Лекция 8. Скрытые марковские модели. Часть 2.

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

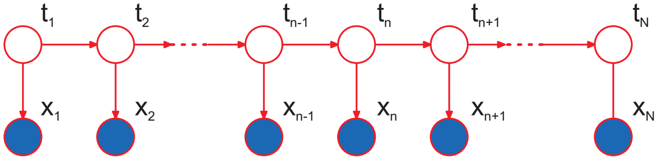
[1]МГУ, ВМиК, лаб. КГ

[2]МГУ, ВМиК, каф. ММП

[3]ВЦ РАН

Спецкурс «Структурные методы анализа изображений и сигналов»

Скрытая Марковская модель (СММ)



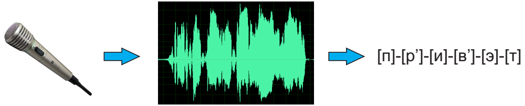
$X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ — произвольные
$T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_N\}$, $\boldsymbol{t}_n \in \{1, \ldots, K\}$ — дискретные

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

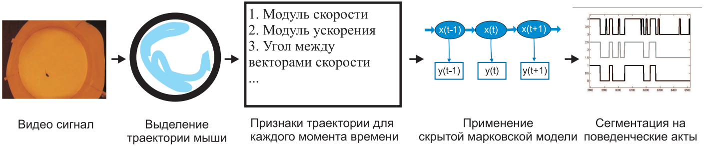
$$\log p(X, T) = \underbrace{\log p(\boldsymbol{t}_1) + \sum_{n=1}^{N} \log p(\boldsymbol{x}_n | \boldsymbol{t}_n)}_{\text{унарные потенциалы}} + \underbrace{\sum_{n=2}^{N} \log p(\boldsymbol{t}_n | \boldsymbol{t}_{n-1})}_{\text{бинарный потенциал}}$$

Примеры использования

- Распознавание речи



- Анализ поведения



- Распознавание генов в ДНК
- Машинный перевод
- Криптография
- И др.

Задачи в СММ

- Сегментация (Алгоритм Витерби, предыдущая лекция).

$$p(T|X,\Theta) \to \max_T$$

- Обучение с учителем (предыдущая лекция).

$$p(T,X|\Theta) \to \max_\Theta$$

- Обучение без учителя (EM-алгоритм).

$$p(X|\Theta) = \sum_T p(X,T|\Theta) \to \max_\Theta$$

  Подзадача: нахождение маргинального распределения $p(\boldsymbol{t}_n|X,\Theta)$.

- Прогнозирование.

$$p(\boldsymbol{x}_{N+1}|X) \to \max_{\boldsymbol{x}_{N+1}}$$

Спецификация вероятностной модели I

$0-1$ кодировка состояний: $\boldsymbol{t}_n = (t_{n1}, \dots, t_{nK})$, где

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

Пусть $A_{ij} = p(t_{nj} = 1 | t_{n-1,i} = 1)$, $\sum_j A_{ij} = 1$, $A \in \mathbb{R}_{K\times K}$. Тогда

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

Пусть $p(t_{1j} = 1) = \pi_j$, $\sum_j \pi_j = 1$. Тогда $p(\boldsymbol{t}_1) = \prod_{j=1}^{K} \pi_j^{t_{1j}}$
Пусть унарные вероятности заданы параметрически: $p(\boldsymbol{x}_n|\phi_j)$. Тогда

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

Спецификация вероятностной модели II

Полный набор параметров СММ $\Theta = \{\boldsymbol{\pi}, A, \boldsymbol{\phi}\}$.
Правдоподобие вычисляется как

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

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

$$\log p(X,T|\Theta) = \sum_{j=1}^{K} t_{1j}\log\pi_j + \sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} t_{n-1,i}t_{nj}\log A_{ij} +$$

$$\sum_{n=1}^{N}\sum_{j=1}^{K} t_{nj}\log p(\boldsymbol{x}_n|\boldsymbol{\phi}_j)$$

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Метод максимального правдоподобия

Для оценки параметров СММ Θ воспользуемся методом максимального правдоподобия:

$$\Theta_* = \arg\max_{\Theta} p(X|\Theta) = \arg\max_{\Theta} \sum_T p(X, T|\Theta)$$

Прямая максимизация правдоподобия затруднительна, т.к. оптимизируемая функция является многоэкстремальной, имеет большие области постоянных значений и, кроме того, для вычисления функции требуется суммирование K^N слагаемых. Можно воспользоваться итерационным EM-алгоритмом.

ЕМ-алгоритм в общем виде

$$p(X|\Theta) = \sum_T p(X, T|\Theta) \to \max_\Theta \Leftrightarrow \log\left(\sum_T p(X, T|\Theta)\right) \to \max_\Theta$$

- Е-шаг. Фиксируется значение параметров Θ_{old}. Оценивается апостериорное распределение на скрытые переменные $p(T|X, \Theta_{old})$, и полное правдоподобие усредняется по полученному распределению:

$$\mathbb{E}_{T|X,\Theta_{old}} \log p(X, T|\Theta) = \sum_T \log p(X, T|\Theta) p(T|X, \Theta_{old})$$

- М-шаг. Фиксируется апостериорное распределение $p(T|X, \Theta_{old})$, и производится поиск новых значений параметров Θ_{new}:

$$\Theta_{new} = \arg\max_\Theta \mathbb{E}_{T|X,\Theta_{old}} \log p(X, T|\Theta)$$

- Шаги Е и М повторяются до сходимости.

