

# Прикладной статистический анализ данных

## Марковские модели

Олег Бахтеев  
psad@phystech.edu

2021

# Марковская цепь

Последовательность дискретных случайных величин  $X_1, \dots, X_T$ , принимающих некоторый набор значений  $\{O_1, \dots, O_m\}$ , называется простой однородной цепью Маркова, если

$$P(X_{t+1} = O_{t+1} | X_t = O_t, \dots, X_1 = O_1) = P(X_{t+1} = O_{t+1} | X_t = O_t),$$

$P(X_{t+1} = O_{t+1} | X_t = O_t)$  не зависит от номера шага  $t$ .

Марковская цепь задается:

- множеством наблюдаемых состояний  $\{O_1, \dots, O_m\}$ ;
- начальными значениями вероятности состояний  $P(X_1 = O_i) = P_i$ ;
- вероятностью перехода между состояниями  $P(X_t = O_i | X_{t-1} = O_j) = P_{ij}$ .

## Пример: погода

Задан набор из трех состояний:

- ①  $O_1$  = дождливая погода;
  - ②  $O_2$  = пасмурная погода;
  - ③  $O_3$  = солнечная погода.
- Какова вероятность, что в следующие четыре дня погода будет меняться как “солнце-солнце-дождь-дождь”?

$$P(O_3, O_3, O_1, O_1) = P_3 P_{33} P_{31} P_{11}$$

- Какова вероятность, что ровно  $N$  дней будет пасмурная погода?

$$P(X_2 = O_2, \dots, X_t = O_2, X_N \neq O_2 | X_1 = O_2) = P_{22}^{N-1} (1 - P_{22}).$$

- Ожидаемая продолжительность постоянной пасмурной погоды:

$$E = \sum_{t=1}^{\infty} t \cdot P(X_2 = O_2, \dots, X_t = O_2, X_{t+1} \neq O_2 | X_1 = O_2) = \frac{1}{1 - P_{22}}.$$

# Языковая модель

Примером марковской цепи выступает языковая  $n$ -грамм модель.

Под  $n$ -граммой понимается последовательность из  $n$  подряд идущих слов.

**Пример:**

*Шла Саша по шоссе* содержит три 2-граммы:

- ① Шла Саша;
- ② Саша по;
- ③ По шоссе.

# Языковая модель

Языковая модель позволяет оценить вероятность появления предложения на основе марковской модели языка.

Для удобства при построении языковой модели вводятся два специальных символа: *BOS (Begin Of Sentence)* и *EOS (End Of Sentence)*.

**Пример** для 3-граммной языковой модели:

$$\begin{aligned} p(w_1, \dots, w_n) = & p(SOS) \times \\ & \times p(w_1|SOS)p(w_2|w_1, SOS)p(w_3|w_2, w_1) \dots p(w_n|w_{n-1}, w_{n-2}) \\ & \times p(EOS|w_n, w_{n-1}). \end{aligned}$$

# Языковая модель: измерение качества

Как оценить качество модели?

**Кросс-Энтропия.**

Оценка на основе заданной выборки  $w_1, \dots, w_n$ :

$$H = -\frac{1}{n} \log p(w_1, \dots, w_n).$$

**Перплексия:**

$$PP = 2^H = p(w_1, \dots, w_n)^{-\frac{1}{n}}.$$

- $PP = \infty \rightarrow$  марковская цепь не описывает выборку;
- $PP = 1 \rightarrow$  марковская цепь идеально описывает выборку.

## Языковая модель: незнакомые слова

В случае, если языковой модели встретится неизвестное слово,  $p(w_1, \dots, w_n) = 0$ .

**Варианты работы с незнакомыми словами:**

- Сглаживание Лапласа:

$$p(w_i) = \frac{c_i + 1}{\sum_{j=1}^v c_j + v},$$

где  $c_i$  — встречаемость слова  $w_i$  в тексте,  $v$  — мощность словаря.

