EM-алгоритм.

Ссылка на видеокурс от МИАН

Сегодня рассматриваются методы EM-алгоритм (**E**xpectation-**M**aximization — ожидание-максимизация) и k-средних (k-means), который является упрощением первого. Оба метода из информации, представленной в виде многомерных данных, выделяют классы, на которую ее можно разбить. Классификация выполняется автоматически (без учителя).

Постановка задачи и идея

Имеем множество X, на котором определена некоторая метрика ρ. X — метрическое пространство. В качестве X удобно рассматривать пространство $\mathbb{R}^N$ с евклидовой метрикой $\|\cdot\|$.

Имеем множество образцов $X_L \subset X$ — обучающая выборка. Задача состоит в том, чтобы по множеству X_L построить метод классификации, который будет разбивать все множество X по аналогии с X_L на K кластеров. Количество классов K задано заранее.

Пусть X_L^k, $k = 1, 2, \ldots, K$ представляет собой множество с элементами k-го класса, тогда

$$X_L = \bigcup_{k=1}^{K} X_L^k.$$

Пусть имеется **центральный элемент** $\mu_k \in M_k$.

Рассмотрим произвольный $x \in X_L$. Если этот $x \in X_L^{k^*}$ $k^* \in \{1, 2, \ldots, K\}$, то

$$\rho(x, \mu_{k^*}) = \min_{k=1,\ldots,K} \rho(x, \mu_k). \tag{1}$$

То есть элемент обучающей выборки x принадлежит классу k^*, если x **ближе всего** к его центру тяжести μ_{k^*}.

Если окажется так, что $\rho(x, \mu_{k^*}) = \rho(x, \mu_{k^{**}})$, то нужно принять дополнительное соглашение о выборе класса для x. Пусть, например, из равносильных классов будет выбран тот, что имеет меньший номер. Тогда запись (1) будет однозначно определять класс, к которому принадлежит выбранный x.

Алгоритм k-средних

1. Инициализируем $\{\mu_k\}$ произвольно, таким образом, что $\mu_k \neq \mu_l$ при $k \neq l$. Это могут быть случайные элементы множества X_L. Выбор центров является наиболее важным шагом, определяющим исход кластеризации. Если этот выбор неудачен, то следует улучшить схему инициализации.

2. Выбираем произвольный элемент $x \in X_L$. На этом элементе происходит обучение.

3. Как только выбраны центры — получена разбивка на классы

   $$X_L^k = \{\mu_k\}.$$

   Находим k^*, такой, что

   $$\rho(x, \mu_{k^*}) = \min_k \rho(x, \mu_k),$$

   то есть $x \in X_L^{k^*}$.

4. Добавляем x в найденное подмножество

   $$X_L^{k^*} = X_L^{k^*} \cup \{x\}.$$

5. Проводим процедуру усреднения. Находим новый центр класса

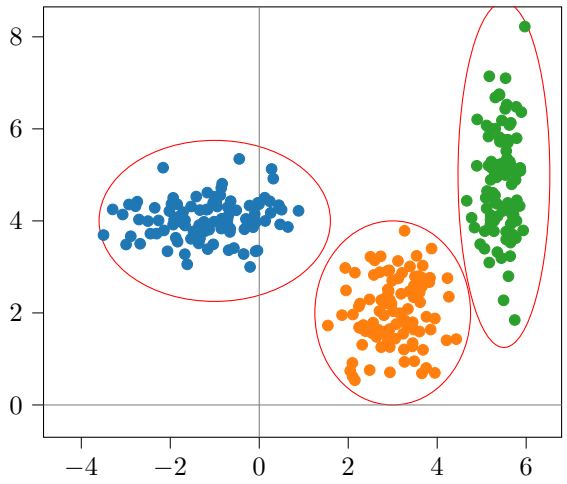
$$\mu_{k^*} = \frac{x^1_{k^*} + x^2_{k^*} + \cdots + x^{|X^{k^*}_L|}_{k^*}}{|X^{k^*}_L|}.$$

6. Переход к шагу 2.

На выходе получаем усредненные $\{\mu_k\}$, которые занимают точные положения центров классов.

Многомерное нормальное распределение

EM-алгоритм применяется в предположении, что классифицируемые многомерные данные представляют собой смесь многомерных **нормальных распределений**, то есть каждый класс описывается некоторым нормальным распределением.



На графике представлен пример смеси из двухкомпонентных нормальных распределений. **Математическое ожидание** — центр эллипса (эллипсоида, если $N > 2$) рассеивания, а полуоси — **среднеквадратичные отклонения** (корень квадратный из **дисперсии**) каждой из компонент.

Задача обучения состоит в отнесении объекта к соответствующему кластеру, иными словами, нахождения параметров каждого из нормальных распределений.

