

Разделение смеси распределений. EM-алгоритм для классификации и кластеризации

Воронцов Константин Вячеславович

vokov@forecsys.ru

<http://www.MachineLearning.ru/wiki?title=User:Vokov>

Этот курс доступен на странице вики-ресурса

<http://www.MachineLearning.ru/wiki>

«Машинное обучение (курс лекций, К.В.Воронцов)»

Видеолекции: <http://shad.yandex.ru/lectures>

1 Разделение смеси распределений

- EM-алгоритм
- Определение числа компонент
- Модификации EM-алгоритма

2 Классификация

- Разделение гауссовской смеси
- Эмпирические оценки средних и дисперсий
- Сеть радиальных базисных функций

3 Кластеризация

- Задача кластеризации
- Метод k -средних
- Задача частичного обучения

Напоминание. Байесовская теория классификации

X — объекты, Y — ответы, $X \times Y$ — в.п. с плотностью $p(x, y)$;

Две подзадачи:

① Дано:

$X^\ell = (x_i, y_i)_{i=1}^\ell$ — обучающая выборка.

Найти:

эмпирические оценки $\hat{P}(y)$ и $\hat{p}(x|y)$, $y \in Y$

(восстановить плотность каждого класса по выборке).

② Дано:

априорные вероятности $P(y)$ и плотности $p(x|y)$, $y \in Y$.

Найти:

классификатор $a: X \times Y$, минимизирующий риск $R(a)$.

Решение:

$$a(x) = \arg \max_{y \in Y} \lambda_y P(y) p(x|y).$$

Задача восстановления 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 -й компоненты;

$w_j = p(j)$ — априорная вероятность j -й компоненты.

Задача 1: при фиксированном k ,

имея простую выборку $X^m = \{x_1, \dots, x^m\} \sim p(x)$,

оценить вектор параметров $(w, \theta) = (w_1, \dots, w_k, \theta_1, \dots, \theta_k)$.

Задача 2: оценить ещё и k .

Максимизация правдоподобия и EM-алгоритм

Задача максимизации логарифма правдоподобия

$$L(w, \theta) = \ln \prod_{i=1}^m p(x_i) = \sum_{i=1}^m \ln \sum_{j=1}^k w_j \varphi(x_i, \theta_j) \rightarrow \max_{w, \theta}.$$

при ограничениях $\sum_{j=1}^k w_j = 1$; $w_j \geq 0$.

Итерационный алгоритм Expectation–Maximization:

- 1: начальное приближение параметров (w, θ) ;
- 2: **повторять**
- 3: оценка скрытых переменных $G = (g_{ij})$, $g_{ij} = p(j|x_i)$:
 $G := E\text{-шаг}(w, \theta)$;
- 4: максимизация правдоподобия, отдельно по компонентам:
 $(w, \theta) := M\text{-шаг}(w, \theta, G)$;
- 5: **пока** w, θ и G не стабилизируются.

EM-алгоритм как способ решения системы уравнений

Теорема (необходимые условия экстремума)

Точка $(w_j, \theta_j)_{j=1}^k$ локального экстремума $L(w, \theta)$ удовлетворяет системе уравнений относительно w_j, θ_j и g_{ij} :

$$\text{E-шаг: } g_{ij} = \frac{w_j \varphi(x_i, \theta_j)}{\sum_{s=1}^k w_s p_s(x_i, \theta_s)}, \quad i = 1, \dots, m, \quad j = 1, \dots, k;$$

$$\text{M-шаг: } \theta_j = \arg \max_{\theta} \sum_{i=1}^m g_{ij} \ln \varphi(x_i, \theta), \quad j = 1, \dots, k;$$

$$w_j = \frac{1}{m} \sum_{i=1}^m g_{ij}, \quad j = 1, \dots, k.$$

EM-алгоритм — это метод простых итераций для её решения

Вероятностная интерпретация

E-шаг — это формула Байеса:

$$g_{ij} = P(j|x_i) = \frac{P(j)p(x_i|j)}{p(x_i)} = \frac{w_j\varphi(x_i, \theta_j)}{p(x_i)} = \frac{w_j\varphi(x_i, \theta_j)}{\sum_{s=1}^k w_s p_s(x_i, \theta_s)}.$$

Очевидно, выполнено условие нормировки: $\sum_{j=1}^k g_{ij} = 1$.

