Прикладной статистический анализ данных
Анализ последовательностей

Олег Бахтеев
mipt.psad19@gmail.com

2019

Марковская цепь

Последовательность дискретных случайных величин $X_1, \ldots, X_T$ называется простой однородной цепью Маркова, если

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

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

Таким образом, марковская цепь задается:

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

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”?

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

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

$$P(X_2 = O_2, \ldots, X_t = O_2, X_{t+1} \neq O_2 | X_1 = O_2) = P_{22}^{t-1}(1 - P_{22}).$$

- Ожидаемая продолжительность постоянной пасмурной погоды:

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

Языковая модель

n-граммой назовем последовательность из n подряд идущих слов.
Пример:
Шла Саша по шоссе содержит три 2-граммы:

1. Шла Саша;
2. Саша по;
3. По шоссе.

n-граммная языковая модель позволяет оценить вероятность появления предложения на основе марковской модели языка.

Пример для 3-граммной языковой модели $p(w_1, \ldots, w_n) =$

$$= p(SOS)p(w_1|SOS)p(w_2|w_1, SOS)p(w_3|w_2, w_1) \ldots p(w_n|w_{n-1}, w_{n-2})p(EOS|w_n, w_{n-1})$$

Языковая модель: особенности

- Как оценить качество модели?
  Перплексия:

$$PP = P(w_1, \ldots, w_n)^{-\frac{1}{n}}.$$

$$PP = b^{-\frac{1}{n}\sum_w p(w)\log_b p(w)}$$

.

- Что делать с незнакомыми словами?
  - Сглаживание Лапласа:

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

    где c_i — встречаемость слова в тексте, v — мощность словаря.
  - Интерполяция:

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

$$\sum_i \lambda_i = 1.$$

Марковские модели, проверки гипотез

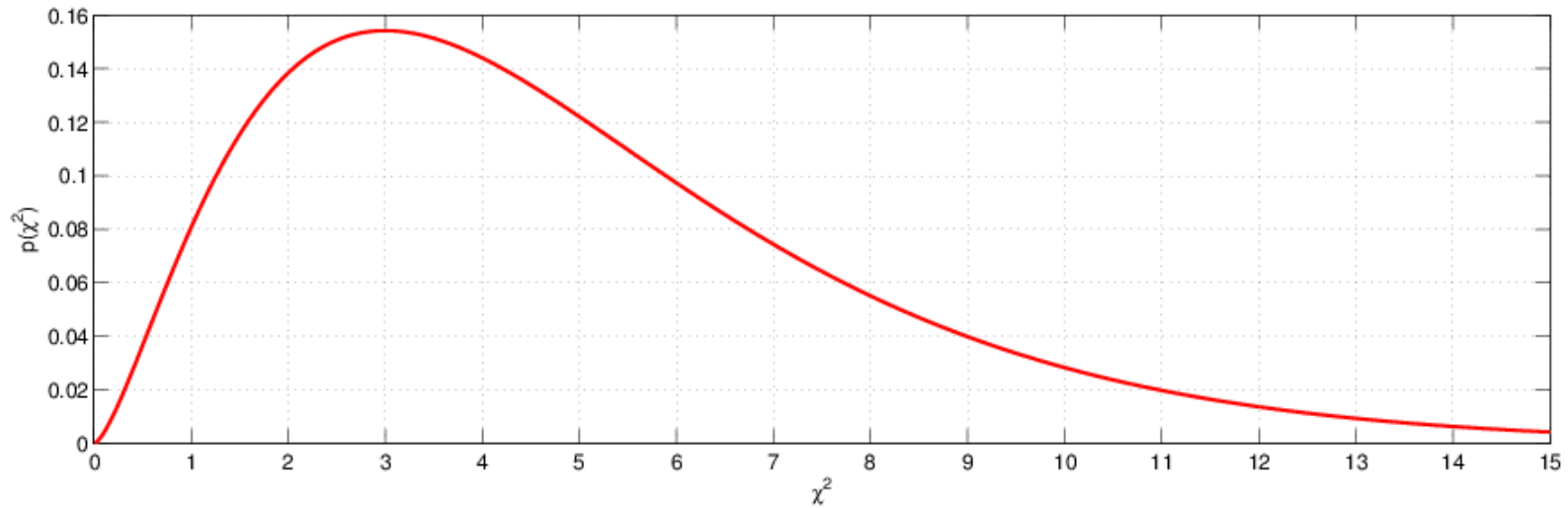
выборка: $X_1, \ldots, X_T$

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

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

статистика: $n_i \sum_j \frac{(\hat{p}_{ij} - p^0_{ij})^2}{p^0_{ij}}, n_i = |\{X_t : X_t = O_i, t = 1, \ldots, T-1\}|.$

