EM-алгоритм II

Сергей Николенко

Академия MADE — Mail.Ru

11 мая 2020 г.

Random facts:

- 11 мая 1926 г. со Шпицбергена вылетел дирижабль «Норвегия»; Руал Амундсен и Умберто Нобиле пролетели над Северным полюсом
- 11 мая 1950 г. в Théâtre des Noctambules (Театре полуночников) прошла премьера пьесы Эжена Ионеско «Лысая певица»
- 11 мая 1997 г. Гарри Каспаров в 19 ходов проиграл последнюю партию матча против Deep Blue и весь матч; впрочем, до этого счёт был равный
- 11 мая 2000 г. в Индии родилась миллиардная жительница, а 11 мая 2013 г. в русскоязычной Википедии появилась миллионная статья
- 11 мая 2020 г. в Таиланде проходит ежегодная церемония Первой Борозды; в своё время в ней участвовал молодой Будда

Алгоритм EM

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.

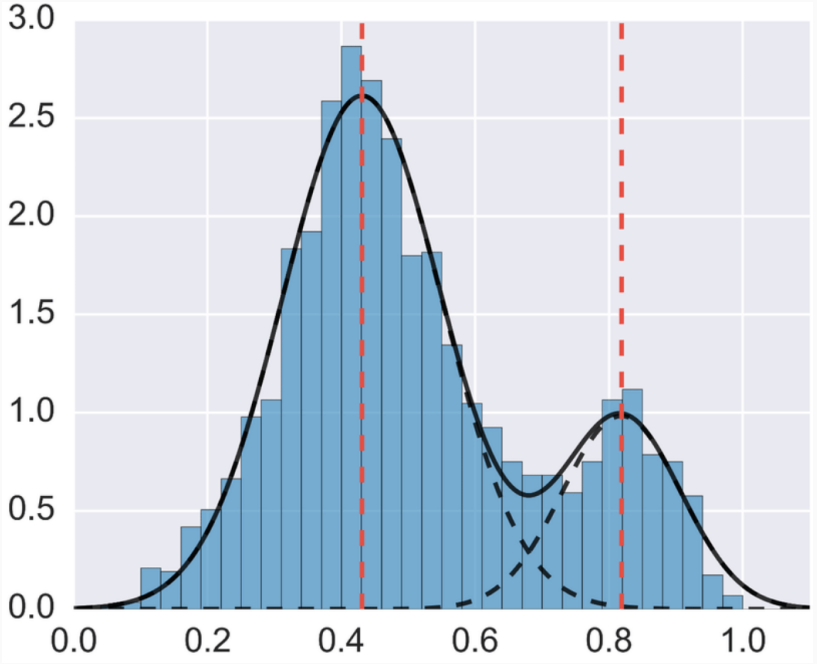
Постановка задачи

- 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 y сэмплируется из суммы двух нормальных распределений. Дисперсии даны (одинаковые), нужно найти только средние μ_1, μ_2.



- Какое тут правдоподобие? Как его оптимизировать?

Два распределения

- Нельзя понять, какие y_i были порождены каким распределением — классический пример *скрытых переменных*.
- Один тестовый пример полностью описывается как тройка $\langle y_i, z_{i1}, z_{i2} \rangle$, где $z_{ij} = 1$ iff y_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:

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

- Мы подсчитываем эти ожидания, а потом подправляем гипотезу:

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

- Звучит логично, но с какой стати это всё работает?

Обоснование алгоритма EM

- Дадим формальное обоснование алгоритма EM.
- Мы решаем задачу максимизации правдоподобия по данным $\mathsf{y} = \{y_1, \ldots, y_N\}$.

$$L(\boldsymbol{\theta} \mid \mathcal{Y}) = p(\mathcal{Y} \mid \boldsymbol{\theta}) = \prod p(y_i \mid \boldsymbol{\theta})$$

  или, что то же самое, максимизации $\ell(\boldsymbol{\theta} \mid \mathcal{Y}) = \log L(\boldsymbol{\theta} \mid \mathcal{Y})$.
- EM может помочь, если этот максимум трудно найти аналитически.

Обоснование алгоритма EM

- Давайте предположим, что в данных есть *скрытые компоненты*, такие, что если бы мы их знали, задача была бы проще.
- Замечание: совершенно не обязательно эти компоненты должны иметь какой-то физический смысл. :) Может быть, так просто удобнее.
- В любом случае, получается набор данных $\mathcal{X} = (\mathcal{Y}, \mathcal{Z})$ с совместной плотностью

$$p(\mathsf{x} \mid \boldsymbol{\theta}) = p(\mathsf{y}, \mathsf{z} \mid \boldsymbol{\theta}) = p(\mathsf{z} \mid \mathsf{y}, \boldsymbol{\theta}) p(\mathsf{y} \mid \boldsymbol{\theta}).$$

- Получается полное правдоподобие $L(\boldsymbol{\theta} \mid \mathcal{X}) = p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta})$. Это случайная величина (т.к. $\mathcal{Z}$ неизвестно).

Обоснование алгоритма EM

- Заметим, что настоящее правдоподобие $L(\boldsymbol{\theta}) = E_Z\left[p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}\right]$.
- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии $\mathcal{Y}$ и текущих оценок параметров $\boldsymbol{\theta}_n$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right].$$

- Здесь $\boldsymbol{\theta}_n$ — текущие оценки, а $\boldsymbol{\theta}$ — неизвестные значения (которые мы хотим получить в конечном счёте); т.е. $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$ — это функция от $\boldsymbol{\theta}$.

Обоснование алгоритма EM

- E-шаг алгоритма EM вычисляет условное ожидание (логарифма) полного правдоподобия при условии $\mathcal{Y}$ и текущих оценок параметров $\boldsymbol{\theta}$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right].$$

- Условное ожидание — это

$$E\left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta}) \mid \mathcal{Y}, \boldsymbol{\theta}_n\right] = \int_{\mathsf{z}} \log p(\mathcal{Y}, \mathsf{z} \mid \boldsymbol{\theta}) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \mathrm{d}\mathsf{z},$$

  где $p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)$ — маргинальное распределение скрытых компонентов данных.
- EM лучше всего применять, когда это выражение легко подсчитать, может быть, даже аналитически.
- Вместо $p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)$ можно подставить $p(\mathsf{z}, \mathcal{Y} \mid \boldsymbol{\theta}_n) = p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) p(\mathcal{Y} \mid \boldsymbol{\theta}_n)$, от этого ничего не изменится.

Обоснование алгоритма EM

- В итоге после E-шага алгоритма EM мы получаем функцию $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$.
- На M-шаге мы максимизируем

$$\boldsymbol{\theta}_{n+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n).$$

- Затем повторяем процедуру до сходимости.
- В принципе, достаточно просто находить $\boldsymbol{\theta}_{n+1}$, для которого $Q(\boldsymbol{\theta}_{n+1}, \boldsymbol{\theta}_n) > Q(\boldsymbol{\theta}_n, \boldsymbol{\theta}_n)$ — Generalized EM.
- Осталось понять, что значит $Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)$ и почему всё это работает.

