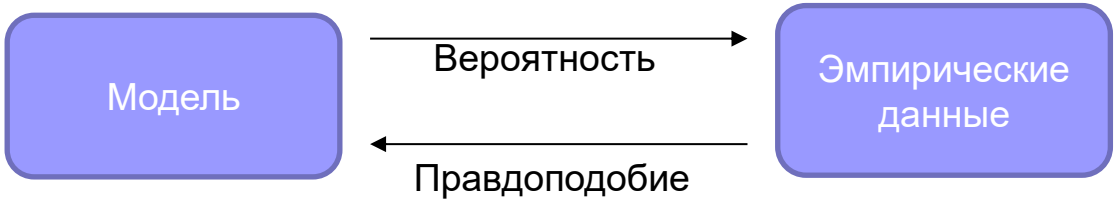
Лекция 10: Восстановление плотности распределения

Основные подходы к восстановлению плотности распределения

- Параметрические методы восстановления плотности:
  - Задача восстановления плотности распределения
  - Восстановление многомерной гауссовской плотности
  - Проблема мультиколлинеарности
- Непараметрическое восстановление плотности:
  - Восстановление одномерных плотностей
  - Восстановление многомерных плотностей
  - Выбор ядра и ширины окна
  - Аппроксимация восстановленной плотности
- Разделение смеси распределений:
  - Задача разделения смеси распределений
  - EM-алгоритм
  - Обобщения и модификации EM-алгоритма

Параметрическое восстановление плотности распределения



- Постановка задачи:
  - Дано: простая (**i.i.d.**) выборка $X^l = \{x_1, \dots, x_l\} \sim p(x)$.
  - Найти параметрическую модель плотности распределения: $p(x) = \varphi(x;\, \theta)$, где θ – параметр, φ – фиксированная функция.
- Критерий – максимум правдоподобия выборки:

$$L(\theta; X^l) = \ln \prod_{i=1}^{l} \varphi(x_i; \theta) = \sum_{i=1}^{l} \ln \varphi\,(x_i; \theta) \to max_\theta$$

- Необходимое условие оптимума:

$$\frac{\partial}{\partial \theta} L(\theta; X^l) = \sum_{i=1}^{l} \frac{\partial}{\partial \theta} \ln \varphi(x_i; \theta) = 0$$

  - где функция $\varphi(x; \theta)$ достаточно гладкая по параметру θ.

Простой пример с распределением Бернулли

- Дана выборка размера 8 случайной бинарной переменной, распределенной по закону Бернулли с неизвестным параметром p:

$$\begin{aligned} L(p) &= P(0,1,1,0,0,1,0,1|p) \\ &= P(0|p)P(1|p)\dots P(1|p) \\ &= (1-p)p\dots p \\ &= p^4(1-p)^4 \end{aligned}$$

- Надо найти p, максимизирующий логарифмическое правдоподобие $l(p) = \log[P(X|B(p))]$:

$$\begin{aligned} \ell(p) &= logL(p) = 4log(p) + 4log(1-p) \\ \frac{d\ell(p)}{dp} &= \frac{4}{p} - \frac{4}{1-p} \equiv 0 \\ \rightarrow p &= \frac{1}{2} \end{aligned}$$

Восстановление многомерной гауссовской плотности

- Пусть объекты x описываются n признаками и выборка порождена n-мерной гауссовской плотностью:

$$p(x) = N(x; \mu, \Sigma) = \frac{\exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}{\sqrt{(2\pi)^n \det \Sigma}}$$

$\mu \in \mathbb{R}^n$ – вектор математического ожидания, $\mu = Ex$

$\Sigma \in \mathbb{R}^{n \times n}$ – ковариационная матрица, $\Sigma = E(x-\mu)(x-\mu)^T$
(симметричная, невырожденная, положительно определенная)

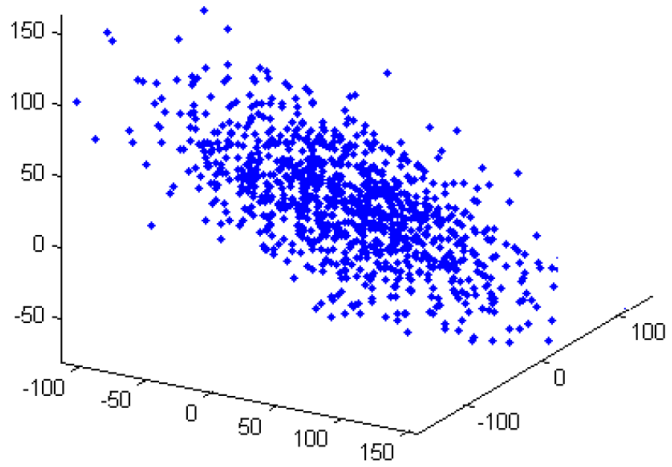
- Выборочные оценки максимального правдоподобия:

