Краткий конспект

Лекция 7. Скрытые марковские модели в биоинформатике
draft version

Д. Ищенко* Б. Коварский* И. Алтухов* Д. Алексеев*

7 марта, 2016

*МФТИ

На предыдущих лекциях мы научились выравнивать последовательности, сравнивать их, собирать короткие в большие, но пока что последовательность для нас это всего лишь набор символом, смысла которого мы не понимаем. Важная задачу в вычислительной биологии - это зная биологическую последовательность, понять структурные и функциональные характеристики более высокого порядка, проаннотировать последовательность.

В случае геномной последовательности, нас интересует определение экзон-интронной структуры, поиск открытых рамок считывания, поиск сайтов связывания рибосом, транскрипционных факторов, геномных островов. Для последовательностей отдельных белков нас интересуют структурные особенности белка, его укладка.

Оказывается, с разной эффективностью перечисленные задачи могут быть решены при помощи скрытых марковских моделей. Более того, выравнивать последовательности также можно с использованием HMM

Для начала, вспомним, что такое марковская цепь.

1 Марковские цепи

1.1 Определения

Цепь Маркова — последовательность дискретных случайных величин (состояний) $Q = q_0, q_1, \ldots, q_n, \ldots$, обладающая *марковскми свойством*: значение случайной величины на любом шаге, зависит только от предыдущего состояния q_{n-1}:

$$\mathrm{P}(q_n|q_{n-1}, q_{n-2}, \ldots, q_n) = \mathrm{P}(q_n|q_{n-1})$$

$$q_n \in S$$

$$S = \{s_1, \ldots, s_k, \ldots\}$$

- конечное или счетное множество возможных состояний марковской цепи.

Марковская модель - вероятностная модель, описывающая последовательность событий, обладающим марковским свойством. Таким образом, марковская модель - минимально возможное усложнение модели независимых испытаний.

Марковская модель $\lambda = (S, A, \pi)$ однозначно задается следующим набором параметров:

1. множество состояний

$$S = \{s_1, \ldots, s_k, \ldots\}$$

2. матрица перехода между состояниями

$$A = a_{ij}(n) = \mathrm{P}(q_{n+1} = s_j | q_n = s_i)$$

   Матрица перехода является стохастической матрицей: для ее строк выполняется условие $\sum_j a_{ij} = 1$

3. начальное распределение

$$\pi = (pi_i); \quad \pi_i = \mathrm{P}(q_0 = s_i)$$

Реализацией марковской модели $\lambda = (S, A, \pi)$ служит последовательность: $Q = q_0, q_1, \ldots, q_k, \ldots$, иными словами марковская модель генерирует последовательность событий Q:

$$\lambda = (S, A, \pi) \rightsquigarrow Q$$

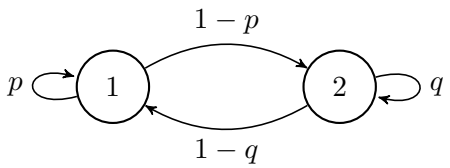
1.2 Примеры

Рассмотрим простейшую марковскую цепь с двумя состояниями:

$$\lambda = (S, A, \pi)$$

$$S = \{1, 2\}; \quad A = \begin{pmatrix} p & (1-p) \\ (1-q) & q \end{pmatrix}; \quad \pi = \begin{pmatrix} r \\ 1-r \end{pmatrix}$$

Граф переходов для такой марковской цепи:



Тогда, вероятность наблюдать последовательность состояний (траекторию) $1 \to 1 \to 2$, при заданной марковской модели λ будет определяться как:

$$\mathrm{P}(Q = 1, 1, 2|\lambda) = \mathrm{P}(q_0 = 1)\mathrm{P}(q_1 = 1|q_0 = 1)\mathrm{P}(q_2 = 2|q_1 = 1) = \pi_1 a_{11} a_{12} = rp(1-p) \tag{1}$$

Если траектория не определена, то вероятность, что мы окажемся после двух шагов в состоянии 2:

$$\mathrm{P}(q_2 = 2|\lambda) = \mathrm{P}(Q = 1, 1, 2) + \mathrm{P}(Q = 2, 1, 2) + \mathrm{P}(Q = 1, 2, 2) + \mathrm{P}(Q = 2, 2, 2) = \tag{2}$$

$$= \pi_1 a_{11} a_{12} + \pi_2 a_{21} a_{12} + \pi_1 a_{12} a_{22} + \pi_2 a_{22} a_{22} \tag{3}$$

Обобщим эти наблюдения для произвольной марковской цепи.