Е-шаг

Обозначим

$$\gamma(t_{nj}) = \mathbb{E}_{T|X,\Theta} t_{nj} = p(t_{nj} = 1|X, \Theta_{old}),$$
$$\xi(t_{n-1,i}, t_{nj}) = \mathbb{E}_{T|X,\Theta}(t_{n-1,i}t_{nj}) = p(t_{n-1,i} = 1, t_{nj} = 1|X, \Theta_{old}).$$

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

Тогда

$$\mathbb{E}_{T|X,\Theta_{old}} \log p(X, T|\Theta) = \sum_{j=1}^{K} \gamma(t_{1j}) \log \pi_j +$$
$$\sum_{n=2}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} \xi(t_{n-1,i}, t_{nj}) \log A_{ij} + \sum_{n=1}^{N}\sum_{j=1}^{K} \gamma(t_{nj}) \log p(\boldsymbol{x}_n|\phi_j)$$

M-шаг

$$\boldsymbol{\pi}: \qquad \sum_{j=1}^{K}\gamma(t_{1j})\log\pi_j + \lambda\left(\sum_{j=1}^{K}\pi_j - 1\right) \to \underset{\boldsymbol{\pi},\lambda}{\mathrm{extr}}$$

$$\frac{\gamma(t_{1j})}{\pi_j} + \lambda = 0 \Rightarrow \pi_j = -\frac{\gamma(t_{1j})}{\lambda}$$

$$\sum_{j=1}^{K}\pi_j = 1 \Rightarrow \lambda = -\sum_{i=1}^{K}\gamma(t_{1i})$$

$$\pi_j^{new} = \frac{\gamma(t_{1j})}{\sum_{i=1}^{K}\gamma(t_{1i})}$$

Действуя аналогично для A, принимая во внимание, что $\sum_{j=1}^{K}A_{ij} = 1 \ \forall\ i$, получаем:

$$A_{ij}^{new} = \frac{\sum_{n=2}^{N}\xi(t_{n-1,i}t_{nj})}{\sum_{k=1}^{K}\sum_{n=2}^{N}\xi(t_{n-1,i}t_{nk})}$$

М-шаг для компонент ϕ

М-шаг для компонент генерации данных $p(\boldsymbol{x}_n|\boldsymbol{\phi}_k)$ абсолютно аналогичен М-шагу для оценки параметров при восстановлении смесей распределений. В частности, если в качестве компонент выступают многомерные нормальные распределения

$$p(\boldsymbol{x}_n|\boldsymbol{\phi}_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \Sigma_k),$$

то задача оптимизации для параметров $\boldsymbol{\mu}_k, \Sigma_k$ может быть решена в явном виде:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(t_{nk})\boldsymbol{x}_n}{\sum_{n=1}^N \gamma(t_{nk})}$$

$$\Sigma_k = \frac{\sum_{n=1}^N \gamma(t_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N \gamma(t_{nk})}$$

Инициализация параметров Θ

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Для начала работы EM-алгоритма необходимо задать начальные значения параметров $\Theta = (\boldsymbol{\pi}, A, \boldsymbol{\phi})$. Заметим, что если какой-нибудь параметр из $(\boldsymbol{\pi}, A)$ инициализирован нулем, то в процессе итераций EM его значение не изменится.

- Значения параметров $\boldsymbol{\pi}$ и A обычно выбираются случайными при соблюдении ограничений $\sum_j \pi_j = 1$ и $\sum_j A_{ij} = 1\ \forall i$.
- Инициализация $\boldsymbol{\phi}$ зависит от формы распределений $p(\boldsymbol{x}|\boldsymbol{\phi})$. В случае нормальных распределений можно провести кластеризацию данных на K кластеров и выбрать в качестве $\boldsymbol{\mu}_k$ и Σ_k центр и разброс соответствующего кластера.

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Вычисление апостериорных распределений на скрытые компоненты

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

На Е-шаге алгоритма обучения СММ требуется вычисление апостериорных распределений на скрытые компоненты

$$\gamma(t_{nj}) = p(t_{nj} = 1|X, \Theta), \quad \xi(t_{n-1,i}, t_{nj}) = p(t_{n-1,i} = 1, t_{nj} = 1|X, \Theta)$$

Алгоритм «вперед-назад» (Баума-Уэлша) позволяет эффективно вычислять эти величины для всех n, i, j за линейное по N время.
В дальнейшем для удобства будем опускать Θ во всех формулах, считая набор параметров фиксированным.

Идея алгоритма Витерби (max-sum)

$$\max_T p(T|X,\Theta) = \max_T p(X,T|\Theta) = \max_T \log p(X,T|\Theta) =$$

$$= \max_T \left[\log p(\boldsymbol{t}_1) + \sum_{n=2}^N \log p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) + \sum_{n=1}^N \log p(\boldsymbol{x}_n|\boldsymbol{t}_n)\right] =$$

$$\left\{ \begin{array}{c} \max(a+b, a+c) = \\ a + \max(b,c) \end{array} \right\} = \max_{\boldsymbol{t}_1}\Bigg[\log p(\boldsymbol{x}_1|\boldsymbol{t}_1) + \log p(\boldsymbol{t}_1) +$$

