EM-алгоритм (1D)

Владимир Яшин

17 октября 2016 г.

1 Задача Expectation-Maximization алгоритма (1D)

Часто у нас нет возможности аналитически вычислить параметры некоторого распределения методом Максимального правдоподобия. В этом случае часто прибегают к итеративному поиску этих параметров, используя EM-алгоритм (Expectation-Maximization algorithm).

Рассмотрим два вектора размеров n и m: $\mathbf{X}_1 \in \mathbb{R}^n$ и $\mathbf{X}_2 \in \mathbb{R}^m$, набранные из нормальных Гауссовских распределений с разными параметрами среднего и дисперсии (например, рост мужчин и рост женщин)

$$\mathbf{X}_1 = \left\{x_i | x_i \sim \mathcal{N}(\mu_1, \sigma_1^2) \quad \forall i \in \{1..n\}\right\}$$
$$\mathbf{X}_2 = \left\{x_j | x_j \sim \mathcal{N}(\mu_2, \sigma_2^2) \quad \forall j \in \{1..m\}\right\}.$$

Оба вектора значений обычно собраны в один вектор длинны $n + m$: $\mathbf{X} \in \mathbb{R}^{n+m}$, распределенный по некоторому совместному распределению (не читать как joint. Мы рассматриваем 1D вариант).

$$\mathbf{X} = \left\{x_k | x_k \quad \forall k \in \{1..(m+n)\}\right\}$$

Желательно, чтобы распределение $\mathbf{X}$ было двумодальным, то есть средние исходных распределений отличались друг от друга. Также, в практических целях важно знать, какое из средних в исходных распределениях больше. Последнее не является необходимым для исполнения алгоритма, но такое знание будет полезным при интерпретации результатов. Например, если бы мы не имели апостериорного знания о том, что мужчины зачастую выше женщин, то результатом исполнения EM-алгоритма было бы то, что рост в исходных выборках отличается, но нельзя было бы сделать вывод, кто, в среднем, выше мужчины или женщины.

Задачей EM-алгоритма в данном случае является поиск четырёх параметров исходных распределений (μ_1, σ_1^2, μ_2 и σ_2^2). Алгоритм получает на вход: вектор $\mathbf{X}$ данные о длине двух векторов (n и m) и какое-то начальное приближение параметров исходных распределений.

Указанная задача решается в два шага: Expectaction и Maximization шаги.

Expectation-step

$$\mathbb{P}(\mathbf{X}_1|x_k) = \frac{\mathbb{P}(x_k|\mathbf{X}_1)\cdot\mathbb{P}(\mathbf{X}_1)}{\mathbb{P}(x_k|\mathbf{X}_1)\cdot\mathbb{P}(\mathbf{X}_1)+\mathbb{P}(x_k|\mathbf{X}_2)\cdot\mathbb{P}(\mathbf{X}_2)},$$

$$\mathbb{P}(\mathbf{X}_2|x_k) = \frac{\mathbb{P}(x_k|\mathbf{X}_2)\cdot\mathbb{P}(\mathbf{X}_2)}{\mathbb{P}(x_k|\mathbf{X}_1)\cdot\mathbb{P}(\mathbf{X}_1)+\mathbb{P}(x_k|\mathbf{X}_2)\cdot\mathbb{P}(\mathbf{X}_2)} \quad \forall k\in\{1..(m+n)\},$$

где $\mathbb{P}(\mathbf{X}_1)$ и $\mathbb{P}(\mathbf{X}_2)$ — доля первого и второго векторов в данном $\mathbf{X}$;
$\mathbf{X}_1 \sim \mathcal{N}(\mu_1,\sigma_1^2)$ и $\mathbf{X}_2 \sim \mathcal{N}(\mu_2,\sigma_2^2)$, тогда

$$\mathbb{P}(x_k|\mathbf{X}_1) = \frac{1}{\sqrt{2\pi\sigma_1^2}}\exp\left[-\frac{(x_k-\mu_1)^2}{2\sigma_1^2}\right],$$

$$\mathbb{P}(x_k|\mathbf{X}_2) = \frac{1}{\sqrt{2\pi\sigma_2^2}}\exp\left[-\frac{(x_k-\mu_2)^2}{2\sigma_2^2}\right].$$

