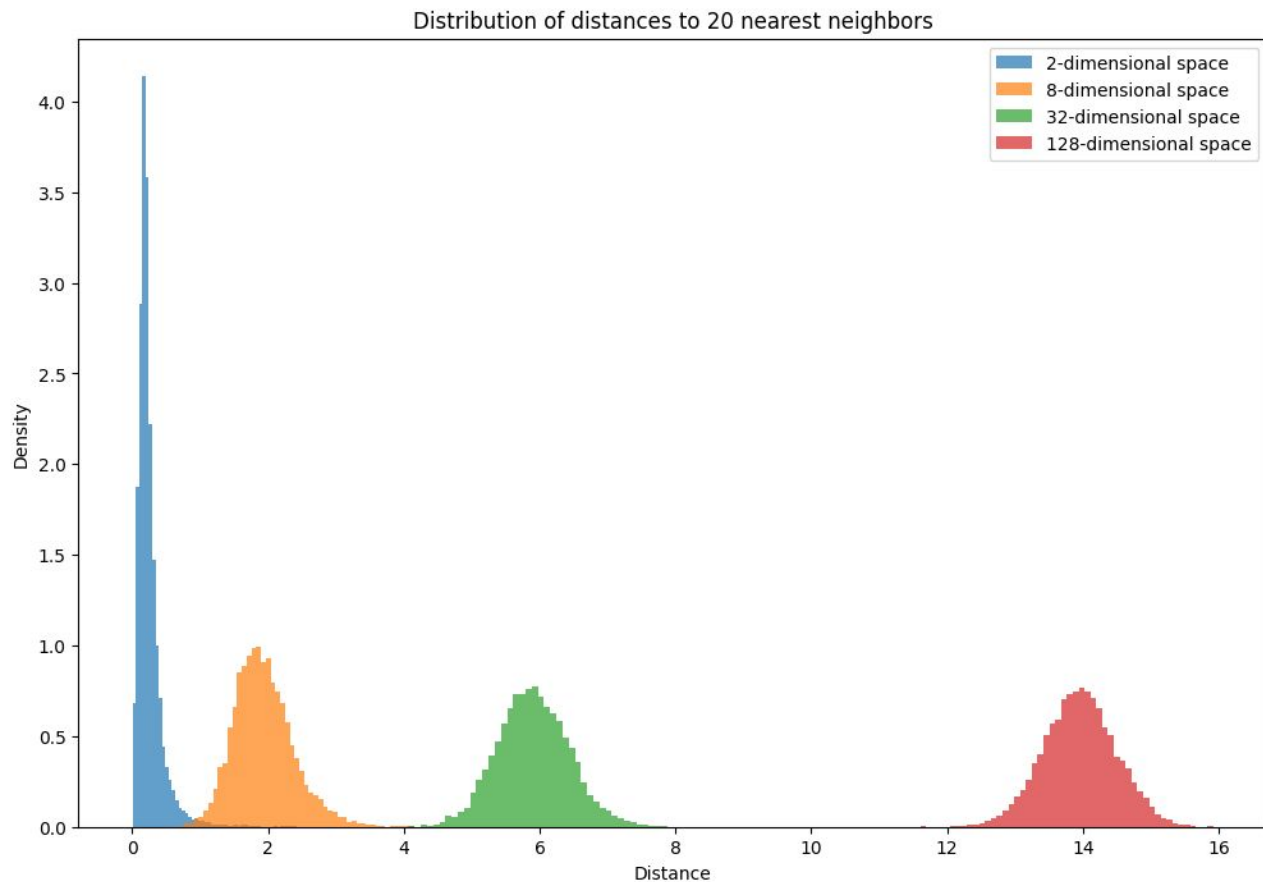
Неудачная инициализация центров



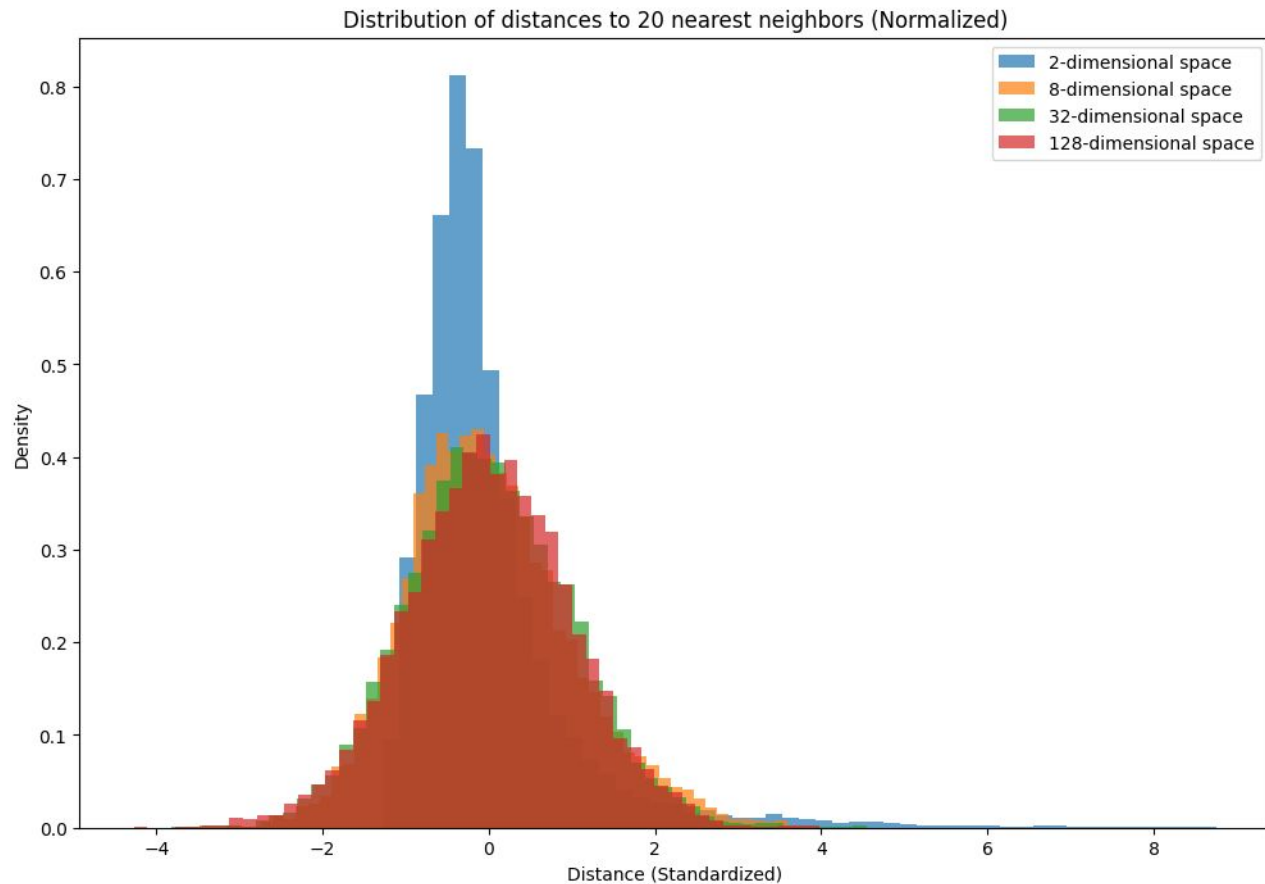
результат неудачной кластеризации

1. Инициализация центров
2. Гиперпараметр - кол-во кластеров
3. Предположение о распределении
4. Скорость работы не постоянна

Недостатки обоих методов - проклятие размерности



Проклятие размерности - решение



Повторяемость результатов

https://github.com/LeonardCaceres/IITP/blob/main/IPPI_enter.ipynb



Вывод

Ссылки

1. [Лекция — Прикладная статистика в машинном обучении \(v-marco.github.io\)](https://v-marco.github.io)
2. [Семинар — Прикладная статистика в машинном обучении \(v-marco.github.io\)](https://v-marco.github.io)
3. [EM_Algorithm.pdf \(columbia.edu\)](https://columbia.edu)
4. [Учебник по машинному обучению \(yandex.ru\)](https://yandex.ru)
5. [Википедия — свободная энциклопедия \(wikipedia.org\)](https://wikipedia.org)
6. [Mathematics for Machine Learning | Companion webpage to the book “Mathematics for Machine Learning”. Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press. \(mml-book.github.io\)](https://mml-book.github.io)
7. [ЕМ-алгоритм — Викиконспекты \(ifmo.ru\)](https://ifmo.ru)
8. [ЕМ — масштабируемый алгоритм кластеризации | Loginom](https://loginom.com)

<https://disk.yandex.ru/i/2jE5GbNsMhkPzQ> - ссылка на резюме(через неделю пропадет)