нулевое распределение: χ^2_{m-1}



Марковские модели, проверки гипотез

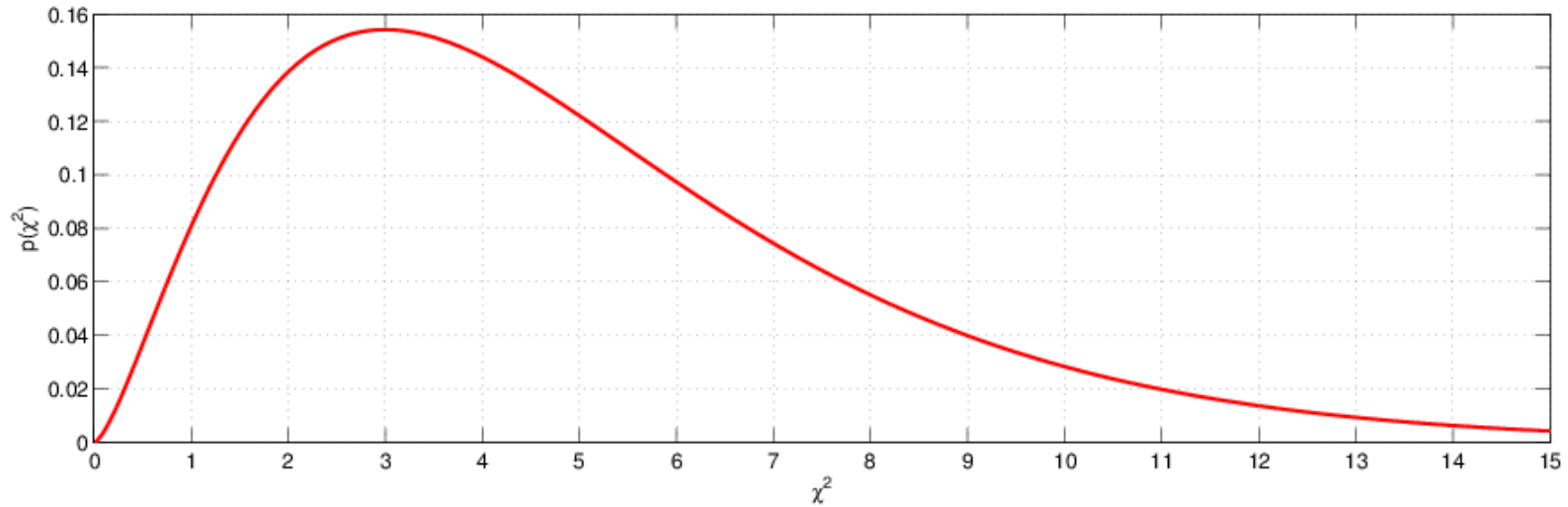
выборка: $X_1, \ldots, X_T$, задана марковская модель порядка 2:
$P(X_t = O_i | X_{t-1} = O_j, X_{t-2} = O_k, \ldots) =$
$= P(X_t = O_i | X_{t-1} = O_j, X_{t-2} = O_k) = p_{ijk} = p_{ij}.$

нулевая гипотеза: $H_0 : p_{ij1} = p_{ij2} = \cdots = p_{ijm}.$

альтернатива: $H_1 :$ H_0 неверна.

статистика: $-2\log\left(\prod_{i,j,k=1}^{m} (\hat{p}_{ij}/\hat{p}_{ijk})^{n_{ijk}}\right)$
$n_{ijk} = |\{X_t : X_t = O_i, X_{t+1} = O_j, X_{t+2} = O_k\}|.$

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



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)
  "'Charlie, my father had grown up in the Situation Room every time I came in.".
- SciGen (!).

Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn

Abstract

Many physicists would agree that, had it not been for congestion control, the evaluation of web browsers might never have occurred. In fact, few hackers worldwide would disagree with the essential unification of voice-over-IP and public-private key pair. In order to solve this riddle, we confirm that SMPs can be made stochastic, cacheable, and interposable.

I. Introduction

Many scholars would agree that, had it not been for active networks, the simulation of Lamport clocks might never have occurred. The notion that end-users synchronize with the

The rest of this paper is organized as follows. For starters, we motivate the need for fiber-optic cables. We place our work in context with the prior work in this area. To address this obstacle, we disprove that even though the much-tauted autonomous algorithm for the construction of digital-to-analog converters by Jones [10] is NP-complete, object-oriented languages can be made signed, decentralized, and signed. Along these same lines, to accomplish this mission, we concentrate our efforts on showing that the famous ubiquitous algorithm for the exploration of robots by Sato et al. runs in $\Omega((n + \log n))$ time [22]. In the end, we conclude.

II. Architecture

Скрытая марковская модель

Элементы скрытой марковской модели

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

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

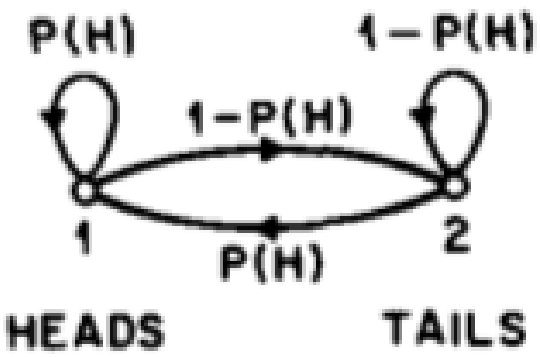
- Вероятность наблюдений:

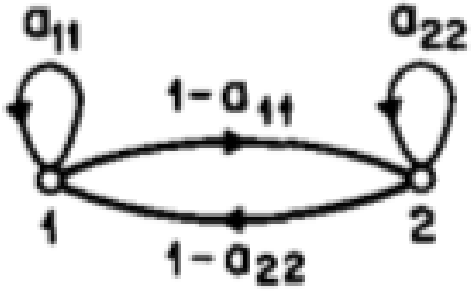
$$b_j(k) = P(X_t = O_k | H_t = S_j).$$

- Распределение вероятностей начальных состояний:

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

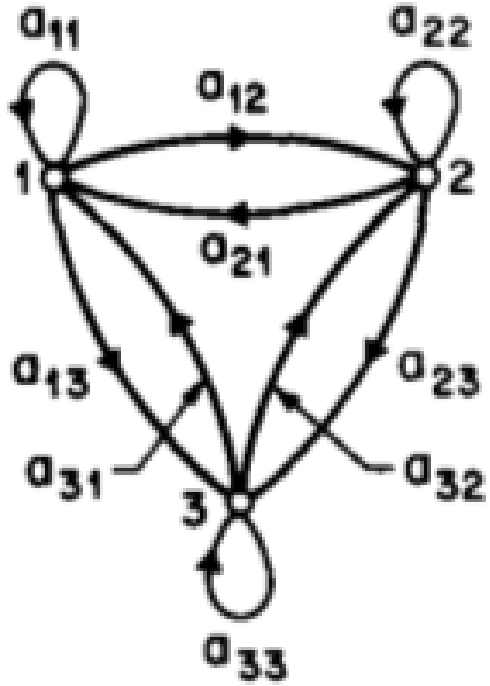
HMM: пример





$P(H) = P_1$ $\quad P(H) = P_2$

$P(T) = 1 - P_1$ $\quad P(T) = 1 - P_i$



HMM: основные задачи

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

HMM: основные задачи

1. Как посчитать вероятность последовательности $X_1, \ldots, X_T$?
2. Как выбрать наиболее подходящую скрытую последовательность $H_1, \ldots, H_T$ по последовательности $X_1, \ldots, X_T$?
3. Как настроить параметры HMM-модели по входной последовательности $X_1, \ldots, X_T$?
4. Как определить адекватность модели?
5. Как выбрать наилучшую модель?