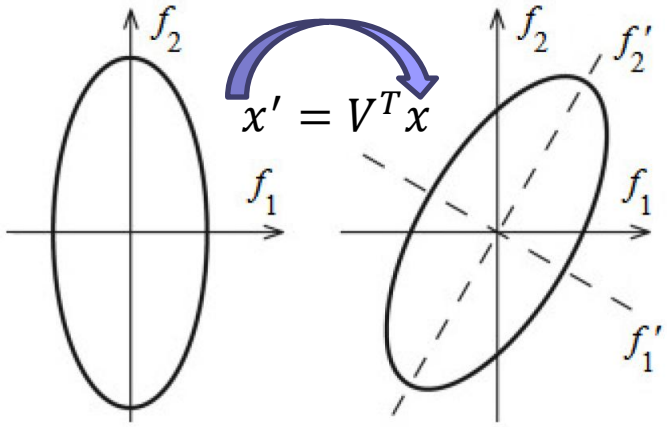
$$\frac{\partial}{\partial \mu} \ln L\left(\mu, \Sigma; X^l\right) = 0 \quad \Rightarrow \hat{\mu} = \frac{1}{l}\sum_{i=1}^{l} x_i$$

$$\frac{\partial}{\partial \Sigma} \ln L\left(\mu, \Sigma; X^l\right) = 0 \quad \Rightarrow \hat{\Sigma} = \frac{1}{l}\sum_{i=1}^{l} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

Геометрический смысл многомерной нормальной плотности

- Эллипсоид рассеяния – облако точек эллиптической формы.





- При $\Sigma = diag(\sigma_1^2, \dots, \sigma_n^2)$ оси эллипсоида параллельны осям.
- В общем случае: $\Sigma = VSV^T$ – спектральное разложение:
  - $V = (v_1, \dots, v_n)$ – ортогональные собственные векторы, $V^T V = I_n$
  - $S = diag(\lambda_1, \dots, \lambda_n)$ – собственные значения матрицы Σ
  - $(x-\mu)^T \Sigma^{-1} (x-\mu) = (x-\mu)^T V S^{-1} V^T (x-\mu) = (x'-\mu')^T S^{-1} (x'-\mu')$
  - $x' = V^T x$ – ортогональное преобразование поворот / отражение.

Проблема мультиколлинеарности

- Проблема:
  - при $l < n$ матрица $\hat{\Sigma}$ вырождена, но даже при $l \geq n$ она может оказаться плохо обусловленной.
- Регуляризация ковариационной матрицы $\hat{\Sigma} + \tau I_n$
  - увеличивает собственные значения на τ, сохраняя собственные векторы (параметр τ можно подбирать по скользящему контролю).
- Диагонализация ковариационной матрицы
  - оценивание n одномерных плотностей признаков $f_j(x), j = 1, \dots, n$:

$$\hat{p}_j(\xi) = \frac{1}{\sqrt{2\pi}\hat{\sigma}_j} \exp\left(-\frac{\left(\xi - \hat{\mu}_j\right)^2}{2\hat{\sigma}_j^2}\right), j = 1, \dots, n$$

  - где $\hat{\mu}_j$ и $\hat{\sigma}_j^2$ – оценки среднего и дисперсии признака j:

$$\hat{\mu}_j = \frac{1}{l}\sum_{i=1}^{l} f_j(x_i),\ \hat{\sigma}_j^2 = \frac{1}{l}\sum_{i=1}^{l}\left(f_j(x_i) - \hat{\mu}_j\right)^2$$

Задача непараметрического восстановления плотности

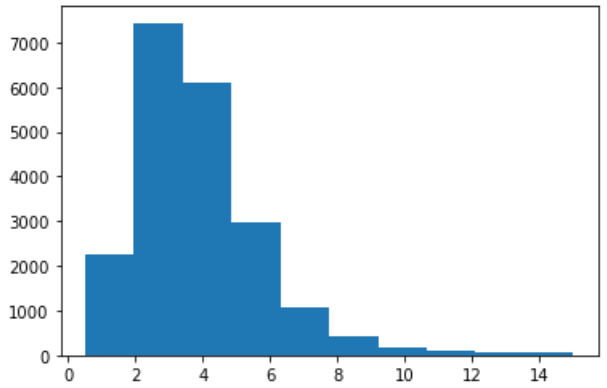
- Задача:
  - по выборке $X^l = \{x_i\}_{i=1}^l$ оценить плотность $\hat{p}(x)$, **без параметрической модели**
- Дискретный случай $x_i \in D, |D| \ll l$.
  - Гистограмма частот: $\hat{p}(x) = \frac{1}{l}\sum_{i=1}^{l}[x_i = x]$
- Одномерный непрерывный случай: $x_i \in \mathbb{R}$.
  - По определению плотности, если $P[a,b]$ – вероятностная мера отрезка $[a,b]$: $p(x) = \lim_{h\to 0}\frac{1}{2h}P[x-h, x+h]$
- Эмпирическая оценка плотности по окну ширины h (заменяем вероятность долей объектов выборки):

$$\hat{p}_h(x) = \frac{1}{2h}\frac{1}{l}\sum_{i=1}^{l}[|x - x_i| < h]$$

Локальная непараметрическая оценка Парзена-Розенблатта

- Эмпирическая оценка плотности по окну ширины h:

