

# Скрытые марковские модели

Сергей Николенко

Распознавание речи — ИТМО, осень 2008

# Outline

- 1 Марковские процессы
  - Марковские цепи и процессы
  - Задачи, которые нужно решать
- 2 Определения и обозначения
  - Обозначения в скрытых марковских моделях
  - Задачи формально
- 3 Решения задач
  - Первая задача
  - Вторая задача
  - Третья задача
  - Обоснование алгоритма Баума-Велха

## Марковские цепи

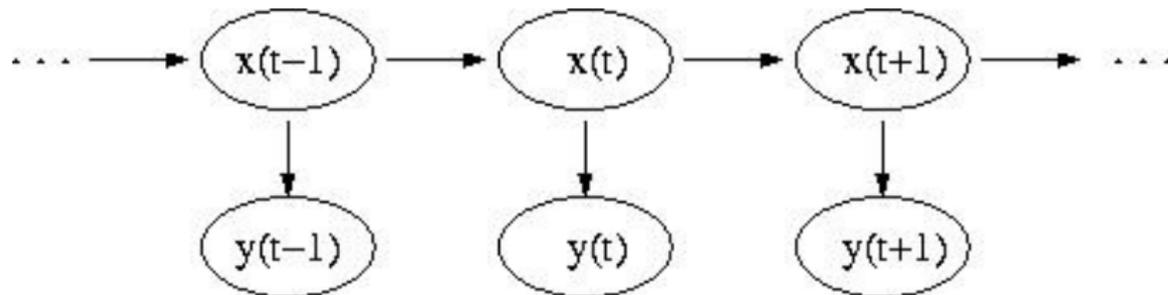
- Марковская цепь задаётся начальным распределением вероятностей  $p^0(x)$  и вероятностями перехода  $T(x'; x)$ .
- $T(x'; x)$  — это распределение следующего элемента цепи в зависимости от следующего; распределение на  $(t + 1)$ -м шаге равно

$$p^{t+1}(x') = \int T(x'; x)p^t(x)dx.$$

- В дискретном случае  $T(x'; x)$  — это матрица вероятностей  $p(x' = i | x = j)$ .

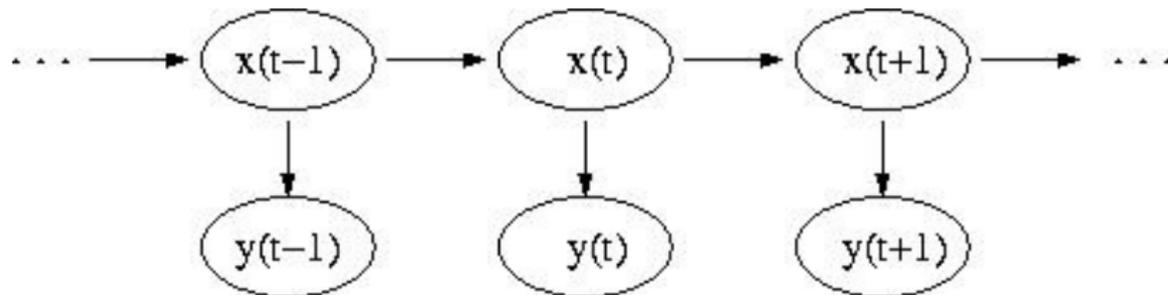
## Дискретные марковские цепи

- Мы будем находиться в дискретном случае.
- Марковская модель — это когда мы можем наблюдать какие-то функции от марковского процесса.



## Дискретные марковские цепи

- Здесь  $x(t)$  — сам процесс (модель), а  $y(t)$  — то, что мы наблюдаем.
- Задача — определить скрытые параметры процесса.



## Дискретные марковские цепи

- Главное свойство — следующее состояние не зависит от истории, только от предыдущего состояния.

$$\begin{aligned} p(x(t) = x_j | x(t-1) = x_{j_{t-1}}, \dots, x(1) = x_{j_1}) = \\ = p(x(t) = x_j | x(t-1) = x_{j_{t-1}}). \end{aligned}$$

- Более того, эти вероятности  $a_{ij} = p(x(t) = x_j | x(t-1) = x_i)$  ещё и от времени  $t$  не зависят.
- Эти вероятности и составляют матрицу перехода  $A = (a_{ij})$ .

## Вероятности перехода

- Естественные свойства:
- $a_{ij} \geq 0$ .
- $\sum_j a_{ij} = 1$ .

## Прямая задача

- Естественная задача: с какой вероятностью выпадет та или иная последовательность событий?
- Т.е. найти нужно для последовательности  $Q = q_{i_1} \dots q_{i_k}$

$$p(Q|\text{модель}) = p(q_{i_1})p(q_{i_2}|q_{i_1}) \dots p(q_{i_k}|q_{i_{k-1}}).$$

- Кажалось бы, это тривиально.
- Что же сложного в реальных задачах?