- Интерполяция моделей разных порядков:

$$\hat{p}(w_n | w_{n-1}, w_{n-2}) = \lambda_1 p(w_n | w_{n-1}, w_{n-2}) + \lambda_2 p(w_n | w_{n-1}) + \lambda_3 p(w_n),$$

$$\sum_i \lambda_i = 1.$$

# Марковские модели, проверки гипотез

Проверка гипотезы о соответствии вектора вероятностей  $p_{i1}, \dots, p_{im}$  перехода из состояния  $i$  заданному:

выборка:  $X_1, \dots, X_T$

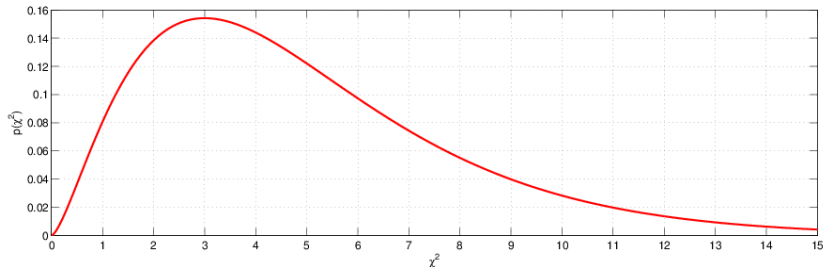
нулевая гипотеза:  $H_0: p_{i1}, \dots, p_{im} = \mathbf{p}^0$

альтернатива:  $H_1: p_{i1}, \dots, p_{im} \neq \mathbf{p}^0$

статистика:  $n_i \sum_j \frac{(p_{ij} - p_{ij}^0)^2}{p_{ij}^0},$

где  $n_i$  — встречаемость наблюдения  $O_i$   
в последовательности  $X_1, \dots, X_{T-1}$

нулевое распределение:  $\chi_{m-1}^2$





# Марковские модели, проверки гипотез

Проверка гипотезы о том, что марковскую цепь второго порядка можно “свернуть” в цепь первого порядка:

выборка:  $X_1, \dots, X_T$ , задана марковская модель порядка 2:

$$P(X_t = O_k | X_{t-1} = O_j, X_{t-2} = O_i, \dots) = p_{ijk}$$

нулевая гипотеза:  $H_0: p_{1jk} = p_{2jk} = \dots = p_{mjk}$ .

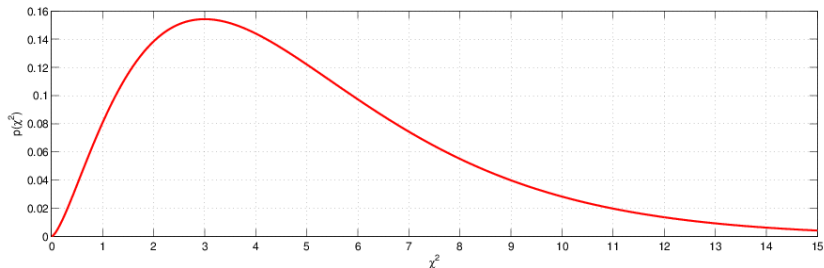
альтернатива:  $H_1: H_0$  неверна.

статистика:  $-2\log\left(\prod_{i,j,k=1}^m (\hat{p}_{ij}/p_{ijk})^{n_{ijk}}\right)$ ,

$\hat{p}_{ij}$  — оценка МП,

$$n_{ijk} = |\{X_t : X_t = O_i, X_{t+1} = O_j, X_{t+2} = O_k\}|.$$

нулевое распределение:  $\chi_{m(m-1)^2}^2$



## Проверка гипотез, комментарии

- Вероятностное распределение  $p_{ij}$  представимо как мультиномиальное распределение события  $j$  при условии события  $i$ , поэтому для проверки гипотез применимы критерии для мультиномиальных величин (в случае  $m = 2$  — критерии для распределения Бернулли).
- Предполагается, что все вероятности переходов при проверке гипотез строго больше нуля.
- Критерии можно обобщить на случай моделей более высокого порядка (например, полагать  $p_{ijk}$  моделью первого порядка с событием  $X_t = O_k$  при условии *единого* события  $\langle X_{t-1} = O_j, X_{t-2} = O_i \rangle$ ).
- Возможна проверка критериев по нескольким последовательностям, а не по одной. Статистики и нулевая гипотеза от этого не меняются.
- Подробнее — см. Anderson et al. (в списке литературы).

# Марковские модели как порождающие модели

## Примеры порождающих моделей:

- Генераторы поведения ветра (используются для изучения климата).
- Генераторы текста (см.  
<https://hackernoon.com/automated-text-generator-using-markov-chain-de999a41e047>)
- SciGen: генератор псевдонаучных текстов
  - ▶ В России известен, благодаря сгенерированной статье “Router” (“Корчеватель”). Подробнее см. на вики:  
<https://en.wikipedia.org/wiki/SCIgen>

# Скрытая марковская модель

Скрытая марковская модель — обобщение марковской цепи, в котором разделяются наблюдаемые и ненаблюдаемые (скрытые) переменные.