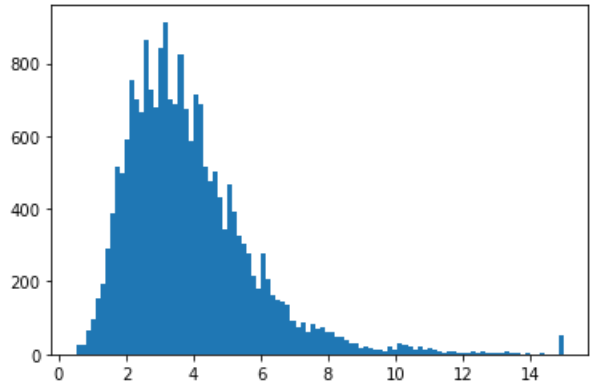
$$\hat{p}_h(x) = \frac{1}{lh}\sum_{i=1}^{l}\frac{1}{2}\left[\frac{|x - x_i|}{h} < 1\right]$$



- Обобщение:
  - оценка Парзена-Розенблатта по окну ширины h:

$$\hat{p}_h(x) = \frac{1}{lh}\sum_{i=1}^{l} K\left(\frac{x - x_i}{h}\right),$$

- $K(r)$ – ядро, удовлетворяющее требованиям:
  - четная функция;
  - нормированная функция: $\int K(r)\,dr = 1$;
  - невозрастающая при $r > 0$, неотрицательная функция.
  - В частности, при $K(r) = \frac{1}{2}[|r| < 1]$ имеем эмпирическую оценку (см выше)



Обоснование оценки Парзена-Розенблатта (Kernel Density Estimate)

- Теорема (одномерный случай, $x_i \in \mathbb{R}$). Пусть выполнены:
  - X^l – простая выборка из распределения $p(x)$;
  - Ядро $K(z)$ непрерывно и ограничено: $\int K^2(z)dz < \infty$;
  - Последовательность h_l: $\lim_{l\to\infty} h_l = 0$ и $\lim_{l\to\infty} lh_l = \infty$
  - Тогда: $\hat{p}_{h_l}(x) \to p(x)$ при $l \to \infty$ для почти всех $x \in X$ со скоростью $O(l^{-\frac{2}{5}})$
- Многомерный случай ($x_i \in \mathbb{R}^n$)
  - Если объекты описываются n признаками f_j: $X \to \mathbb{R}$:

$$\hat{p}_{h_1,\dots h_n}(x) = \frac{1}{l}\sum_{i=1}^{l}\prod_{j=1}^{n}\frac{1}{h_j}K\left(\frac{f_j(x) - f_j(x_i)}{h_j}\right)$$

  - Если на X задана функция расстояния $\rho(x, x^l)$:

$\hat{p}_h(x) = \frac{1}{lV(h)}\sum_{i=1}^{l} K\left(\frac{\rho(x,x_i)}{h}\right)$, где $V(h) = \int K\left(\frac{\rho(x,x_i)}{h}\right)dx$ – нормировочный множитель

Выбор ядра

- Функционал качества восстановления плотности (MI(ntegrated)SE):

$$J(K) = \int_{-\infty}^{+\infty} E\left(\hat{p}_h(x) - p(x)\right)^2 dx$$

- Популярные ядра:
  - оптимальное (Епанечникова), $J(K^*)/J(K)$ =1, $E(r) = \frac{3}{4}\left(1 - r^2\right)[|r| \le 1]$
  - квартическое, $J(K^*)/J(K)$ = 0.995, $Q(r) = \frac{15}{16}\left(1 - r^2\right)^2[|r| \le 1]$
  - треугольное, $J(K^*)/J(K)$ = 0.989

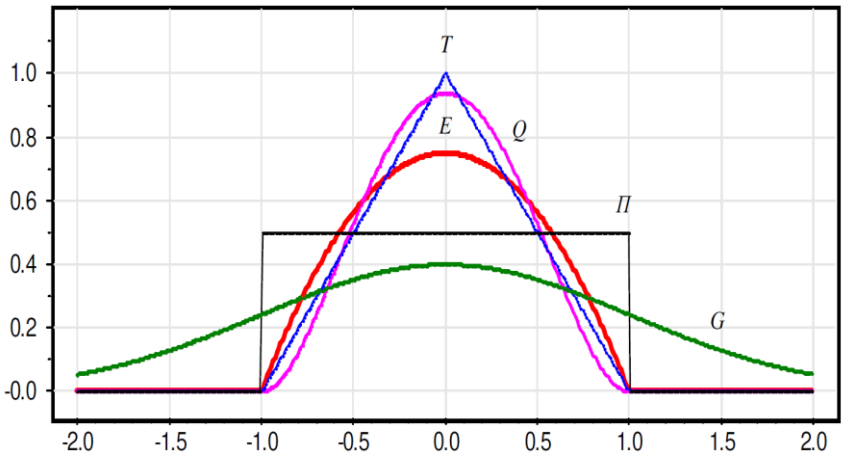
$T(r) = (1 - |r|)[|r| \le 1]$

  - гауссовское, $J(K^*)/J(K)$ = 0.961

$G(r) = (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}r^2\right)$

  - прямоугольное, $J(K^*)/J(K)$ = 0.943

$\Pi(r) = \frac{1}{2}[|r| \le 1]$



  - Асимптотические значения отношения $J(K^*)/J(K)$ при $l \to \infty$ **не зависят от вида распределения** $p(x)$

