МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

Л.Ю.Ковригина, И.А.Шилин

Руководство по выполнению лабораторных работ по теме “Вероятностные языковые модели”

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 09.04.04
в качестве учебно-методического пособия для реализации основных профессиональных образовательных программ высшего образования магистратуры

Санкт-Петербург

2018

УДК 81’32+004.8

Ковригина Л.Ю., Шилин И.А. Руководство по выполнению лабораторных работ по теме «Вероятностные языковые модели» – СПб: Университет ИТМО, 2018. – 33 с.

Рецензенты: Муромцев Д.И., к.т.н, доцент, заведующий кафедрой информатики и прикладной математики Университета ИТМО

В учебном пособии приводится базовая теория, примеры расчетов и лабораторных заданий, необходимых для разработки и применения вероятностных языковых моделей в задачах автоматической обработки естественного языка. В первой части рассматриваются *n*-граммы как средство языкового моделирования, во второй части – скрытые марковские модели и их применение для задач частеречной разметки. Указанные методы относятся к классическим подходам и могут быть использованы в процессе оценивания качества усовершенствованных моделей автоматической обработки естественного языка.

Учебное-пособие адресовано студентам высших учебных заведений, изучающим автоматическую обработку естественного языка.

При оформлении обложки использован пример из: URL: http://www.cse.unsw.edu.au/~billw/cs9414/notes/nlp/ambiguity/ambiguity-2009.html

УНИВЕРСИТЕТ ИТМО

Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2018

©Ковригина Л.Ю., Шилин И.А., 2018

Оглавление

Введение

В учебно-методическом пособии приводится базовая теория, примеры расчетов и лабораторных заданий, необходимых для разработки и применения вероятностных языковых моделей в задачах автоматической обработки естественного языка. В первой части рассматриваются *n*-граммы как средство языкового моделирования, во второй части – скрытые марковские модели и их применение для задач частеречной разметки. Указанные методы относятся к классическим подходам и могут быть использованы в процессе оценивания качества усовершенствованных моделей автоматической обработки естественного языка.

N-граммы и скрытые марковские модели широко применяются в задачах интеллектуального анализа текстов при разработке алгоритмов автоматического извлечения терминологии, алгоритмов частеречной разметки, шаблонов снятия омонимии, построении языковых моделей, извлечении формальных грамматик, извлечении моделей процессов из неструктурированных и структурированных данных и т.д.

N-граммы и скрытые марковские модели к настоящему времени вытесняются более эффективными методами анализа текста, однако, важно уметь их применять к поставленным задачам и использовать в качестве базового тестового уровня при оценке качества новых моделей и методов.

1. Методы разработки языковых моделей

1.1 Языковые модели. *N*-граммы

Модели, которые приписывают последовательности слов вероятность ее появления в тексте, называются языковыми моделями (*language models*, *LMs*) (Jurafsky, 2017). Можно моделировать и вероятность появления предложения в тексте, однако для этого нужны сверхбольшие объемы текстовых коллекций. Для многих практических задач крайне желательно иметь хорошую языковую модель, поэтому постоянно появляются новые методы создания языковых моделей. Лучшие результаты в настоящее время показывают рекуррентные нейронные сети с механизмом внимания, при этом им требуется меньшее количество контекстов при построении языковых моделей, чем классическим методам.

Задача языкового моделирования формально может быть сведена к вычислению вероятности появления слова w_i при условии, что до этого появилась цепочка слов $w_1 \ldots w_{i-1}$ (история). Например, для изречения Демокрита "*Враг не тот, кто наносит обиду, а тот, кто делает это преднамеренно*" вероятностная языковая модель должна уметь предсказывать, например, вероятность

P = (*преднамеренно* | *враг не тот, кто наносит обиду, а тот, кто делает это*)

Простые методы порождения вероятностных языковых моделей, к которым относятся *n*-граммы, вычисляют вероятность P напрямую, как

$$P = \frac{F(\text{враг не тот, кто наносит обиду, а тот, кто делает это преднамеренно})}{F(\text{враг не тот, кто наносит обиду, а тот, кто делает это})} \quad (1)$$

где F – абсолютная частота встречаемости выражения в корпусе текстов.

Однако такой способ расчета не всегда возможен, так как язык непрерывно меняется (появляются новые предложения, интересующее нас предложение может вообще отсутствовать в корпусе), а вычислительные затраты при расчете вероятности появления предложения *s* велики.

По этим причинам вычисление вероятности появления слова w_i при условии появления "истории" $w_1 \ldots w_{i-1}$ в *n*-граммных моделях заменяется вычислением "истории" на *k* предыдущих шагах. Так, для последовательности

из двух слов (биграммы) будет учитываться история только для одного предыдущего слова.

N*-граммой** (n*-gram**, *multi-word unit*, *MWU*) называется последовательность из *n* структурных единиц (токенов), на которые сегментирован исходный текст (слова, символы алфавита, числа, небуквенные символы и их последовательности и т.д.). *N*-граммы, извлеченные из коллекций текстовых документов, чаще всего состоят из словоформ, лексем или стем (псевдооснов). Однако, если исходные тексты собраны из социальных сетей, форумов, чатов и т.п. web-ресурсов, то в состав *n*-грамм могут входить эмотиконы, URI, фрагменты *html*-разметки и других метаданных и другие элементы, не относящиеся к лексической системе языка, на котором написаны анализируемые тексты.

Если рассматривать *n*-граммы как последовательности слов, то отдельное слово называется униграммой, два слова – биграммой (“ушел домой”), три – триграммой (“пришел в гости”) и т.д. Слова внутри *n*-граммы могут не иметь синтаксических связей между собой, единственное, что их связывает – это совместная встречаемость. Противопоставление по наличию/отсутствию структурной связи между элементами и является признаком, отличающим *n*-грамму от коллокации: *n*-грамма – (статистически устойчивая) последовательность из *n* соседних слов, а коллокация – устойчивое выражение (словосочетание), слова в котором связаны друг с другом, при этом они не обязательно располагаются друг за другом.

Пример ниже показывает разницу между *n*-граммой и коллокацией:

Предложение “Дальнейшие события припоминаю, как в тумане” содержит (с учетом того, что знаки пунктуации были удалены из текста)

5 биграмм:

- дальнейшие события
- события припоминаю
- припоминаю как
- как в
- в тумане

но только 2 коллокации:

- дальнейшие события
- в тумане

1.2 Как обнаружить статистически устойчивое сочетание?

К настоящему времени разработано множество методов извлечения *n*-грамм и коллокаций и проведена оценка этих методов (Banerjee and Pedersen

2003, Evert 2004, Baldwin 2004, Хохлова, Захаров, 2010, Baldwin and Kim, 2010). Для них существует общее название – меры ассоциативной связанности.

Меры ассоциативной связанности (*association measures*) – меры, вычисляющие силу связи между элементами в составе коллокации.

Параметрами функций для вычисления меры ассоциативной связанности являются чаще всего частота совместной встречаемости, частота слова в корпусе, размер корпуса и др., а значение функции следует интерпретировать как силу синтагматической связи между элементами *n*-грамм / словосочетаний.