M-шаг — это максимизация взвешенного правдоподобия, с весами объектов g_{ij} для j -й компоненты смеси:

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

$$w_j = \frac{1}{m} \sum_{i=1}^m g_{ij}.$$

Доказательство. Условия Каруша–Куна–Таккера

Лагранжиан оптимизационной задачи « $L(w, \theta) \rightarrow \max$ »:

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

Приравниваем нулю производные:

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

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

EM-алгоритм

Вход: $X^m = \{x_1, \dots, x_m\}$, k , δ , начальные $(w_j, \theta_j)_{j=1}^k$;

Выход: $(w_j, \theta_j)_{j=1}^k$ — параметры смеси распределений

1: **повторять**

2: E-шаг (expectation):

для всех $i = 1, \dots, m$, $j = 1, \dots, k$

$$g_{ij}^0 := g_{ij}; \quad g_{ij} := \frac{w_j \varphi(x_i, \theta_j)}{\sum_{s=1}^k w_s \rho_s(x_i, \theta_s)};$$

3: M-шаг (maximization):

для всех $j = 1, \dots, k$

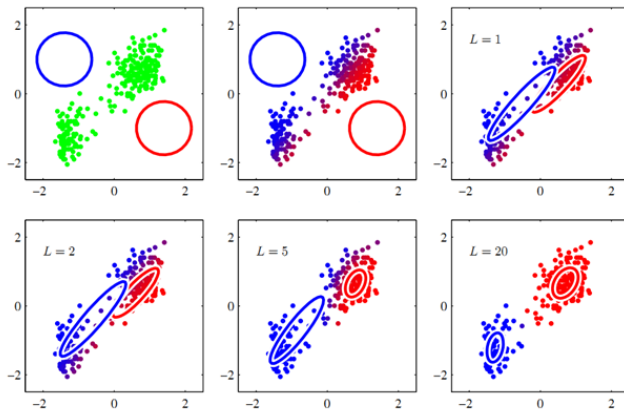
$$\theta_j := \arg \max_{\theta} \sum_{i=1}^m g_{ij} \ln \varphi(x_i, \theta); \quad w_j := \frac{1}{m} \sum_{i=1}^m g_{ij};$$

4: **пока** $\max_{i,j} |g_{ij} - g_{ij}^0| > \delta$;

5: **вернуть** $(w_j, \theta_j)_{j=1}^k$;

Пример

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



EM-алгоритм с последовательным добавлением компонент

Проблемы базового варианта EM-алгоритма:

- Как выбирать начальное приближение?
- Как определять число компонент?
- Как ускорить сходимость?

Вход:

выборка $X^m = \{x_1, \dots, x_m\}$;

R — допустимый разброс правдоподобия объектов;

m_0 — минимальная длина выборки, по которой можно восстанавливать плотность;

δ — параметр критерия останова;

Выход:

k — число компонент смеси;

$(w_j, \theta_j)_{j=1}^k$ — веса и параметры компонент;

EM-алгоритм с последовательным добавлением компонент

1: начальное приближение — одна компонента:

$$\theta_1 := \arg \max_{\theta} \sum_{i=1}^m \ln \varphi(x_i, \theta); \quad w_1 := 1; \quad k := 1;$$

2: **для всех** $k := 2, 3, \dots$

3: выделить объекты с низким правдоподобием:

$$U := \{x_i \in X^m \mid p(x_i) < \frac{1}{R} \max_j p(x_j)\};$$

4: **если** $|U| < m_0$ **то**

5: **выход** из цикла по k ;

6: начальное приближение для k -й компоненты:

$$\theta_k := \arg \max_{\theta} \sum_{x_i \in U} \ln \varphi(x_i, \theta); \quad w_k := \frac{1}{m} |U|;$$

$$w_j := w_j(1 - w_k), \quad j = 1, \dots, k - 1;$$

7: выполнить EM $(X^m, k, \delta, w, \theta)$;

Регуляризация с априорным распределением Дирихле

Гипотеза. Вектор весов $w = (w_j)_{j=1}^k$ порождается распределением Дирихле с вектором параметров $\alpha \in \mathbb{R}^k$:

$$\text{Dir}(w|\alpha) = \frac{\Gamma(\alpha_0)}{\prod_j \Gamma(\alpha_j)} \prod_j w_j^{\alpha_j-1}, \quad \alpha_0 = \sum_{j=1}^k \alpha_j, \quad \alpha_j \geq 0;$$