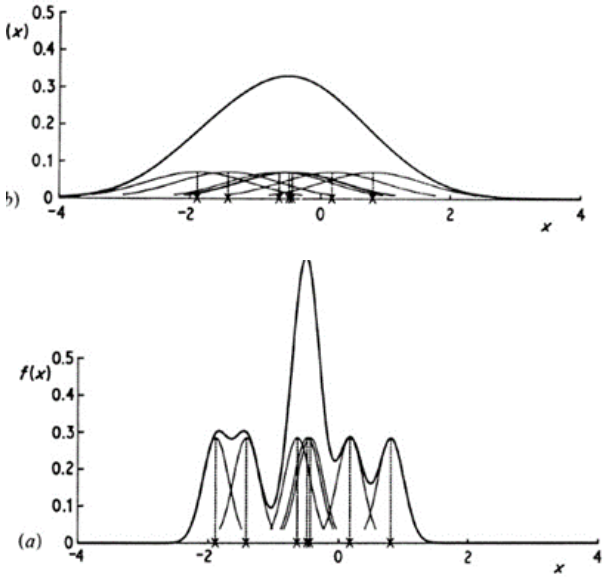
Зависимость оценки плотности от ширины окна

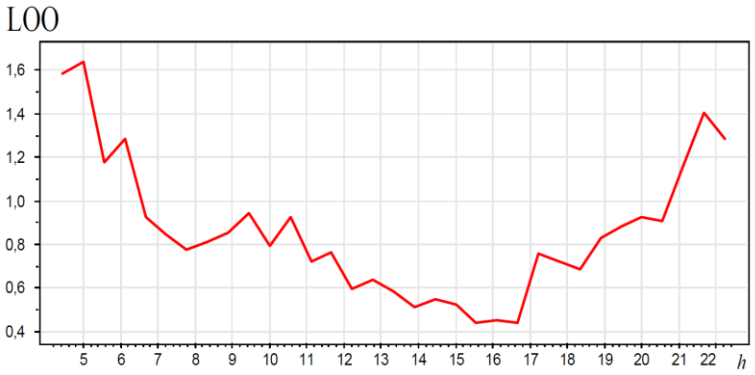
- Оценка $\hat{p}_h(x)$ при различных значениях ширины окна h
- Качество восстановления плотности зависит от ширины окна h, но слабо зависит от вида ядра K
- Можно выбирать ширину ядра через k – число соседей:

$$h_k(x) = \rho\left(x, x^{k+1}\right)$$

- Выбор ширины ядра (или числа соседей k) через CV или LO:

$$LOO(h) = -\sum_{i=1}^{l} \ln \hat{p}_h(x_i; X^l \backslash \mathrm{x}_i) \to min_h$$





Ядерные непараметрические методы анализа данных

- Восстановление плотности. Метод Парзена-Розенблатта:

$$\hat{p}_h(x) = \frac{1}{lV(h)} \sum_{i=1}^{l} K\left(\frac{\rho(x, x_i)}{h}\right)$$

- Классификация. Метод парзеновского окна:

$$a_h(x) = \arg\max_{y \in Y} \sum_{i=1}^{l} [y_i = y]\, K\left(\frac{\rho(x, x_i)}{h}\right)$$

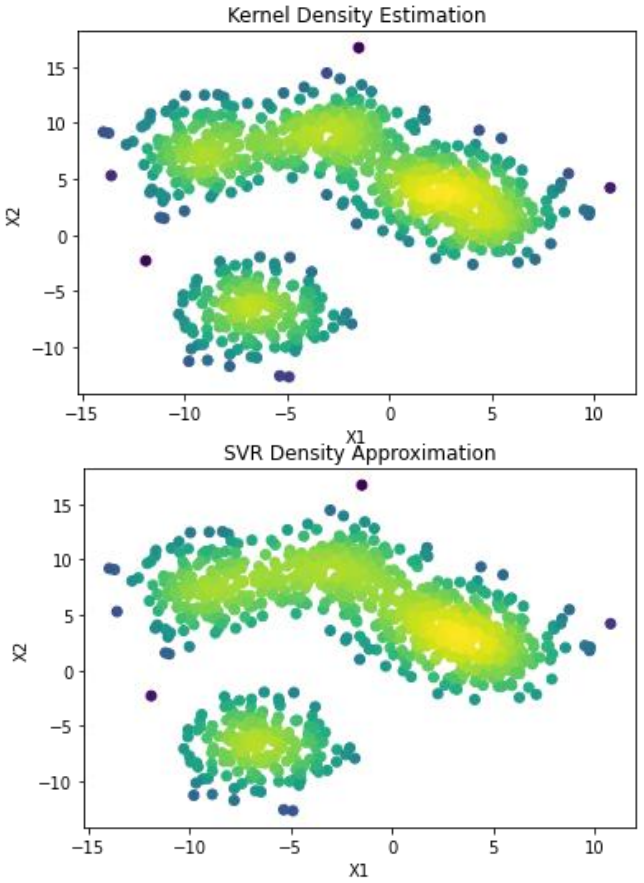
- Регрессия. Метод ядерного сглаживания Надарая-Ватсона:

$$a_h(x) = \frac{\sum_{i=1}^{l} y_i K\left(\frac{\rho(x, x_i)}{h}\right)}{\sum_{i=1}^{l} K\left(\frac{\rho(x, x_i)}{h}\right)}$$