Ни одна из мер ассоциативной связанности не является универсальной. Так, *MI* чувствительна к низкочастотным словам, а *t*-score полезна для нахождения высокочастотных коллокаций.

Ниже приведены формулы для расчета некоторых мер ассоциативной связанности (цит. по (Захаров, Хохлова, 2010):

1. Взаимная информация (*Mutual Information*, *MI*) рассчитывается по формуле:

$$MI(n,c) = \frac{log_2 f(n,c) \times N}{f(n) \times f(c)}; \quad (2)$$

2. *t*-score рассчитывается по формуле:

$$t-score = \frac{f(n,c) - \frac{f(n) \times f(c)}{N}}{\sqrt{f(n,c)}} \quad (3)$$

где *n* – ключевое слово (*node*); *c* – коллокат (*collocate*); *f(n,c)* – частота встречаемости ключевого слова *n* в паре с коллокатом; *f(n), f(c)* – абсолютные (независимые) частоты ключевого слова *n* и слова *c* в корпусе (тексте); *N* – общее число словоупотреблений в корпусе (тексте).

3. Мера максимального правдоподобия рассчитывается по формуле:

$$log-likelihood = 2 \sum_{ij} O_{ij} \times log \frac{O_{ij}}{E_{ij}} \quad (4)$$

где O_{ij}, E_{ij} – наблюдаемая и ожидаемая частоты.

Для извлечения и обработки *n*-грамм есть множество сервисов (Sketch Engine[1], Google *N*-gram Viewer[2]) и функций в библиотеках обработки текста (например, NLTK[3], Apache OpenNLP[4], SRILM[5], ngram package на языке R[6]).

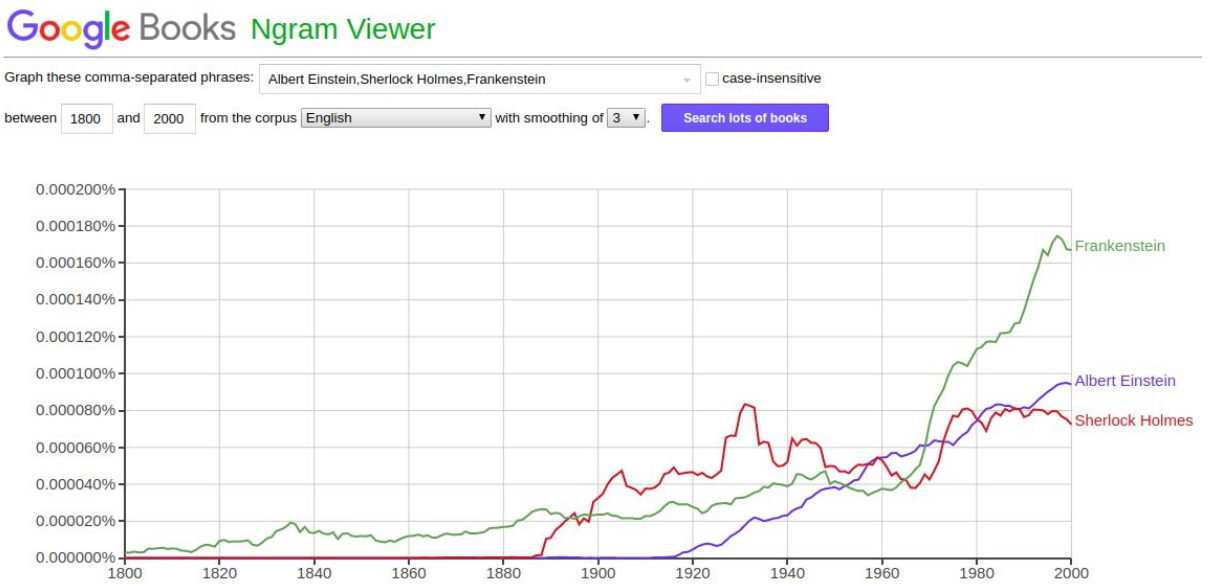

Рисунок 1 – Пример использования Ngram Viewer (Google Books)

Лабораторная работа №1. Функции для поиска *n*-грамм в библиотеке NLTK

Задание. Реализовать алгоритм расчета меры ассоциации для триграмм (в зависимости от доставшегося варианта) для веб-корпуса Aranea[7]:

1. словоформы, *MI*;
2. словоформы, *t*-score;
3. словоформы, *log-likelihood*;
4. лексемы, *MI*;
5. лексемы, *t*-score;
6. лексемы, *log-likelihood*.

Отчет. В файл с отчетом вывести 50 триграмм с наибольшим значением меры ассоциативной связанности.

Указания по выполнению лабораторной работы. В некоторых вариантах требуется предварительная лемматизация входных данных. Для проведения

[1] https://www.sketchengine.eu/
[2] https://books.google.com/ngrams
[3] https://www.nltk.org/
[4] https://opennlp.apache.org
[5] http://www.speech.sri.com/projects/srilm/
[6] https://cran.r-project.org/web/packages/ngram/index.html
[7] http://sketch.juls.savba.sk/aranea_about/index.html

лемматизации можно воспользоваться морфологическими анализаторами mystem, pymorphy2 или извлечь лексемы из разметки корпуса Aranea.

2 Скрытые марковские модели

2.1 Вводная теория по конечным автоматам и цепям Маркова

Абстрактный автомат – модель дискретного устройства, имеющего один вход, один выход и в каждый момент времени находящегося в одном состоянии из множества возможных. На вход этому устройству поступают символы одного алфавита, результатом работы абстрактного автомата является цепочка символов, генерируемых функцией выходов.

Конечный автомат – абстрактный автомат, число возможных внутренних состояний которого конечно.

Конечный автомат может задается упорядоченной пятеркой элементов некоторых множеств $K = (Q, T, \delta, q_0, F)$, где:

– Q – множество внутренних состояний;

– T – входной алфавит (конечное множество входных символов), из которого формируются входные слова, воспринимаемые конечным автоматом;

– δ – функция переходов, определенная как отображение $\delta : Q \times (T \cup \{\lambda\}) \to Q$, такое, что $\delta(q, a) = \{r : q \to_a r\}$, то есть значение функции переходов на упорядоченной паре (состояние, входной символ или пустая цепочка) есть множество всех состояний, в которые из данного состояния возможен переход по данному входному символу или пустой цепочке λ;

– q_0 – начальное состояние $(q_0 \in Q)$;

– F – множество заключительных, или конечных состояний $F \subset Q$ (Карпов, 2002).

Рассмотрим пример: конечный автомат, допускающий цепочки из 0 и 1, в которых имеется подцепочка 11 (Карпов, 2002):

$K = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\})$

Функция переходов:

$\delta(q_0, 0) = \{q_0\}, \delta(q_0, 1) = \{q_1\}, \delta(q_1, 0) = \{q_0\}$

$\delta(q_1, 1) = \{q_2\}, \delta(q_2, 0) = \{q_2\}, \delta(q_2, 1) = \{q_2\}$

Функция переходов может быть задана как таблица переходов или как диаграмма переходов. Если функция δ – однозначная, то конечный автомат

называется детерминированным. Если функция δ – многозначная, то конечный автомат называется недетерминированным.