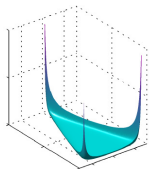
Распределение Дирихле порождает нормированные неотрицательные векторы заданной размерности k

Пример:

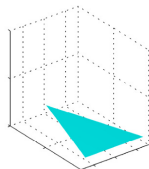
$\text{Dir}(w|\alpha)$

$k = 3$

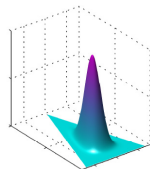
$w, \alpha \in \mathbb{R}^3$



$\alpha_1 = \alpha_2 = \alpha_3 = 0.1$

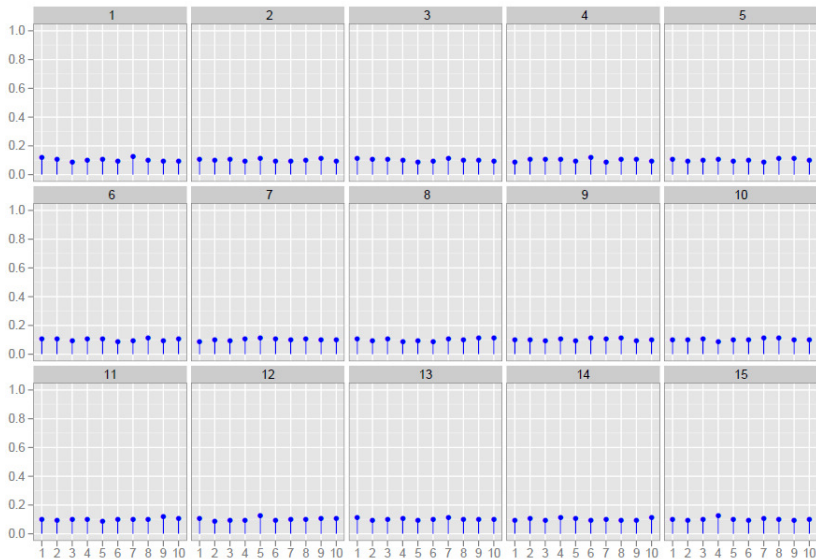


$\alpha_1 = \alpha_2 = \alpha_3 = 1$