Вероятность траектории:

i_k - номер состояния из числа возможных

$$\mathrm{P}(Q) = \mathrm{P}(q_0 = i_0, \ldots, q_n = i_n) = \prod_{k=0}^{n} \mathrm{P}(q_k = i_k|q_{k-1} = i_{k-1}, \ldots, q_0 = i_0) = \prod_{k=0}^{n} \mathrm{P}(q_k = i_k|q_{k-1} = i_{k-1}) = \tag{4}$$

$$= \prod_{k=0}^{n} \mathrm{P}(q_k = i_k|q_{k-1} = i_{k-1}) = \pi_{i_0} \prod_{k=1}^{n} a_{i_{k-1} i_k} \tag{5}$$

Вероятность перехода из состояния i_0 в i_n после n шагов

$$\mathrm{P}(q_n = i_n|q_0 = i_0) = \sum_{i_1, \ldots, i_{n-1}} \pi_{i_0} \prod k = 1^n a i_{k-1} i_k = (A^n) i_0, i_n \pi_{i_0}$$

Вероятности наблюдения состояний после n шагов.

$$p_n = \mathrm{P}(q_n) = A^n \pi$$

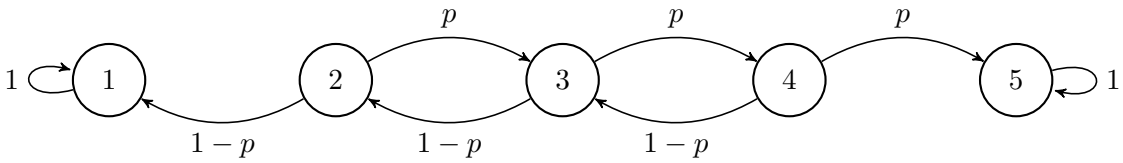
То есть матрица переходных вероятностей за n шагов однородной цепи Маркова есть n-я степень матрицы переходных вероятностей за 1 шаг

Еще один пример - случайное блуждание с отражением и с поглощением. Пускай есть частица, которая в первый момент времени имеет координату $q_0 = k$. С

вероятностью p она движется на единицу вверх и с $1-p$ на единицу вниз. Состояния 1 и N - являются поглощающими, т.е. после их достижения уровня блуждание прекращается. Марковская модель для такого процесса при $N = 5, k = 3$ будет иметь вид:

$$S = \{1,2,3,4,5\}; \quad A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1-p & 0 & p & 0 & 0 \\ 0 & 1-p & 0 & p & 0 \\ 0 & 0 & 1-p & 0 & p \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}; \quad \pi = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix};$$

Граф переходов:



Перейдем теперь к рассмотрению скрытой марковской модели.

2 Скрытые марковские модели

2.1 Определения

Скрытая марковская модель [первого порядка] — это вероятностная модель последовательности $(q_0, d_0), \ldots, (q_n, d_n), \ldots$, которая состоит из набора наблюдаемых переменных и набора скрытых переменных состояния q_n . При этом значения наблюдаемой переменной d_n на шаге n зависит только от скрытого состояния q_n, которое в свою очередь зависит лишь от скрытого состояния q_{n-1} на предыдущем шаге $n-1$.

Что поменялось в сравнение с обычной марковским процессом? Теперь состояния q_n скрыты от наблюдателя, а судить о становится возможным по наблюдаемым d_n.

(нужна картинка происходящего)

Скрытая марковская модель $\lambda = (S, \Sigma, A, B, \pi)$ описывается следующим набором параметров:

1. конечное множество состояний скрытой марковской модели:

$$S = \{s_1, \ldots, s_k\}$$

2. матрица вероятностей переходов между скрытыми состояниями

$$A = (a_{ij}); \quad a_{ij} = \mathrm{P}(s_j | s_i)$$

3. алфавит, множество наблюдаемых скрытой марковской модели

$$\Sigma = \{x_1, \ldots, x_m\}$$

4. матрица вероятностей эмиссий, т.е. вероятностей получить наблюдаемую x_k, находясь в состоянии s_j

$$B = (b_{jk}); \quad b_{ik} = p(x_k | s_j)$$

5. начальное распределение

$$\pi = (\pi_i); \quad \pi_i = \mathrm{P}(q_0 = s_i)$$

Реализацией HMM является набор из двух последовательностей:

- наблюдения (данные):
$$D = (d_0, d_1, \ldots, d_n)$$
- скрытые состояния (траектория цепи)

$$Q = (q_0, q_1, \ldots, q_n)$$

$$\lambda = (S, \Sigma, A, B) \rightsquigarrow (D, Q)$$

2.2 Задача о подмене монеты

Для иллюстрации рассмотрим задачу о подмене монеты. Допустим некто играет с вами в орлянку, подбрасывает монету, сообщая вам лишь результат - орел или решка, самой монеты вы не видите. При этом ваш противник не слишком честен и изредка он делает подмену правильной монеты на фальшивую у которой одна из сторон выпадает чаще (скажем, решка в 75% случаев) и наоборот. Можете ли вы глядя на результат - последовательность орлов и решек, определить в какой момент наиболее вероятно были произведены подмены?

Опишем процесс в терминах СММ.

Имеется два скрытых состояния - фальшивая монета (F, *false*) либо нет (N, *normal*) и две наблюдаемых - орел (H, *heads*) либо решка (T, *tails*)

$$S = \{F, N\}; \quad \Sigma = \{H, T\}$$

Вероятности переходов, допустим:

$$A = \begin{matrix} \\ F \\ N \end{matrix} \begin{matrix} \begin{matrix} F & N \end{matrix} \\ \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix} \end{matrix}$$

Вероятности наблюдаемых:

$$B = \begin{matrix} \\ F \\ N \end{matrix} \begin{matrix} \begin{matrix} H & T \end{matrix} \\ \begin{pmatrix} 0.25 & 0.75 \\ 0.5 & 0.5 \end{pmatrix} \end{matrix}$$

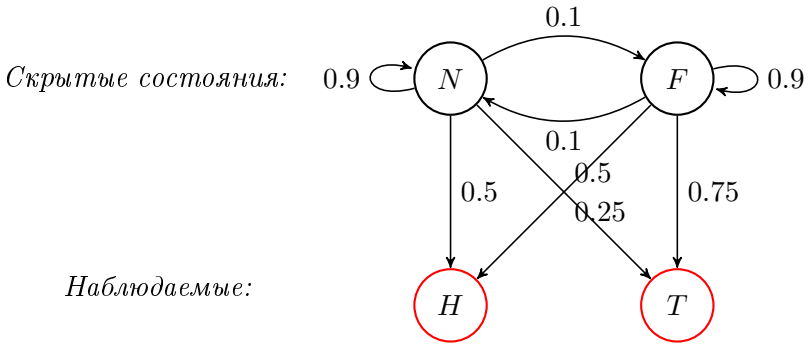
Начальное распределение:

$$\pi = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

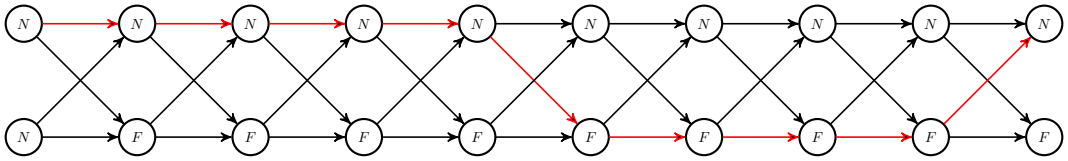
Последовательность наблюдений: $D = (H, H, T, T, H, T, T, T, T, H)$

Последовательность скрытых состояний: $Q = (N, N, N, N, N, F, F, F, F, N)$

Данную СММ можно изобразить в виде графа переходов:



Последовательность скрытых состояний может быть изображена как траектория на направленном ациклическом графе всевозможных траекторий переходов между скрытыми состояниями:



Для определения моментов подмены монеты мы должным подобрать среди всех возможных последовательностей скрытых состояний (траекторий) такую, при которых вероятность наблюдать последовательность D максимальна, т.е. мы должны решить следующую задачу:

$$Q^* = \underset{Q}{\operatorname{argmax}} P(D, Q|\lambda)$$

2.3 Анализ биологических последовательностей при помощи CMM

Какое отношение орлянка имеет к анализу биологических последовательностей? Биологическую последовательность можно воспринимать как последовательность наблюдений (аминокислот и нуклеотидов), за которыми скрываются неизвестные биологические свойства (скрытые состояния). Рассмотрим два примера.

