Лекция 6. Скрытые марковские модели. Часть 1.

А. С. Конушин[1] Д. П. Ветров[2] Д. А. Кропотов[3]
В. С. Конушин[1] О. В. Баринова[1]

[1]МГУ, ВМиК, лаб. КГ

[2]МГУ, ВМиК, каф. ММП

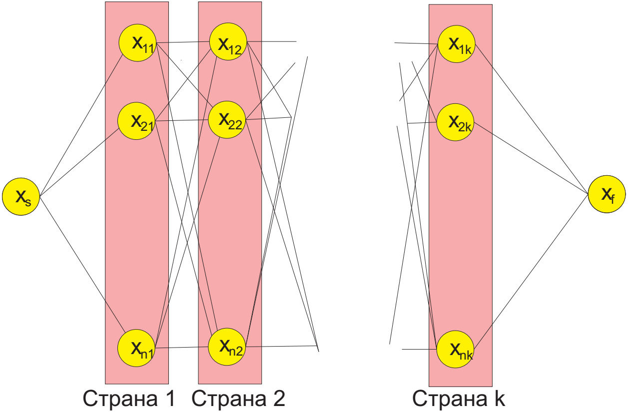
[3]ВЦ РАН

Спецкурс «Структурные методы анализа изображений и сигналов»

План

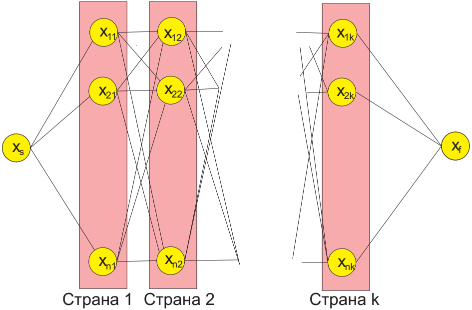
План

Задача объезда стран



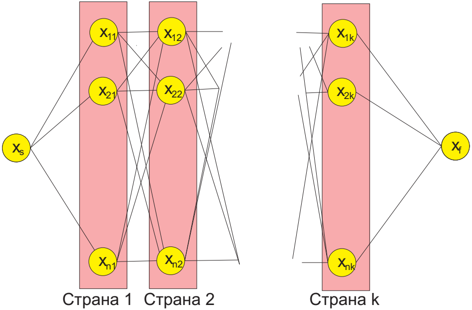
- Рассмотрим такую задачу: необходимо в определенной последовательности объехать k стран, в каждой из которых провести одну ночь в отеле, потратив минимум денег
- Если в каждой стране имеется n городов, то задача сводится к перебору n^k вариантов

Функция Беллмана



- Пусть цена билета между городами x_{ij} и $x_{i+1,l}$ задается функцией $f(x_{ij}, x_{i+1,l})$, а цена ночлега в городе — функцией $h(x_{ij})$
- Подсчитаем функцию Беллмана $V(x)$, определяемую рекуррентно: $V(x_s) = 0$
$$V(x_{i+1,l}) = \min_j \left[V(x_{ij}) + f(x_{ij}, x_{i+1,l}) + h(x_{i+1,l}) \right]$$
- Физический смысл функции Беллмана это наименьшая сумма денег, которую нужно потратить, чтобы добраться из города x_s в город $x_{i+1,l}$ (легко показать по индукции)

Динамическое программирование



- Определим также функцию $S(x)$, возвращающую город, откуда мы приехали в город x: $S(x_s) = \emptyset$

$$S(x_{i+1,l}) = \arg\min_j \left[V(x_{ij}) + f(x_{ij}, x_{i+1,l}) + h(x_{i+1,l})\right]$$

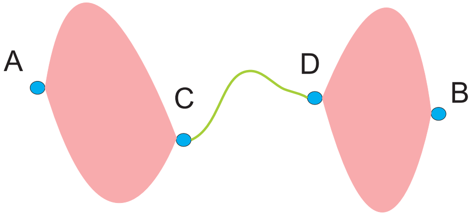
- Тогда оптимальный путь $(x_1^*, x_2^*, \ldots, x_k^*)$ может быть получен путем рекуррентного вызова функции $S(x)$: $x_n^* = S(x_f)$

$$x_m^* = S(x_{m+1}^*)$$

Особенности динамического программирования

Метод динамического программирования эффективен, если имеет место

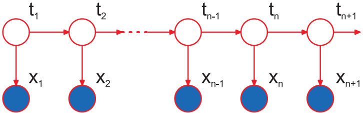
- Перекрывающиеся подзадачи, которые необходимо решить, чтобы решить исходную задачу
- Оптимальная подструктура (принцип кусочной оптимальности)
- Возможность запомнить решения подзадач



