Алгоритмы кластеризации II

Сергей Николенко

Машинное обучение — ИТМО, осень 2006

Outline

Постановка задачи

- Часто возникает ситуация, когда в имеющихся данных некоторые переменные присутствуют, а некоторые — отсутствуют.
- Даны результаты сэмплирования распределения вероятностей с несколькими параметрами, из которых известны не все.

Постановка задачи

- Эти неизвестные параметры тоже расцениваются как случайные величины.
- Задача — найти наиболее вероятную гипотезу, то есть ту гипотезу h, которая максимизирует

$$E[\ln p(D|h)].$$

Частный случай

Построим один из простейших примеров применения алгоритма EM. Пусть случайная переменная x сэмплируется из суммы двух нормальных распределений. Дисперсии даны (одинаковые), нужно найти только средние μ_1, μ_2.

Два распределения

- Теперь нельзя понять, какие x_i были порождены каким распределением — классический пример *скрытых переменных*.
- Один тестовый пример полностью описывается как тройка $\langle x_i, z_{i1}, z_{i2} \rangle$, где $z_{ij} = 1$ iff x_i был сгенерирован j-м распределением.

Суть алгоритма EM

- Сгенерировать какую-нибудь гипотезу $h = (\mu_1, \mu_2)$.
- Пока не дойдем до локального максимума:
  - Вычислить ожидание $E(z_{ij})$ в предположении текущей гипотезы (E–шаг).
  - Вычислить новую гипотезу $h' = (\mu_1', \mu_2')$, предполагая, что z_{ij} принимают значения $E(z_{ij})$ (M–шаг).

В примере с гауссианами

В примере с гауссианами:

$$E(z_{ij}) = \frac{p(x = x_i|\mu = \mu_j)}{p(x = x_i|\mu = \mu_1) + p(x = x_i|\mu = \mu_2)} = \\ = \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{e^{-\frac{1}{2\sigma^2}(x_i - \mu_1)^2} + e^{-\frac{1}{2\sigma^2}(x_i - \mu_2)^2}}.$$

Мы подсчитываем эти ожидания, а потом подправляем гипотезу:

$$\mu_j \leftarrow \frac{1}{m}\sum_{i=1}^{m} E(z_{ij})x_i.$$

Мысли?

- Какие есть мысли о применении алгоритма EM к задачам кластеризации?

Гипотезы

- Чтобы воспользоваться статистическим алгоритмом, нужно сформулировать гипотезы о распределении данных.
- *Гипотеза о природе данных*: тестовые примеры появляются случайно и независимо, согласно вероятностному распределению, равному смеси распределений кластеров

$$p(x) = \sum_{c \in C} w_c p_c(x), \quad \sum_{c \in C} w_c = 1,$$

где w_c — вероятность появления объектов из кластера c, p_c — плотность распределения кластера c.

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?
- Часто берут сферические гауссианы, но это не слишком гибкий вариант: кластер может быть вытянут в ту или иную сторону.

Гипотезы cont'd

- Остается вопрос: какими предположить распределения p_c?
- Часто берут сферические гауссианы, но это не слишком гибкий вариант: кластер может быть вытянут в ту или иную сторону.
- Мы будем брать эллиптические гауссианы.
- *Гипотеза* 2: Каждый кластер c описывается d–мерной гауссовской плотностью с центром $\mu_c = \{\mu_{c1}, \ldots, \mu_{cd}\}$ и диагональной матрицей ковариаций $\Sigma_c = \mathrm{diag}(\sigma_{c1}^2, \ldots, \sigma_{c2}^2)$ (т.е. по каждой координате своя дисперсия).

Постановка задачи и общий вид алгоритма

- В этих предположениях получается в точности задача разделения смеси вероятностных распределений. Для этого и нужен EM–алгоритм.
- Каждый тестовый пример описывается своими координатами $(f_1(x), \ldots, f_n(x))$.
- Скрытые переменные в данном случае — вероятности g_{ic} того, что объект x_i принадлежит кластеру $c \in C$.

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–шаг: с использованием g_{ic} уточняются параметры кластеров w, μ, σ:

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–шаг: с использованием g_{ic} уточняются параметры кластеров w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic},$$

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–шаг: с использованием g_{ic} уточняются параметры кластеров w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic}, \quad \mu_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} f_j(x_i),$$

Идея алгоритма

- E–шаг: по формуле Байеса вычисляются скрытые переменные g_{ic}:

$$g_{ic} = \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}.$$

- M–шаг: с использованием g_{ic} уточняются параметры кластеров w, μ, σ:

$$w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic}, \quad \mu_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} f_j(x_i),$$

$$\sigma_{cj}^2 = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} \left(f_j(x_i) - \mu_{cj}\right)^2.$$

Алгоритм

$\texttt{EMCluster}(X, |C|)$:

- Инициализировать $|C|$ кластеров; начальное приближение:
  $w_c := 1/|C|$, $\mu_c :=$ случайный x_i,
  $\sigma^2_{cj} := \frac{1}{n|C|} \sum_{i=1}^{n} (f_j(x_i) - \mu_{cj})^2$.
- Пока принадлежность кластерам не перестанет изменяться:
  - E–шаг: $g_{ic} := \frac{w_c p_c(x_i)}{\sum_{c' \in C} w_{c'} p_{c'}(x_i)}$.
  - M–шаг: $w_c = \frac{1}{n} \sum_{i=1}^{n} g_{ic}$, $\mu_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} f_j(x_i)$,

$$\sigma^2_{cj} = \frac{1}{n w_c} \sum_{i=1}^{n} g_{ic} (f_j(x_i) - \mu_{cj})^2 .$$

  - Определить принадлежность x_i к кластерам:

$$\text{clust}_i := \text{argmax}_{c \in C} g_{ic}.$$

Упражнение

Упражнение

Реализовать алгоритм EM для кластеризации точек в евклидовом пространстве размерности d, получая на вход координаты точек и желаемое количество кластеров.