$$+ \max_{T\backslash \boldsymbol{t}_1}\left[\log p(\boldsymbol{t}_2|\boldsymbol{t}_1) + \sum_{n=3}^N \log p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) + \sum_{n=2}^N \log p(\boldsymbol{x}_n|\boldsymbol{t}_n)\right]\Bigg] =$$

$$= \max_{\boldsymbol{t}_1}\Big[\log p(\boldsymbol{x}_1|\boldsymbol{t}_1) + \log p(\boldsymbol{t}_1) + \max_{\boldsymbol{t}_2}\big[\log p(\boldsymbol{t}_2|\boldsymbol{t}_1) + \log p(\boldsymbol{x}_2|\boldsymbol{t}_2) + \dots +$$

$$+ \max_{\boldsymbol{t}_N}[\log p(\boldsymbol{t}_N|\boldsymbol{t}_{N-1}) + \log p(\boldsymbol{x}_N|\boldsymbol{t}_N)]\big]\Big]$$

Идея алгоритма «вперед-назад» (sum-product) I

$$p(\boldsymbol{t}_i|X) = \frac{p(\boldsymbol{t}_i, X)}{p(X)} = \frac{1}{p(X)} \sum_{T\setminus \boldsymbol{t}_i} p(X, T)$$

$$\sum_{T\setminus \boldsymbol{t}_i} p(X, T) = \sum_{T\setminus \boldsymbol{t}_i} \left[p(\boldsymbol{t}_1) \prod_{n=2}^{N} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{t}_n) \right] =$$

$$= \{ab + ac = a(b + c)\} =$$

$$= \underbrace{p(\boldsymbol{x}_i|\boldsymbol{t}_i) \sum_{\boldsymbol{t}_1,\dots,\boldsymbol{t}_{i-1}} \left[p(\boldsymbol{t}_1) \prod_{n=2}^{i} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) \prod_{n=1}^{i-1} p(\boldsymbol{x}_n|\boldsymbol{t}_n) \right]}_{\boldsymbol{\alpha}(\boldsymbol{t}_i)} \times$$

$$\underbrace{\sum_{\boldsymbol{t}_{i+1},\dots,\boldsymbol{t}_N} \left[\prod_{n=i+1}^{N} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}) p(\boldsymbol{x}_n|\boldsymbol{t}_n) \right]}_{\boldsymbol{\beta}(\boldsymbol{t}_i)}$$

Идея алгоритма «вперед-назад» (sum-product) II

$$\boldsymbol{\alpha}(\boldsymbol{t}_i) = p(\boldsymbol{x}_i|\boldsymbol{t}_i)\sum_{\boldsymbol{t}_1,\dots,\boldsymbol{t}_{i-1}} p(\boldsymbol{t}_1)\prod_{n=2}^{i} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})\prod_{n=1}^{i-1} p(\boldsymbol{x}_n|\boldsymbol{t}_n) =$$

$$= \{ab + ac = a(b+c)\} =$$

$$p(\boldsymbol{x}_i|\boldsymbol{t}_i)\sum_{\boldsymbol{t}_{i-1}} p(\boldsymbol{t}_i|\boldsymbol{t}_{i-1})p(\boldsymbol{x}_{i-1}|\boldsymbol{t}_{i-1})\sum_{\boldsymbol{t}_1,\dots,\boldsymbol{t}_{i-2}} p(\boldsymbol{t}_1)\prod_{n=2}^{i-1} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})\prod_{n=1}^{i-2} p(\boldsymbol{x}_n|\boldsymbol{t}_n) =$$

$$= p(\boldsymbol{x}_i|\boldsymbol{t}_i)\sum_{\boldsymbol{t}_{i-1}} p(\boldsymbol{t}_i|\boldsymbol{t}_{i-1})p(\boldsymbol{x}_{i-1}|\boldsymbol{t}_{i-1})\sum_{\boldsymbol{t}_{i-2}} p(\boldsymbol{t}_{i-1}|\boldsymbol{t}_{i-2})p(\boldsymbol{x}_{i-2}|\boldsymbol{t}_{i-2}) \times \cdots$$

$$\times \sum_{\boldsymbol{t}_1} p(\boldsymbol{t}_2|\boldsymbol{t}_1)p(\boldsymbol{x}_1|\boldsymbol{t}_1)p(\boldsymbol{t}_1)$$

Итеративная формула пересчета:

$$\boldsymbol{\alpha}(\boldsymbol{t}_i) = p(\boldsymbol{x}_i|\boldsymbol{t}_i)\sum_{\boldsymbol{t}_{i-1}} p(\boldsymbol{t}_i|\boldsymbol{t}_{i-1})\boldsymbol{\alpha}(\boldsymbol{t}_{i-1})$$

Идея алгоритма «вперед-назад» (sum-product) III

$$\boldsymbol{\beta}(\boldsymbol{t}_i) = \sum_{\boldsymbol{t}_{i+1},\dots,\boldsymbol{t}_N} \prod_{n=i+1}^{N} p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})p(\boldsymbol{x}_n|\boldsymbol{t}_n) = \sum_{\boldsymbol{t}_{i+1}} p(\boldsymbol{t}_{i+1}|\boldsymbol{t}_i)p(\boldsymbol{x}_{i+1}|\boldsymbol{t}_{i+1}) \times \sum_{\boldsymbol{t}_{i+2}} p(\boldsymbol{t}_{i+2}|\boldsymbol{t}_{i+1})p(\boldsymbol{x}_{i+2}|\boldsymbol{t}_{i+2}) \cdots \sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_N|\boldsymbol{t}_{N-1})p(\boldsymbol{x}_N|\boldsymbol{t}_N)$$