Обоснование алгоритма EM

- Мы хотели перейти от $\boldsymbol{\theta}_n$ к $\boldsymbol{\theta}$, для которого $\ell(\boldsymbol{\theta}) > \ell(\boldsymbol{\theta}_n)$.

$$
\begin{aligned}
&\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}_n) = \\
&\qquad = \log\left(\int_{\mathsf{z}} p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta}) \mathrm{d}\mathsf{z}\right) - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) = \\
&= \log\left(\int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)} \mathrm{d}\mathsf{z}\right) - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) \geq \\
&\geq \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log\left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)}\right) \mathrm{d}\mathsf{z} - \log p(\mathcal{Y} \mid \boldsymbol{\theta}_n) = \\
&\qquad = \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log\left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathcal{Y} \mid \boldsymbol{\theta}_n) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)}\right) \mathrm{d}\mathsf{z}.
\end{aligned}
$$

Обоснование алгоритма EM

- Получили

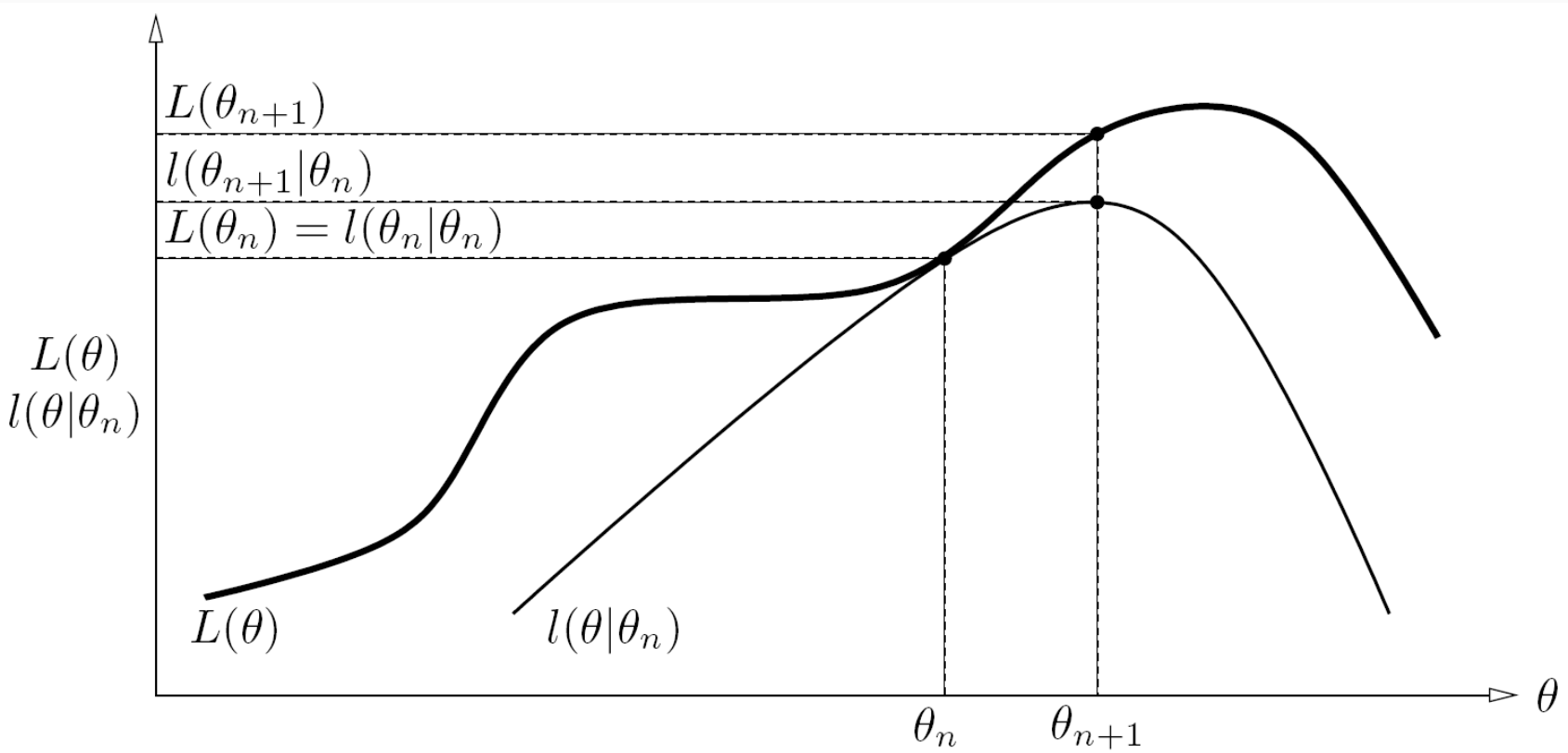
$$\ell(\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_n) =$$
$$= \ell(\boldsymbol{\theta}_n) + \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n) \log \left(\frac{p(\mathcal{Y} \mid \mathsf{z}, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta})}{p(\mathcal{Y} \mid \boldsymbol{\theta}_n) p(\mathsf{z} \mid \mathcal{Y}, \boldsymbol{\theta}_n)} \right) \mathrm{d}\mathsf{z}.$$

Упражнение. Докажите, что $\mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_n) = \ell(\boldsymbol{\theta}_n)$.

Обоснование алгоритма EM

- Иначе говоря, мы нашли нижнюю оценку на $\ell(\boldsymbol{\theta})$ везде, касание происходит в точке $\boldsymbol{\theta}_n$.
- Т.е. мы нашли нижнюю оценку для правдоподобия и смещаемся в точку, где она максимальна (или хотя бы больше текущей).
- Такая общая схема называется *MM-алгоритм* (minorization-maximization). Мы к ним ещё вернёмся.

Обоснование алгоритма EM



Обоснование алгоритма EM

- Осталось только понять, что максимизировать можно Q.

$$\boldsymbol{\theta}_{n+1} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}, \boldsymbol{\theta}_n) = \arg\max_{\boldsymbol{\theta}} \{\ell(\boldsymbol{\theta}_n) + \\ + \int_{\mathsf{z}} f(y \mid \mathcal{X}, \boldsymbol{\theta}_n) \log \left(\frac{p(\mathcal{X} \mid y, \boldsymbol{\theta}) f(y \mid \boldsymbol{\theta})}{p(\mathcal{X} \mid \boldsymbol{\theta}_n) f(y \mid \mathcal{X}, \boldsymbol{\theta}_n)} \right) \mathrm{d}\mathsf{z} \Big\} = \\ = \arg\max_{\boldsymbol{\theta}} \left\{ \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{X}, \boldsymbol{\theta}_n) \log \left(p(\mathcal{X} \mid y, \boldsymbol{\theta}) p(\mathsf{z} \mid \boldsymbol{\theta}) \right) \mathrm{d}\mathsf{z} \right\} = \\ = \arg\max_{\boldsymbol{\theta}} \left\{ \int_{\mathsf{z}} p(\mathsf{z} \mid \mathcal{X}, \boldsymbol{\theta}_n) \log p(\mathcal{X}, y \mid \boldsymbol{\theta}) \mathrm{d}\mathsf{z} \right\} = \\ = \arg\max_{\boldsymbol{\theta}} \{Q(\boldsymbol{\theta}, \boldsymbol{\theta}_n)\},$$

а остальное от $\boldsymbol{\theta}$ не зависит. Вот и получился EM.