- Чем они плохи?
  - **Нужно хранить всю выборку! Много места и долго считать** ☹
  - Вариант «разреженного» решения – аппроксимация плотности $\hat{p}_h(x)$, полученной через метод Парзена-Розенблатта, с помощью сплайнов или SVM регрессии

Пример - оценки и аппроксимация плотности распределения

```python
from sklearn.datasets import make_blobs
from sklearn.neighbors import KernelDensity
from sklearn.metrics import mean_squared_error
from sklearn.svm import SVR

n_samples = 1000
X_one, y_one = make_blobs(n_samples=n_samples,
                          centers=5, cluster_std=2,
                          random_state=42)

kde = KernelDensity(kernel='gaussian', bandwidth=1).fit(X_one)
kde_scr=kde.score_samples(X_one)

plt.scatter(X_one[:, 0], X_one[:, 1], c=kde_scr)
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('Kernel Density Estimation')
plt.show()

svr = SVR(C=10, kernel="rbf", epsilon=0.1).fit(X_one, kde_scr)
svr_scr = svr.predict(X_one)

plt.scatter(X_one[:, 0], X_one[:, 1], c=svr_scr)
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('SVR Density Approximation')
plt.show()

dev=mean_squared_error(svr_scr,kde_scr)
print(f"1000 points KDE vs {int(svr.n_support_)} points SVR")
print(f"Approximation error={dev}")
```



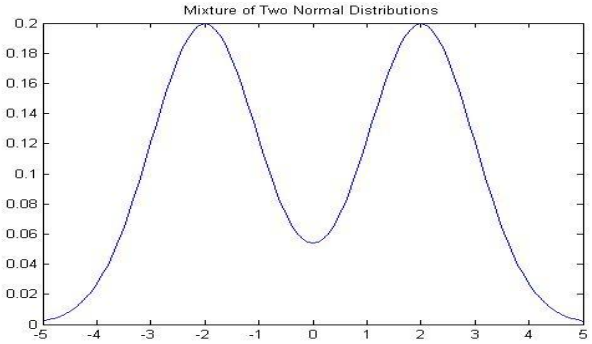

```
1000 points KDE vs 372 points SVR
Approximation error=0.015639321720188372
```

Задача разделения смеси распределений

- Порождающая модель смеси распределений:

$$p(x) = \sum_{j=1}^{k} w_j \varphi(x, \theta_j), \quad \sum_{j=1}^{k} w_j = 1, \quad w_j \geq 0$$

  - k – число компонент смеси;
  - $\varphi(x, \theta_j) = p(x|j)$ – функция правдоподобия j-ой компоненты;
  - $w_j = P(j)$ – априорная вероятность j-ой компоненты.
- Задачи:
  - Основная: **при фиксированном $\boldsymbol{k}$**, имея простую выборку $X^l = \{x_1, .., x_l\} \sim p(x)$, оценить вектор параметров $(\boldsymbol{w}, \boldsymbol{\theta}) = (w_1, \ldots, w_k, \theta_1, \ldots, \theta_k)$.
  - Дополнительно: **найти $\boldsymbol{k}$**.



Максимизация правдоподобия и EM-алгоритм

- Задача максимизации логарифма правдоподобия

$$L(w,\theta) = \ln \prod_{i=1}^{l} p(x_i) = \sum_{i=1}^{l} \ln \left[\sum_{j=1}^{k} w_j \varphi(x_i, \theta_j) \right] \to max_{w,\theta}$$

  - при ограничениях $\sum_{j=1}^{k} w_j = 1; w_j \geq 0$
  - вводим **скрытые переменные** $g_{ij} = P(j|x_i)$, их семантика – распределение ненаблюдаемых «меток» компонент (кластеров) для каждого наблюдения, позволяет максимизировать «взвешенное» правдоподобие **отдельно по каждой компоненте**
- Итерационный алгоритм Expectation-Maximization:
  - Начальное приближение параметров (w, θ);
  - **E-шаг** $(w, \theta) \to G = (g_{ij})$**:** оценка скрытых переменных
  - **M-шаг** $(w, \theta, G) \to (w, \theta)$: максимизация взвешенного правдоподобия отдельно по компонентам (w, θ)
  - Пока w, θ и G не стабилизируются.