## Скрытые марковские модели

- А сложно то, что никто нам не скажет, что модель должна быть именно такой.
- И, кроме того, мы обычно наблюдаем не  $x(t)$ , т.е. реальные состояния модели, а  $y(t)$ , т.е. некоторую функцию от них (данные).
- Давайте приведём пример.

## Скрытые марковские модели: пример

- Пусть кто-то бросает монетку и сообщает нам результаты — последовательность орлов и решек.
- Но мы не знаем, что он бросает монетку, мы только знаем, что есть вот такая последовательность битов.
- Если мы предположим, что он бросает одну монетку, модель будет одна: два состояния, вероятности перехода между ними  $p$  и  $1 - p$ , вероятности остаться  $1 - p$  и  $p$ .

## Скрытые марковские модели: пример

- Но мы же можем подумать, что у него две монетки! :)
- Тогда состояния по-прежнему два, но параметров больше.
- А если три монетки?..
- В общем, задачи уже ясны.

## Задачи скрытых марковских моделей

- Первая: найти вероятность последовательности наблюдений в данной модели.
- Вторая: найти «оптимальную» последовательность состояний при условии данной модели и данной последовательности наблюдений.
- Третья: найти наиболее правдоподобную модель (параметры модели).

# Outline

- 1 Марковские процессы
  - Марковские цепи и процессы
  - Задачи, которые нужно решать
- 2 Определения и обозначения
  - Обозначения в скрытых марковских моделях
  - Задачи формально
- 3 Решения задач
  - Первая задача
  - Вторая задача
  - Третья задача
  - Обоснование алгоритма Баума-Велха

## Состояния и наблюдаемые

- $X = \{x_1, \dots, x_n\}$  — множество состояний.
- $V = \{v_1, \dots, v_m\}$  — алфавит, из которого мы выбираем наблюдаемые  $y$  (множество значений  $y$ ).
- $q_t$  — состояние во время  $t$ ,  $y_t$  — наблюдаемая во время  $t$ .

## Распределения

- $a_{ij} = p(q_{t+1} = x_j | q_t = x_i)$  — вероятность перехода из  $i$  в  $j$ .
- $b_j(k) = p(v_k | x_j)$  — вероятность получить данные  $v_k$  в состоянии  $j$ .
- Начальное распределение  $\pi = \{\pi_j\}$ ,  $\pi_j = p(q_1 = x_j)$ .
- Данные будем обозначать через  $D = d_1 \dots d_T$  (последовательность наблюдаемых,  $d_i$  принимают значения из  $V$ ).

## Комментарий

- Проще говоря, вот как работает HMM (hidden Markov model).
- Выберем начальное состояние  $x_1$  по распределению  $\pi$ .
- По  $t$  от 1 до  $T$ :
  - Выберем наблюдаемую  $d_t$  по распределению  $p(v_k|x_j)$ .
  - Выберем следующее состояние по распределению  $p(q_{t+1} = x_j|q_t = x_i)$ .
- Таким алгоритмом можно выбрать случайную последовательность наблюдаемых.

## Задачи

- Теперь можно формализовать постановку задач.
- Первая задача: по данной модели  $\lambda = (A, B, \pi)$  и последовательности  $D$  найти  $p(D|\lambda)$ . Фактически, это нужно для того, чтобы оценить, насколько хорошо модель подходит к данным.
- Вторая задача: по данной модели  $\lambda$  и последовательности  $D$  найти «оптимальную» последовательность состояний  $Q = q_1 \dots q_T$ . Как и раньше, будет два решения: «побитовое» и общее.
- Третья задача: оптимизировать параметры модели  $\lambda = (A, B, \pi)$  так, чтобы максимизировать  $p(D|\lambda)$  при данном  $D$  (найти модель максимального правдоподобия). Эта задача — главная, в ней и заключается обучение скрытых марковских моделей.

# Outline

- 1 Марковские процессы
  - Марковские цепи и процессы
  - Задачи, которые нужно решать
- 2 Определения и обозначения
  - Обозначения в скрытых марковских моделях
  - Задачи формально
- 3 Решения задач
  - Первая задача
  - Вторая задача
  - Третья задача
  - Обоснование алгоритма Баума-Велха

## Постановка

- Формально, первая задача выглядит так. Нужно найти

$$\begin{aligned} p(D|\lambda) &= \sum_Q p(D|Q, \lambda) p(Q|\lambda) = \\ &= \sum_{q_1, \dots, q_T} b_{q_1}(d_1) \dots b_{q_T}(d_T) \pi_{q_1} a_{q_1 q_2} \dots a_{q_{T-1} q_T}. \end{aligned}$$

- Ничего не напоминает?