## Элементы скрытой марковской модели

- $X_1, \dots, X_T$  — наблюдаемая последовательность;
- $H_1, \dots, H_T$  — скрытая последовательность;
- $S_1, \dots, S_n$  — множество скрытых состояний;
- $O_1, \dots, O_m$  — алфавит наблюдений;
- Вероятности перехода из одного состояния в другое:

$$a_{ij} = P(H_{t+1} = S_j | H_t = S_i);$$

- Вероятность наблюдений:

$$b_j(k) = P(X_t = O_k | H_t = S_j).$$

- Распределение вероятностей начальных состояний:

$$\pi_i = P(H_1 = S_i).$$

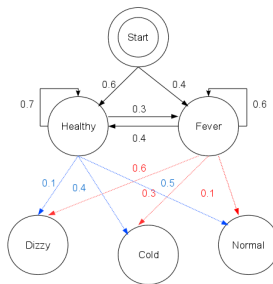
# HMM: пример

## Пример: wikipedia

Доктор опрашивает потенциально больных людей о своем самочувствии и фиксирует ответы. Люди отвечают могут ответить, что они чувствуют себя нормально (normal), что у них кружится голова (dizzy), что у них озноб (cold).

Наблюдаемые величины  $\{O_1, O_2, O_3\} = \{\text{normal}, \text{dizzy}, \text{cold}\}$ .

Скрытые величины — наличие простуды  $\{H_1, H_2\} = \{\text{healthy}, \text{fever}\}$ .



# НММ: основные задачи

- ① Как посчитать вероятность последовательности  $X_1, \dots, X_T$ ?
- ② Как выбрать наиболее подходящую скрытую последовательность  $H_1, \dots, H_T$  по последовательности  $X_1, \dots, X_T$ ?
- ③ Как настроить параметры НММ-модели по входной последовательности  $X_1, \dots, X_T$ ?

# НММ: основные задачи

- ① Как посчитать вероятность последовательности  $X_1, \dots, X_T$ ?
- ② Как выбрать наиболее подходящую скрытую последовательность  $H_1, \dots, H_T$  по последовательности  $X_1, \dots, X_T$ ?
- ③ Как настроить параметры НММ-модели по входной последовательности  $X_1, \dots, X_T$ ?

**Что интересует нас:**

- ① Как определить адекватность модели?
- ② Как выбрать наилучшую модель?

# НММ, основные задачи, наивное решение

**Вычисление вероятности последовательности** Вычисление полной вероятности с полным перебор скрытых состояний:

$$P(X_1, \dots, X_N) = \sum_{i_1=1}^n \cdots \sum_{i_T=1}^n \pi_{i_1} b_{i_1}(X_1) a_{i_1 i_2} b_{i_2}(X_2) \dots a_{i_{T-1} i_T} b_{i_T}(X_T).$$

**Проблема:** высокая сложность:  $2T \cdot n^T$ .

**Вычисление оптимальной последовательности скрытых состояний**

Будем максимизировать вероятность каждого скрытого состояния по отдельности:

$$S_i = \arg \max_{i'} P(H_t = S_{i'} | X_1, \dots, X_T), \forall t.$$

**Проблема:** не учитываются вероятности перехода между скрытыми состояниями  $a_{ij}$ .



# НММ, основные задачи

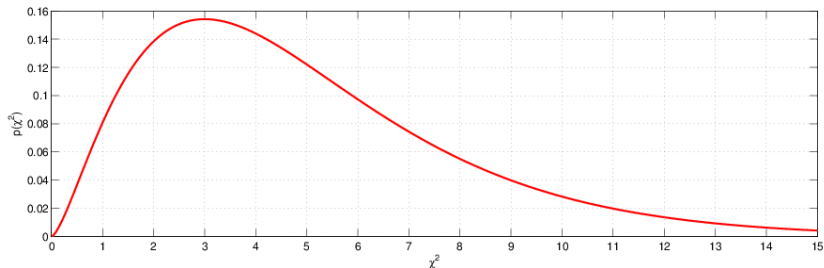
Общепринятые решения основных задач:

- Вычисление вероятности последовательности: Forward-Backward алгоритм
  - ▶ Основан на динамическом программировании
  - ▶ Сложность:  $O(n^2T)$
- Вычисление оптимальной последовательности скрытых состояний: алгоритм Витерби
  - ▶ Основан на динамическом программировании, схож с Forward-Backward алгоритмом
- Оптимизация параметров НММ-модели
  - ▶ EM-алгоритм Баума — Велша

Подробнее см. Rabiner (в списке литературы).