EM-алгоритм

- Теорема (необходимые условия экстремума):
  - точка $(w_j, \theta_j)_{j=1}^{k}$ локального экстремума логарифмического правдоподобия $L(w, \theta)$ удовлетворяет системе уравнений относительно w_j, θ_j, g_{ij}:
  - E-шаг: $g_{ij} = \frac{w_j \varphi(x_i, \theta_j)}{\sum_{s=1}^{k} w_s \varphi(x_i, \theta_s)}, \ i = 1, \dots, l, \ j = 1, \dots, k;$
  - M-шаг: $\theta_j = \arg\max_{\theta} \sum_{i=1}^{l} \boldsymbol{g_{ij}} \ln \varphi(x_i, \theta), \ j = 1, \dots, k;$
  - $w_j = \frac{1}{l} \sum_{i=1}^{l} g_{ij}, \quad j = 1, \dots, k.$
- Вероятностная интерпретация:
  - **E-шаг** – формула Байеса: $g_{ij} = P(j|x_i) = \frac{P(j)p(x_i|j)}{p(x_i)} = \frac{w_j \varphi(x_i, \theta_j)}{p(x_i)} = \frac{w_j \varphi(x_i, \theta_j)}{\sum_{s=1}^{k} w_s \varphi(x_i, \theta_s)},$ при условии нормировки $\sum_{j=1}^{k} g_{ij} = 1$
  - **M-шаг** – это максимизация **взвешенного** правдоподобия, с весами объектов g_{ij} для j-ой компоненты смеси:

$$\theta_j = \arg\max_{\theta} \sum_{i=1}^{l} \boldsymbol{g_{ij}} \ln \varphi(x_i, \theta), \ w_j = \frac{1}{l} \sum_{i=1}^{l} g_{ij}$$

Доказательство (через условия ККТ)

- Лагранжиан оптимизационной задачи $\mathcal{L}(w,\theta) \to max$:

$$\mathcal{L}(w,\theta) = \sum_{i=1}^{l} \ln\left(\underbrace{\sum_{j=1}^{k} w_j \varphi(x_i,\theta_j)}_{p(x_i)}\right) - \lambda\left(\sum_{j=1}^{k} w_j - 1\right)$$

- Приравниваем нулю производные:

$$\frac{\partial \mathcal{L}}{\partial w_j} = 0 \quad \Rightarrow \sum_{i=1}^{l} \frac{\varphi(x_i,\theta_j)}{p(x_i)} = \lambda \Rightarrow \sum_{i=1}^{l} \underbrace{\frac{\boldsymbol{w_j}\varphi(x_i,\theta_j)}{p(x_i)}}_{g_{ij}} = \lambda \boldsymbol{w_j};$$

суммируем по $j \Rightarrow \boldsymbol{\lambda = l} \Rightarrow \ \boldsymbol{w_j = \frac{1}{l}\sum_{i=1}^{l} g_{ij}}$

$$\frac{\partial \mathcal{L}}{\partial \theta_j} = \sum_{i=1}^{l} \underbrace{\frac{w_j \boldsymbol{\varphi(x_i,\theta_j)}}{p(x_i)}}_{g_{ij}} \frac{\frac{\partial \varphi(x_i,\theta_j)}{\partial \theta_j}}{\boldsymbol{\varphi(x_i,\theta_j)}} = \frac{\partial}{\partial \theta_j} \sum_{i=1}^{l} \boldsymbol{g_{ij}} \ln \varphi(x_i,\theta_j) = 0$$

EM-алгоритм для разделения смеси распределений

- Вход: $X^l = \{x_1, \dots, x_l\},\ k$;
- Выход: $(w_j,\ \theta_j)_{j=1}^{k}$ – параметры смеси распределений;
- Инициализировать $(\theta_j)_{j=1}^{k},\ w_j := \frac{1}{k}$
- Повторять
  - E-шаг (expectation): для всех $i = 1, \dots, l,\quad j = 1, \dots, k$

$$g_{ij} := \frac{w_j \varphi(x_i, \theta_j)}{\sum_{s=1}^{k} w_s \varphi(x_i, \theta_s)}$$

  - M-шаг (maximization): для всех $j = 1, \dots, k$

$$w_j := \frac{1}{l} \sum_{i=1}^{l} g_{ij}\,, \theta_j := \arg\max_{\theta} \sum_{i=1}^{l} g_{ij} \ln \varphi(x_i, \theta)$$

- Пока w_j, θ_j и / или g_{ij} не сошлись.

Gaussian Mixture Model (GMM)

- Вход: $X^l = \{x_1, \dots, x_l\}, \quad k;$
- Выход: $\left(w_j, \boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right)_{j=1}^{k}$ – параметры смеси гауссиан;
- Инициализировать $\left(\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}\right)_{j=1}^{k}, w_j := \frac{1}{k}$
- Повторять
  - E-шаг (expectation): для всех $i = 1, .., l, \quad j = 1, \dots, k$

$$g_{ij} := \frac{w_j \boldsymbol{N}(\boldsymbol{x_i}; \boldsymbol{\mu_j}, \boldsymbol{\Sigma_j})}{\sum_{s=1}^{k} w_s \boldsymbol{N}(\boldsymbol{x_i}; \boldsymbol{\mu_s}, \boldsymbol{\Sigma_s})}$$