Конечный преобразователь анализирует цепочку символов на входной ленте и записывает другую цепочку символов на выходной ленте. По определению $M = (Q, T, D, \delta, q_0, F)$, где:

– Q – множество внутренних состояний;

– T – алфавит входных символов;

– D – алфавит выходных символов;

– δ – функция переходов (отображение $Q \times T \to Q$;

– q_0 – начальное состояние ($q_0 \in Q$);

– F – множество заключительных, или конечных состояний ($F \subset Q$).

Цепь Маркова – последовательность случайных событий, в которой вероятность каждого события зависит только от состояния, в котором процесс находится в текущий момент и не зависит от более ранних состояний.

Конечная дискретная цепь определяется:

– множеством состояний $S = s_1, ..., s_N, s_t$ - состояние цепи в момент времени t;

– вектором начальных вероятностей (начальным распределением) $\pi = P(s_1 = i),\ 1 \le i \le N$, определяющим вероятности того, что в начальный момент времени $t = 1$ процесс находился в состоянии s_i;

– матрицей переходных вероятностей A; a_{ij} – вероятность перехода процесса с текущим состоянием s_i в следующее состояние s_j, при этом сумма вероятностей переходов из одного состояния равна 1:

$$\sum_{j=1}^{n} a_{ij} = 1$$ (Романовский, 2008, Huang, Acero, 2001).

2.2 Скрытая марковская модель

Скрытая марковская модель определяется пятеркой:

– алфавитом наблюдаемых символов $O = o_1, o_2, ..., o_M$;

– множеством состояний $S = s_1, ..., s_N$;

– матрицей переходных вероятностей A; a_{ij} – вероятность перехода из состояния i в следующее состояние j;

– матрицей вероятностей эмиссии B; $b_i(k)$ – вероятность появления наблюдаемого символа o_k, если модель находится в состоянии i;

– вектором начальных вероятностей (начальным распределением) $\pi = P(s_1 = i),\ 1 \le i \le N$.

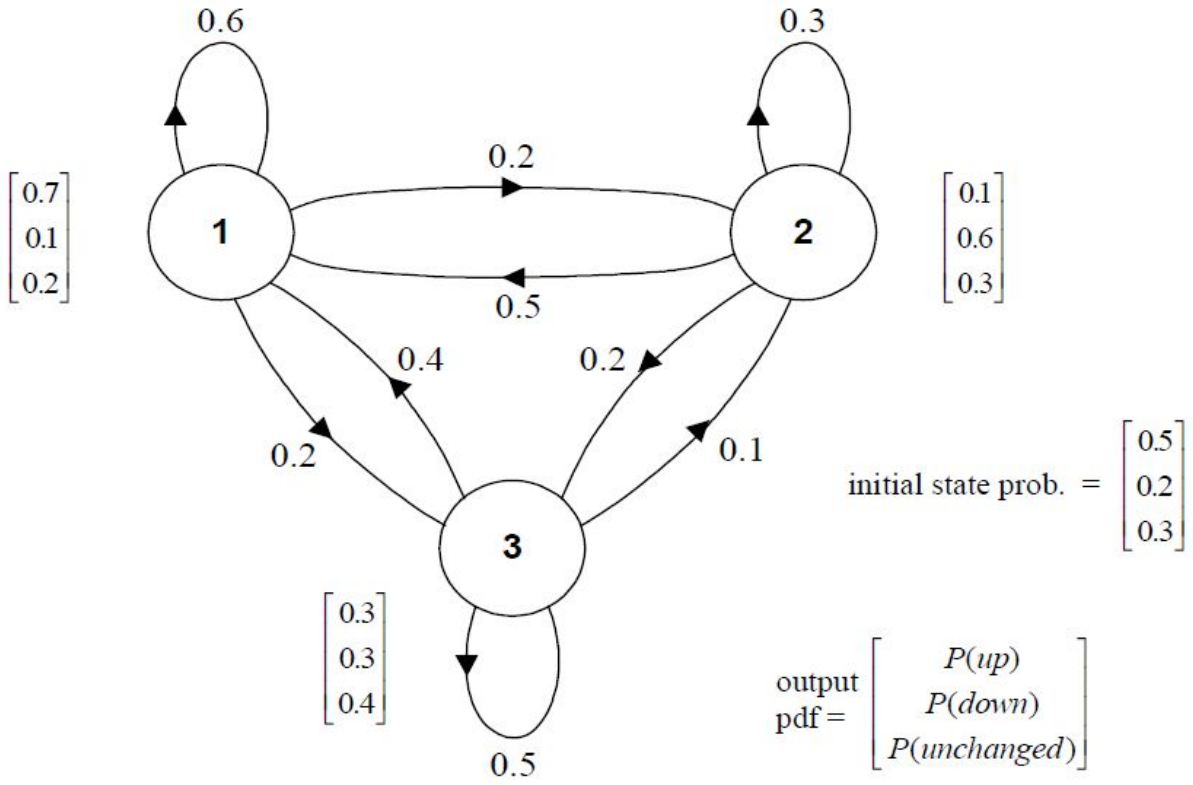

Рисунок 2. Пример скрытой марковской модели (из [Huang, Acero, 2001])

На рис. 2 (пример из монографии [Huang, Acero, 2001]) приведена скрытая марковская модель промышленного индекса Доу-Джонса. В конце дня рынок в зависимости от динамики промышленного индекса Доу-Джонса может находиться в одном из трех состояний *по отношению к значению индекса за предыдущий день*:

состояние **1** – рынок “быков” (индекс вырос),

состояние **2** – рынок “медведей” (индекс снизился),

состояние **3** – стабильный рынок (индекс не менялся).

Вероятности перехода из состояния в состояние определяются матрицей переходных вероятностей A :

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

и вектором начальных вероятностей: $\pi_1 = 0.5,\ \pi_2 = 0.2,\ \pi_3 = 0.3$.

Каждое из трех скрытых состояний может характеризоваться следующими наблюдаемыми (динамикой индекса Доу-Джонса): *up*, *down*, *unchanged*. Заданы следующие вероятности эмиссии наблюдаемых из каждого состояния (см. тж. рис.2):

$b_1(up) = 0.7,\ b_1(down) = 0.1,\ b_1(unchanged) = 0.2$;

$b_2(up) = 0.1,\ b_2(down) = 0.6,\ b_2(unchanged) = 0.3$;

$b_3(up) = 0.3,\ b_3(down) = 0.3,\ b_3(unchanged) = 0.4$.

3 Алгоритмы для скрытых марковских моделей

При применении скрытых марковских моделей выделяется три типа задач:

1) оценивание (evaluation): найти вероятность последовательности наблюдений в данной модели (алгоритм просмотра вперед);
2) декодирование (decoding): найти последовательность скрытых состояний, которая с наибольшей вероятностью породила последовательность наблюдаемых состояний (алгоритм Витерби);
3) обучение (learning): найти наиболее правдоподобную модель по последовательности наблюдаемых состояний (алгоритм Баума-Уэлша).