Maximization-step

$$\mu_1 = \frac{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_1|x_k)\cdot x_k}{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_1|x_k)} \qquad \mu_2 = \frac{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_2|x_k)\cdot x_k}{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_2|x_k)},$$

$$\sigma_1^2 = \frac{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_1|x_k)\cdot(x_k-\mu_1)^2}{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_1|x_k)} \qquad \sigma_2^2 = \frac{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_2|x_k)\cdot(x_k-\mu_2)^2}{\sum_k^{m+n}\mathbb{P}(\mathbf{X}_2|x_k)}.$$

2 Expectation-Maximization-алгоритм

Пусть нам даны: вектор $\mathbf{X}$ из $\mathbb{R}^{n+m}$; начальное приближение четырёх параметров ($\overline{\mu_1}, \overline{\mu_2}$ и $\overline{\sigma_1^2}, \overline{\sigma_2^2}$); и мощности[1] исходных множеств (N_1 и N_2).

[1] Вообще используется доля $N_1/(N_1+N_2)$, чтобы нормализовать на единицу, но если, и в числителе, и в знаменателе вынести за скобку $1/(N_1+N_2)$, то можно будет сократить дробь и мощностей каждого из множеств достаточно для исполнения алгоритма.

Нулевая итерация:

$$\mu_{1_0} := \overline{\mu_1}$$
$$\sigma^2_{1_0} := \overline{\sigma^2_1}$$
$$\mu_{2_0} := \overline{\mu_2}$$
$$\sigma^2_{2_0} := \overline{\sigma^2_2}$$

E-step:

$$\mathbb{P}(x_k|\mathbf{X}_1)_0 := \frac{1}{\sqrt{2\pi\sigma^2_{1_0}}} \exp\left[-\frac{(x_k-\mu_{1_0})^2}{2\sigma^2_{1_0}} \right],$$
$$\mathbb{P}(x_k|\mathbf{X}_2)_0 := \frac{1}{\sqrt{2\pi\sigma^2_{2_0}}} \exp\left[-\frac{(x_k-\mu_{2_0})^2}{2\sigma^2_{2_0}} \right]$$

$$\mathbb{P}(\mathbf{X}_1|x_k)_0 := \frac{\mathbb{P}(x_k|\mathbf{X}_1)_0 \cdot N_1}{\mathbb{P}(x_k|\mathbf{X}_1)_0 \cdot N_1 + \mathbb{P}(x_k|\mathbf{X}_2)_0 \cdot N_2},$$
$$\mathbb{P}(\mathbf{X}_2|x_k)_0 := \frac{\mathbb{P}(x_k|\mathbf{X}_2)_0 \cdot N_2}{\mathbb{P}(x_k|\mathbf{X}_1)_0 \cdot N_1 + \mathbb{P}(x_k|\mathbf{X}_2)_0 \cdot N_2} \quad \forall k \in \{1..(m+n)\},$$

M-step:

$$\sigma^2_{1_1} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_0 \cdot (x_k-\mu_{1_0})^2}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_0}$$
$$\sigma^2_{2_1} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_0 \cdot (x_k-\mu_{2_0})^2}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_0}.$$
$$\mu_{1_1} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_0 \cdot x_k}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_0}$$
$$\mu_{2_1} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_0 \cdot x_k}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_0},$$

S-ая итерация:

E-step:

$$\mathbb{P}(x_k|\mathbf{X}_1)_s := \frac{1}{\sqrt{2\pi\sigma_{1_s}^2}} \exp\left[-\frac{(x_k-\mu_{1_s})^2}{2\sigma_{1_s}^2}\right],$$
$$\mathbb{P}(x_k|\mathbf{X}_2)_s := \frac{1}{\sqrt{2\pi\sigma_{2_s}^2}} \exp\left[-\frac{(x_k-\mu_{2_s})^2}{2\sigma_{2_s}^2}\right]$$

$$\mathbb{P}(\mathbf{X}_1|x_k)_s := \frac{\mathbb{P}(x_k|\mathbf{X}_1)_s \cdot N_1}{\mathbb{P}(x_k|\mathbf{X}_1)_s \cdot N_1 + \mathbb{P}(x_k|\mathbf{X}_2)_s \cdot N_2},$$
$$\mathbb{P}(\mathbf{X}_2|x_k)_s := \frac{\mathbb{P}(x_k|\mathbf{X}_2)_s \cdot N_2}{\mathbb{P}(x_k|\mathbf{X}_1)_s \cdot N_1 + \mathbb{P}(x_k|\mathbf{X}_2)_s \cdot N_2} \quad \forall k \in \{1..(m+n)\},$$