История и первые применения

История EM-алгоритма

- Всё это появилось в работе Dempster–Laird–Rubin; доклад на Royal Statistical Society в 1976, статья вышла в 1977

Maximum Likelihood from Incomplete Data via the *EM* Algorithm

By A. P. Dempster, N. M. Laird and D. B. Rubin

Harvard University and Educational Testing Service

[Read before the Royal Statistical Society at a meeting organized by the Research Section on Wednesday, December 8th, 1976, Professor S. D. Silvey in the Chair]

Summary

A broadly applicable algorithm for computing maximum likelihood estimates from incomplete data is presented at various levels of generality. Theory showing the monotone behaviour of the likelihood and convergence of the algorithm is derived. Many examples are sketched, including missing value situations, applications to grouped, censored or truncated data, finite mixture models, variance component estimation, hyperparameter estimation, iteratively reweighted least squares and factor analysis.

Keywords: Maximum likelihood; Incomplete data; EM algorithm; Posterior mode

- Но были и более ранние аналоги (сами DLR тоже пишут, что было много примеров раньше, и ссылаются на них)...

История EM-алгоритма

- (Ceppellini et al., 1955): подсчёт частот генов в популяции:
  - в популяции есть k аллелей с частотами $p_1, \ldots, p_k$ и генами $G_1, \ldots, G_k$;
  - например, в рассмотренной MN-системе есть два аллеля $G_1 = M$ и $G_2 = N$, и у человека диплоидные клетки, т.е. бывают варианты MM, MN и NN; если бы мы умели различать все три варианта, то было бы легко подсчитать гены каждого типа;
  - но что если M доминантный, а N рецессивный, т.е. гомозиготные MM- и гетерозиготные MN-организмы неразличимы?
  - есть закон Харди-Вайнберга, который говорит, что гомозиготы и гетерозиготы встречаются с частотами $\frac{p^2}{p^2+2pq}$ и $\frac{2pq}{p^2+2pq}$, где p и $q = 1 - p$ — частоты генов;
  - можно было бы всё подсчитать, но получается замкнутый круг: нужно знать частоты генов, чтобы посчитать долю MM и MN, но чтобы посчитать частоты, нужно знать долю MM и MN...

История EM-алгоритма

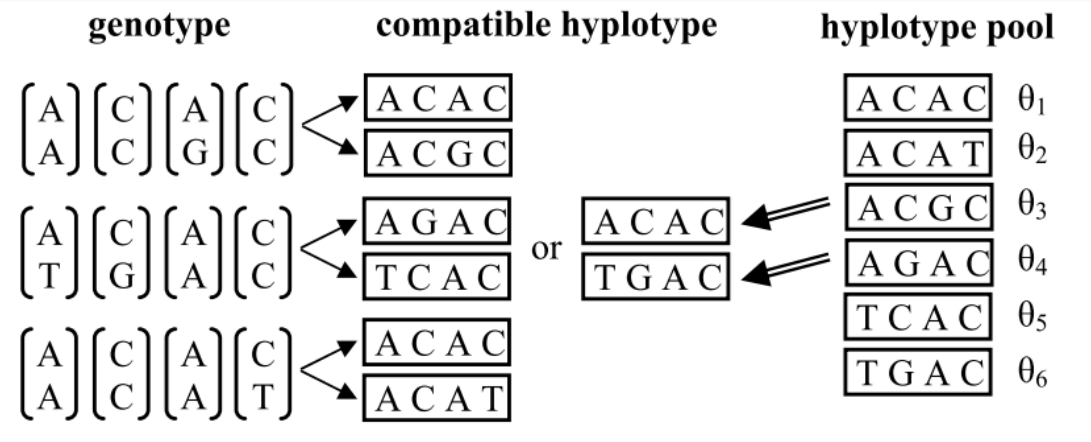
- И тут Ceppellini et al. говорят:

> by counting genes we can obtain estimates p' and q' of the gene frequencies. Unfortunately, this argument is still circular, since it presupposes a knowledge of the gene frequencies in order to obtain the estimate. But if any value is provisionally assumed for p, say $p(1)$, the new estimate $p' = p(2)$ obtained by gene counting will be rather more accurate, since the number of genes in the recessive individuals is known for certain, and the provisional value $p(1)$ is used only in estimating the number of genes among the dominants. This new value $p(2)$ can be taken as a new provisional value, and a further estimate $p'(2) = p(3)$ obtained by gene counting. The last process can be continued, giving a series of values $p(1)$, $p(2)$, $p(3)$, ..., each more accurate than the last; when two successive values are equal to the order of accuracy desired their common value can be taken as the final estimate.

- Это, видимо, одно из самых ранних применений EM-алгоритма, и такие применения актуальны, конечно, до сих пор.

История EM-алгоритма

- Похожий пример из современного источника (Fan et al., 2010)



- Гаплотип – совокупность аллелей (вариантов), которые наследуются вместе; у диплоидных организмов две копии каждой хромосомы, но возможны варианты
- Т.е. на картинке три генотипа, подходящие им гаплотипы и общий набор гаплотипов с частотами θ_i.

История EM-алгоритма

- Простейшая модель: каждый генотип порождается двумя гаплотипами, выбранными с вероятностью θ_i:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{N} \left(\sum_{j,k:\ h_j \oplus h_k = y_i} \theta_j \theta_k \right).$$

- Для известных пар гаплотипов $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)$, где $\gamma_i = (\gamma_i^+, \gamma_i^-)$, можно просто оценить $\theta_j = \frac{N_j}{2N}$.
- Так что это самый простой вариант EM:

$$\theta_j^{(t+1)} = \frac{1}{2N} \mathbb{E}_{\boldsymbol{\gamma}|\boldsymbol{\theta}^{(t)},\mathbf{y}} \left[N_j \right].$$

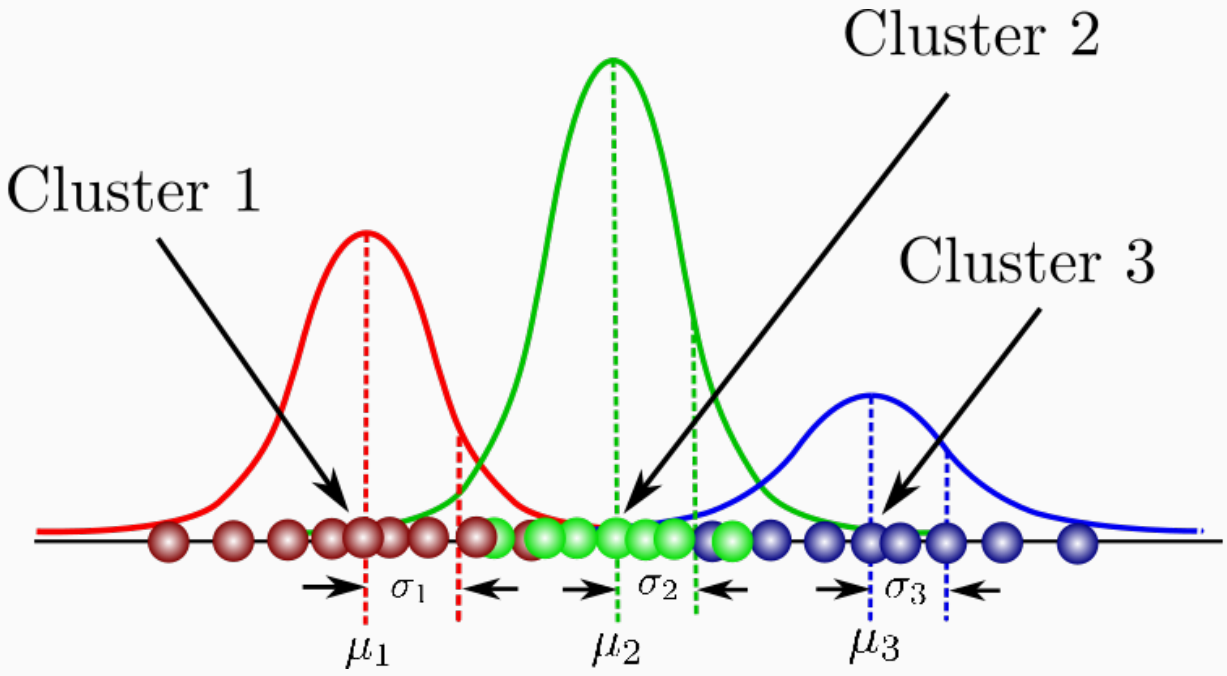
EM для кластеризации

Мысли?