# НММ, проверка гипотезы

выборка:  $X_1, \dots, X_T$   
нулевая гипотеза:  $H_0: \mathbf{a} = \mathbf{a}^0, \mathbf{b} = \mathbf{b}^0, \boldsymbol{\pi} = \boldsymbol{\pi}^0$ .  
альтернатива:  $H_1: H_0$  неверна.  
статистика:  $2\log(\hat{p}(X_1, \dots, X_T) - p^0(X_1, \dots, X_T))$ .  
нулевое распределение:  $\chi^2_{n+mn+m^2}$



# НММ: сравнение моделей

Как определить понятие эквивалентности на моделях?

Дивергенция Кульбака-Лейблера:

$$D_{KL}(p_1, p_2) = \mathbb{E}_{X \sim p_2} (\log p_1(X) - \log p_2(X)).$$

- $D_{KL}(p_1, p_2) > 0$ .
- $D_{KL}(p_1, p_2) \neq D_{KL}(p_2, p_1)$ .
- $D_{KL}(p_1, p_2) = 0 \iff p_1 = p_2$ .

Модификация для НММ:

$$D'_{KL}(p_1, p_2) = \frac{1}{N} \mathbb{E}_{X_1, \dots, X_T \sim p_2} (\log p_1(X_1, \dots, X_T) - \log p_2(X_1, \dots, X_T)).$$

Симметричная версия:

$$D''_{KL}(p_1, p_2) = \frac{D'_{KL}(p_1, p_2) + D'_{KL}(p_2, p_1)}{2}.$$

# HMM: разновидности

- left-right-модели
  - ▶ Вводится порядок на множестве скрытых наблюдений
  - ▶ Переход между наблюдениями “от большего к меньшему” запрещен
  - ▶ Используется в распознавании речи
- С непрерывным распределением на наблюдениях
- Авторегрессионные HMM-модели.

# HMM: эксперимент Cave and Neuwirth

HMM обучена на большом наборе английских текстов. Размерность множества скрытых состояний — 2. Наблюдаемые величины — символы в тексте. На выходе получается распределение переходов, при котором скрытую переменную можно интерпретировать как гласную или согласную букву.

	Initial		Final	
a	0.03735	0.03909	0.13845	0.00075
b	0.03408	0.03537	0.00000	0.02311
c	0.03455	0.03537	0.00062	0.05614
d	0.03828	0.03909	0.00000	0.06937
e	0.03782	0.03583	0.21404	0.00000
f	0.03922	0.03630	0.00000	0.03559
g	0.03688	0.04048	0.00081	0.02724
h	0.03408	0.03537	0.00066	0.07278
i	0.03875	0.03816	0.12275	0.00000
j	0.04062	0.03909	0.00000	0.00365
k	0.03735	0.03490	0.00182	0.00703
l	0.03968	0.03723	0.00049	0.07231
m	0.03548	0.03537	0.00000	0.03889
n	0.03735	0.03909	0.00000	0.11461
o	0.04062	0.03397	0.13156	0.00000
p	0.03595	0.03397	0.00040	0.03674
q	0.03641	0.03816	0.00000	0.00153
r	0.03408	0.03676	0.00000	0.10225
s	0.04062	0.04048	0.00000	0.11042
t	0.03548	0.03443	0.01102	0.14392
u	0.03922	0.03537	0.04508	0.00000
v	0.04062	0.03955	0.00000	0.01621
w	0.03455	0.03816	0.00000	0.02303
x	0.03595	0.03723	0.00000	0.00447
y	0.03408	0.03769	0.00019	0.02587
z	0.03408	0.03955	0.00000	0.00110
space	0.03688	0.03397	0.33211	0.01298

# HMM: примеры применения

- Назначение соответствий между словами в исходном и переведенном предложении (наблюдения — множество слов в переведенном предложении, скрытые состояния — исходные слова).
- Анализ частей речи (наблюдения — слова, скрытые состояния — части речи).
- Распознавание речи (наблюдения — представления звуковых сегментов, скрытые состояния — слова или буквы).
- Выравнивание биологических последовательностей (наблюдения — элементы последовательности, скрытые состояния — экзоны).

# Сэмплирующие методы

Марковские модели являются генеративными моделями. Они позволяют “сэмплировать” (порождать) объекты из распределения, описываемого марковской моделью.

**Что делать с более сложными распределениями?**

- Как сэмплировать?
- Как вычислять интегралы по этим распределениям?

## Простые случаи

**Интегрирование** Метод Монте-Карло: проклятье размерности: нужно уметь сэмплировать из нашего распределения

**Сэмплирование** Пусть существует обратимая функция  $T$  из  $x \in \mathcal{U}(0, 1)$  в некоторое распределение  $z$ . Тогда

$$F_z(t) = p(z \leq t) = p(T(t') \leq t) = p(t' \leq T^{-1}(t)) = T^{-1}(t).$$

Отсюда  $F_z^{-1} = T$ .