  - M-шаг (maximization): для всех $j = 1, \dots, k$

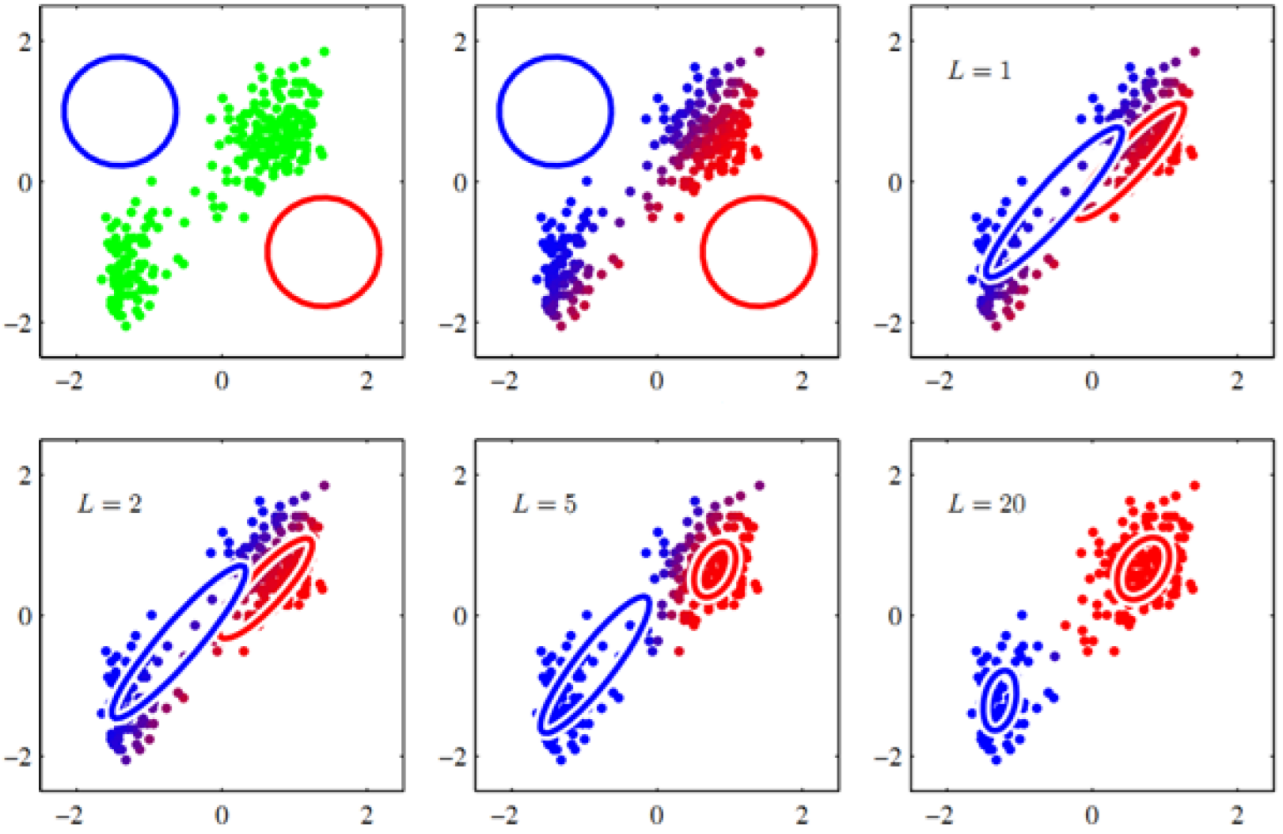
$$w_j := \frac{1}{l} \sum_{i=1}^{l} g_{ij},$$

$$\boldsymbol{\mu_j} := \frac{\boldsymbol{1}}{\boldsymbol{l w_j}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{l}} \boldsymbol{g_{ij} x_i}, \boldsymbol{\Sigma_j} := \frac{\boldsymbol{1}}{\boldsymbol{l w_j}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{l}} \boldsymbol{g_{ij}} \left(\boldsymbol{x_i} - \boldsymbol{\mu_j}\right)\left(\boldsymbol{x_i} - \boldsymbol{\mu_j}\right)^{\boldsymbol{T}}$$

- Пока (w_j, μ_j, Σ_j) и / или g_{ij} не сошлись.