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

Мысли?

- Какие есть мысли о применении алгоритма EM к задачам кластеризации?
- Кластеризацию можно формализовать как всё ту же задачу разделения смеси распределений:



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

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

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.
- Можно взять, например, эллиптические гауссианы.
- *Гипотеза 2*: Каждый кластер c описывается d–мерной гауссовской плотностью с центром $\boldsymbol{\mu}_c = \{\mu_{c1}, \ldots, \mu_{cd}\}$ 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 $(\mathbf{y}_1, \ldots, \mathbf{y}_N)$.
- Скрытые переменные $\mathbf{z}_n$ в данном случае — это one-hot вектор $\mathbf{z}_n = (z_{nc})$ того, какому кластеру принадлежит $\mathbf{y}_n$.
- Обозначим вероятности того, что объект $\mathbf{y}_n$ принадлежит кластеру $c \in C$, через g_{nc}.

Идея алгоритма

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

Идея алгоритма

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

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

Идея алгоритма

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

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- *M*–шаг: с использованием g_{nc} уточняются параметры кластеров $\mathbf{w}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$:

Идея алгоритма

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

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- *M*–шаг: с использованием g_{nc} уточняются параметры кластеров $\mathbf{w}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$:

$$w_c = \frac{1}{N} \sum_{n=1}^{N} g_{nc}, \quad \boldsymbol{\mu}_c = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \mathbf{y}_n,$$

Идея алгоритма

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

$$g_{nc} = \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}.$$

- *M*–шаг: с использованием g_{nc} уточняются параметры кластеров $\mathbf{w}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$:

$$w_c = \frac{1}{N} \sum_{n=1}^{N} g_{nc}, \quad \boldsymbol{\mu}_c = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \mathbf{y}_n,$$

$$\sigma_{cj}^2 = \frac{1}{n w_c} \sum_{n=1}^{N} g_{nc} \left(y_{nj} - \mu_{cj}\right)^2.$$

Алгоритм

`EMCluster`$(\mathcal{Y}, |C|)$:

- Инициализировать $|C|$ кластеров; начальное приближение: $w_c := 1/|C|$, $\boldsymbol{\mu}_c :=$ случайный $\mathbf{y}_n$, $\sigma_{cj}^2 := \frac{1}{n|C|}\sum_{i=1}^{N}\left(y_{nj} - \mu_{cj}\right)^2$.
- Пока принадлежность кластерам не перестанет изменяться:
  - E–шаг: $g_{nc} := \frac{w_c p_c(\mathbf{y}_n)}{\sum_{c' \in C} w_{c'} p_{c'}(\mathbf{y}_n)}$.
  - M–шаг: $w_c = \frac{1}{N}\sum_{i=1}^{N} g_{nc}$, $\mu_{cj} = \frac{1}{nw_c}\sum_{i=1}^{N} g_{nc} y_{nj}$,

$$\sigma_{cj}^2 = \frac{1}{nw_c}\sum_{i=1}^{N} g_{nc}\left(y_{nj} - \mu_{cj}\right)^2.$$

- После сходимости определить принадлежность x_i к кластерам:

$$\mathrm{clust}_i := \arg\max_{c \in C} g_{nc}.$$

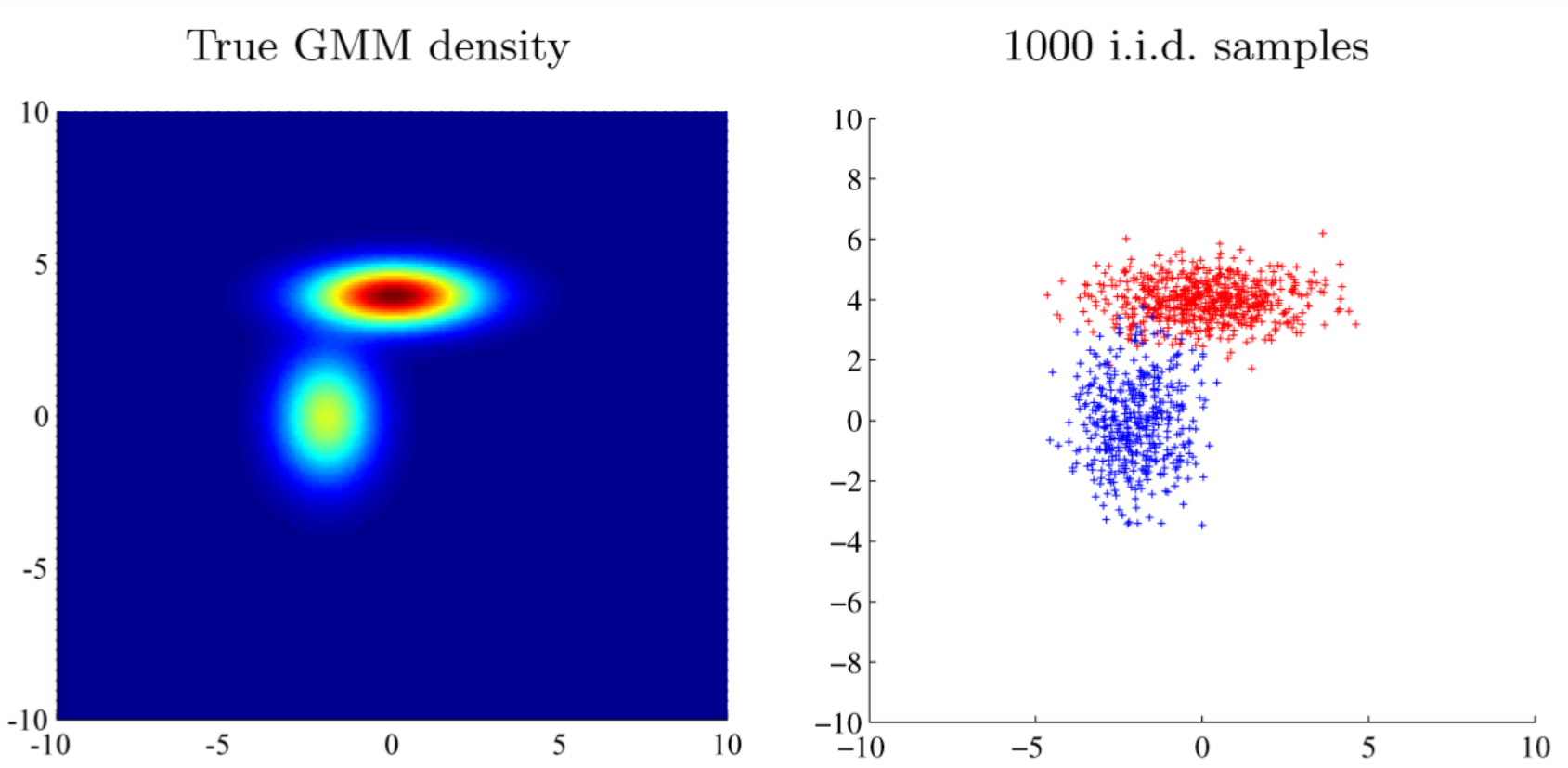
Почему так?