M-step:

$$\sigma_{1_{s+1}}^2 := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_s \cdot (x_k-\mu_{1_s})^2}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_s}$$
$$\sigma_{2_{s+1}}^2 := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_s \cdot (x_k-\mu_{2_s})^2}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_s}.$$
$$\mu_{1_{s+1}} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_s \cdot x_k}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_1|x_k)_s}$$
$$\mu_{2_{s+1}} := \frac{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_s \cdot x_k}{\sum_k^{m+n} \mathbb{P}(\mathbf{X}_2|x_k)_s},$$

Код на [vanilla] Питоне с исполнением EM-алгоритма можно найти в Листинге 1 или на ГитХабе в более потребном виде и вместе с примером.

Листинг 1: EM-algorithm for two one-dimensional Gaussians on (vanilla) Python.

```python
def em_alg(X, mu1_, sigma1_, mu2_, sigma2_, n1, n2, verbose=True):
    # function that returns a probability for a given quantile (norm dist)
    def dnorm(x, mu, sigma_sq):
        pi = 3.14159265359
        e  = 2.71828182846
        # calculating prob according to gaussian formula for normal pdf
        p = 1 / ((2 * pi * sigma_sq) ** 0.5) * e ** (-((x - mu) ** 2) / (2 * sigma_sq))
        return p

    # we need params list for calculating the diff between iterations
    params = [[mu1_, sigma1_, mu2_, sigma2_]]
    # setting up the initial values
    mu1, sigma1, mu2, sigma2 = params[0]
    i = 1
    delta = 1
    # while delta is too large impliment the algorithm
    while delta >= 0.1 and i <= 100:

        # E-step
        # calc the probability Pr(X_i | first dist) and Pr(X_i | second dist)
        P_X_1 = [dnorm(X[i], mu1, sigma1) for i in range(len(X))]
        P_X_2 = [dnorm(X[i], mu2, sigma2) for i in range(len(X))]
        # calc the Bayes' probability Pr(first dist | x_i) and Pr(second dist | x_i)
        # note: n1 should be the ratio of n1 in (n1 + n2) as well as n2
        # but you may factorize (n1 + n2) and eliminate this factor from fraction
        assert len(P_X_1) == len(P_X_2)
        P_1_X = [(P_X_1[i] * n1) / (P_X_1[i] * n1 + P_X_2[i] * n2) for i in range(len(P_X_1))]
        P_2_X = [(P_X_2[i] * n2) / (P_X_1[i] * n1 + P_X_2[i] * n2) for i in range(len(P_X_2))]

        # M-step
        # calc new mu and new sigma for both distributions
        assert len(P_1_X) == len(X)
        sigma1 = sum([P_1_X[i] * (X[i] - mu1) ** 2 for i in range(len(X))]) / sum(P_1_X)
        sigma2 = sum([P_2_X[i] * (X[i] - mu2) ** 2 for i in range(len(X))]) / sum(P_2_X)
        # taking the square root from sigmas
        sigma1, sigma2 = sigma1 ** 0.5, sigma2 ** 0.5
        mu1 = sum([P_1_X[i] * X[i] for i in range(len(X))]) / sum(P_1_X)
        mu2 = sum([P_2_X[i] * X[i] for i in range(len(X))]) / sum(P_2_X)
        # calc delta: the previous state - the new state and pop out the prev state
        params.append([mu1, sigma1, mu2, sigma2])
        delta = sum([abs(params[0][i] - params[1][i]) for i in range(len(params[0]))])
        params.pop(0)
    # if a user choose to see the progress
    if verbose == True:
        print('iteration:', i, 'delta =', delta)
    # add count after one iteration
    i += 1

    return mu1, sigma1, mu2, sigma2
```