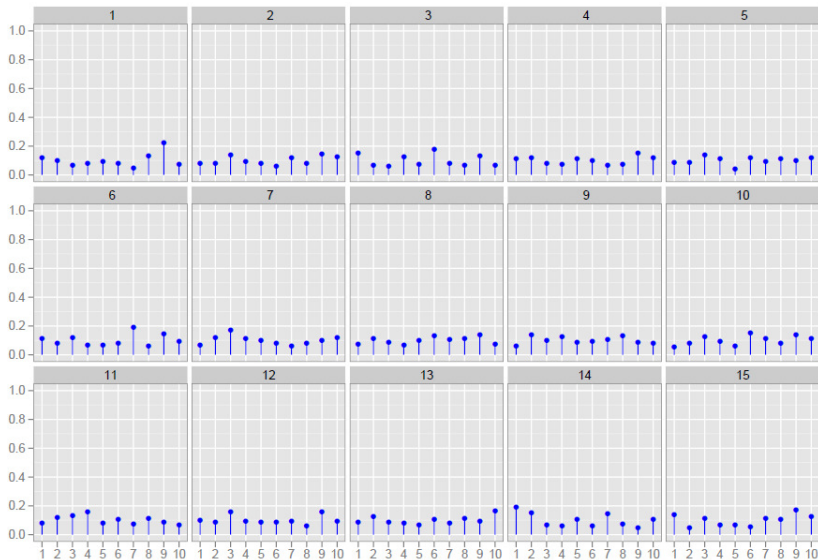


$\alpha_1 = \alpha_2 = \alpha_3 = 10$

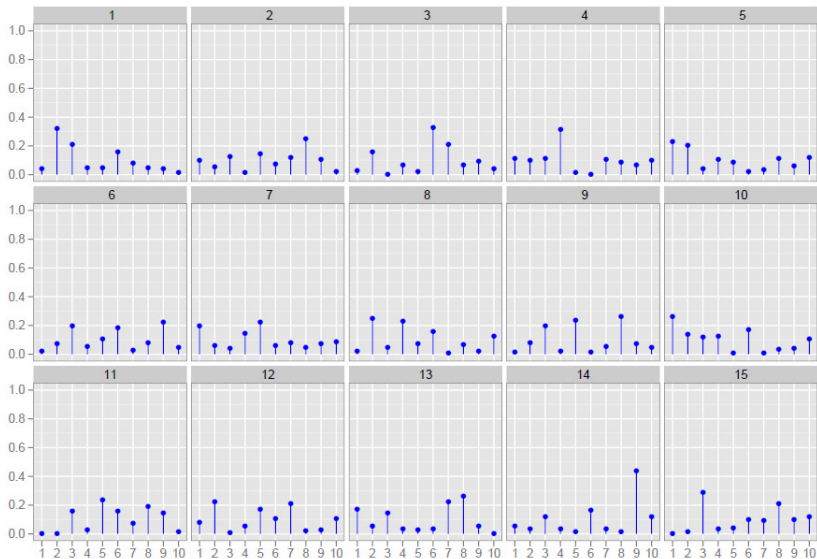
Распределение Дирихле при $\alpha \equiv 100$, 10 компонент



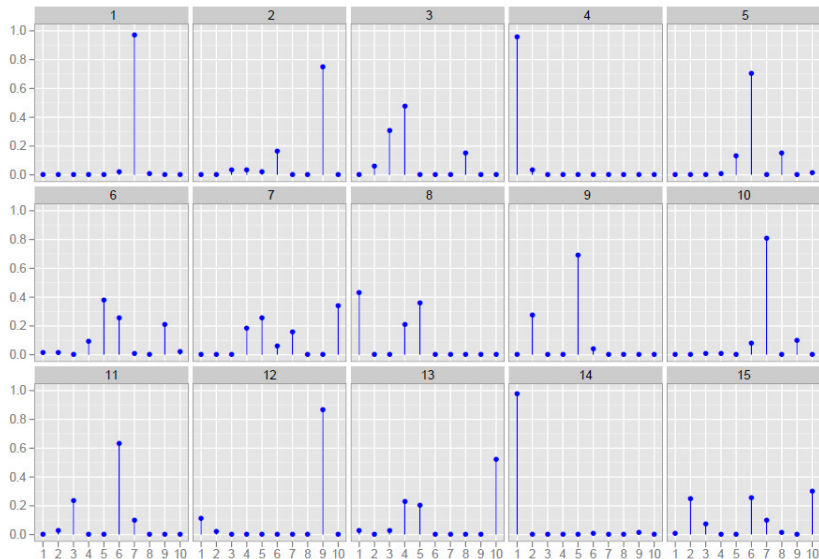
Распределение Дирихле при $\alpha \equiv 10$, 10 компонент



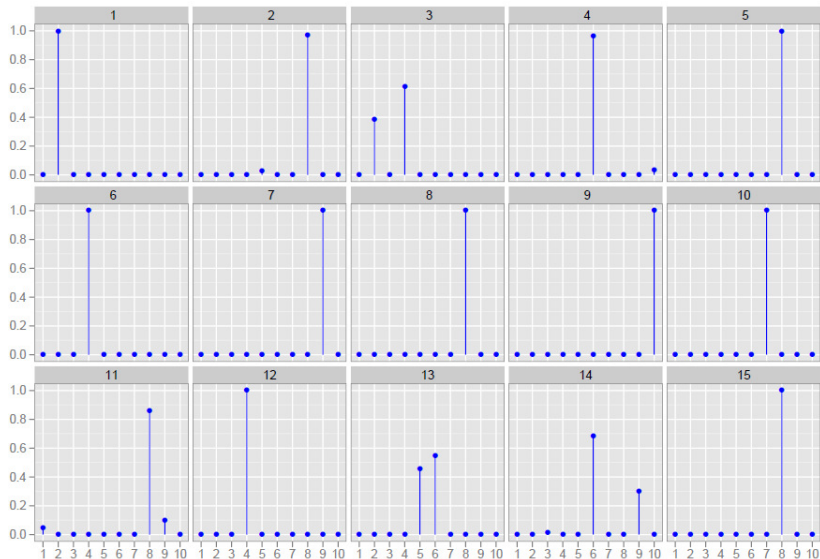
Распределение Дирихле при $\alpha \equiv 1$, 10 компонент