Итеративная формула пересчета:

$$\boldsymbol{\beta}(\boldsymbol{t}_i) = \sum_{\boldsymbol{t}_{i+1}} \boldsymbol{\beta}(\boldsymbol{t}_{i+1})p(\boldsymbol{x}_{i+1}|\boldsymbol{t}_{i+1})p(\boldsymbol{t}_{i+1}|\boldsymbol{t}_i)$$

Понятие полукольца

Полукольцо — множество R с двумя бинарными операциями $\oplus$ и $\otimes$:

- $(R, \oplus)$ — коммутативный моноид с единицей:
  - $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
  - $0 \oplus a = a \oplus 0 = a$
  - $a \oplus b = b \oplus a$
- $(R, \otimes)$ — моноид с единичным элементом:
  - $a \otimes (b \otimes c) = (a \otimes b) \otimes c$
  - $1 \otimes a = a \otimes 1 = a$
- Дистрибутивность:
  - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
  - $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$
- $0 \otimes a = a \otimes 0 = 0$

Примеры: $(\mathbb{R}, +, \times)$, $(\mathbb{R}, \max, +)$, $(\mathbb{R}, \min, \times)$, $(\mathbb{R}, \max, \min)$, $(\mathbb{R}, \min, \max)$ и др.

Алгоритм $(\oplus - \otimes)$ для произвольного полукольца

Пусть все клики в графе размера 2. Тогда с помощью алгоритма $\oplus - \otimes$ можно эффективно вычислять выражения вида:

$$\bigoplus_{X \backslash \boldsymbol{x}_i} \bigotimes_{(\boldsymbol{x}_k, \boldsymbol{x}_l) \in \mathcal{E}} \psi(\boldsymbol{x}_k, \boldsymbol{x}_l)$$

Вычисление $\gamma(t_{nj})$

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

$$\gamma(\boldsymbol{t}_i) = p(\boldsymbol{t}_i|X) = \frac{p(X, \boldsymbol{t}_i)}{p(X)} = \frac{\boldsymbol{\alpha}(\boldsymbol{t}_i)\boldsymbol{\beta}(\boldsymbol{t}_i)}{p(X)}$$

Итерационный процесс:

$$\boldsymbol{\alpha}(\boldsymbol{t}_1) = p(\boldsymbol{t}_1)p(\boldsymbol{x}_1|\boldsymbol{t}_1)$$
$$\boldsymbol{\alpha}(\boldsymbol{t}_i) = p(\boldsymbol{x}_i|\boldsymbol{t}_i)\sum_{\boldsymbol{t}_{i-1}} p(\boldsymbol{t}_i|\boldsymbol{t}_{i-1})\boldsymbol{\alpha}(\boldsymbol{t}_{i-1})$$
$$\boldsymbol{\beta}(\boldsymbol{t}_N) = 1$$
$$\boldsymbol{\beta}(\boldsymbol{t}_i) = \sum_{\boldsymbol{t}_{i+1}} \boldsymbol{\beta}(\boldsymbol{t}_{i+1})p(\boldsymbol{x}_{i+1}|\boldsymbol{t}_{i+1})p(\boldsymbol{t}_{i+1}|\boldsymbol{t}_i)$$

На каждом шаге рекурсии вектор $\boldsymbol{\alpha}(\boldsymbol{t}_i)$ длины K умножается на матрицу $p(\boldsymbol{t}_i|\boldsymbol{t}_{i-1})$ размера $K \times K$. Поэтому сложность всего рекуррентного процесса составляет $O(NK^2)$.

Формула для $\xi(\boldsymbol{t}_{i-1}, \boldsymbol{t}_i)$

$$\xi(\boldsymbol{t}_{i-1}, \boldsymbol{t}_i) = \frac{p(X, \boldsymbol{t}_{i-1}, \boldsymbol{t}_i)}{p(X)}$$

$$p(X, \boldsymbol{t}_{i-1}, \boldsymbol{t}_i) = \sum_{T \backslash \{\boldsymbol{t}_{i-1}, \boldsymbol{t}_i\}} p(\boldsymbol{t}_1) \prod_{n=2}^{N} p(\boldsymbol{t}_n | \boldsymbol{t}_{n-1}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{t}_n) =$$

$$= p(\boldsymbol{t}_i | \boldsymbol{t}_{i-1}) p(\boldsymbol{x}_i | \boldsymbol{t}_i) \underbrace{p(\boldsymbol{x}_{i-1} | \boldsymbol{t}_{i-1}) \sum_{\boldsymbol{t}_1, \dots, \boldsymbol{t}_{i-2}} p(\boldsymbol{t}_1) \prod_{n=2}^{i-1} p(\boldsymbol{t}_n | \boldsymbol{t}_{n-1}) \prod_{n=1}^{i-2} p(\boldsymbol{x}_n | \boldsymbol{t}_n)}_{\boldsymbol{\alpha}(\boldsymbol{t}_{i-1})} \times$$

$$\underbrace{\sum_{\boldsymbol{t}_{i+1}, \dots, \boldsymbol{t}_N} \prod_{n=i+1}^{N} p(\boldsymbol{t}_n | \boldsymbol{t}_{n-1}) p(\boldsymbol{x}_n | \boldsymbol{t}_n)}_{\boldsymbol{\beta}(\boldsymbol{t}_i)} =$$

$$= p(\boldsymbol{t}_i | \boldsymbol{t}_{i-1}) p(\boldsymbol{x}_i | \boldsymbol{t}_i) \boldsymbol{\alpha}(\boldsymbol{t}_{i-1}) \boldsymbol{\beta}(\boldsymbol{t}_i)$$