Декодирование и обучение скрытой марковской модели применяется в задаче построения языковых моделей и частеречной разметки. Когда морфологический анализатор учится приписывать входному предложению последовательность меток частей речи, слова считаются наблюдаемыми, а соответствующие им метки частей речи – скрытыми состояниями.

Рассмотрим алгоритмы для каждой из задач. Алгоритмы приводятся по (Rabiner, 1989, Huang, Acero, 2001).

3.1 Задача оценивания: алгоритм просмотра вперед

Задача оценивания скрытой марковской модели сводится к вычислению вероятности $P = P(X|\Phi)$ последовательности наблюдений $X = (X_1, X_2, \ldots X_T)$ для заданной модели Φ.

Для этого нужно суммировать вероятности всех последовательностей скрытых состояний, которые могли породить данную последовательность наблюдений:

$$P(X|\Phi) = \sum_S P(S|\Phi)P(X|S,\Phi) \qquad (5)$$

Согласно марковскому предположению, вероятность последовательности состояний $S = (s_1, s_2, \ldots s_T)$ с начальным состоянием s_1 рассчитывается по формуле (6):

$$P(S|\Phi) = P(s_1|\Phi)\prod_{t=2}^{T} P(s_t|s_{t-1},\Phi) = a_{s_0 s_1} a_{s_1 s_2} \ldots a_{s_{T-1} s_T} \qquad (6)$$

Используя допущение о том, что вероятность эмиссии наблюдаемого символа зависит только от текущего состояния и не зависит от предыдущих наблюдений (output-independent assumption), рассчитаем вероятность эмиссии символа X в состоянии S для модели Φ:

$$P(X|S,\Phi) = P({X_1}^T|{S_1}^T,\Phi) = \prod_{t=1}^{T} P(X_t|s_t,\Phi) = \\ = b_{s_1}(X_1)b_{s_2}(X_2)...b_{s_T}(X_T) \quad (7)$$

$$P(X|\Phi) = \sum_S P(S|\Phi)P(X|S,\Phi) = \\ = \sum_S a_{s_0 s_1} b_{s_1}(X_1) a_{s_1 s_2} b_{s_2}(X_2)...a_{s_{T-1} s_T} b_{s_T}(X_T) \quad (8)$$

В формуле (8) приведен простой, но неэффективный способ вычислений. Если мы будем хранить промежуточные результаты, то уменьшим сложность вычислений до $O(N^2 T)$.

Введем понятие прямой вероятности (forward probability) $\alpha_t(i)$ – вероятности того, что модель, находясь в состоянии i в момент времени t, породила подцепочку наблюдаемых символов $X_1, X_2, ...X_t$. Теперь можно рассмотреть алгоритм просмотра вперед.

Алгоритм просмотра вперед (forward algorithm)

Шаг 1: Инициализация.

$\alpha_1(i) = \pi_i b_i(X_1) \qquad 1 \le i \le N$, где i - состояние, t - момент времени.

Шаг 2: Рекурсия.

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(X_t) \qquad 2 \le t \le T;\ 1 \le j \le N \quad (9)$$

Шаг 3: Завершение.

$$P(X|\Phi) = \sum_{i=1}^{N} \alpha_T(i).$$

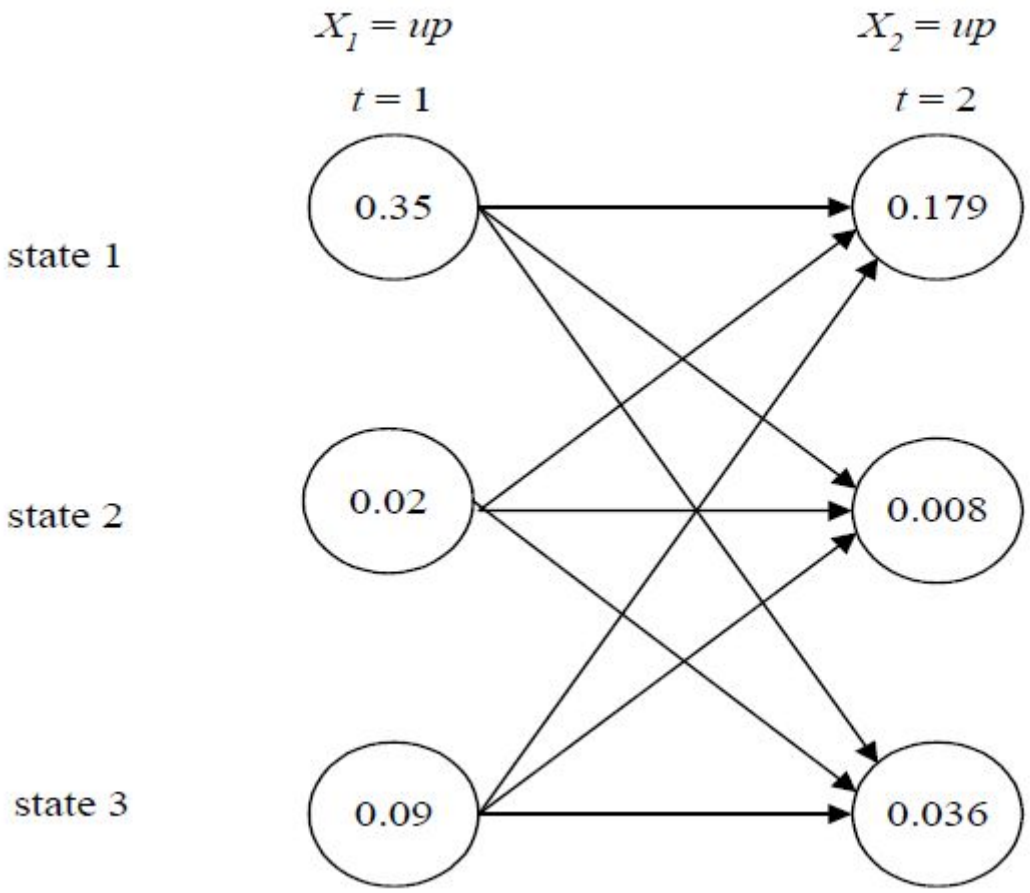

Рисунок 3. Прямые вероятности $\alpha_t(i)$ цепочки X = (*up*, *up*) в скрытой марковской модели, изображенной на рис.2

Практическое задание № 1. Алгоритм просмотра вперед

Задание. Какова вероятность порождения цепочки X = (*up*, *up*) моделью, изображенной на рисунке 2. Какова вероятность, что при порождении данной цепочки модель находилась в состоянии 1 в оба момента времени?

Решение.

1. Вычислим вероятности $\alpha_t(i)$ эмиссии символа X_1 каждым из скрытых состояний в момент времени $t = 1$:

$\alpha_1(1) = 0.5 \times 0.7 = 0.35$,

$\alpha_1(2) = 0.2 \times 0.1 = 0.02$,

Рассчитайте самостоятельно $\alpha_1(3)$. Ответ на рис. 3.

2. Вычислим рекуррентно вероятности $\alpha_t(i)$ эмиссии цепочки X_1^T = =(*up, up*) моделью, находящейся в скрытом состоянии j в момент времени t.