Распределение Дирихле при $\alpha \equiv 0.1$, 10 компонент



Распределение Дирихле при $\alpha \equiv 0.01$, 10 компонент



Принцип максимума апостериорной вероятности

Регуляризация log-правдоподобия:

$$\begin{aligned}
 L(w, \theta) + R(w) &= \ln \prod_{i=1}^m p(x_i) + \ln \text{Dir}(w|\alpha) = \\
 &= \sum_{i=1}^m \ln \sum_{j=1}^k w_j \varphi(x_i, \theta_j) + \ln \frac{\Gamma(\alpha_0)}{\prod_j \Gamma(\alpha_j)} \sum_{j=1}^k (\alpha_j - 1) \ln w_j \rightarrow \max_{w, \theta}
 \end{aligned}$$

Модификация формулы M-шага в случае разреженного симметричного распределения Дирихле, $\tau = 1 - \alpha_j \in (0, 1)$:

$$\begin{aligned}
 L(w, \theta) - \tau \sum_{j=1}^k \ln w_j &\rightarrow \max_{w, \theta}; \\
 w_j &\propto \left(\frac{1}{m} \sum_{i=1}^m g_{ij} - \tau \right)_+
 \end{aligned}$$

Подбор τ — по максимуму правдоподобия тестовой выборки.

GEM — обобщённый EM-алгоритм

Идея:

Не обязательно добиваться высокой точности на M-шаге. Достаточно лишь сместиться в направлении максимума, сделав одну или несколько итераций, и затем выполнить E-шаг.

Преимущества:

- сохраняется свойство слабой локальной сходимости (в смысле увеличения правдоподобия на каждом шаге)
- повышается скорость сходимости при сопоставимом качестве решения

SEM — стохастический EM-алгоритм

Идея: на M -шаге вместо максимизации

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

максимизируется обычное, невзвешенное, правдоподобие

$$\theta_j := \arg \max_{\theta} \sum_{x_i \in X_j} \ln \varphi(x_i, \theta),$$

выборки X_j строятся путём стохастического моделирования:
для каждого $i = 1, \dots, m$ генерируется $j \sim P(\theta_j | x_i) \equiv g_{ij}$
и объект x_i помещается в X_j .

Преимущества:

ускорение сходимости, предотвращение закливаний.

HEM — иерархический EM-алгоритм

Идея:

«Плохо описанные» компоненты расщепляются на две или более *дочерних* компонент.

Преимущество:

автоматически выявляется иерархическая структура каждого класса, которую затем можно интерпретировать содержательно.

Гауссовская смесь с диагональными матрицами ковариации

Гауссовская смесь GMM — Gaussian Mixture Model

Допущения:

1. Функции правдоподобия классов $p(x|y)$ представимы в виде смесей k_y компонент, $y \in Y = \{1, \dots, M\}$.
2. Компоненты имеют n -мерные гауссовские плотности с некоррелированными признаками:

$$\mu_{yj} = (\mu_{yj1}, \dots, \mu_{yjn}), \quad \Sigma_{yj} = \text{diag}(\sigma_{yj1}^2, \dots, \sigma_{yjn}^2), \quad j = 1, \dots, k_y:$$

$$p(x|y) = \sum_{j=1}^{k_y} w_{yj} p_{yj}(x), \quad p_{yj}(x) = \mathcal{N}(x; \mu_{yj}, \Sigma_{yj}),$$

$$\sum_{j=1}^{k_y} w_{yj} = 1, \quad w_{yj} \geq 0;$$

Эмпирические оценки средних и дисперсий

Числовые признаки: $f_d: X \rightarrow \mathbb{R}$, $d = 1, \dots, n$.

Решение задачи M-шага:

для всех классов $y \in Y$ и всех компонент $j = 1, \dots, k_y$,

$$w_{yj} = \frac{1}{\ell_y} \sum_{i: y_i=y} g_{yij}$$

для всех размерностей (признаков) $d = 1, \dots, n$

$$\hat{\mu}_{yjd} = \frac{1}{\ell_y w_{yj}} \sum_{i: y_i=y} g_{yij} f_d(x_i);$$

$$\hat{\sigma}_{yjd}^2 = \frac{1}{\ell_y w_{yj}} \sum_{i: y_i=y} g_{yij} (f_d(x_i) - \hat{\mu}_{yjd})^2;$$

Замечание: компоненты «наивны», но смесь не «наивна».