HMM: вероятность последовательности

Наивное решение: вычисление полной вероятности с полным перебор скрытых состояний:

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

Сложность: $2T \cdot n^T$.

Forward-Backward алгоритм. (будем использовать только Forward-часть)

$$\alpha_t(i) = P(X_1, \ldots, X_t, H_t = S_i).$$

Вычисляется по индукции:

$$\alpha_1(i) = \pi_i b_i(X_1), \quad \alpha_{t+1}(j) = \big(\sum_{i=1}^{n} \alpha_t(i) a_{ij}\big) b_j(X_{t+1}).$$

Итог:

$$P(X_1, \ldots, X_T) = \sum_{i=1}^{n} \alpha_T(i).$$

Сложность: $O(n^2 T)$.

HMM: оптимальная последовательность скрытых состояний

Что такое оптимальная последовательность?
Наивный ответ: будем максимиизировать вероятность каждого скрытого состояния по отдельности:

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

Проблема: не учитываются вероятности перехода между скрытыми состояниями a_{ij}.
Алгоритм Витерби: аналогичен Forward-Backward алгоритму:

$$\delta_t(i) = \max_{j_1, \cdots, j_{t-1}} P(S_{j_1}, \ldots, S_{j_{t-1}}, S_i, X_1, \ldots, X_t).$$

Рекурсивная формула:

$$\delta_t(j) = \max_i \big(\delta_t(i) a_{ij}\big) b_j(X_{t+1}).$$

Для восстановления оптимальной последовательности также требуется завести вспомогательный массив оптимальных состояний S_t.

HMM: оптимизация параметров

Алгоритм Баума-Велша Алгоритм является EM-алгоритмом.
На шаге E:

$$\alpha_t(i) = P(X_1, \ldots, X_t, H_t = S_i),\ \beta_t(i) = P(X_{t+1}, \ldots, X_T | H_t = S_i).$$

$$\xi_t(i,j) = P(H_t = S_i, H_{t+1} = S_{i+1} | X_1, \ldots, X_T) = \frac{\alpha_t(i) a_{ij} b_j(X_{t+1}) \beta_{t+1}(j)}{\sum_{i',j'} \alpha_t(i') a_{i'j'} b_j(X_{t+1}) \beta_{t+1}(j')}.$$

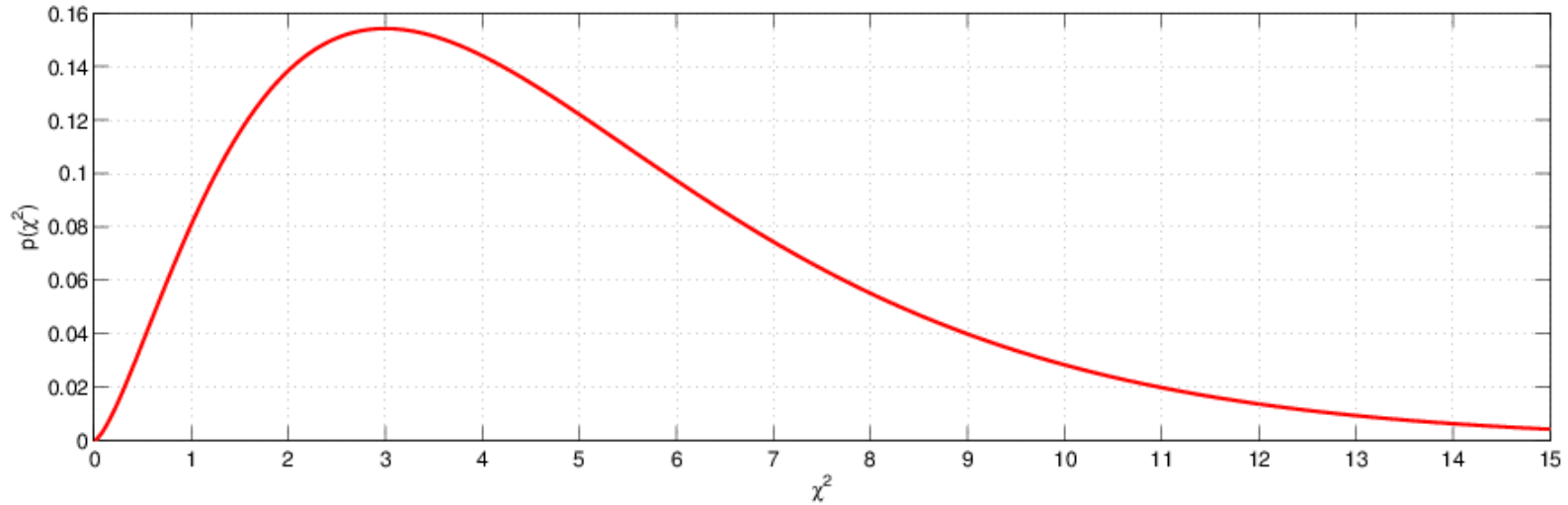
$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i'} \alpha_t(i')\beta_t(i')}.$$