## Суть решения

- Правильно, это такая же задача маргинализации, как мы решаем всё время.
- Мы воспользуемся так называемой forward–backward procedure, по сути — вычислением на решётке, как в декодировании.
- Будем последовательно вычислять промежуточные величины вида

$$\alpha_t(i) = p(d_1 \dots d_t, q_t = x_i | \lambda),$$

т.е. искомые вероятности, но ещё с учётом текущего состояния.

## Решение

- Инициализируем  $\alpha_1(i) = \pi_i b_i(d_1)$ .
- Шаг индукции:

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^n \alpha_t(i) a_{ij} \right] b_j(d_{t+1}).$$

- После того как дойдём до шага  $T$ , подсчитаем то, что нам нужно:

$$p(D|\lambda) = \sum_{i=1}^n \alpha_T(i).$$

- Фактически, это только прямой проход, обратный нам здесь не понадобился.
- Что вычислял бы обратный проход?

## Обратный проход

- Он вычислял бы условные вероятности  $\beta_t(i) = p(d_{t+1} \dots d_T | q_t = x_i, \lambda)$ .
- Их можно вычислить, проинициализировав  $\beta_T(i) = 1$ , а затем по индукции:

$$\beta_t(i) = \sum_{j=1}^n a_{ij} b_j(d_{t+1}) \beta_{t+1}(j).$$

- Это нам пригодится чуть позже, при решении второй и третьей задачи.

## Два варианта

- Как мы уже упоминали, возможны два варианта.
- Первый: решать «побитово», отвечая на вопрос «какое наиболее вероятное состояние во время  $j$ ?».
- Второй: решать задачу «какая наиболее вероятная последовательность состояний?».

## Побитовое решение

- Рассмотрим вспомогательные переменные

$$\gamma_t(i) = p(q_t = x_i | D, \lambda).$$

- Наша задача – найти

$$q_t = \operatorname{argmax}_{1 \leq i \leq n} \gamma_t(i), \quad 1 \leq t \leq T.$$

- Как это сделать?

## Побитовое решение

- Выражаем через  $\alpha$  и  $\beta$ :

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{p(D|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^n \alpha_t(i)\beta_t(i)}.$$

- На знаменатель можно не обращать внимания — нам нужен  $\operatorname{argmax}$ .

## Решение относительно последовательности

- Чтобы ответить на вопрос о наиболее вероятной последовательности, мы будем использовать уже знакомый алгоритм Витерби.
- То есть, по сути, то же самое динамическое программирование.
- Наши вспомогательные переменные — это

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} p(q_1 q_2 \dots q_t = x_i, d_1 d_2 \dots d_t | \lambda).$$

## Решение относительно последовательности

- Т.е.  $\delta_t(i)$  — максимальная вероятность достичь состояния  $x_i$  на шаге  $t$  среди всех путей с заданными наблюдаемыми.
- По индукции:

$$\delta_{t+1}(j) = \left[ \max_i \delta_t(i) a_{ij} \right] b_j(d_{t+1}).$$

- И надо ещё запоминать аргументы, а не только значения; для этого будет массив  $\psi_t(j)$ .

## Решение относительно последовательности: алгоритм

- Проинициализируем  $\delta_1(i) = \pi_i b_i(d_1)$ ,  $\psi_1(i) = \square$ .
- Индукция:

$$\delta_t(j) = \max_{1 \leq i \leq n} [\delta_{t-1}(i) a_{ij}] b_j(d_t),$$

$$\psi_t(j) = \operatorname{argmax}_{1 \leq i \leq n} [\delta_{t-1}(i) a_{ij}].$$

- Когда дойдём до шага  $T$ , финальный шаг:

$$p^* = \max_{1 \leq i \leq n} \delta_T(i), \quad q_T^* = \operatorname{argmax}_{1 \leq i \leq n} \delta_T(i).$$

- И вычислим последовательность:  $q_t^* = \psi_{t+1}(q_{t+1}^*)$ .

## Общая суть

- Аналитически найти глобальный максимум  $p(D|\lambda)$  у нас никак не получится.
- Зато мы рассмотрим итеративную процедуру (по сути — градиентный подъём), которая приведёт к локальному максимуму.
- Это называется алгоритм Баума–Велха (Baum–Welch algorithm). Он является на самом деле частным случаем алгоритма EM.

## Вспомогательные переменные

- Теперь нашими вспомогательными переменными будут вероятности того, что мы во время  $t$  в состоянии  $x_i$ , а во время  $t + 1$  — в состоянии  $x_j$ :

$$\xi_t(i, j) = p(q_t = x_i, q_{t+1} = x_j | D, \lambda).$$

- Если переписать через уже знакомые переменные:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}{p(D | \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_t(i) a_{ij} b_j(d_{t+1}) \beta_{t+1}(j)}.$$

- Отметим также, что  $\gamma_t(i) = \sum_j \xi_t(i, j)$ .