При $t = 2$:

$$\alpha_2(1) = \left[\alpha_1(1)a_{11} + \alpha_1(2)a_{21} + \alpha_1(3)a_{31}\right] \times b_1(X_2) =$$
$$= (0.35 \times 0.6 + 0.02 \times 0.5 + 0.09 \times 0.4) \times 0.7 = 0.179$$

Рассчитайте самостоятельно: $t = 2,\ \alpha_2(2) = \ ?$

Рассчитайте самостоятельно: $t = 2,\ \alpha_2(3) = ?$

3. Завершение алгоритма. В конечном состоянии $P(X|\Phi) = \alpha_T(s_F)$ вероятность порождения цепочки $X = (up, up)$ моделью, находившейся в оба момента времени в состоянии 1: $\alpha_2(s_F = s_1) = 0.179 / (0.179 + 0.008 + 0.036) = 0.8$

3.2 Задача декодирования: Алгоритм Витерби

Задача поиска оптимального пути. Алгоритм запоминает наилучший путь на каждом шаге $V_t(i)$ – вероятность наиболее возможной последовательности состояний в момент времени t, породившей последовательность наблюдений $X_1^{\,t}$.

Шаг 1: Инициализация.

$$V_1(i) = \pi_i b_i(X_1) \qquad 1 \le i \le N$$
$$B_1(i) = 0$$

Шаг 2: Рекурсия.

$$V_t(j) = Max_{1 \le i \le N} \left[V_{t-1}(i) a_{ij} \right] b_j(X_t)$$
$$\text{где } 2 \le t \le T;\ 1 \le j \le N \qquad (10)$$
$$B_t(j) = Arg\ max_{1 \le i \le N} \left[V_{t-1}(i) a_{ij} \right]$$
$$\text{где } 2 \le t \le T;\ 1 \le j \le N \qquad (11)$$

Шаг 3: Завершение.

Лучший результат $= Max_{1 \le i \le N} \left[V_t(i) \right]$

$$s_T{}^* = Arg\ max_{1 \le i \le N} \left[B_T(i) \right]$$

Шаг 4: Обратный проход.

Находим путь, который соответствует наибольшей вероятности:

$$s_T{}^* = B_{t+1}(s_{t+1}{}^*) \qquad t = T-1,\ T-2, ..., 1$$
$$S^* = (s_1{}^*,\ s_2{}^*,\ \ldots,\ s_T{}^*) \qquad \text{(лучшая последовательность)}$$

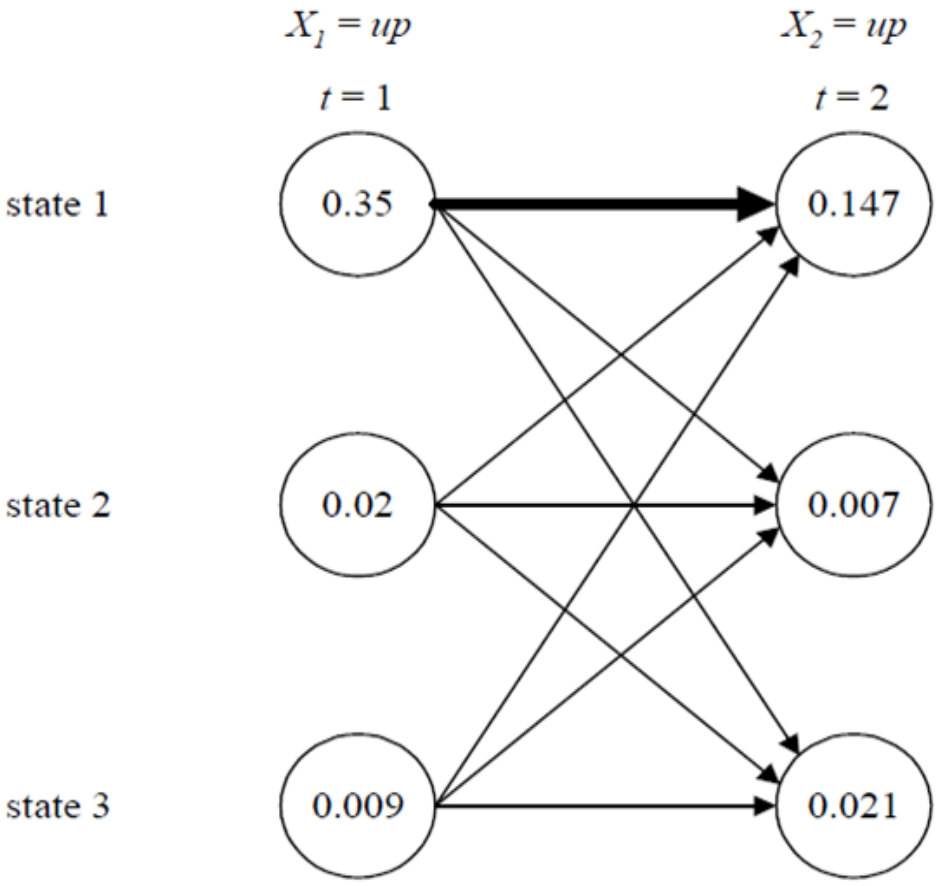

Рисунок 3. Иллюстрация алгоритма Витерби (пример из [Huang, Acero, 2001])

На рисунке 3 изображен путь, соответствующий наилучшей последовательности состояний, породившей цепочку $X = (up, up)$, см. условие к практическому заданию 1.

Рассчитаем $\alpha_1(1) = 0.49 \times 0.7 = 0.35$

Вероятность того, что элемент X_t в момент времени t порожден состоянием j : $V_t(j) = V_2(1) = 0.35 \times 0.6 \times 0.7 = 0.147$.

3.3 Задача обучения модели: Алгоритм Баума-Уэлша

Алгоритм Баума-Уэлша является частным случаем EM-алгоритма. При обучении скрытой марковской модели необходимо найти оптимальные параметры модели $\Phi(A, B, \pi)$.

Данная задача относится к аналитически неразрешимым, т.к. можно найти только локальный максимум при заданной конечной последовательности наблюдаемых, а не оптимальные параметры модели.

Нахождение локального максимума производится с помощью итеративной процедуры.

Введем для этого необходимые определения.

3.3.1. Обратная вероятность β

$$\beta_t(i) = P(X_{t+1}{}^T \mid s_t = i, \Phi),$$

где $\beta_t(i)$ – вероятность порождения подцепочки $X_{t+1}{}^T$, (т.е. от $t+1$ до последнего символа) при условии, что модель находилась в состоянии i в момент времени t (см. рисунок 5).

Формулы для расчета обратной вероятности β :

Шаг 1. Инициализация.

$\beta_i(T) = 1$; (произвольная инициализация, можно задать как $\beta_i(T) = 1/N$), где N – количество состояний, $1 \le i \le N$.

Шаг 2. Рекурсия.