Проблема

- Остается проблема: нужно задавать количество кластеров. А что, если оно неизвестно?
- Этим мы займемся позже.

Outline

Суть алгоритма k–средних

- Это фактически упрощение алгоритма EM.
- Разница в том, что мы не считаем вероятности принадлежности кластерам, а жестко приписываем каждый объект одному кластеру.
- Кроме того, в алгоритме k–средних форма кластеров не настраивается (но это не так важно).

Цель

- Цель алгоритма k–средних — минимизировать меру ошибки

$$E(X, C) = \sum_{i=1}^{n} \|x_i - \mu_i\|^2,$$

где μ_i — ближайший к x_i центр кластера.

- Т.е. мы не относим точки к кластерам, а двигаем центры, а принадлежность точек определяется автоматически.

Алгоритм неформально

- Идея та же, что в EM:
  - Проинициализировать.
  - Классифицировать точки по ближайшему к ним центру кластера.
  - Перевычислить каждый из центров.
  - Если ничего не изменилось, остановиться, если изменилось — повторить.

Алгоритм

$\texttt{kMeans}(X, |C|)$:

- Инициализировать центры $|C|$ кластеров $\mu_1, \ldots, \mu_{|C|}$.
- Пока принадлежность кластерам не перестанет изменяться:
  - Определить принадлежность x_i к кластерам:

$$\text{clust}_i := \text{argmin}_{c \in C} \rho(x_i, \mu_c).$$

  - Определить новое положение центров кластеров:

$$\mu_c := \frac{\sum_{\text{clust}_i = c} f_j(x_i)}{\sum_{\text{clust}_i = c} 1}.$$

Главные недостатки

- Необходимо точно знать число кластеров заранее.
- Качество результата зависит от разбиения.
- Поэтому часто применяют сначала какой-нибудь другой (дешевый) метод кластеризации, получая число кластеров и начальное разбиение, а затем уже с этими начальными данными запускают алгоритм k–средних.

Semi–supervised clustering

- И EM, и k–means хорошо обобщаются на случай частично обученных кластеров.
- То есть про часть точек уже известно, какому кластеру они принадлежат.
- Как это учесть?

Semi–supervised clustering

- Чтобы учесть информацию о точке x_i, достаточно для EM положить скрытую переменную g_{ic} равной тому кластеру, которому нужно, с вероятностью 1, а остальным — с вероятностью 0, и не пересчитывать.
- Для k–means то же самое, но для clust_i.

Outline

Что такое нечеткий кластер?

- Во всех вышеприведенных алгоритмах один объект принадлежал строго одному кластеру (возможно, с какой-то вероятностью).
- Теперь вводим *меру принадлежности* кластеру, и тем самым вводим нечеткость.
- Точки на краю кластера точки «меньше принадлежат» кластеру, чем в центре.

Принадлежность

- Будем обозначать принадлежность кластеру c через $u_c(x)$.
- Обычно выбирают меры так, чтобы

$$\sum_{c \in C} u_c(x) = 1.$$

Функция ошибки

- Нечеткие алгоритмы кластеризации минимизируют некоторую меру ошибки. Часто применяется такая мера:

$$E(C) = \sum_{c \in C} \sum_{x \in X} u_c^m(x) \rho^2(x, \mathrm{Center}_c),$$

где m — некоторый вещественный параметр.

- Доказывать, что именно она минимизируется, мы не будем, но функция ошибки вполне естественная.

Центры кластеров

- Как определить центр кластера? Для этого обычно рассматривают взвешенную посредством u_c сумму по всем точкам:

$$\mathrm{Center}_c = \frac{\sum_x u_c(x)^m x}{\sum_x u_c(x)^m},$$

где m — некоторый вещественный параметр.

- Затем можно перевзвесить относительно новых центров; будет похоже на

$$u_c(x) := \frac{1}{\rho(\mathrm{Center}_c, x)},$$

но нужно еще фаззифицировать немножко. В общем, все тот же EM–принцип.

Алгоритм

$\texttt{cMeans}(X, |C|)$:

- Случайно выбрать коэффициенты $u_c(x)$ для всех $x \in X$ и $c \in C$.
- Пока алгоритм не сойдется:
  - $\forall c \in C: \quad \text{Center}_c := \frac{\sum_x u_c(x)^m x}{\sum_x u_c(x)^m}$.
  - $\forall c \in C, x \in X: \quad u_c(x) := \frac{1}{\sum_{c' \in C} \left(\frac{\rho(\text{Center}_c, x)}{\rho(\text{Center}_{c'}, x)} \right)^{2/(m-1)}}$.

Обсуждение

- Если $m = 2$, то перевзвешивание эквивалентно линейной нормализации коэффициентов так, чтобы их сумма была равна 1.
- При $m \longrightarrow 1$ все больший и больший вес придается самому близкому кластеру, и алгоритм становится все более похож на алгоритм k–средних.

Упражнение

Упражнение

Реализовать алгоритм нечеткой кластеризации c–средних.

Спасибо за внимание!

- Lecture notes, слайды и коды программ появятся на моей homepage:

  `http://logic.pdmi.ras.ru/~sergey/index.php?page=teaching`

- Присылайте любые замечания, коды программ на других языках, решения упражнений, новые численные примеры и прочее по адресам:
  `sergey@logic.pdmi.ras.ru`, `smartnik@inbox.ru`