Основанием для такого предположения, как правило, являются предельные законы в теории вероятностей, в частности **центральная предельная теорема**. Утверждение состоит в том, что **сумма** большого количества слабо зависимых (независимых) случайных величин (которые либо одинаково распределены, либо распределены по-разному, но так, что ни одна из них не превалирует над другими) имеет распределение, близкое к нормальному. На практике это часто применяется так, что сумма независимых и часто неизвестных факторов аппроксимируется нормальным распределением.

EM-алгоритм

Будем рассматривать элементы из N-мерного пространства $x \in \mathbb{R}^N$. Обучающая выборка — это некоторые, выбранные нами элементы $\{x^m\}_{m=1}^M$, где $x^m = (x^m_1, x^m_2, \ldots, x^m_N)$ $m = 1, \ldots, M$. Традиционно, $K > 1$ — количество классов, на которое требуется разбить исходное множество.

Каждый класс описывается многомерным нормальным распределением. Математическое ожидание для каждого кластера есть N-мерный вектор

$$\mu_k = (\mu_{k1}, \mu_{k2}, \ldots, \mu_{kN}), \quad k = 1, 2, \ldots, K,$$

у которого компоненты суть математические ожидания соответствующих компонент некоторого вектора x, принадлежащего этому кластеру.

Многомерное нормальное распределение описывается **ковариационной матрицей**. Для простоты будем считать, что каждая компонента вектора x будет независимой друг от друга случайной величиной. Поэтому ковариации между компонентами равны 0 и матрица ковариации диагональная — на диагонали дисперсии каждой из компонент

$$\sigma_k^2 = (\sigma_{k1}^2, \sigma_{k2}^2, \ldots, \sigma_{kN}^2), \quad k = 1, 2, \ldots, K.$$

Целью алгоритма является построение вероятностей, что объект x^m принадлежит кластеру k. Эти вероятности будем обозначать через g_{mk}. Ясно, что

$$\sum_{k=1}^{K} g_{mk} = 1, \quad m = 1, 2, \ldots, M.$$

Как и для сети Кохонена, следует проводить процедуру нормировки исходных данных.

Приведем сам алгоритм.

1. Инициализируем веса
$$w_k = \frac{1}{K}, \quad k = 1, 2, \ldots, K.$$
2. Инициализируем математические ожидания для каждого кластера
$$\mu_{kn} = x_n^k, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, N.$$
3. Инициализируем дисперсии для каждого кластера
$$\sigma_{kn}^2 = \frac{1}{MK} \sum_{m=1}^{M} (x_n^m - \mu_{kn})^2, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, N.$$
Для нахождения дисперсии центрируем на мат. ожидание.
Как и в предыдущем методе, исход кластеризации зависит от удачного выбора инициализирующих значений.
4. **E-шаг**
Проводим оценку вероятностей
$$g_{mk} = \frac{w_k p_k(x^m)}{w_1 p_1(x^m) + w_2 p_2(x^m) + \cdots + w_K p_K(x^m)}, \quad m = 1, 2, \ldots, M, \quad k = 1, 2, \ldots, K,$$
где $p_k(x)$ — плотность многомерного нормального распределения, соответствующее k-ому кластеру
$$p_k(x) = \frac{1}{\sqrt{\det \sum_k} (2\pi)^{N/2}} \exp\left\{ -\frac{1}{2} \sum_{n=1}^{N} \left(\frac{x_n - \mu_{kn}}{\sigma_{kn}} \right)^2 \right\},$$
где $\sum_k$ — матрица ковариаций k-го класса, $\det \sum_k = \sigma_{k1}^2 \cdot \sigma_{k2}^2 \ldots \sigma_{kN}^2$. Ясно, что матрица ковариаций не должна вырождаться.
5. **M-шаг**
Обновление коэффициентов
$$w_k = \frac{1}{M} \sum_{m=1}^{M} g_{mk}, \quad k = 1, 2, \ldots, K,$$
$$\mu_{kn} = \frac{1}{M w_k} \sum_{m=1}^{M} g_{mk} x_n^m, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, N,$$
$$\sigma_{kn}^2 = \frac{1}{M w_k} \sum_{m=1}^{M} g_{mk} (x_n^m - \mu_{kn})^2, \quad k = 1, 2, \ldots, K, \quad n = 1, 2, \ldots, N.$$