2.3.1 Поиск геномных островов

Первый пример - поиск геномных островов. Геномные острова - кластеры генов в геномах прокариот, приобретаемые посредством горизонтального переноса.

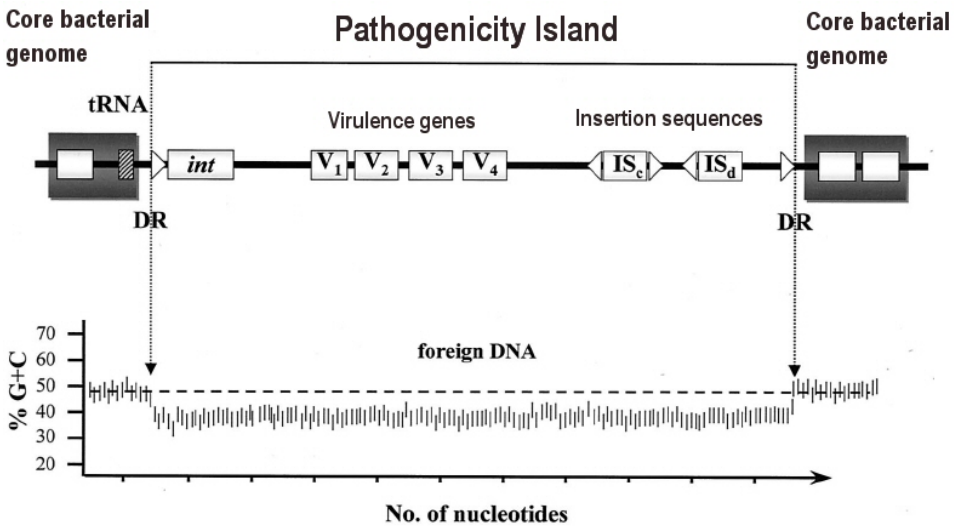


Рис. 1: Упрощенная схема строения острова патогенности. IS-insertion sequence, *int*-интеграза, $V_1, \ldots, V_4$ - гены вирулентности, DR-прямые повторы. Иллюстрация из *Herbert Schmidt and Michael Hensel, Clinical Microbiology Reviews, January 2004, Vol. 17, p. 14-56.*

Когда в 1990х исследовали геномы кишечной палочки оказалось, что патогенные и непатогенные штаммы в сущности отличаются наличием или отсутствием определенных кластеров генов, расположенными в нестабильных регионах хромосом, из чего следует что кластеры скорее всего были приобретены горизонтально. Подобные кластеры получили название острова патогенности. В островах патогенности могут находятся разнообразные токсины (гемолизин E.coli, пестицин чумной палочки), адгезины, позволяющие бактерии прикрепляться, суперантигены (вызывающие массовую активацию неспецифическую активацию Т-лимфоцитов и как следствие инфекционно-токсический шок). Но островками патогенности все не ограничилось. Затем были обнаружены иные острова: метаболические острова, симбиотические острова, острова резистентности.

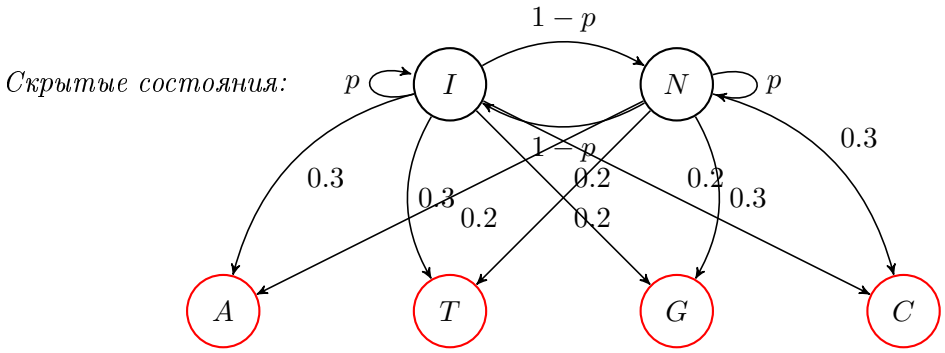
Геномные острова имеют вполне узнаваемую структуру. Их длина несколько десятков тысяч нуклеотидов, они часто фланкированы прямыми повторами, внутри островов присутствуют гены обеспечивающие мобильность (интегразы или транспозазы) и специфичные гены (токсины, гены резистентности). Для нас же важно то, что k-мерный спектр (частоты встречаемости нуклеотидных подстрок длины k) в островах, отличается от геномного. В частности у геномных островов иной GC-состав