- Как доказать, что E-шаг и M-шаг действительно в данном случае так выглядят?
- На E-шаге для параметров $\boldsymbol{\theta} = (\mathsf{w}, \boldsymbol{\mu}, \boldsymbol{\sigma})$:

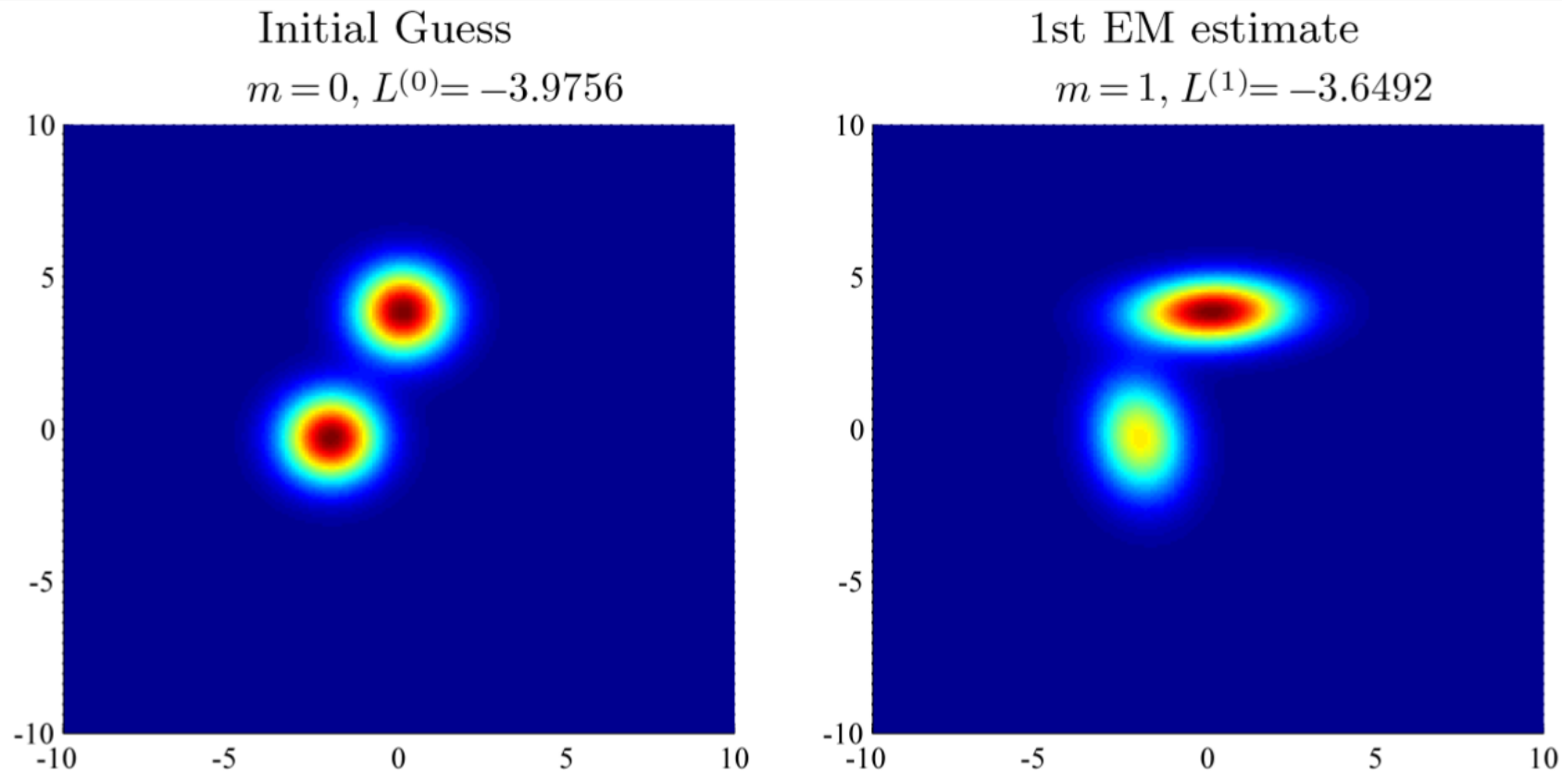
$$
\begin{aligned}
Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) &= \mathbb{E}_{\mathcal{Z} \mid \mathcal{Y}, \boldsymbol{\theta}^{(m)}} \left[\log p(\mathcal{Y}, \mathcal{Z} \mid \boldsymbol{\theta})\right] = \\
&= \mathbb{E}_{\boldsymbol{\theta}^{(m)}} \left[\log \prod_{n=1}^{N} \prod_{c=1}^{C} p(\mathsf{y}_n, z_{nc} \mid \boldsymbol{\theta})^{z_{nc}}\right] = \\
&= \mathbb{E}_{\boldsymbol{\theta}^{(m)}} \left[\sum_{n=1}^{N} \sum_{c=1}^{C} z_{nc} \left(\log p(z_{nc} \mid w_c) + \log p(\mathsf{y}_n \mid \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c)\right)\right] = \\
&= \sum_{n=1}^{N} \sum_{c=1}^{C} \left(\mathbb{E}_{\boldsymbol{\theta}^{(m)}} \left[z_{nc}\right] \log w_c + \mathbb{E}_{\boldsymbol{\theta}^{(m)}} \left[z_{nc}\right] \log p(\mathsf{y}_n \mid \boldsymbol{\mu}_c, \boldsymbol{\sigma}_c)\right).
\end{aligned}
$$

- Отсюда и получается обучение каждого гауссиана независимо, но с весами $g_{nc} = \mathbb{E}_{\boldsymbol{\theta}^{(m)}} \left[z_{nc}\right]$.