Если известно, что оптимальный путь проходит через точки C и D и известен оптимальный путь между ними, то этот путь станет частью оптимального пути между A и B

План

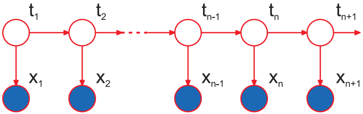
Скрытая Марковская модель (СММ)



Скрытая Марковская модель [первого порядка] — это вероятностная модель последовательности, которая

- Состоит из набора наблюдаемых переменных $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_n \in \mathbb{R}^d$ и латентных (скрытых) переменных $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N\}$, $\boldsymbol{t}_n \in \{0, 1\}^K$, $\sum_{j=1}^K t_{nj} = 1$
- Латентные переменные T являются бинарными и кодируют K состояний, поэтому их иногда называют переменными состояния
- Значение наблюдаемого вектора $\boldsymbol{x}_n$, взятого в момент времени n, зависит только от скрытого состояния $\boldsymbol{t}_n$, которое в свою очередь зависит только от скрытого состояния в предыдущий момент времени $\boldsymbol{t}_{n-1}$

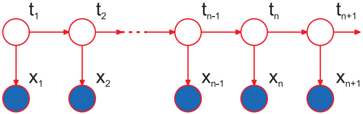
Примеры использования СММ



Что можно анализировать с помощью СММ

- Речь
- Видео
- Поведение
- Фондовые рынки
- Естественный язык
- ДНК
- и др.

Скрытая Марковская модель (СММ)



- Скрытая марковская модель является частным случаем байесовской сети (графической модели, задаваемой ориентированным графом)
- Граф, задающий СММ, является ациклическим, поэтому для СММ существуют эффективные алгоритмы вывода
- Для полного задания модели достаточно задать все условные распределения вида $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$, $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$ и априорное распределение $p(\boldsymbol{t}_1)$

Спецификация вероятностной модели

Пусть имеется K возможных состояний. Закодируем состояние в каждый момент времени n бинарным вектором $\boldsymbol{t}_n = (t_{n1}, \ldots, t_{nK})$, где

$$t_{nj} = \begin{cases} 1, & \text{если в момент } n \text{ модель находится в состоянии } j \\ 0, & \text{иначе} \end{cases}$$

Тогда распределение $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$ можно задать матрицей перехода A размера $K \times K$, где $A_{ij} = p(t_{nj} = 1|t_{n-1,i} = 1)$, $\sum_j A_{ij} = 1$, т.е.

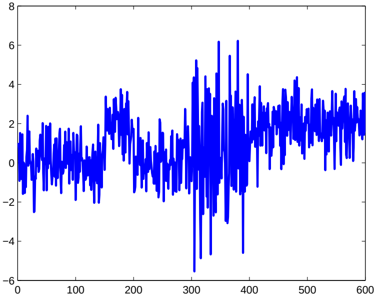
$$p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) = \prod_{i=1}^{K} \prod_{j=1}^{K} A_{ij}^{t_{n-1,i}t_{nj}}$$

Пусть в первый момент времени $p(t_{1j} = 1) = \pi_j$. Тогда

$$p(\boldsymbol{t}_1) = \prod_{j=1}^{K} \pi_j^{t_{1j}}$$

Замечание

- Хотя матрица A может быть произвольного вида с учетом ограничений на неотрицательность и сумму элементов строки, с точки зрения СММ представляет интерес диагональное преобладание матрицы перехода
- В этом случае можно ожидать, что процесс находится в некотором состоянии на протяжении какого-то отрезка времени
- Появляется простая физическая интерпретация СММ: имеется процесс, который иногда (относительно редко) скачкообразно меняет свои характеристики