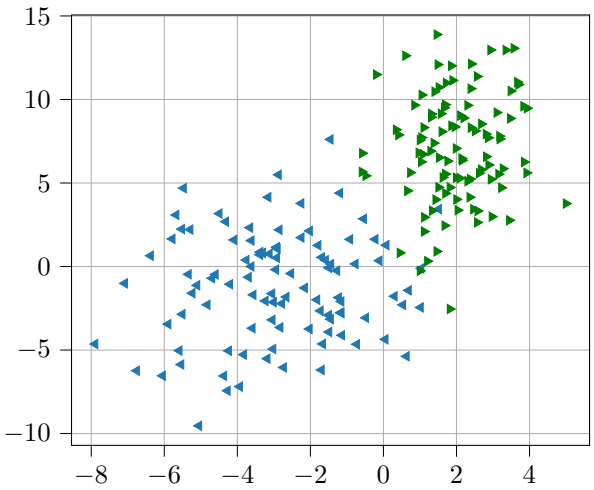
6. Повторяем шаги 4 и 5 заданное количество раз.

На выходе алгоритма получаем вероятности g_{mk}. Объект x^m принадлежит кластеру k^*, если

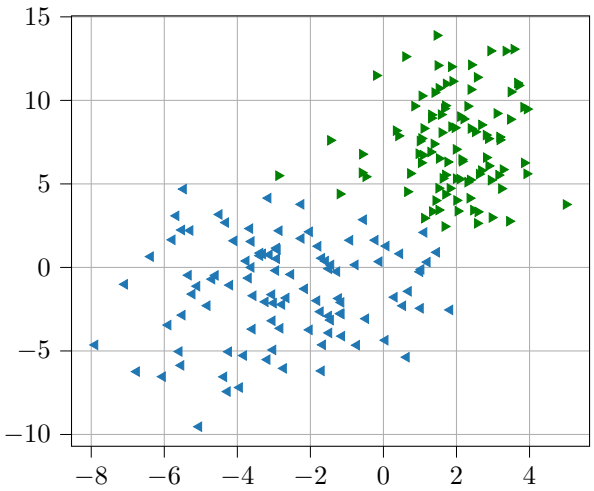
$$g_{mk^*} = \max_{k=1,2,\ldots,K} g_{mk}.$$

Пример 1. Смесь гауссиан

Сначала создадим искусственную выборку исходных данных (точек на плоскости), из смеси двух гауссиан.



Теперь разделим это множество на два кластера с помощью алгоритма k-means (за 1000 итераций).



Видим, что с высокой точностью алгоритм разделил заданную случайную смесь. Ошибся только в спорных точках.

Алгоритм хорошо себя показывает именно на смесях из ярко выраженных нормальных распределений. Если рассматривать более сложные кластерные структуры, то разбиение, скорее всего, будет некорректным.

Пример 2. Классификация предприятий

В качестве другого примера применим EM-алгоритм для кластеризации предприятий.

При $K = 2$ матрица вероятностей примет следующий вид:

Название	Гигантское	Крупное и среднее
Газпром	1.00	$3.75 \cdot 10^{-116}$
Новатэк	$1.75 \cdot 10^{-213}$	1.00
Лукойл	$6.46 \cdot 10^{-92}$	1.00
Сбер	1.00	$1.48 \cdot 10^{-100}$
Яндекс	$3.72 \cdot 10^{-199}$	1.00
МТС	$3.47 \cdot 10^{-247}$	1.00
Уралкалий	$2.82 \cdot 10^{-273}$	1.00
Казаньоргсинтез	$2.43 \cdot 10^{-271}$	1.00
ТМК	$3 \cdot 10^{-265}$	1.00
РКК “Энергия”	$4.51 \cdot 10^{-273}$	1.00

При $K = 3$:

Название	Гигантское	Крупное	Среднее
Газпром	1.00	$1.92 \cdot 10^{-69}$	0.00
Новатэк	0.00	$2.51 \cdot 10^{-2}$	0.974916613
Лукойл	$6.07 \cdot 10^{-245}$	1.00	$2.61 \cdot 10^{-22}$
Сбер	1.00	$1.34 \cdot 10^{-59}$	0.00
Яндекс	0.00	$1.2 \cdot 10^{-6}$	0.999998799
МТС	0.00	$5.99 \cdot 10^{-6}$	0.999994008
Уралкалий	0.00	$6.01 \cdot 10^{-8}$	0.99999994
Казаньоргсинтез	0.00	$6.19 \cdot 10^{-8}$	0.999999938
ТМК	0.00	$6.65 \cdot 10^{-7}$	0.999999335
РКК “Энергия”	0.00	$5.56 \cdot 10^{-8}$	0.999999944

Видим, что EM-алгоритм справился с классификацией предприятий также, как и сеть Кохонена.

EM-алгоритм использует совершенно иной подход к кластеризации данных, чем сеть Кохонена, но является не менее эффективным и рабочим.

Приложение

Код алгоритмов и комментарии к нему в приложении.