Нормировочная константа $p(X)$

Значения $\boldsymbol{\gamma}(\boldsymbol{t}_n)$ и $\xi(\boldsymbol{t}_{n-1}, \boldsymbol{t}_n)$ определены с точностью до нормировочной константы $p(X)$. Однако, на М-шаге эти значения, как правило, входят и в числитель, и в знаменатель. Таким образом, нормировочная константа сокращается. Например, при вычислении центра нормального распределения:

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(t_{nk})\boldsymbol{x}_n}{\sum_{n=1}^N \gamma(t_{nk})} = \frac{\sum_{n=1}^N \alpha(t_{nk})\beta(t_{nk})\boldsymbol{x}_k}{\sum_{n=1}^N \alpha(t_{nk})\beta(t_{nk})}$$

Тем не менее, сама нормировочная константа $p(X)$ — это значение правдоподобия, которое может представлять отдельный интерес (например, можно отслеживать возрастание правдоподобия при итерациях EM-алгоритма). Зная $\boldsymbol{\alpha}(\boldsymbol{t}_n)$ и $\boldsymbol{\beta}(\boldsymbol{t}_n)$, правдоподобие может быть легко вычислено как

$$p(X) = \sum_{\boldsymbol{t}_n} \boldsymbol{\alpha}(\boldsymbol{t}_n)\boldsymbol{\beta}(\boldsymbol{t}_n), \ \forall n \ \Rightarrow \ p(X) = \sum_{\boldsymbol{t}_N} \boldsymbol{\alpha}(\boldsymbol{t}_N)$$

Итоговый ЕМ-алгоритм для случая, когда $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ — нормальные распределения

- Начальная инициализация параметров $\Theta = (\boldsymbol{\pi}, A, \boldsymbol{\phi})$. Параметры $\boldsymbol{\pi}$ и A можно инициализировать случайно, а $\boldsymbol{\phi}$ с помощью кластеризации данных.
- Е-шаг. Вычисление $\boldsymbol{\alpha}(\boldsymbol{t}_n)$ и $\boldsymbol{\beta}(\boldsymbol{t}_n)$ с помощью рекуррентного алгоритма «вперед-назад». Вычисление величин $\gamma(t_{nj})$ и $\xi(t_{n-1,i}, t_{nj})$ и, возможно, правдоподобия $p(X)$.
- М-шаг. Вычисление новых значений параметров:

$$\pi_j^{new} = \frac{\gamma(t_{1j})}{\sum_{i=1}^K \gamma(t_{1i})}, \qquad A_{ij}^{new} = \frac{\sum_{n=2}^N \xi(t_{n-1,i} t_{nj})}{\sum_{k=1}^K \sum_{n=2}^N \xi(t_{n-1,i} t_{nk})}$$

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{n=1}^N \gamma(t_{nk})\boldsymbol{x}_n}{\sum_{n=1}^N \gamma(t_{nk})}, \quad \Sigma_k^{new} = \frac{\sum_{n=1}^N \gamma(t_{nk})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N \gamma(t_{nk})}$$

- Повторять шаги Е и М до сходимости (пока значение $p(X)$ или Θ не стабилизируется).

Задача прогнозирования

$$p(\boldsymbol{x}_{N+1}|X) = \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}, \boldsymbol{t}_{N+1}|X) = \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1}) p(\boldsymbol{t}_{N+1}|X) =$$

$$= \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1}) \left(\sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_{N+1}, \boldsymbol{t}_N|X) \right) =$$

$$= \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1}) \left(\sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_N|X) p(\boldsymbol{t}_{N+1}|\boldsymbol{t}_N) \right) =$$

$$= \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1}) \left(\sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_{N+1}|\boldsymbol{t}_N) \frac{p(\boldsymbol{t}_N, X)}{p(X)} \right) =$$

$$= \frac{1}{p(X)} \sum_{\boldsymbol{t}_{N+1}} p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1}) \left(\sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_{N+1}|\boldsymbol{t}_N) \boldsymbol{\alpha}(\boldsymbol{t}_N) \right)$$

Это фактически смесь распределений с компонентами $p(\boldsymbol{x}_{N+1}|\boldsymbol{t}_{N+1})$ и весами $w_k = \frac{1}{p(X)} \sum_{\boldsymbol{t}_N} p(\boldsymbol{t}_{N+1}|\boldsymbol{t}_N) \boldsymbol{\alpha}(\boldsymbol{t}_N)$. Для получения точечного прогноза $\boldsymbol{x}^*_{N+1}$ можно воспользоваться MCMC.