В терминах CMM в данной задачи наблюдаемые - нуклеотиды, взятые из алфавита $\Sigma = \{A, T, G, C\}$, а скрытые состояния, находится ли данный нуклеотид внтури острова или нет $S = \{I, N\}$. Геномные острова и части генома не относящиеся к ним состоят из целого блока нуклеотидов, а потому вероятности переходов между скрытыми состояниями малы, это позволяет эффективно применять CMM.

Допустим, GC-состав острова $\sim 40\%$, генома $\sim 60\%$, тогда:

$$\mathrm{P}(G|I) = \mathrm{P}(C|I) = 0.2; \quad \mathrm{P}(A|I) = \mathrm{P}(T|I) = 0.3$$
$$\mathrm{P}(G|N) = \mathrm{P}(C|N) = 0.3; \quad \mathrm{P}(A|N) = \mathrm{P}(T|N) = 0.2$$

Граф переходов будет выглядеть как:



Для определения границ острова будет решаться та же самая задача:

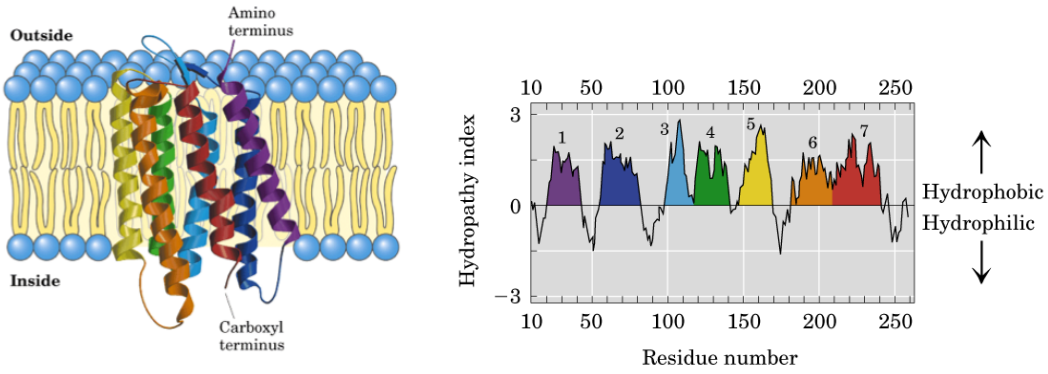
$$Q^* = \underset{Q}{\operatorname{argmax}}\, P(D, Q|\lambda)$$

, что и в примере с фальшивой монетой.

2.3.2 Определение структуры трансмембранных белков

Второй пример - это определение доменов трансмембранных белков. Трансмембранные белки насквозь пронизывают липидный бислой, при этом часть их находится внутри мембраны, а часть - снаружи клетки, часть внутри цитоплазмы. Внутри мембраны чаще располагаются гидрофобные белки, снаружи - гидрофильные, как видно на примере бактериородопсина (рис. 2).

В простейшей CMM для определения внутримембранных регионов двадцатисимвольный аминокислотный алфавит можно редуцировать до двухбуквенного:

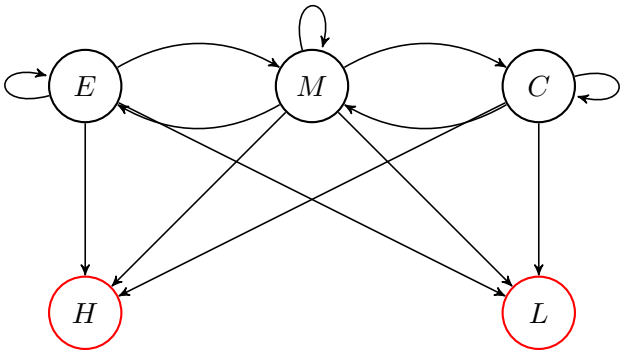


(a) (b)

Рис. 2: (a) структура бактериородопсина; (b) индекс гидропатии для разных аминокислотных остатков бактериородопсина. Иллюстрация из *Lehninger Principles of Biochemistry, 3rd ed., 2000*

$\Sigma = \{H(hydrophobic), L(hydrophylic)\}$ и рассматривать три скрытых состояния: $S = \{E(extracellular), M(membrane), C(cytoplasmic)\}$

Граф переходов:



Как видно, особенность данной CMM, что вероятности переходов $p(E|C) = p(C|E) = 0$, так как мы не можем оказаться внутри клетки, не пройдя через мембрану.