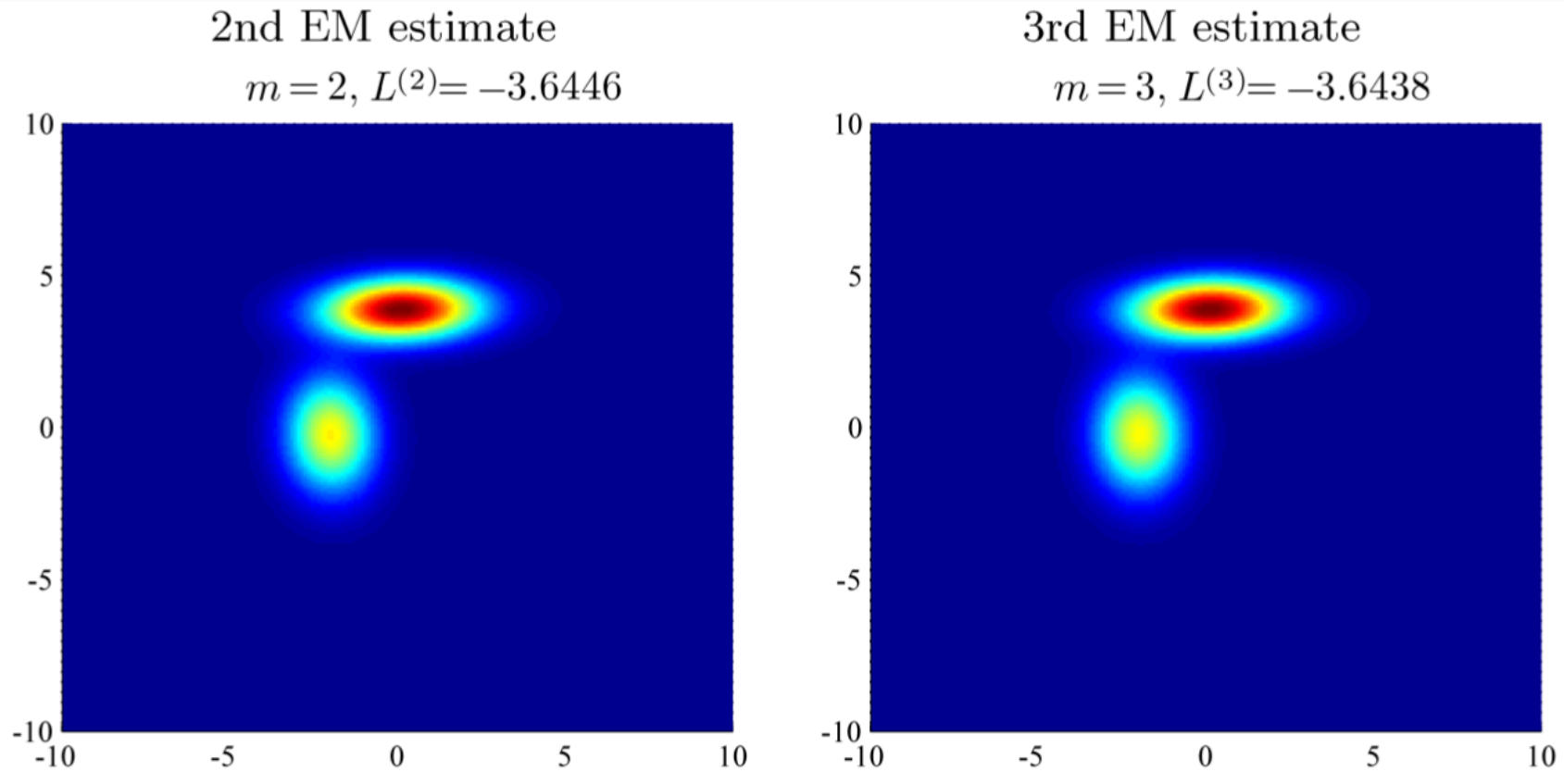
Пример


True GMM density
10
5
0
-5
-10
-10
-5
0
5
10
1000 i.i.d. samples
10
8
6
4
2
0
−2
−4
−6
−8
−10
−10
−5
0
5
10

Пример


Initial Guess
$m = 0,\ L^{(0)} = -3.9756$
1st EM estimate
$m = 1,\ L^{(1)} = -3.6492$
10
5
0
-5
-10
-10
-5
0
5
10

Пример


2nd EM estimate
$m = 2$, $L^{(2)} = -3.6446$
3rd EM estimate
$m = 3$, $L^{(3)} = -3.6438$
10
5
0
-5
-10
-10
-5
0
5
10

Проблема

- Остается проблема: нужно задавать количество кластеров.
- Как её решать?

Свойства и простые расширения EM

Суть алгоритма k–средних

- Один из самых известных алгоритмов кластеризации – алгоритм k-средних – это фактически упрощение алгоритма EM.
- Формальная цель алгоритма k–средних — минимизировать меру ошибки

$$E(\mathcal{Y}, C) = \sum_{n=1}^{n} ||\mathbf{y}_n - \boldsymbol{\mu}_i||^2,$$

  где $\boldsymbol{\mu}_i$ — ближайший к $\mathbf{y}_n$ 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

kMeans($\mathcal{Y}, |C|$):

- Инициализировать центры $|C|$ кластеров $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{|C|}$.
- Пока принадлежность кластерам не перестанет изменяться:
  - Определить принадлежность y_n к кластерам:

$$\mathrm{clust}_n := \arg\min_{c \in C} \rho(\mathsf{y}_n, \boldsymbol{\mu}_c).$$

  - Определить новое положение центров кластеров:

$$\boldsymbol{\mu}_c := \frac{\sum_{\mathrm{clust}_n = c} \mathsf{y}_n}{\sum_{\mathrm{clust}_n = c} 1}.$$

И чем же это от EM отличается?

Point-estimate variant of EM

- Разница в том, что мы не считаем вероятности принадлежности кластерам, а жестко приписываем каждый объект одному кластеру.
- Это на самом деле вариант EM, в котором вместо полного распределения $p(\mathcal{X} \mid \boldsymbol{\theta})$, которое в Q-функции используется, мы берём просто точку максимального правдоподобия
- Point-estimate variant of EM, или Classification EM:

$$\mathsf{z}^{(m)} = \arg\max_{\mathsf{z}} p(\mathsf{z} \mid \mathsf{y}, \boldsymbol{\theta}^{(m)}),$$
$$\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} p(\mathsf{z}^{(m)} \mid \boldsymbol{\theta}^{(m)}).$$

Требования к $\mathcal{X}$ и $\mathcal{Y}$

- Что требуется, чтобы EM-алгоритм работал?
- Неформально нужно, чтобы $p(\mathcal{X} \mid \boldsymbol{\theta})$ было легко максимизировать.
- А формально нужно, чтобы $p(\mathsf{y} \mid \mathsf{x}, \boldsymbol{\theta}) = p(\mathsf{y} \mid \mathsf{x})$, т.е. чтобы выполнялось марковское свойство для $\theta \to \mathsf{x} \to \mathsf{y}$.
- На самом деле обычно EM применяется тогда, когда $\mathsf{y} = f(\mathsf{x})$ для детерминированной функции f, и это свойство тривиально выполняется.
- Более того, обычно EM применяется, когда f — это просто проекция, т.е. когда $\mathsf{x} = (\mathsf{y}, \mathsf{z})$, как мы изначально и рассматривали.

Разложение Q-функции по точкам

- Важное простое свойство — если данные состоят из независимо порождённых $\mathcal{X} = \{\mathsf{x}_1, \ldots, \mathsf{x}_N\}$ (как всегда и бывает), то

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathcal{X}\mid\mathcal{Y},\boldsymbol{\theta}^{(m)}} \left[\log \prod_{n=1}^{N} p(\mathsf{x}_n \mid \boldsymbol{\theta}) \right] =$$

$$= \mathbb{E}_{\mathcal{X}\mid\mathcal{Y},\boldsymbol{\theta}^{(m)}} \left[\sum_{n=1}^{N} \log p(\mathsf{x}_n \mid \boldsymbol{\theta}) \right] = \sum_{n=1}^{N} \mathbb{E}_{\mathsf{x}_n\mid\mathsf{y}_n,\boldsymbol{\theta}^{(m)}} \left[\log p(\mathsf{x}_n \mid \boldsymbol{\theta}) \right],$$

  потому что $p(\mathsf{x}_n \mid \mathcal{Y}, \boldsymbol{\theta}) = p(\mathsf{x}_n \mid \mathsf{y}_n, \boldsymbol{\theta})$
- Упражнение: докажите это!