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Необходимость устойчивых вычислений для $\boldsymbol{\alpha}(\boldsymbol{t}_n)$ и $\boldsymbol{\beta}(\boldsymbol{t}_n)$

Формулы пересчета для $\boldsymbol{\alpha}(\boldsymbol{t}_n)$ и $\boldsymbol{\beta}(\boldsymbol{t}_n)$:

$$\boldsymbol{\alpha}(\boldsymbol{t}_n) = p(\boldsymbol{x}_n|\boldsymbol{t}_n)\sum_{\boldsymbol{t}_{n-1}}\boldsymbol{\alpha}(\boldsymbol{t}_{n-1})p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$$

$$\boldsymbol{\beta}(\boldsymbol{t}_n) = \sum_{\boldsymbol{t}_{n+1}}\boldsymbol{\beta}(\boldsymbol{t}_{n+1})p(\boldsymbol{t}_{n+1}|\boldsymbol{t}_n)p(\boldsymbol{x}_{n+1}|\boldsymbol{t}_{n+1})$$

На практике значения вероятностей $p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$ и $p(\boldsymbol{x}_n|\boldsymbol{t}_n)$ могут быть существенно меньше единицы. В процессе пересчета эти вероятности умножаются друг на друга, и получающиеся значения перестают укладываться в машинную точность.

Выбор устойчивых переменных

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

$$p(\boldsymbol{t}_n|X) = \frac{\boldsymbol{\alpha}(\boldsymbol{t}_n)\boldsymbol{\beta}(\boldsymbol{t}_n)}{p(X)} = \frac{\boldsymbol{\alpha}(\boldsymbol{t}_n)}{p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)}\frac{\boldsymbol{\beta}(\boldsymbol{t}_n)}{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)}$$

Рассмотрим вместо $\boldsymbol{\alpha}(\boldsymbol{t}_n)$ следующую величину:

$$\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n) = \frac{\boldsymbol{\alpha}(\boldsymbol{t}_n)}{p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n,\boldsymbol{t}_n)}{p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} = p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)$$

Вычисление $\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n)$ будет устойчивым, т.к. $\sum_{\boldsymbol{t}_n}\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n) = 1$. Вместо $\boldsymbol{\beta}(\boldsymbol{t}_n)$ рассмотрим величину:

$$\hat{\boldsymbol{\beta}}(\boldsymbol{t}_n) = \frac{\boldsymbol{\beta}(\boldsymbol{t}_n)}{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{t}_n)}{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} =$$

$$\frac{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{t}_n)}{\sum_{\boldsymbol{t}_n} p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N,\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)} = \frac{p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{t}_n)}{\sum_{\boldsymbol{t}_n} p(\boldsymbol{x}_{n+1},\dots,\boldsymbol{x}_N|\boldsymbol{t}_n)p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots}$$

В знаменателе стоит взвешенная сумма выражений в числителе, причем сумма весов равна единице.

Устойчивые формулы для $\boldsymbol{\alpha}(\boldsymbol{t}_n)$

Введем следующее обозначение:

$$c_i = p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\dots,\boldsymbol{x}_1)$$

Тогда

$$p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \prod_{i=1}^n p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\dots,\boldsymbol{x}_1) = \prod_{i=1}^n c_i$$

$$\boldsymbol{\alpha}(\boldsymbol{t}_n) = p(\boldsymbol{t}_n|\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)p(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n) = \hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n)\left(\prod_{i=1}^n c_i\right)$$

Подставляя это выражение в формулу пересчета для $\boldsymbol{\alpha}(\boldsymbol{t}_n)$

$$\boldsymbol{\alpha}(\boldsymbol{t}_n) = p(\boldsymbol{x}_n|\boldsymbol{t}_n)\sum_{\boldsymbol{t}_{n-1}}\boldsymbol{\alpha}(\boldsymbol{t}_{n-1})p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1}),$$

получаем

$$c_n\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n) = p(\boldsymbol{x}_n|\boldsymbol{t}_n)\sum_{\boldsymbol{t}_{n-1}}\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_{n-1})p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})$$

Значение c_n определяется из условия нормировки для $\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n)$.

Устойчивые формулы для $\boldsymbol{\beta}(\boldsymbol{t}_n)$

$$\hat{\boldsymbol{\beta}}(\boldsymbol{t}_n) = \frac{p(\boldsymbol{x}_{n+1}, \dots, \boldsymbol{x}_N | \boldsymbol{t}_n)}{p(\boldsymbol{x}_{n+1}, \dots, \boldsymbol{x}_N | \boldsymbol{x}_1, \dots, \boldsymbol{x}_n)} = \frac{\boldsymbol{\beta}(\boldsymbol{t}_n)}{\prod_{i=n+1}^N c_i}$$

Подставляя выражение для $\hat{\boldsymbol{\beta}}(\boldsymbol{t}_n)$ в формулу пересчета

$$\boldsymbol{\beta}(\boldsymbol{t}_n) = \sum_{\boldsymbol{t}_{n+1}} \boldsymbol{\beta}(\boldsymbol{t}_{n+1}) p(\boldsymbol{t}_{n+1} | \boldsymbol{t}_n) p(\boldsymbol{x}_{n+1} | \boldsymbol{t}_{n+1}),$$

получаем

$$c_{n+1} \hat{\boldsymbol{\beta}}(\boldsymbol{t}_n) = \sum_{\boldsymbol{t}_{n+1}} \hat{\boldsymbol{\beta}}(\boldsymbol{t}_{n+1}) p(\boldsymbol{t}_{n+1} | \boldsymbol{t}_n) p(\boldsymbol{x}_{n+1} | \boldsymbol{t}_{n+1})$$

Значения c_n определяются из формул пересчета для $\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n)$.