2.4 Алгоритмические задачи, связанные с CMM

С CMM связаны три основные задачи

1. Уже упомянутая *задача распознавания*: определение наиболее вероятной последовательности скрытых состояний при заданных параметрах СММ λ и последовательности наблюдаемых:

$$Q^* = \underset{Q}{\operatorname{argmax}} \mathrm{P}(D, Q|\lambda)$$

Задача решается при помощи *алгоритма Витерби*.

2. *Определение правдоподобия модели*: определение вероятности получить последовательность наблюдаемых Q при заданных параметрах СММ λ:

$$L = \mathrm{P}(D|\lambda)$$

Оценка правдоподобия модели позволяет из конечного набора конкурирующих моделей $\lambda_1, \lambda_2, \ldots, \lambda_n$ выбирать из них наиболее правдоподбную:

$$\lambda^* = \underset{\lambda_1,\lambda_2,\ldots,\lambda_n}{\operatorname{argmax}} \mathrm{P}(D|\lambda_i)$$

Задача решается при помощи *алгоритма просмотра вперед*.

3. *Обучение СММ без учителя*: подобрать набор параметров СММ λ при котором вероятность наблюдать последовательность Q максимальна:

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} \mathrm{P}(D|\lambda)$$

В отличие от предыдущей задачи у нас нет конечного набора заданных СММ, вместо этого требуется найти СММ среди всевозможных значений параметров. Задача решается при помощи *алгоритма Баума-Велша*.

3 Алгоритмы

Последовательность скрытых состояний - образует путь в направленном ациклическом графе переходов между скрытыми состояниями. Задачи на направленных ациклических графах как мы знаем хорошо решаются при помощи динамического программирования.

3.1 Задача распознавания. Алгоритм Витерби

$$Q^* = \underset{Q}{\operatorname{argmax}} \mathrm{P}(D, Q|\lambda)$$

Для начала допустим, что последовательности скрытых состояний Q заданы, как и наблюдаемые D:

$$D = (d_0, d_1, d_2, \ldots, d_n)$$
$$Q = (q_0, q_1, q_2, \ldots, q_n)$$

Тогда:

$$\mathrm{P}(D, Q|\lambda) = \mathrm{P}(D|Q, \lambda)\mathrm{P}(Q, \lambda)$$

$$\mathrm{P}(D|Q, \lambda) = \mathrm{P}(d_0|q_0)\mathrm{P}(d_1|q_1)\ldots\mathrm{P}(d_n|q_n) = \prod_{i=0}^{n} \mathrm{P}(d_i|q_i)$$

$$\mathrm{P}(Q|\lambda) = \mathrm{P}(q_0)\mathrm{P}(q_1|q_0)\ldots\mathrm{P}(q_n|q_{n-1}) = \mathrm{P}(q_0)\prod_{i=1}^{n} \mathrm{P}(q_i|q_{i-1})$$

$$\mathrm{P}(D, Q|\lambda) = \mathrm{P}(q_0)\mathrm{P}(d_0|q_0)\prod_{i=1}^{n} \mathrm{P}(d_i|q_i)\mathrm{P}(q_i|q_{i-1})$$

Вместо произведения вероятностей можно рассматривать сумму логарифмов, что вычислительно удобнее :

$$\log \mathrm{P}(D, Q|\lambda) = \log \mathrm{P}(q_0) + \log \mathrm{P}(d_0|q_0) + \sum_{i=1}^{n} \log(\mathrm{P}(d_i|q_i)\mathrm{P}(q_i|q_{i-1}))$$

Проблема в том, что в действительности мы не знаем последовательность Q, перебирать же всевозможные траектории и находить $\mathrm{P}(D, Q|\lambda)$ вычислительно неэффективно.