Спецификация вероятностной модели

- Условное распределение $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ определяется текущим состоянием $\boldsymbol{t}_n$
- Обычно предполагают, что оно нам известно с точностью до параметров ϕ_k, $k \in \{1, \ldots, K\}$, т.е. если $t_{n1} = 1$, то $\boldsymbol{x}_n$ взят из распределения $p(\boldsymbol{x}_n|\phi_1)$, если $t_{n2} = 1$, то $\boldsymbol{x}_n$ взят из распределения $p(\boldsymbol{x}_n|\phi_2)$, и т.д.
- Таким образом

$$p(\boldsymbol{x}_n|\boldsymbol{t}_n) = \prod_{j=1}^{K} (p(\boldsymbol{x}_n|\phi_k))^{t_{nk}}$$

Задачи в СММ

Обозначим полный набор параметров $\Theta = \{\boldsymbol{\pi}, A, \phi\}$. Тогда основные задачи, возникающие в СММ, можно сформулировать следующим образом:

- Обучение с учителем. Известна некоторая последовательность X, для которой заданы T. Задача состоит в оценке по обучающей выборке набора параметров Θ.
- Сегментация. Известна некоторая последовательность X и набор параметров Θ. Задача состоит в получении наиболее правдоподобной последовательности состояний T как $\arg\max_T p(T|X, \Theta)$ (алгоритм Витерби).
- Обучение без учителя. Известна некоторая последовательность X и число состояний K. Задача состоит в оценке параметров Θ (ЕМ-алгоритм).
  - Нахождение маргинального распределения $p(\boldsymbol{t}_n|X, \Theta)$ компоненты $\boldsymbol{t}_n$ по заданным X и Θ
- Прогнозирование. Известна некоторая последовательность X. Задача состоит в оценке наблюдаемого вектора в следующий момент времени $N+1$ — $p(\boldsymbol{x}_{N+1}|X)$.

План

Совместное распределение переменных СММ

- Предположим, нам задана обучающая выборка (X, T), представляющая собой одну или несколько последовательностей, в которых известны значения скрытых компонент
- Требуется оценить вектор параметров Θ
- По построению байесовской сети совместное распределение переменных задается формулой

$$p(X,T|\Theta) = p_{\boldsymbol{\pi}}(\boldsymbol{t}_1)\prod_{n=1}^{N} p_{\boldsymbol{\phi}}(\boldsymbol{x}_n|\boldsymbol{t}_n)\prod_{n=2}^{N} p_A(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) =$$

$$= \prod_{j=1}^{K}\pi_j^{t_{1j}}\left(\prod_{n=2}^{N}\prod_{i=1}^{K}\prod_{j=1}^{K} A_{ij}^{t_{n-1,i}t_{nj}}\right)\left(\prod_{n=1}^{N}\prod_{k=1}^{K}(p(\boldsymbol{x}_n|\phi_k))^{t_{nk}}\right)$$

Метод максимального правдоподобия

- Для оценки параметров Θ воспользуемся методом максимального правдоподобия

$$\Theta_{ML} = \arg\max p(X, T|\Theta) = \arg\max \log p(X, T|\Theta)$$

- Для удобства перейдем к логарифму функции правдоподобия (положение максимума, очевидно, не изменится)

$$\log p(X, T|\Theta) = \left(\sum_{j=1}^{K} t_{1j} \log \pi_j\right) +$$
$$+ \left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i} t_{nj} \log A_{ij}\right) + \left(\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log p(\boldsymbol{x}_n|\phi_k)\right)$$

Функция Лагранжа

- Параметры, входящие в Θ, не могут принимать произвольные значения, следовательно необходима оптимизация при ограничениях

$$\sum_{j=1}^{K} \pi_j = 1, \quad \sum_{j=1}^{K} A_{ij} = 1, \quad \forall i = \overline{1, K}$$

- Воспользуемся правилом множителей Лагранжа и выпишем лагранжиан

$$\mathcal{L}(\Theta, \lambda, \boldsymbol{\mu}) = \log p(X, T|\Theta) + \lambda \left(\sum_{j=1}^{K} \pi_j - 1 \right) + \sum_{i=1}^{K} \mu_i \left(\sum_{j=1}^{K} A_{ij} - 1 \right) \to \mathrm{extr}$$