Окончательные выражения для E-шага

Правдоподобие $p(X)$ вычисляется как

$$p(X) = \prod_{n=1}^{N} c_n$$

Другие необходимые величины вычисляются как

$$\begin{aligned}
\gamma(\boldsymbol{t}_n) &= \hat{\boldsymbol{\alpha}}(\boldsymbol{t}_n)\hat{\boldsymbol{\beta}}(\boldsymbol{t}_n) \\
\xi(\boldsymbol{t}_{n-1}, \boldsymbol{t}_n) &= \frac{1}{c_n}\hat{\boldsymbol{\alpha}}(\boldsymbol{t}_{n-1})p(\boldsymbol{x}_n|\boldsymbol{t}_n)p(\boldsymbol{t}_n|\boldsymbol{t}_{n-1})\hat{\boldsymbol{\beta}}(\boldsymbol{t}_n)
\end{aligned}$$

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

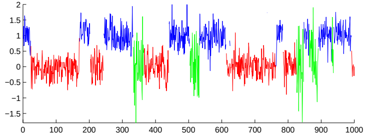
Модельный пример I

В качестве иллюстрации работы ЕМ-алгоритма обучения СММ рассмотрим простой модельный пример. $X \in \mathbb{R}$, $K = 3$, данные сгенерированы из нормальных распределений со следующими параметрами:

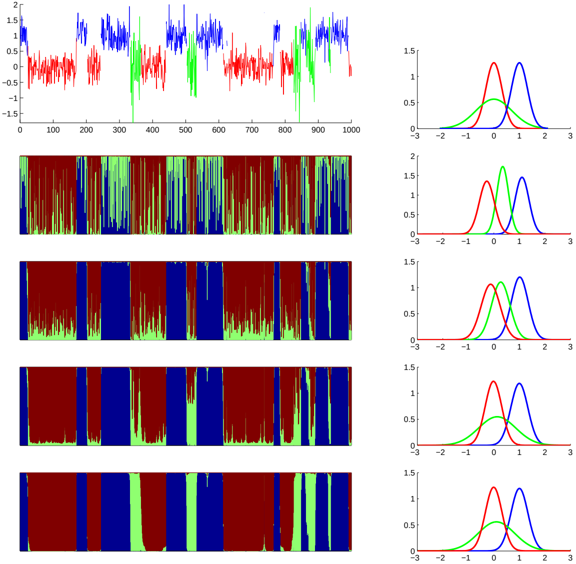
$$\mu_1 = 0, \quad \mu_2 = 0, \quad \mu_3 = 1$$
$$\sigma_1^2 = 0.1, \; \sigma_2^2 = 0.5, \; \sigma_3^2 = 0.1$$

Априорные вероятности и матрица перехода выбраны следующим образом:

$$\boldsymbol{\pi} = [0.3, \; 0.2, \; 0.5], \qquad A = \begin{bmatrix} 0.98 & 0.01 & 0.01 \\ 0.01 & 0.97 & 0.02 \\ 0.01 & 0.01 & 0.98 \end{bmatrix}$$



Модельный пример II



Ит. 1:

Ит. 5:

Ит. 20:

Ит. 54:

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

Модельный пример III

После 54-ой итерации EM-алгоритма значения параметров были следующие:

$$\boldsymbol{\pi} = [10^{-190},\ 10^{-125},\ 1], \qquad A = \begin{bmatrix} 0.984 & 0.004 & 0.012 \\ 0.013 & 0.949 & 0.038 \\ 0.011 & 0.011 & 0.978 \end{bmatrix}$$

$$\mu_1 = -0.01,\ \sigma_1^2 = 0.11$$
$$\mu_2 = 0.1,\ \sigma_2^2 = 0.51$$
$$\mu_3 = 1,\ \sigma_3^2 = 0.11$$

План лекции

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

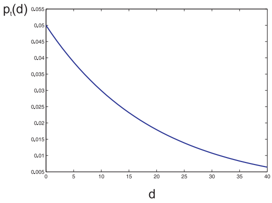
Модификации стандартной СММ

Прямое моделирование длин сегментов

В классической СММ априорная вероятность нахождения в состоянии t ровно d моментов времени составляет

$$p_t(d) = (A_{tt})^d (1 - A_{tt})$$

Здесь A_{tt} — вероятность остаться в состоянии t в каждый момент времени.



Такое распределение может быть неадекватно практической ситуации.

Прямое моделирование длин сегментов

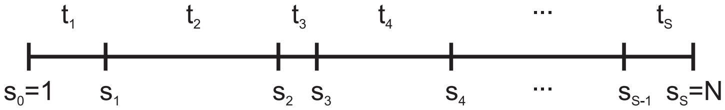
Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ



Пусть наблюдаемая последовательность разбита на S сегментов, на интервале $[s_{i-1}+1, s_i]$ система находится в состоянии t_i. Правдоподобие такой конфигурации:

$$p(X,T,S|\Theta) =$$

$$= p(t_1)p_{t_1}(s_1)\prod_{n=1}^{s_1} p(\boldsymbol{x}_n|t_1)\prod_{i=2}^{S} p(t_i|t_{i-1})p_{t_i}(s_i-s_{i-1})\prod_{n=s_{i-1}+1}^{s_i} p(\boldsymbol{x}_n|t_i)$$

Алгоритм «вперед-назад»

Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ

$\alpha_n(t) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \text{сост. } t \text{ заканчивается в момент времени } n)$