**Пример**

$$z = \lambda \exp(-\lambda t).$$

$$F_z(t) = 1 - \exp(-\lambda t).$$

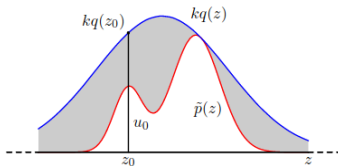
$$F_z^{-1}(t') = -\frac{1}{\lambda} \log(1 - t').$$



# Сэмплирование с отклонением

- Задана плотность  $p(z)$  (может быть задана с точностью до нормировочной константы)
- Введем распределение  $q$
- Подберем множитель  $k$  таким образом, чтобы  $kq(z) \geq p(z)$  для всех  $z$
- В цикле
  - ▶ Просэмплируем  $z_0 \sim kq$
  - ▶ Просэмплируем  $u \sim \mathcal{U}(0, z_0)$
  - ▶ Если  $u \leq p(z_0)$  — считать его сэмплом из  $p(z)$

Идея метода: сэмплы  $u$  равномерно распределены в регионе, ограниченном кривой  $p(z)$ .



Bishop, 2006

## Сэмплирование по значимости

Пусть мы не можем сэмплировать из  $p(z)$ , но можем оценивать правдоподобие в каждой точке, и хотим получить интеграл

$$Ef = \int f(z)p(z)dz.$$

Тогда введем распределение  $q$ :

$$Ef = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}dz \approx \frac{1}{L} \sum_{l=1}^L \frac{p(z^l)}{q(z^l)} f(z^l).$$

**Основная идея:** Сэмплируем аналогично сэмплированию с отклонениями, но  $q$  — марковское распределение, обусловленное на предыдущий успешный шаг  
Хотим, чтобы предельное (стационарное) распределение соответствовало нашему распределению  $p(z)$ .  
Достаточное условие

$$p(z)T(z|z') = p(z')T(z'|z).$$

## Алгоритм Метрополиса — Гастингса

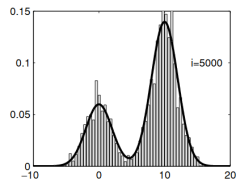
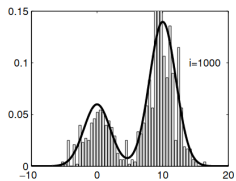
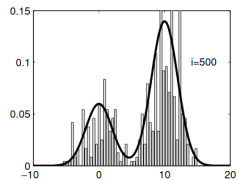
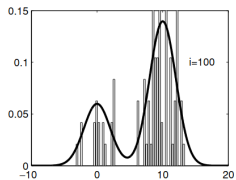
- Сэмплируем новое значение  $z' \sim q(z|z^t)$ .
- Принимаем его с вероятностью  $A(z'|z^t) = \min\left(1, \frac{p(z')q(z^t|z')}{p(z^t)q(z'|z^t)}\right)$ .
- Если приняли:  $z^{t+1} = z'$ ,
- иначе:  $z^{t+1} = z^t$ .

Условие предельного распределения выполняется:

$$p(z)T(z|z') = p(z)T(z'|z) = p(z')T(z'|z^t) = p(z')q(z'|z^t)A(z'|z^t) = p(z^t)q(z^t|z')A(z^t|z').$$

- Сэмплы скоррелированы. Если требуется декоррелировать сэмплы, можно брать каждый  $k$ -й сэмпл.
- Работает в пространствах высокой размерности значительно лучше, чем сэмплирование с отклонением.

## Пример работы, Andrieu et al.



# Литература

- Tutorial: L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition
- Tutorial: M. Stamp, A Revealing Introduction to Hidden Markov Models
- Проверка гипотез: T. W. Anderson, Leo A. Goodman, Statistical Inference about Markov Chains
- Языковые модели: D. Jurafsky, J. H. Martin, Speech and Language Processing
- Машинный перевод: P. Koehn, Statistical Machine Translation
- IBM M1 & HMM:  
[http://www.cs-114.org/wp-content/uploads/2016/04/CS114\\_L25PMachineTranslation-IBM.pdf](http://www.cs-114.org/wp-content/uploads/2016/04/CS114_L25PMachineTranslation-IBM.pdf)
- Bishop C. M. Pattern recognition and machine learning. – springer, 2006.
- Andrieu C. et al. An introduction to MCMC for machine learning //Machine learning. – 2003. – Т. 50. – №. 1. – С. 5-43.