Оценка $\boldsymbol{\pi}$

$$\log p(X,T|\Theta) = \left(\sum_{j=1}^{K} t_{1j}\log \pi_j\right) +$$

$$+\left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij}\right) + \left(\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)\right)$$

$$\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu}) = \log p(X,T|\Theta) + \lambda\left(\sum_{j=1}^{K}\pi_j - 1\right) + \sum_{i=1}^{K}\mu_i\left(\sum_{j=1}^{K}A_{ij} - 1\right) \to \text{extr}$$

$$\frac{\partial\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu})}{\partial\pi_j} = \frac{t_{1j}}{\pi_j} + \lambda = 0 \;\Rightarrow\; \pi_j = -\frac{t_{1j}}{\lambda}$$

$$\sum_{j=1}^{K}\pi_j = 1 \;\Rightarrow\; \lambda = -\sum_{j=1}^{K} t_{1j} = -1$$

$$\pi_j = t_{1j}$$

Оценка матрицы A

$$\log p(X,T|\Theta)=\left(\sum_{j=1}^{K}t_{1j}\log\pi_j\right)+$$
$$+\left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K}t_{n-1,i}t_{nj}\log A_{ij}\right)+\left(\sum_{n=1}^{N}\sum_{k=1}^{K}t_{nk}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)\right)$$

$$\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu})=\log p(X,T|\Theta)+\lambda\left(\sum_{j=1}^{K}\pi_j-1\right)+\sum_{i=1}^{K}\mu_i\left(\sum_{j=1}^{K}A_{ij}-1\right)\to\mathrm{extr}$$

$$\frac{\partial\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu})}{\partial A_{ij}}=\sum_{n=2}^{N}\frac{t_{n-1,i}t_{nj}}{A_{ij}}+\mu_i=0\ \Rightarrow\ A_{ij}=-\sum_{n=2}^{N}\frac{t_{n-1,i}t_{1j}}{\mu_i}$$

$$\sum_{j=1}^{K}A_{ij}=1\ \Rightarrow\ \mu_i=-\sum_{n=2}^{N}\sum_{j=1}^{K}t_{n-1,i}t_{nj}=-\sum_{n=2}^{N}t_{n-1,i}$$

$$A_{ij}=\frac{\sum_{n=2}^{N}t_{n-1,i}t_{nj}}{\sum_{n=2}^{N}t_{n-1,i}}$$

Оценка условной плотности $p(\boldsymbol{x}|\boldsymbol{t})$

$$\log p(X,T|\Theta) = \left(\sum_{j=1}^{K} t_{1j}\log \pi_j\right) +$$

$$+\left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij}\right) + \left(\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)\right)$$

$$\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu}) = \log p(X,T|\Theta) + \lambda\left(\sum_{j=1}^{K}\pi_j - 1\right) + \sum_{i=1}^{K}\mu_i\left(\sum_{j=1}^{K}A_{ij} - 1\right) \to \mathrm{extr}$$

$$\frac{\partial\mathcal{L}(\Theta,\lambda,\boldsymbol{\mu})}{\partial\boldsymbol{\phi}_k} = \sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\frac{\partial\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)}{\boldsymbol{\phi}_k} = \sum_{\{n:\ t_{nk}=1\}}\frac{\partial\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)}{\boldsymbol{\phi}_k} = 0$$

Получили стандартную задачу максимизации правдоподобия по выборке независимых одинаково-распределенных объектов

$$\boldsymbol{\phi}_k = \arg\max \sum_{\{n:\ t_{nk}=1\}}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)$$

Для оценки параметров $\boldsymbol{\phi}_k$ можно воспользоваться методами восстановления плотностей, например, ЕМ-алгоритмом

План

Сегментация сигнала

- Пусть известна некоторая последовательность наблюдений X и набор параметров СММ Θ. Требуется определить наиболее вероятную последовательность состояний T, т.е. найти $\arg\max_T p(T|X,\Theta)$
- Заметим, что $p(X|\Theta)$ не зависит от T, поэтому

$$\arg\max_T p(T|X,\Theta) = \arg\max_T \frac{p(X,T|\Theta)}{p(X|\Theta)} = \\ = \arg\max_T p(X,T|\Theta) = \arg\max_T \log p(X,T|\Theta)$$

- Но это же классическая задача динамического программирования!