Байесовский классификатор

Подставим гауссовскую смесь в байесовский классификатор:

$$a(x) = \arg \max_{y \in Y} \underbrace{\lambda_y P_y}_{\Gamma_y(x)} \sum_{j=1}^{k_y} \underbrace{w_{yj} \mathcal{N}_{yj} \exp\left(-\frac{1}{2} \rho_{yj}^2(x, \mu_{yj})\right)}_{\rho_{yj}(x)}$$

$\mathcal{N}_{yj} = (2\pi)^{-\frac{n}{2}} (\sigma_{yj1} \cdots \sigma_{yjn})^{-1}$ — нормировочные множители;
 $\rho_{yj}(x, \mu_{yj})$ — взвешенная евклидова метрика в $X = \mathbb{R}^n$:

$$\rho_{yj}^2(x, \mu_{yj}) = \sum_{d=1}^n \frac{1}{\sigma_{yjd}^2} (f_d(x) - \mu_{yjd})^2.$$

Интерпретация — как у метрического классификатора:

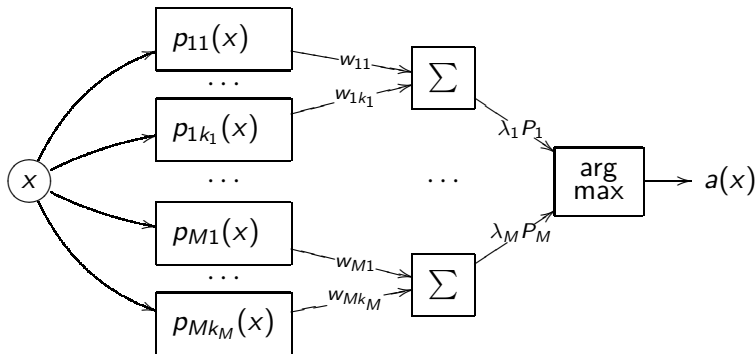
$\rho_{yj}(x)$ — близость объекта x к j -й компоненте класса y ;

$\Gamma_y(x)$ — близость объекта x к классу y .

Байесовский классификатор — сеть RBF

Radial Basis Functions (RBF) — трёхуровневая суперпозиция:

$$a(x) = \arg \max_{y \in Y} \lambda_y P_y \sum_{j=1}^{k_y} w_{yj} p_{yj}(x)$$



Преимущества EM-RBF

EM — один из лучших алгоритмов обучения радиальных сетей.

Преимущества EM-алгоритма по сравнению с SVM:

- 1 EM-алгоритм легко сделать устойчивым к шуму
- 2 EM-алгоритм довольно быстро сходится
- 3 автоматически строится *структурное описание* каждого класса в виде совокупности компонент — *кластеров*

Недостатки EM-алгоритма:

- 1 EM-алгоритм чувствителен к начальному приближению
- 2 Определение числа компонент — трудная задача (простые эвристики могут плохо работать)

Постановка задачи кластеризации

Дано:

X — пространство объектов;

$X^\ell = \{x_i\}_{i=1}^\ell$ — обучающая выборка;

$\rho: X \times X \rightarrow [0, \infty)$ — функция расстояния между объектами.

Найти:

Y — множество кластеров,

$a: X \rightarrow Y$ — алгоритм кластеризации.

Критерии слабо формализованные:

— каждый кластер состоит из близких объектов;

— объекты разных кластеров существенно различны.

Кластеризация — это *обучение без учителя*.

Вероятностная формализация задачи кластеризации

Гипотеза о вероятностной природе данных:

Выборка X^ℓ случайна, независима, из смеси распределений

$$p(x) = \sum_{y \in Y} w_y p_y(x), \quad \sum_{y \in Y} w_y = 1,$$

$p_y(x)$ — плотность, w_y — априорная вероятность кластера y .