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(X_{t+1}) \beta_{t+1}(j), \text{ где } t = T-1,\ T-2,\ \ldots,\ 1 \qquad (12)$$

3.3.2. Параметр γ

Параметр γ (13) дает оценку вероятности перехода из состояния i в состояние j в момент времени t при заданной модели Ф и порожденной последовательности наблюдаемых $X_1{}^T$:

$$\gamma_t(i,j) = P(s_t = i,\ s_{t+1} = j \mid X_1{}^T,\ \Phi) =$$

$$\frac{P(s_t = i,\ s_{t+1} = j,\ X_1{}^T \mid \Phi)}{P(X_1{}^T \mid \Phi)} = \frac{\alpha_t(i) \times a_{ij} \times b_j(X_{t+1}) \times \beta_{t+1}(j)}{\sum_{k=1}^{N} \alpha_T(k)} \qquad (13)$$

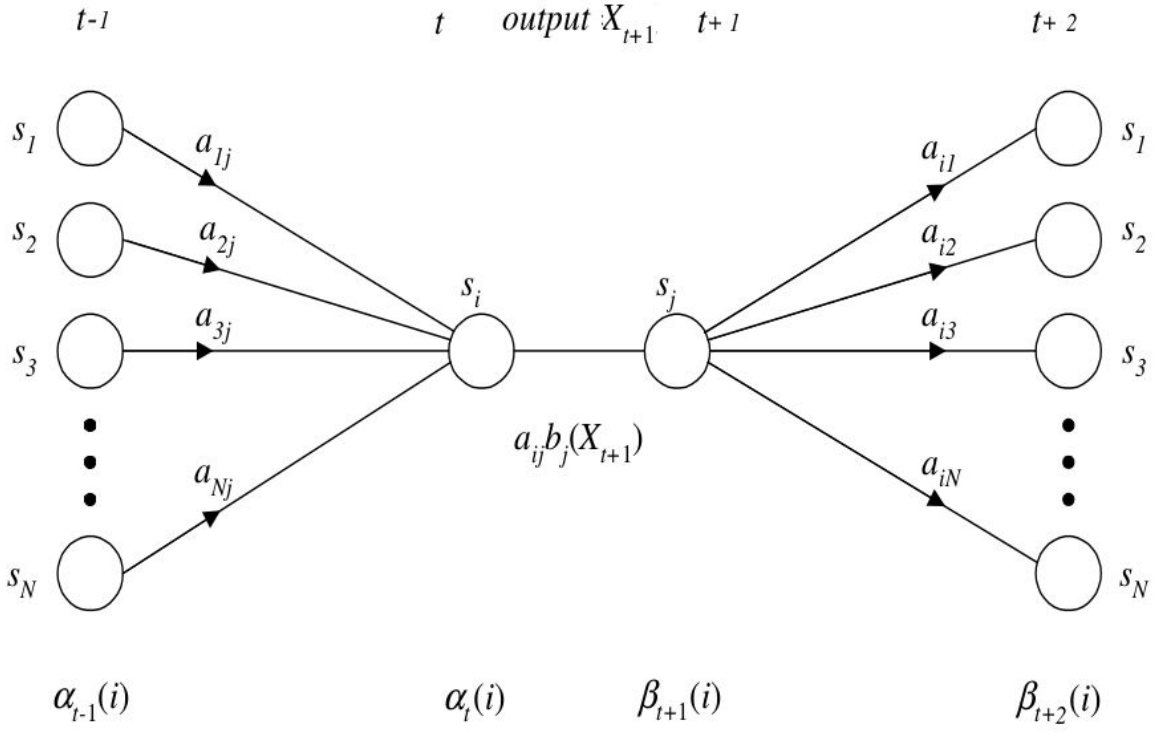

Рисунок 5. Иллюстрация формулы для расчета γ – вероятности перехода из состояния i в состояние j (Huang, Acero, 2001).

3.4 Оптимизация параметров модели

Задача: максимизировать значение Q - функции

$$Q(\Phi, \widehat{\Phi}) = Q_{a_i}(\Phi, \widehat{a}_i) + Q_{bj}(\Phi, \widehat{b}_j) \tag{14}$$

$$Q_{a_i}(\Phi, \widehat{a}_i) = \sum_i \sum_j \sum_t \frac{P(X, s_{t-1}=i, s_t=j \mid \Phi)}{P(X \mid \Phi)} log\, \widehat{a}_{ij} \tag{15}$$

$$Q_{b_j}(\Phi, \widehat{b}_j) = \sum_j \sum_k \sum_{t \in X_t = o_k} \frac{P(X, s_t=j \mid \Phi)}{P(X \mid \Phi)} log\, \widehat{b}_j(k) \tag{16}$$

Параметры модели на каждой итерации можно оценить по следующим формулам (17, 18):

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \gamma_t(i,k)} \tag{17}$$

$\widehat{a}_{ij}$ может быть проинтерпретирована как отношение количества переходов из состояния s_i в состояние s_j к общему количеству переходов из состояния s_i:

$$\widehat{a}_{ij} = \frac{\text{количество переходов из } s_i \text{ в } s_j}{\text{количество переходов из состояния } s_i}.$$

$$\widehat{b}_j(k) = \frac{\sum\limits_{t \in X_t = o_k} \sum\limits_{i} \gamma_t(i,j)}{\sum\limits_{t=1}^{T} \sum\limits_{i} \gamma_t(i,j)} \qquad (18)$$

$\widehat{b}_j(k)$ может быть проинтерпретирована как отношение частоты эмиссии наблюдаемой O_k из состояния j к общему количеству переходов в состояние j:

$$\widehat{b}_j(k) = \frac{\text{частота выбросов наблюдаемой } O_k \text{ из } j}{\text{количество переходов в } j}$$

3.5 Алгоритм Баума-Уэлша

Шаг 1: Инициализация.
Установить случайные параметры модели.

Шаг 2: *E*-шаг.
Вычислить значения функции $Q(\Phi, \widehat{\Phi})$, используя значения текущей модели Φ.

Шаг 3: *M*-шаг.
Вычислить параметры модели $\widehat{\Phi}$, используя уравнения (17) и (18), чтобы максимизировать значения Q-функции.

Шаг 4: Итерация.
$\Phi = \widehat{\Phi}$, вернуться на шаг 2, повторять вычисления до сходимости.

Лабораторная работа №2. Обучение скрытой марковской модели[8]

1. Эмпирические данные

Дан корпус c. В корпусе имеется Y слов:
$ABBA,\ BAB,\ F(ABBA) = 10,\ F(BAB) = 20$.

Количество (абсолютная частота) наблюдаемых последовательностей в корпусе:

$$\sum_{u \in Y} F(u) = 10 + 20 \ = \ 30 .$$

2. Первая скрытая марковская модель h_1

Инициализируем скрытую Марковскую модель с произвольными параметрами $\Phi_1(A, B, \pi)$, см. рис. 6.

Вектор начальных вероятностей: $\pi_s = 0.85,\ \pi_t = 0.15$.

Вероятности перехода из состояния i в состояние j:
$a_{ss} = 0.3,\ a_{st} = 0.7,\ a_{ts} = 0.1,\ a_{tt} = 0.9$

Вероятности эмиссии:
$b_s(A) = 0.4,\ b_s(B) = 0.6,\ b_t(A) = 0.5,\ b_t(B) = 0.5$.