Generalized EM

- Другие обобщения могут пригодиться, если всё-таки подсчитать $\mathbb{E}_{\mathcal{X}|\mathcal{Y},\boldsymbol{\theta}^{(m)}}[\log p(\mathcal{X} \mid \boldsymbol{\theta})]$ или оптимизировать $p(\mathcal{X} \mid \boldsymbol{\theta})$ нелегко.
- Обобщённый EM (Generalized EM, GEM): вместо $\arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(m)})$ нам достаточно просто выбирать такую $\boldsymbol{\theta}^{(m+1)}$, чтобы

$$Q\left(\boldsymbol{\theta}^{(m+1)}, \boldsymbol{\theta}^{(m)}\right) > Q\left(\boldsymbol{\theta}^{(m)}, \boldsymbol{\theta}^{(m)}\right).$$

Stochastic EM

- Стохастический EM (Stochastic EM): если Q-функцию не получается посчитать в замкнутой форме, но и просто максимум брать не хочется, как в Classification EM, можно попробовать брать $\mathbf{x}$ случайным образом на E-шаге:

$$\mathbf{x}^{(m)} \sim p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)}),$$

а потом использовать его на M-шаге как обычно:

$$\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{x}^{(m)} \mid \boldsymbol{\theta}^{(m)}).$$

Monte Carlo EM

- Монте-Карло EM (Monte Carlo EM): саму Q-функцию тоже можно попытаться приблизить, если подсчитать сложно; можно использовать приближение ожидания в Q-функции через сэмплирование:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) \approx \frac{1}{R} \sum_{r=1}^{R} \log p(\mathbf{x}^{(m,r)} \mid \boldsymbol{\theta}), \text{ где } \mathbf{x}^{(m,r)} \sim p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta}^{(m)}).$$

- Впрочем, в таких случаях часто можно вообще забить на EM и аппроксимировать напрямую апостериорное распределение.

EM с априорным распределением

- А можно и априорное распределение в EM добавить, конечно же:

$$\boldsymbol{\theta}_{MAP} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \mathcal{Z}) = \arg\max_{\boldsymbol{\theta}} \left(\log p(\mathcal{Y} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right).$$

- При этом базовая схема особо не меняется:

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) = \mathbb{E}_{\mathcal{X} \mid \mathcal{Y}, \boldsymbol{\theta}^{(m)}} \left[\log p(\mathcal{X} \mid \boldsymbol{\theta})\right],$$
$$\boldsymbol{\theta}^{(m+1)} = \arg\max_{\boldsymbol{\theta}} \left(Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(m)}) + \log p(\boldsymbol{\theta})\right).$$

- Например, так можно избежать вырожденных случаев (кластер из одной точки).

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

- 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?
- Чтобы учесть информацию о точке $\mathbf{x}_i$, достаточно для EM положить скрытую переменную g_{nc} 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.

Пример EM: presence-only data

Presence-only data

- Пример из экологии: пусть мы хотим оценить, где водятся те или иные животные.
- Как определить, что суслики тут водятся, понятно: видишь суслика — значит, он есть.
- Но как определить, что суслика нет? Может быть, ты не видишь суслика, и я не вижу, а он есть?..

Presence-only data

- Формально говоря, есть переменные $\mathbf{x}$, определяющие некий регион (квадрат на карте), и мы моделируем вероятность того, что нужный вид тут есть, $p(y = 1 \mid \mathbf{x})$, при помощи логит-функции:

$$p(y = 1 \mid \mathbf{x}) = \sigma(\eta(\mathbf{x})) = \frac{1}{1 + e^{-\eta(\mathbf{x})}},$$

  где $\eta(\mathbf{x})$ может быть линейной (тогда получится логистическая регрессия), но может, в принципе, и не быть.
- Заметим, что даже если бы мы знали настоящие y, это было бы ещё не всё: сэмплирование положительных и отрицательных примеров неравномерно, перекошено в пользу положительных.

Presence-only data

- Это значит, что ещё есть пропорции сэмплирования (sampling rates)

$$\gamma_0 = p(s = 1 \mid y = 0), \quad \gamma_1 = p(s = 1 \mid y = 1),$$

  т.е. вероятности взять в выборку положительный/отрицательный пример.

- Их можно оценить как

$$\gamma_0 = \frac{n_0}{(1-\pi)N}, \quad \gamma_1 = \frac{n_1}{\pi N},$$

  где π — истинная доля положительных примеров (встречаемость, occurrence).

- Кстати, эту π было бы очень неплохо оценить в итоге.

Presence-only data

- Тогда, если знать истинные значения всех y, то в принципе можно обучить:

$$p(y = 1 \mid s = 1, \mathbf{x}) =$$
$$= \frac{p(s = 1 \mid y = 1, \mathbf{x})p(y = 1 \mid \mathbf{x})}{p(s = 1 \mid y = 0, \mathbf{x})p(y = 0 \mid \mathbf{x}) + p(s = 1 \mid y = 1, \mathbf{x})p(y = 1 \mid \mathbf{x})} =$$
$$= \frac{\gamma_1 e^{\eta(\mathbf{x})}}{\gamma_0 + \gamma_1 e^{\eta(\mathbf{x})}} = \frac{e^{\eta^*(\mathbf{x})}}{1 + e^{\eta^*(\mathbf{x})}},$$

где $\eta^*(\mathbf{x}) = \eta(\mathbf{x}) + \log(\gamma_1/\gamma_0)$, т.е.

$$\eta^*(\mathbf{x}) = \eta(\mathbf{x}) + \log\left(\frac{n_1}{n_0}\right) - \log\left(\frac{\pi}{1-\pi}\right).$$

Presence-only data

- Таким образом, если π неизвестно, то $\eta(\mathbf{x})$ можно найти с точностью до константы.
- А у нас не n_0 и n_1, а naive presence n_p и background n_u, т.е.

$$p(y=1 \mid s=1) = \frac{n_p+\pi n_u}{n_p+n_u}, \quad p(y=1 \mid s=0) = \frac{(1-\pi)n_u}{n_p+n_u},$$
$$\gamma_1 = \frac{p(y=1|s=1)p(s=1)}{p(y=1)} = \frac{n_p+\pi n_u}{\pi(n_p+n_u)}p(s=1),$$
$$\gamma_0 = \frac{p(y=0|s=1)p(s=1)}{p(y=0)} = \frac{n_u}{n_p+n_u}p(s=1),$$

  и в нашей модели $\log \frac{n_1}{n_0} = \log \frac{n_p+\pi n_u}{\pi n_u}$, т.е. всё как раньше, но $n_1 = n_p + \pi n_u$, $n_0 = (1-\pi)n_u$.
- Обучать по y можно так: обучить модель, а потом вычесть из $\eta(\mathbf{x})$ константу $\log \frac{n_p+\pi n_u}{\pi n_u}$.