Аналогия с задачей объезда стран

- В самом деле логарифм совместной плотности по определению

$$\log p(X, T|\Theta) = \left(\sum_{j=1}^{K} t_{1j} \log \pi_j\right) + \\ + \left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj} \log A_{ij}\right) + \left(\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \log p(\boldsymbol{x}_n|\phi_k)\right)$$

- Первое слагаемое определяет «пункт отбытия», второе слагаемое — стоимость перезда из города страны $n-1$ в город в стране n, а третье слагаемое отражает «стоимость ночлега» в выбранном городе страны n

Алгоритм Витерби

$$\log p(X,T|\Theta) = \left(\sum_{j=1}^{K} t_{1j}\log\pi_j\right) + \\ + \left(\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij}\right) + \left(\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)\right)$$

Функция Беллмана $V(t_{1j}) = \log\pi_j$

$$V(t_{nj}) = \max_i \left[V(t_{n-1,i}) + \left(\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij}\right) + \log p(\boldsymbol{x}_n|\boldsymbol{\phi}_j)\right]$$

Функция $S(t_{nj})$ определяется аналогично: $S(t_{1j}) = \emptyset$

$$S(t_{nj}) = \arg\max_i \left[V(t_{n-1,i}) + \left(\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij}\right) + \log p(\boldsymbol{x}_n|\boldsymbol{\phi}_j)\right]$$

Алгоритм Витерби

$$V(t_{nj}) = \max_i \left[V(t_{n-1,i}) + \left(\sum_{i=1}^{K} \sum_{j=1}^{K} t_{n-1,i} t_{nj} \log A_{ij} \right) + \log p(\boldsymbol{x}_n | \boldsymbol{\phi}_j) \right]$$

$$S(t_{nj}) = \arg\max_i \left[V(t_{n-1,i}) + \left(\sum_{i=1}^{K} \sum_{j=1}^{K} t_{n-1,i} t_{nj} \log A_{ij} \right) + \log p(\boldsymbol{x}_n | \boldsymbol{\phi}_j) \right]$$

Выполнив прямой проход по сигналу, мы оцениваем $V(t_{nj})$ и $S(t_{nj})$, а выполнив обратный проход, мы получаем оптимальные номера оптимальных состояний $(i^*(1), \dots, i^*(N))$: $i^*(N) = \arg\max_i V(t_{Ni})$

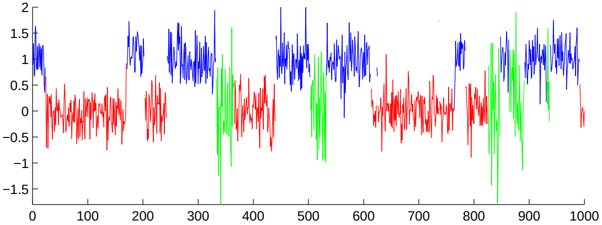
$$i^*(n) = S(t_{n+1, i^*(n+1)})$$

Легко видеть, что значения переменных $\boldsymbol{t}_n$ определяются так: $t_{n,i^*(n)} = 1$, $t_{ni} = 0$, $\forall i \neq i^*(n)$

- Алгорим Витерби позволяет быстро проводить сегментацию очень длинных сигналов
- Существует версия алгоритма Витерби, позволяющая осуществлять сегментацию в реальном времени (точнее, с небольшой задержкой)

Пример использования

Пример разметки сигнала на три состояния с помощью алгоритма Витерби

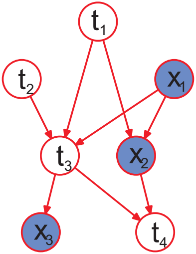


План

Максимум неполного правдоподобия

- Предположим имеется графическая модель, в которой известна только часть значений переменных
- Атомарные распределения известны с точностью до вектора параметров $\boldsymbol{\theta}$
- Требуется оценить параметры по наблюдаемым величинам с помощью метода максимального правдоподобия, т.е. найти

$$\boldsymbol{\theta}_{ML} = \arg\max p(X|\boldsymbol{\theta})$$



Трудности оптимизации неполного правдоподобия

- По правилу суммирования вероятностей неполное правдоподобие может быть получено в виде суммирования по скрытым переменным полного правдоподобия

$$p(X|\boldsymbol{\theta}) = \sum_T p(X, T|\boldsymbol{\theta})$$

- Во многих случаях (в частности, в байесовских сетях) подсчет полного правдоподобия тривиален
- При оптимизации правдоподобия удобно переходить к логарифму, в частности выше мы получили явные формулы для $\arg\max_{\boldsymbol{\theta}} p(X, T|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log p(X, T|\boldsymbol{\theta})$ в СММ
- Прямая оптимизация логарифма неполного правдоподобия очень затруднительна даже в итерационной форме, т.к. функционал имеет вид «логарифм суммы», в то время как удобно оптимизировать «сумму логарифмов»