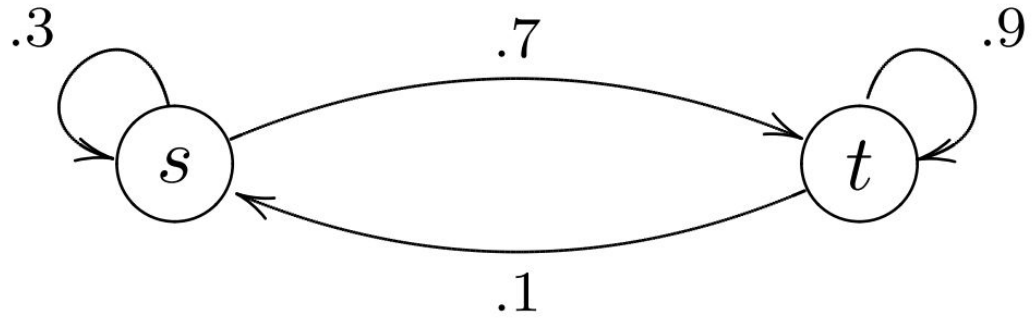

Рисунок 6. Скрытая марковская модель h_1 с произвольными параметрами

3. Прямая вероятность

Рассчитаем прямую вероятность: $\alpha_t(i) = P(X_1^{\ t},\ s_t = i \mid \Phi)$
Последовательность наблюдаемых $ABBA$.

[8] За основу взят пример из http://www.indiana.edu/~iulg/moss/hmmcalculations.pdf

$X_1 = A,\ \alpha_1(s) = \pi_s b_s(A) = 0.85 \times 0.4 = 0.34$

$X_1 = A,\ \alpha_1(t) = \pi_t b_t(A) = 0.15 \times 0.5 = 0.075$

$X_1^{\,2} = AB,$

$$\alpha_2(s) = [\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}]\ b_j(X_t) = \alpha_1(s) a_{ss} b_s(B) + \alpha_1(t) a_{ts} b_s(B) =$$

$$= 0.34 \times 0.3 \times 0.6 + 0.075 \times 0.1 \times 0.6 =$$

$$= 0.0612 + 0.0045 = 0.0657$$

Задание 3.1

Рассчитайте самостоятельно: $X_1^{\,2} = AB,\ \alpha_2(t) - ?$, $X_1^{\,3} = ABB,\ \alpha_3(s) - ?$, $\alpha_3(t) - ?$, $X_1^{\,4} = ABBA,\ \alpha_4(s) - ?,\ \alpha_4(t) - ?$

Ответы:

$X_1^{\,2} = AB,$

$\alpha_2(t) = 0.1528$

$X_1^{\,3} = ABB,$

$\alpha_3(s) = 0.021$

$\alpha_3(t) = 0.0917$

$X_1^{\,4} = ABBA,$

$\alpha_4(s) = 0.0062$

$\alpha_4(t) = 0.0486$

Задание 3.2

Рассчитайте самостоятельно все прямые вероятности α цепочки BAB.

Ответы:

$X_1 = B,$

$\alpha_1(s) = 0.51$

$\alpha_1(t) = 0.075$

$X_1^{\,2} = BA,$

$\alpha_2(s) = 0.0642$

$\alpha_2(t) = 0.2122$

$X_1^{\,3} = BAB,$

$\alpha_3(s) = 0.0243$

$\alpha_3(t) = 0.118$

Получаем общую вероятность для $ABBA$:

$\alpha(ABBA, 4, s) + \alpha(ABBA, 4, t) = 0.0062 + 0.0486 = 0.0548$

Получаем общую вероятность для BAB:

$\alpha(BAB, 3, s) + \alpha(BAB, 3, t) = 0.0243 + 0.118 = 0.1423$

Метод наибольшего правдоподобия – метод поиска модели, наилучшим образом описывающей обучающую выборку, полученную с некоторым неизвестным распределением.

Оценка максимального правдоподобия для корпуса c при модели Φ_1:

$L(c, \Phi) = Pr(ABBA)^{c(ABBA)} \times Pr(BAB)^{c(BAB)} = 0.0548^{10} \times 0.1423^{20}$

Логарифмируя, получим:

$log\ L(c, h_1) = 10 \times log\ 0.0548 + 20 \times log\ 0.1423 = -68.0370$

4. Обратная вероятность $\beta_t(i)$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(X_{t+1}) \beta_{t+1}(j)$$

Последовательность наблюдаемых $ABBA$.

$X_T{}^T,\ \beta_T(s) = 1$

$X_T{}^T,\ \beta_T(t) = 1$

$X_4{}^T = A,$

$\beta_3(s) = a_{ss} b_s(A) \beta_T(s) + a_{st} b_t(A) \beta_T(t) =$

$= 0.3 \times 0.4 \times 1 + 0.7 \times 0.5 \times 1 = 0.47$

$\beta_3(t) = a_{ts} b_s(A) \beta_T(s) + a_{tt} b_t(A) \beta_T(t) =$

$= 0.1 \times 0.4 \times 1 + 0.9 \times 0.5 \times 1 = 0.49$

$X_3{}^T = BA,$

$\beta_2(s) = a_{ss} b_s(B) \beta_3(s) + a_{st} b_t(B) \beta_3(t) =$

$= 0.3 \times 0.6 \times 0.47 + 0.7 \times 0.5 \times 0.49 = 0.2561$

$\beta_2(t) = a_{ts} b_s(B) \beta_3(s) + a_{tt} b_t(B) \beta_3(t) =$

$= 0.1 \times 0.6 \times 0.47 + 0.9 \times 0.5 \times 0.49 = 0.2487$

Задание 4.1

Вычислите самостоятельно $\beta_1(s)$, $\beta_1(t)$ для цепочки *ABBA*.

Ответы:

$X_2^T = BBA$,
$\beta_1(s) = 0.1331$
$\beta_1(t) = 0.1273$

Задание 4.2

Вычислите самостоятельно все необходимые значения $\beta_t(i)$ для цепочки *BAB*.

Ответы:

X_T^T, $\beta_T(s) = 1$
X_T^T, $\beta_T(t) = 1$
$X_3^T = B$,
$\beta_2(s) = 0.53$
$\beta_2(t) = 0.51$
$X_2^T = AB$,
$\beta_1(s) = 0.2421$
$\beta_1(t) = 0.2507$

5. Параметр γ

Параметр γ рассчитывается по формуле (13):

$$\gamma_t(i,j) = P(s_{t-1} = i, s_t = j \mid X_1^T, \Phi) =$$

$$= \frac{\alpha_t(i) \times a_{ij} \times b_j(X_{t+1}) \times \beta_{t+1}(j)}{\sum_{k=1}^{N} \alpha_T(k)}$$

$$\gamma_1(t,s) = \frac{\alpha_t(t) a_{ts} b_s(B) \beta_2^{ABBA}(s)}{Pr_{\Phi_1}(ABBA)} = \frac{0.0750 \times 0.1 \times 0.6 \times 0.2561}{0.0548} = 0.0210$$

Задание 5.1.