$\beta_n(t) = p(\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_N | \text{сост. } t \text{ заканчивается в момент времени } n)$

$\alpha_n^*(t) = p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \text{сост. } t \text{ начинается в момент времени } n+1)$

$\beta_n^*(t) = p(\boldsymbol{x}_{n+1}, \ldots, \boldsymbol{x}_N | \text{сост. } t \text{ начинается в момент времени } n+1)$

$$\alpha_n(t) = \sum_{d=1}^{D} \sum_{k \neq t} \alpha_{n-d}(k) p(t|k) p_t(d) \prod_{j=n-d+1}^{n} p(\boldsymbol{x}_j|t)$$

$$\alpha_n^*(t) = \sum_{k \neq t} \alpha_n(k) p(t|k)$$

$$\beta_n(t) = \sum_{d=1}^{D} \sum_{k \neq t} \beta_{n+d}(k) p(k|t) p_k(d) \prod_{j=n+1}^{n+d} p(\boldsymbol{x}_j|k)$$

$$\beta_n^*(t) = \sum_{d=1}^{D} \beta_{n+d}(t) p_t(d) \prod_{j=n+1}^{n+d} p(\boldsymbol{x}_j|t)$$

D — максимально допустимая длина сегмента

Авторегрессионная СММ (АР СММ)

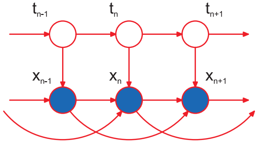
Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ



Классическая СММ плохо моделирует зависимости между переменными на больших временных интервалах. В авторегрессионной СММ предполагается дополнительная зависимость $\boldsymbol{x}_n$ от некоторого количества наблюдений в прошлом $\boldsymbol{x}_{n-1}, \ldots, \boldsymbol{x}_{n-M}$.

В случае дискретных распределений на $\boldsymbol{x}_n$ в АР СММ просто увеличивается таблица распределения с учетом всех возможных значений $\boldsymbol{x}_{n-1}, \ldots, \boldsymbol{x}_{n-M}$. Для нормальных распределений можно использовать линейную модель зависимости:

$$p(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}, \ldots, \boldsymbol{x}_{n-M}, \boldsymbol{t}_n) = \mathcal{N}\left(\boldsymbol{x}_n \,\Big|\, \sum_{j=1}^{M} \alpha_j \boldsymbol{x}_{n-j} + \boldsymbol{\mu}(\boldsymbol{t}_n), \Sigma\right)$$

Метод максимального правдоподобия для СММ

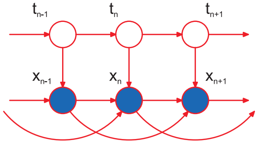
Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

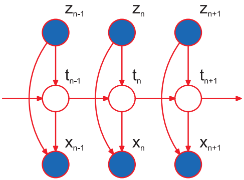
Модификации стандартной СММ

АР СММ



Переменные $\boldsymbol{t}_{n+1}$ и $\boldsymbol{t}_{n-1}$ являются условно независимыми от $\boldsymbol{t}_n$. Поэтому, несмотря на усложнение графической модели, алгоритм «вперед-назад» по-прежнему будет применим. При этом потребуются минимальные изменения на М-шаге ЕМ-алгоритма обучения.

СММ вход-выход (Bengio & Frasconi, 1995)

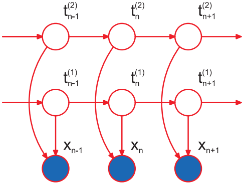


Динамическая система переводит входы $\boldsymbol{z}_n$ в выходы $\boldsymbol{x}_n$:

$$t_n = f(t_{n-1}, \boldsymbol{z}_n)$$
$$\boldsymbol{x}_n = g(\boldsymbol{z}_n, t_n)$$

СММ перестает быть гомогенной, т.к. вероятности переходов из t_{n-1} в t_n зависят от входов $\boldsymbol{z}_n$.
Значения t_{n-1} и t_{n+1} являются условно независимыми от t_n.
Следовательно, алгоритм «вперед-назад» является применимым.

Факториальная СММ (Ghahramani & Jordan, 1997)



Здесь присутствует несколько марковских цепочек скрытых параметров.

Можно свести к классической СММ с числом состояний K^M, где M — количество цепочек, а K — число состояний в каждой цепочке. Тогда вычислительная сложность анализа такой модели составляет $O(NK^{2M})$, и становится недоступной уже при малых значениях M.

Факториальная СММ (Ghahramani & Jordan, 1997)

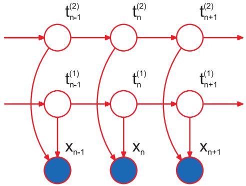
Метод максимального правдоподобия для СММ

Алгоритм «вперед-назад»

Устойчивые формулы для алгоритма «вперед-назад»

Модельный пример

Модификации стандартной СММ



М-шаг для факториальной СММ выполняется напрямую. Однако, на Е-шаге мы не можем применять алгоритм «вперед-назад» для каждой марковской цепочки скрытых состояний независимо. Например, значения $t_n^{(1)}$ и $t_n^{(2)}$ не являются независимыми, т.к. наблюдаемое значение $\boldsymbol{x}_n$ косвенно отражается на значениях обеих величин. В (Ghahramani & Jordan, 1997) предложен приближенный алгоритм для Е-шага, основанный на вариационном подходе.