$$\log p(X|\boldsymbol{\theta}) = \log \sum_T p(X, T|\boldsymbol{\theta})$$

Схема ЕМ-алгоритма

- На входе: выборка X, зависящая от набора параметров $\boldsymbol{\theta}$
- Инициализируем $\boldsymbol{\theta}$ некоторыми начальным приближением
- Е-шаг: Оцениваем распределение скрытой компоненты при фиксированном значении параметров $\boldsymbol{\theta}_{old}$

$$p(T|X, \boldsymbol{\theta}_{old}) = \frac{p(X, T|\boldsymbol{\theta}_{old})}{\sum_T p(X, T|\boldsymbol{\theta}_{old})}$$

- М-шаг: Оптимизируем

$$\mathbb{E}_{T|X,\boldsymbol{\theta}_{old}} \log p(X, T|\boldsymbol{\theta}) = \sum_T p(T|X, \boldsymbol{\theta}_{old}) \log p(X, T|\boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}}$$

  Если бы мы точно знали значение $T = T_0$, то вместо мат. ожидания по всевозможным (с учетом наблюдаемых данных) $T|X, \boldsymbol{\theta}_{old}$, мы бы оптимизировали $\log p(X, T_0|\boldsymbol{\theta})$
- Переход к Е-шагу, пока процесс не сойдется

Замечания

- Оптимизация проводится итерационно методом покоординатного спуска: на каждой итерации последовательно уточняются возможные значения T (Е-шаг), а потом пересчитываются значения $\boldsymbol{\theta}$ (М-шаг)
- Во многих случаях на М-шаге можно получить явные формулы, т.к. там происходит оптимизация выпуклой комбинации логарифмов полных правдоподобий

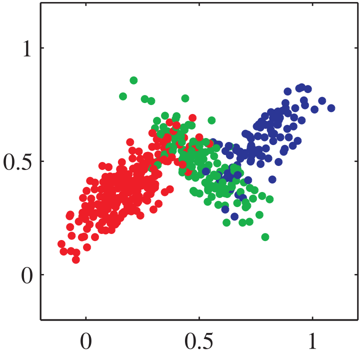
$$\sum_T p(T|X, \boldsymbol{\theta}_{old}) \log p(X, T|\boldsymbol{\theta}) \to \max_{\boldsymbol{\theta}},$$

имеющей вид взвешенной «суммы логарифмов»

План

Смесь гауссовских распределений

- Имеется выборка $X \sim \sum_{j=1}^{l} w_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \Sigma_j)) \in \mathbb{R}^d$, $w_j \geq 0$, $\sum_{j=1}^{l} w_j = 1$
- Требуется восстановить плотность генеральной совокупности



ЕМ-алгоритм

- Выбираем начальное приближение $\boldsymbol{\mu}_j$, w_j, Σ_j
- Е-шаг: Вычисляем распределение скрытых переменных $\boldsymbol{z}_i \in \{0,1\}$, $\sum_j z_{ij} = 1$, которые определяют, к какой компоненте смеси принадлежит объект $\boldsymbol{x}_i$

$$\gamma(z_{ij}) = \frac{w_j \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \Sigma_j))}{\sum_{k=1}^{l} w_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \Sigma_k))}$$

- М-шаг: С учетом новых вероятностей на $\boldsymbol{z_i}$, пересчитываем параметры смеси

$$\boldsymbol{\mu}_j^{new} = \frac{1}{N_j}\sum_{i=1}^{n}\gamma(z_{ij})\boldsymbol{x}_i \qquad w_j^{new} = \frac{N_j}{n} \qquad N_j = \sum_{i=1}^{n}\gamma(z_{ij})$$

$$\Sigma_j^{new} = \frac{1}{N_j}\sum_{i=1}^{n}\gamma(z_{ij})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{new})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{new})^T$$

- Переход к Е-шагу, пока не будет достигнута сходимость

Вывод формул на М-шаге

Выражение для мат. ожидания логарифма правдоподобия по апостериорному распределению имеет вид

$$\mathbb{E}_{Z|X,\boldsymbol{\theta}} \log p(X, Z|\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{l} \gamma(z_{ij}) \left(\log w_j + \log \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right)$$

Дифференцирование по $\boldsymbol{\mu}_j$ и $\boldsymbol{\Sigma}_j$ (точнее по $\boldsymbol{\Sigma}_j^{-1}$) выполняется аналогично случаю одной многомерной гауссианы, только объекты теперь берутся с весом $\gamma(z_{ij})$.