Гипотеза о нормальности плотностей кластеров:

$$p_y(x) = (2\pi)^{-\frac{n}{2}} (\sigma_{y1} \cdots \sigma_{yn})^{-1} \exp\left(-\frac{1}{2} \rho_y^2(x, \mu_y)\right),$$

где $x \equiv (f_1(x), \dots, f_n(x)) \in \mathbb{R}^n$;

$\mu_y = (\mu_{y1}, \dots, \mu_{yn})$ — центр кластера y ;

σ_{yd}^2 — дисперсия значений признака f_d в классе y ;

$\rho_y^2(x, \mu_y) = \sum_{d=1}^n \frac{1}{\sigma_{yd}^2} |f_d(x) - \mu_{yd}|^2$ — взвешенная евклидова метрика.

EM-алгоритм (повторение)

1: начальное приближение w_y , μ_y , Σ_y для всех $y \in Y$;

2: **повторять**

3: E-шаг (expectation):

$$g_{iy} := P(y|x_i) \equiv \frac{w_y p_y(x_i)}{\sum_{z \in Y} w_z p_z(x_i)}, \quad y \in Y, \quad i = 1, \dots, \ell;$$

4: M-шаг (maximization):

$$w_y := \frac{1}{\ell} \sum_{i=1}^{\ell} g_{iy}, \quad y \in Y;$$

$$\mu_{yd} := \frac{1}{\ell w_y} \sum_{i=1}^{\ell} g_{iy} f_d(x_i), \quad y \in Y, \quad d = 1, \dots, n;$$

$$\sigma_{yj}^2 := \frac{1}{\ell w_y} \sum_{i=1}^{\ell} g_{iy} (f_d(x_i) - \mu_{yd})^2, \quad y \in Y, \quad d = 1, \dots, n;$$

5: $y_i := \arg \max_{y \in Y} g_{iy}$, $i = 1, \dots, \ell$;

6: **пока** y_i не перестанут изменяться;

Метод k -средних (k -means)

$X = \mathbb{R}^n$. Упрощённый аналог EM-алгоритма:

- жёсткая кластеризация вместо мягкой,
- метрика фиксирована, дисперсии не настраиваются.

1: начальное приближение центров μ_y , $y \in Y$;

2: **повторять**

3: **аналог E-шага:**

отнести каждый x_i к ближайшему центру:

$$y_i := \arg \min_{y \in Y} \rho(x_i, \mu_y), \quad i = 1, \dots, \ell;$$

4: **аналог M-шага:**

вычислить новые положения центров:

$$\mu_{yd} := \frac{\sum_{i=1}^{\ell} [y_i = y] f_d(x_i)}{\sum_{i=1}^{\ell} [y_i = y]}, \quad y \in Y, \quad d = 1, \dots, n;$$

5: **пока** y_i не перестанут изменяться;

Модификации и обобщения

Основные отличия EM и k -means:

- EM: мягкая кластеризация: $g_{iy} = P\{y_i = y\}$;
 k -means: жёсткая кластеризация: $g_{iy} = [y_i = y]$;
- EM: кластеры эллиптические, настраиваемые;
 k -means: кластеры сферические, не настраиваемые;

Гибриды (упрощение EM — усложнение k -means):

- EM с жёсткой кластеризацией на E-шаге;
- EM без настройки дисперсий (сферические гауссианы);

Недостатки k -means:

- Чувствительность к выбору начального приближения.
- Медленная сходимость (пользуйтесь k -means++)

Частичное обучение (Semi-supervised learning)

Дано:

Y — множество кластеров;

$\{x_i\}_{i=1}^{\ell}$ — обучающая выборка;

$\{x_i, y_i\}_{i=\ell+1}^{\ell+m}$ — размеченная часть выборки, обычно $m \ll \ell$.

Найти:

$a: X \rightarrow Y$ — алгоритм кластеризации.

Как приспособить EM-алгоритм:

Е-шаг: $g_{iy} := [y = y_i]$, $y \in Y$, $i = \ell+1, \dots, \ell+m$;

Как приспособить k -means:

Е-шаг: $y_i := \arg \min_{y \in Y} \rho(x_i, \mu_y)$, $i = 1, \dots, \ell$.

Резюме в конце лекции

- EM-алгоритм — мощный метод восстановления скрытой информации по наблюдаемым данным
- Применяется для разделения смесей распределений, кластеризации, классификации
- Кластеризация k -means — упрощение EM-алгоритма
- Имеет многочисленные модификации, в том числе для определения числа компонент