## Идея

- $\sum_t \gamma_t(i)$  — это ожидаемое количество переходов из состояния  $x_i$ , а  $\sum_t \xi_t(i, j)$  — из  $x_i$  в  $x_j$ .
- Теперь на шаге  $M$  мы будем переоценивать вероятности:

$\bar{\pi}_i =$  ожидаемая частота в  $x_i$  на шаге  $1 = \gamma_1(i)$ ,

$$\bar{a}_{ij} = \frac{\text{к-во переходов из } x_i \text{ в } x_j}{\text{к-во переходов из } x_i} = \frac{\sum_t \xi_t(i, j)}{\sum_t \gamma_t(i)}.$$

$$\bar{b}_j(k) = \frac{\text{к-во появлений в } x_i \text{ и наблюдений } v_k}{\text{к-во появлений в } x_i} = \frac{\sum_{t: d_t=v_k} \gamma_t(i)}{\sum_t \gamma_t(i)}.$$

- EM-алгоритм приведёт к цели: начать с  $\lambda = (A, B, \pi)$ , подсчитать  $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ , снова пересчитать параметры и т.д.

## Расстояние Кульбака–Лейблера

- Kullback–Leibler distance (divergence) — это информационно-теоретическая мера того, насколько далеки распределения друг от друга.

$$D_{KL}(p_1, p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}.$$

- Известно, что это расстояние всегда неотрицательно, равно нулю iff  $p_1 \equiv p_2$ .

## Применительно к НММ

- Мы определим

$$p_1(Q) = \frac{p(Q, D|\lambda)}{p(D|\lambda)}, \quad p_2(Q) = \frac{p(Q, D|\lambda')}{p(D|\lambda')}.$$

- Тогда  $p_1$  и  $p_2$  — распределения, и расстояние Kullback–Leibler:

$$\begin{aligned} 0 \leq D_{LK}(\lambda, \lambda') &= \sum_Q \frac{p(Q, D|\lambda)}{p(D|\lambda)} \log \frac{p(Q, D|\lambda)p(D|\lambda')}{p(Q, D|\lambda')p(D|\lambda)} = \\ &= \log \frac{p(D|\lambda')}{p(D|\lambda)} + \sum_Q \frac{p(Q, D|\lambda)}{p(D|\lambda)} \log \frac{p(Q, D|\lambda)}{p(Q, D|\lambda')}. \end{aligned}$$

## Вспомогательная функция

- Введём вспомогательную функцию

$$Q(\lambda, \lambda') = \sum_Q p(Q|D, \lambda) \log p(Q|D, \lambda').$$

- Тогда из неравенства следует, что

$$\frac{Q(\lambda, \lambda') - Q(\lambda, \lambda)}{p(D|\lambda)} \leq \log \frac{p(D|\lambda')}{p(D|\lambda)}.$$

- Т.е., если  $Q(\lambda, \lambda') > Q(\lambda, \lambda)$ , то  $p(D|\lambda') > p(D|\lambda)$ .
- Т.е., если мы максимизируем  $Q(\lambda, \lambda')$  по  $\lambda'$ , мы тем самым будем двигаться в нужную сторону.

## Функция $Q$

- Нужно максимизировать  $Q(\lambda, \lambda')$ . Перепишем:

$$\begin{aligned} Q(\lambda, \lambda') &= \sum_Q p(Q|D, \lambda) \log p(Q|D, \lambda') = \\ &= \sum_Q p(Q|D, \lambda) \log \pi_{q_1} \prod_t a_{q_{t-1}q_t} b_{q_t}(d_t) = \\ &= \sum_Q p(Q|D, \lambda) \log \pi_{q_1} + \sum_Q p(Q|D, \lambda) \sum_t \log a_{q_{t-1}q_t} b_{q_t}(d_t). \end{aligned}$$

- Последнее выражение легко дифференцировать по  $a_{ij}$ ,  $b_i(k)$  и  $\pi_i$ , добавлять соответствующие множители Лагранжа и решать. Получится именно пересчёт по алгоритму Баума-Велха (проверьте!).

## Спасибо за внимание!

- Lecture notes, слайды и коды программ появятся на моей homepage:  
`http://logic.pdmi.ras.ru/~sergey/index.php?page=teaching`
- Присылайте любые замечания, коды программ на других языках, решения упражнений, новые численные примеры и прочее по адресам:  
`sergey@logic.pdmi.ras.ru, smartnik@inbox.ru`