Чтобы не перебирать все траектории по аналогии с задачей поиска оптимального выравнивания мы можем запоминать на каждом шаге СММ для каждого скрытого состояния какая оптимальная траектория приводит в это состояние. Для этого в алгоритме Витерби определяется следующая вспомогательная величина:

$$v_{l,i} = \max_{q_0,\ldots,q_{l-1}} \mathrm{P}(d_0, d_1, \ldots, d_l, q_0, q_1, \ldots, q_l = s_i|\lambda) = \tag{6}$$

$$= \mathrm{P}(d_l|q_l = s_i, \lambda) \max_{q_0,\ldots,q_{l-1}} \mathrm{P}(d_0, d_1, \ldots, d_{l-1}, q_0, q_1, \ldots, q_l = s_i|\lambda) \tag{7}$$

Она имеет следующий смысл: это максимальная вероятность наблюдать последовательность $d_0, d_1, \ldots, d_l$ среди всевозможных произвольных траекторий длины l: $q_0, q_1, \ldots, q$, заканчивающихся в состоянии s_i. $v_{l,i}$

По индукции можно получить формулу для пересчета $v_{l,i}$ на каждом шаге на основе значений на предыдущем шаге:

$$v_{l+1,j} = \mathrm{P}(d_{l+1}|q_{l+1} = s_j, \lambda) * \max_i(v_{l,i}\mathrm{P}(q_{l+1} = s_j|q_l = s_i, \lambda))$$

Это приводит нас к следующей процедуре для определения наиболее вероятной последовательности скрытых состояний Q^*

procedure VITERBI(A, B, π, D)
  $V \leftarrow [\quad]$ ▷ Массив для хранения вероятностей наиболее вероятных путей
  $T \leftarrow [\quad]$ ▷ Массив для хранения направлений перехода
  for $s \leftarrow 0, k-1$ **do**
    $V[0, s] \leftarrow \pi[s] * B[D[0], s]$
  end for
  for $n \leftarrow 1, N$ **do**
    for $t \leftarrow 0, k-1$ **do**
      $V[n, t] \leftarrow \max_s(V[n-1, s] * A[s, t]) * B[D[n], t]$
      $T[n, t] \leftarrow \mathrm{argmax}_s(V[n-1, s] * A[s, t])$
    end for
  end for
  $S_{fin} \leftarrow \mathrm{argmax}_s(V[N, s])$
  return T, S_{fin}
end procedure

Процедура восстановления пути аналогична задачи выравнивания:

procedure RESTOREHIDDENPATH(T, S_{fin})
  ...
  return Q
end procedure

3.2 Алгоритм просмотра вперед

Задача определения правдоподобия модели решается при помощи алгоритма просмотра вперед.

В задаче необходимо определить вероятность $\mathrm{P}(D|\lambda)$. Для этого нужно просуммировать по всем возможным траекториям Q уже знакомые по прошлой задаче вероятности $\mathrm{P}(D, Q|\lambda)$.

$$\mathrm{P}(D|\lambda) = \sum_{Q} \mathrm{P}(D, Q|\lambda) = \sum_{Q} \mathrm{P}(D|Q, \lambda)\mathrm{P}(Q|\lambda) \tag{8}$$

Сам алгоритм схож с алгоритмом Витерби, но вместо $v_{l,i}$ для эффективного пересчета вводится величина $\alpha_{l,i}$:

$$\alpha_{l,i} = \mathrm{P}(d_0, d_1, \ldots, d_l, q_l = s_i|\lambda) = \sum_{q_0, q_1, \ldots, q_{l-1}} \mathrm{P}(d_0, d_1, \ldots, d_l, q_0, q_1, \ldots, q_l = s_i|\lambda)$$

$\alpha_{l,i}$ - суммарная по всевозможным траекториям длины l: $q_0, q_1, \ldots, q$, заканчивающихся в состоянии s_i вероятность наблюдать последовательность $d_0, d_1, \ldots, d_l$.

Для пересчета используется формула:

$$\alpha_{l+1,j} = \mathrm{P}(d_{l+1}|q_{l+1} = s_j, \lambda) * \sum_{i}(\alpha_{l,i}\mathrm{P}(q_{l+1} = s_j|q_l = s_i, \lambda))$$

Правдоподобие модели: $\mathrm{P}(D|\lambda) = \max_i(\alpha_{n,i})$

4 Ссылки

[1] A tutorial on Hidden Markov Models and selected applications in speech recognition - Proceedings of the IEEE, 1989

[2] Лекции Сергея Николенко в Академическом университете

[3] Лекции Дмитрия Ветрова на ВМК

[4] Jones N., Pevzner P. An Introduction to Bioinformatics Algorithms – MIT Press, 2004.