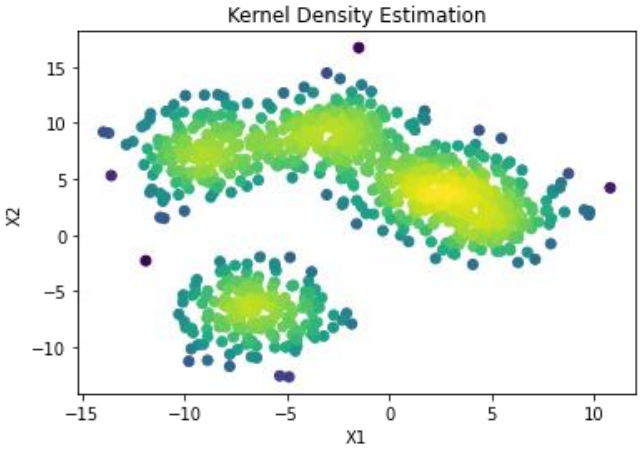
Демо-пример

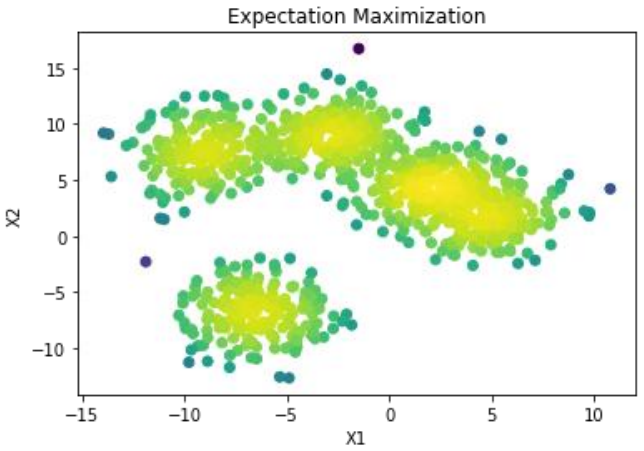
- Две гауссовские компоненты $k = 2$ в пространстве $X = \mathbb{R}^2$.
- Расположение компонент в зависимости от номера итерации L



Пример – параметрическая и непараметрическая оценки плотности распределения

```
from sklearn.mixture import GaussianMixture
from sklearn.datasets import make_blobs
from sklearn.neighbors import KernelDensity
from sklearn.metrics import mean_squared_error

n_samples = 1000
X_one, y_one = make_blobs(n_samples=n_samples,
                          centers=5, cluster_std=2,
                          random_state=42)

kde = KernelDensity(kernel='gaussian', bandwidth=1).fit(X_one)
kde_scr=kde.score_samples(X_one)

plt.scatter(X_one[:, 0], X_one[:, 1], c=kde_scr)
plt.xlabel('X1')
plt.ylabel('X2')
plt.title('Kernel Density Estimation')
plt.show()

gmm_0 = GaussianMixture(n_components=5)
gmm_0.fit(X_one)

E=gmm_0.score_samples(X_one)

plt.xlabel('X1')
plt.ylabel('X2')
plt.scatter(X_one[:, 0], X_one[:, 1],  c=E)
plt.title('Expectation Maximization')
plt.show()

dev=mean_squared_error(E,kde_scr)

print(f"1000 points KDE vs 35 parameters in EM ")
print(f"Approximation error={dev}")
```






```
1000 points KDE vs 35 parameters in EM
Approximation error=0.08319576408750858
```

EM-алгоритм с оценкой числа компонент

- Проблемы базового EM-алгоритма:
  - Как выбирать начальное приближение? На основе «грубых приближений», например, k-means кластеризации
  - Как ускорить сходимость? (будет дальше)
  - Как определять число компонент?
- Эвристики добавления и удаления компонент в EM-алгоритме:
  - Если слишком много объектов x_i имеют слишком низкие правдоподобия $p(x_i)$, то создаем новую $k+1$ – ую компоненту и по этим объектам строим ее начальное приближение.
  - Если у j-ой компоненты слишком низкий w_j, удаляем ее.
  - Регуляризация (сделать максимальную «определенность» неравномерность w_j) $L(w,\theta) - \tau \sum_{j=1}^{k} \ln w_j \to \max$, $w_j \propto \left(\frac{1}{l}\sum_{i=1}^{l} g_{ij} - \tau\right)_+$
  - Комбинация грубого приближения простым алгоритмом кластеризации (например, иерархическим) с выбором числа компонент как числа кластеров на основе статистических оценок качества кластеризации типа псевдо-Фишер, псевдо- t^2, CCC и другие – будут в кластеризации

«Ускоряющие» модификации EM-алгоритма

- Обобщенный EM (GEM):
  - не нужно добиваться точного решения задачи M-шага достаточно сместиться в направлении максимума, сделав одну или **несколько итераций**, затем выполнить E-шаг.
- Стохастический EM (SEM):
  - Идея: на M-шаге вместо взвешенного **максимизируется обычное правдоподобие:**

$$\theta_j := \arg\max_{\theta} \sum_{i=1}^{l} g_{ij} \ln \varphi(x_i, \theta)$$

  - выборки X_j строятся путем **сэмплирования** объектов из X^l l раз **с возвращениями**: $i \sim P(i|j) = \frac{P(j|x_i)P(i)}{P(j)} = \frac{g_{ij}}{lw_j}$
- Преимущества:
  - Ускорение сходимости, предотвращение зацикливаний, при сохранении свойств слабой локальной сходимости (в смысле увеличения правдоподобия на каждом шаге)

Выводы по оценке плотности распределения

- Параметрическое оценивание плотности:
  - Модель плотности + максимизация правдоподобия: $\hat{p}(x) = \varphi(x, \theta)$, но нужно угадать распределение и редко реальные многомерные данные можно описать одним распределением
- Непараметрическое оценивание плотности:
  - Наиболее прост, приводит к методу Парзеновского окна, но вычислительно сложен на этапе применения, требует хранить всю выборку (можно аппроксимировать), нужно выбирать перебором параметры ядра (иногда и само ядро):

$$\hat{p}(x) = \sum_{i=1}^{l} \frac{1}{lV(h)} K\left(\frac{\rho(x,x_i)}{h}\right)$$

- Смеси распределений:
  - Наиболее общий случай, приводит к EM-алгоритму, но нужно угадать распределение(я) и число компонент:

$$\hat{p}(x) = \sum_{j=1}^{k} w_j \varphi(x, \theta_j), k \ll l$$