Presence-only data

- Но у нас нет настоящих данных y, чтобы обучить регрессию, а есть только presence-only z: если $z = 1$, то $y = 1$, но если $z = 0$, то неизвестно, чему равен y.
- (Ward et al., 2009): давайте использовать EM. Правдоподобие:

$$\mathcal{L}(\eta \mid \mathbf{y}, \mathbf{z}, \mathcal{X}) = \prod_i p(y_i, z_i \mid s_i = 1, \mathbf{x}_i) = \\ = \prod_i p(y_i \mid s_i = 1, \mathbf{x}_i) p(z_i \mid y_i, s_i = 1, \mathbf{x}_i).$$

- В нашем случае при n_p положительных примеров и n_u фоновых (неизвестных)

$$\begin{aligned} p(z_i = 0 \mid y_i = 0, s_i = 1, \mathbf{x}_i) &= 1, \\ p(z_i = 1 \mid y_i = 1, s_i = 1, \mathbf{x}_i) &= \frac{n_p}{n_p + \pi n_u}, \\ p(z_i = 0 \mid y_i = 1, s_i = 1, \mathbf{x}_i) &= \frac{\pi n_u}{n_p + \pi n_u}. \end{aligned}$$

Presence-only data

- А максимизировать нам надо сложное правдоподобие, в котором значения $\mathbf{y}$ неизвестны:

$$L(\eta \mid \mathbf{z}, \mathcal{X}) = \prod_i p(z_i \mid s_i = 1, \mathbf{x}) =$$

$$= \prod_i \left(\frac{\frac{n_p}{\pi n_u} e^{\eta(\mathbf{x}_i)}}{1 + \left(1 + \frac{n_p}{\pi n_u}\right) e^{\eta(\mathbf{x}_i)}} \right)^{z_i} \left(\frac{1 + e^{\eta(\mathbf{x}_i)}}{1 + \left(1 + \frac{n_p}{\pi n_u}\right) e^{\eta(\mathbf{x}_i)}} \right)^{1 - z_i}.$$

- Для этого и нужен EM.

Presence-only data

- E-шаг здесь в том, чтобы заменить y_i на его оценку

$$\hat{y}_i^{(k)} = \mathbb{E}\left[y_i \mid \eta^{(k)}\right] = \frac{e^{\eta^{(k)}} + 1}{1 + e^{\eta^{(k)}} + 1}.$$

- M-шаг мы уже видели, это обучение параметров логистической модели с целевой переменной $\mathbf{y}^{(k)}$ на данных $\mathcal{X}$.

(1) Chose initial estimates: $\hat{y}_i^{(0)} = \pi$ for $z_i = 0$.
(2) Repeat until convergence:

- *Maximization step:*
  - Calculate $\hat{\eta}^{*(k)}$ by fitting a logistic model of $\hat{\mathbf{y}}^{(k-1)}$ given X.
  - Calculate $\hat{\eta}^{(k)} = \hat{\eta}^{*(k)} - \log\left(\frac{n_p + \pi n_u}{\pi n_u}\right)$.
- *Expectation step:*

$$\hat{y}_i^{(k)} = \frac{e^{\hat{\eta}^{(k)}}}{1 + e^{\hat{\eta}^{(k)}}} \text{ for } z_i = 0 \quad \text{and} \quad \hat{y}_i^{(k)} = 1 \text{ for } z_i = 1$$

Presence-only data

- У Ward et al. получалось хорошо, но тут вышел любопытный спор.
- Ward et al. писали так: хотелось бы, чтобы можно было оценить π, но "π is identifiable only if we make unrealistic assumptions about the structure of $\eta(\mathbf{x})$ such as in logistic regression where $\eta(\mathbf{x})$ is linear in $\mathbf{x}$: $\eta(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\beta}$".
- Через пару лет вышла статья Royle et al. (2012), тоже очень цитируемая, в которой говорилось: "logistic regression... is hardly unrealistic... such models are the most common approach to modeling binary variables in ecology (and probably all of statistics)... the logistic functions... is customarily adopted and widely used, and even books have been written about it".

Presence-only data

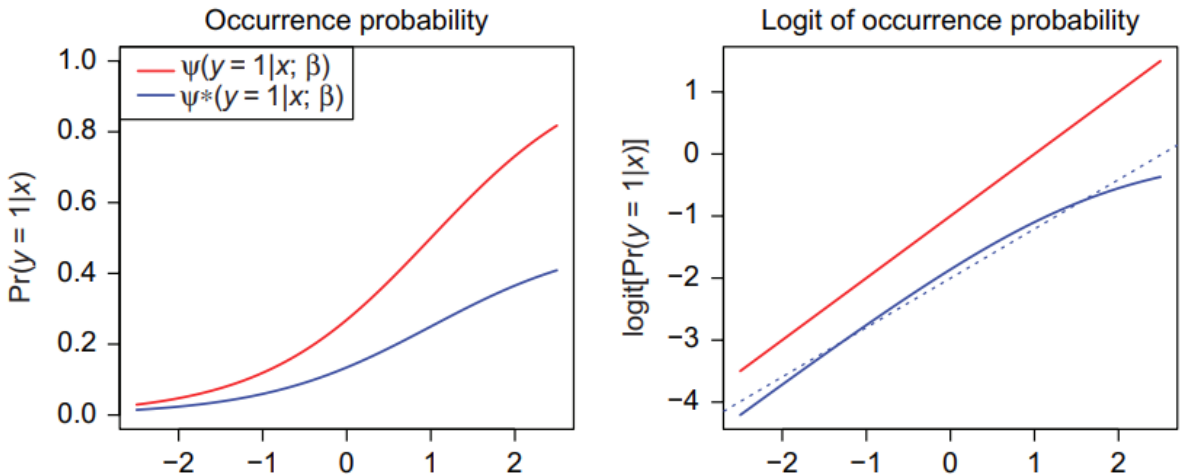
- Они предложили процедуру для оценки встречаемости π:
  - для признаков $\mathbf{x}$ у нас $p(y = 1 \mid \mathbf{x}) = \frac{p(y=1)\pi_1(\mathbf{x})}{p(y=1)\pi_1(\mathbf{x})+(1-p(y=1))\pi_0(\mathbf{x})}$;
  - данные – это выборка из $\pi_1(\mathbf{x})$ и отдельно выборка из $\pi(\mathbf{x})$;
  - как видно, даже если знать π и π_1 полностью, остаётся свобода: надо оценить $p(y = 1 \mid \mathbf{x})$, а $\pi_0(\mathbf{x})$ мы не знаем;
  - Royle et al. вводят предположения для $p(y = 1 \mid \mathbf{x})$ в виде логистической регрессии: $p(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x})$;
  - тогда действительно можно записать $p(y = 1)\pi_1(\mathbf{x}) = p(y = 1 \mid \mathbf{x})\pi(\mathbf{x}) = p(y = 1, \mathbf{x})$, и если $\pi(\mathbf{x})$ равномерно (это логично), то

$$\pi_1(\mathbf{x}_i) = \frac{p(y_i = 1 \mid \mathbf{x}_i)}{\sum_{\mathbf{x}} p(y = 1 \mid \mathbf{x})};$$

    если подставить сюда логистическую регрессию, то можно оценить $\boldsymbol{\beta}$ максимального правдоподобия, и не надо знать $p(y = 1)$!
- Что тут не так?

Presence-only data

- Всё так, но предположение о логистической регрессии здесь выполняет слишком много работы.
- Если рассмотреть две кривые, у которых $p^*(y = 1 \mid \mathbf{x}, \boldsymbol{\beta}) = \frac{1}{2} p(y = 1 \mid \mathbf{x}, \boldsymbol{\beta})$, то у них будет $p^*(y = 1) = \frac{1}{2} p(y = 1)$, но общее правдоподобие $\pi_1(\mathbf{x}_i)$ будет в точности одинаковое, $\frac{1}{2}$ сократится.
- Дело в том, что модель p^* не будет логистической регрессией, с $\boldsymbol{\beta}$ произойдёт что-то нелинейное; но откуда у нас настолько сильное предположение? Как отличить синюю кривую справа от пунктирной прямой?



Спасибо!

Спасибо за внимание!