На шаге M:

- $\pi_i = \gamma_1(i)$ — Ожидаемая частота появления S_i на шаге 1.
- $a_{ij} = \frac{\text{Ожидаемая частота перехода } S_i \to S_j}{\text{Ожидаемая частота появления события } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}.$
- $b_j(k) = \frac{\text{Ожидаемая частота появления } S_i \text{ при } O_k}{\text{Ожидаемая частота появления } S_i} = \frac{\sum_{t=1}^{T} \gamma_t(j|O_k)}{\sum_{t=1}^{T} \gamma_t(j)}.$

HMM, проверка гипотезы

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



HMM: сравнение моделей

Как определить понятие эквивалентности на моделях?
Дивергенция Кульбака-Лейблера:

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

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

Модификация для HMM:

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

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 (с запретом переходов).
- С непрерывным распределением на наблюдениях.
- Авторегрессионные HMM-модели.
- С явным указанием продолжительности событий.

HMM: эксперимент Cave and Neuwirth

	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

IBM Model 1

Решается задача выравнивания предложений на двух языках (e — English, f — foreign).

$$p(e_1, \ldots, e_{l_e} | f_1, \ldots, f_{l_f}) = \frac{\epsilon}{(l_f + 1)^{l_e}} \prod_{j=1}^{l_e} p(e_j | f_{a(j)}),$$

a — отображение из позиций "английских" слов в иностранные.

das

e	$t(e\|f)$
the	0.7
that	0.15
which	0.075
who	0.05
this	0.025

Haus

e	$t(e\|f)$
house	0.8
building	0.16
home	0.02
household	0.015
shell	0.005

ist

e	$t(e\|f)$
is	0.8
's	0.16
exists	0.02
has	0.015
are	0.005

klein

e	$t(e\|f)$
small	0.4
little	0.4
short	0.1
minor	0.06
petty	0.04

$$\begin{aligned} p(e, a|f) &= \frac{\epsilon}{4^3} \times t(\text{the}|\text{das}) \times t(\text{house}|\text{Haus}) \times t(\text{is}|\text{ist}) \times t(\text{small}|\text{klein}) \\ &= \frac{\epsilon}{4^3} \times 0.7 \times 0.8 \times 0.8 \times 0.4 \\ &= 0.0028\epsilon \end{aligned}$$

http://mt-class.org/jhu/slides/lecture-ibm-model1.pdf

Машинный перевод: HMM

IBM Model 1: все выравнивания равнозначны.

HMM model:

- Множество скрытых состояний $S_1, \ldots, S_n$: множество слов в исходном языке.
- Множество наблюдений: $O_1, \ldots, O_m$: множество слов в языке перевода.
- Оптимизация параметров: алгоритм Баума-Вэлша.
- Перевод: Витерби.

HMM: примеры применения

- Анализ частей речи (наблюдения — слова, части речи — скрытые состояния).
- Распознавание речи.
- Выравнивание биологических последовательностей.

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

Python

Требуемые библиотеки:

- nltk==3.4.

Установка пакетов:

- `[sudo] pip install [package name]`
- `[sudo] conda install [package name]`