Оптимизация по w_j проводится с учетом органичения $\sum_{j=1}^{l} w_j = 1$ по правилу множителей Лагранжа

$$\sum_{i=1}^{n} \sum_{j=1}^{l} \gamma(z_{ij}) \log w_j + \lambda \left(\sum_{j=1}^{l} w_j - 1 \right)$$

Дифференцируя лагранжиан по w_j получаем уравнения

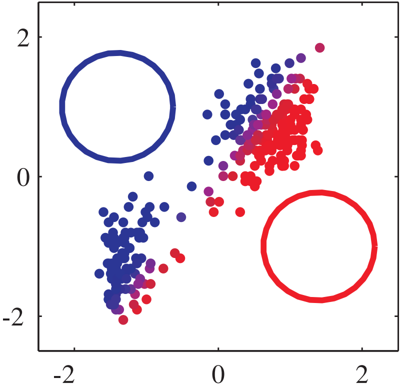
$$\frac{\sum_{i=1}^{n} \gamma(z_{ij})}{w_j} + \lambda = 0, \quad w_j = -\frac{\sum_{i=1}^{n} \gamma(z_{ij})}{\lambda}$$

Учитывая, что $\sum_{j=1}^{l} w_j = 1$ окончательно получаем

$$\lambda = -n, \quad w_j = \frac{\sum_{i=1}^{n} \gamma(z_{ij})}{n} = \frac{N_j}{n}$$

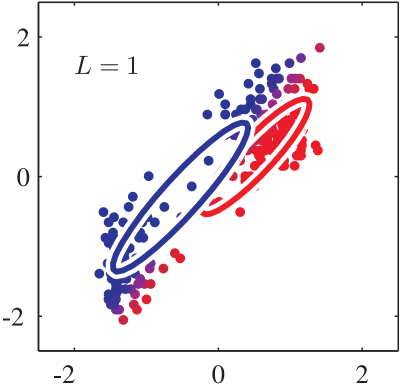
Пример работы ЕМ-алгоритма

Применение ЕМ-алгоритма для разделения смеси двух гауссиан. Итерация 0.



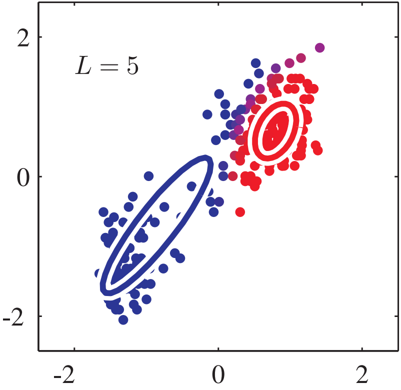
Пример работы EM-алгоритма

Применение EM-алгоритма для разделения смеси двух гауссиан. Итерация 1.



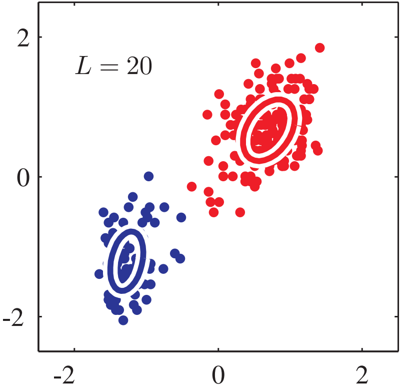
Пример работы ЕМ-алгоритма

Применение ЕМ-алгоритма для разделения смеси двух гауссиан. Итерация 5.



Пример работы EM-алгоритма

Применение EM-алгоритма для разделения смеси двух гауссиан. Итерация 20.



Недостатки ЕМ-алгоритма

- В зависимости от выбора начального приближения может сходиться к разным точкам
- ЕМ-алгоритм находит локальный экстремум, в котором значение правдоподобия может оказаться намного ниже, чем в глобальном максимуме
- ЕМ-алгоритм не позволяет определить количество компонентов смеси l
- !! Величина l является структурным параметром!!

Применение ЕМ-алгоритма для описания сложных плотностей

Смесью гауссиан можно эффективно приближать сложные, плохо-параметризуемые плотности. При этом получаются аналитически заданные приближения