Рассчитайте самостоятельно все остальные необходимые значения параметра γ.

Ответы:

ABBA:

$\gamma_1(s,s) = 0.286$ (0.34*0.3*0.6*0.2561 / 0.0548)

$\gamma_1(s,t) = 0.5401$

$\gamma_1(t,t) = 0.1532$

$\gamma_2(s,s) = 0.1014$

$\gamma_2(s,t) = 0.2056$

$\gamma_2(t,s) = 0.0786$

$\gamma_2(t,t) = 0.6148$

$\gamma_3(s,s) = 0.0460$

$\gamma_3(s,t) = 0.1341$

$\gamma_3(t,s) = 0.0669$

$\gamma_3(t,t) = 0.7530$

BAB:

$\gamma_1(s,s) = 0.2279$

$\gamma_1(s,t) = 0.6397$

$\gamma_1(t,s) = 0.0112$

$\gamma_1(t,t) = 0.1210$

$\gamma_2(s,s) = 0.0812$

$\gamma_2(s,t) = 0.1579$

$\gamma_2(t,s) = 0.0895$

$\gamma_2(t,t) = 0.6710$

6. Оценка параметров модели

Вычислим оценку параметров $\widehat{a}_{ij}$, $\widehat{b}_j, \widehat{\pi}$ по формулам (13) и (14).

Оценка вероятности перехода из состояния *i* в состояние *j* рассчитывается по формуле :

$$\widehat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \gamma_t(i,k)} \qquad (13)$$

Оценка вероятности эмиссии наблюдаемой O_k из состояния j оценивается по формуле:

$$\widehat{b}_j(k) = \frac{\sum\limits_{t \in X_t = o_k} \sum\limits_i \gamma_t(i,j)}{\sum\limits_{t=1}^{T} \sum\limits_i \gamma_t(i,j)} \qquad (14)$$

В случае с разным количеством последовательностей:

$$\widehat{a}_{ij} = \frac{\sum\limits_{m=1}^{M} \sum\limits_{t=1}^{T_m - 1} \gamma_t{}^m(i,j)}{\sum\limits_{m=1}^{M} \sum\limits_{t=1}^{T_m - 1} \sum\limits_{k=1}^{N} \gamma_t{}^m(i,k)}$$

Оценим вероятность остаться на шаге $t+1$ в состоянии s ($\widehat{a}_{ss}$), формула (13):

$$\sum_{m=1}^{M} \sum_{t=1}^{T_m} \gamma_t{}^m(s,s) = F(ABBA) \times (\gamma_1{}^{ABBA}(s,s) + \gamma_2{}^{ABBA}(s,s) +$$

$$+ \gamma_3{}^{ABBA}(s,s)) + F(BAB) \times (\gamma_1{}^{BAB}(s,s) + \gamma_2{}^{BAB}(s,s)) =$$

$$= 10 \times (0.286 + 0.1014 + 0.046) + 20 \times (0.2279 + 0.0812) =$$
$$= 10,516$$

$$\sum_{m=1}^{M} \sum_{t=1}^{T_m} \sum_{k=1}^{N} \gamma_t{}^m(s) =$$
$$= F(ABBA) \times (\gamma_1{}^{ABBA}(s,s) + \gamma_2{}^{ABBA}(s,s) + \gamma_3{}^{ABBA}(s,s) +$$
$$+ \gamma_1{}^{ABBA}(s,t) + \gamma_2{}^{ABBA}(s,t) + \gamma_3{}^{ABBA}(s,t)) + F(BAB) \times$$
$$\times (\gamma_1{}^{BAB}(s,s) + \gamma_2{}^{BAB}(s,s) + \gamma_1{}^{BAB}(s,t) + \gamma_2{}^{BAB}(s,t)) =$$

$$= 10 \times (0.286 + 0.1014 + 0.046 + 0.5401 + 0.2056 + 0.1341)$$
$$+ 20 \times (0.2279 + 0.0812 + 0.6397 + 0.1579) = 35,266$$

$$\widehat{a}_{ss} = \frac{\sum_{m=1}^{M}\sum_{t=1}^{T_m-1}\gamma_t{}^m(s,s)}{\sum_{m=1}^{M}\sum_{t=1}^{T_m-1}\sum_{k=1}^{N}\gamma_t{}^m(s)} = \frac{10{,}516}{35{,}266} = 0.2982$$

$$\widehat{a}_{st} = 1 - \widehat{a}_{ss} = 0.7018$$

Задание 6.1

Закончите самостоятельно расчеты $\widehat{a}$.

Ответы:

$\widehat{a}_{ts} = 0.1059$

$\widehat{a}_{tt} = 0.8941$

Оценим вероятности эмиссии наблюдаемых *A* и *B* по формуле (14):

$$\widehat{b}_j(k) = \frac{\sum_{m=1}^{M}\sum_{t\in X_t=o_k}\sum_i \gamma_t(i,j)}{\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_i \gamma_t(i,j)}$$

Вероятность эмиссии наблюдаемой *A* из состояния *s*:

$$\sum_{m=1}^{M}\sum_{t\in X_t=o_k}\sum_i \gamma_t(i,j) =$$

$$= F(ABBA)\times(\gamma_1{}^{ABBA}(s,s) + \gamma_1{}^{ABBA}(t,s) + \gamma_4{}^{ABBA}(s,s) + \gamma_4{}^{ABBA}(t,s)) + F(BAB)\times(\gamma_2{}^{BAB}(s,s) + \gamma_2{}^{BAB}(t,s)) =$$

$$= 10\times(0.286 + 0.021 + 0.1131) + 20\times(0.0812 + 0.0895) = 7,615$$

Примечание: $\gamma_4{}^{ABBA}(s,s) + \gamma_4{}^{ABBA}(t,s)$ рассчитано как :

$$\gamma_4{}^{ABBA}(s,s) + \gamma_4{}^{ABBA}(t,s) = \alpha_4(s)/Pr(ABBA) = \frac{0.0062}{0.0548} = 0.1131$$

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{i}\gamma_t(i,j) = F(ABBA)\times(\gamma_1{}^{ABBA}(s,s) + \gamma_1{}^{ABBA}(t,s) +$$
$$+\gamma_2{}^{ABBA}(s,s) + \gamma_2{}^{ABBA}(t,s) + \gamma_3{}^{ABBA}(s,s) + \gamma_3{}^{ABBA}(t,s) +$$
$$+\gamma_4{}^{ABBA}(s,s) + \gamma_4{}^{ABBA}(t,s)) + F(BAB)\times(\gamma_1{}^{BAB}(s,s) +$$
$$+\gamma_1{}^{BAB}(t,s) + \gamma_2{}^{BAB}(s,s) + \gamma_2{}^{BAB}(t,s) +$$
$$+\gamma_3{}^{BAB}(s,s) + \gamma_3{}^{BAB}(t,s)) = 18.742$$

$\widehat{b}_s(A) = \frac{7,615}{18.742} = 0.4063$
$\widehat{b}_s(B) = 1 - \widehat{b}_s(A) = 0.5937$

Задание 6.2

Закончите дальнейшие расчеты.

Ответы:

$\widehat{b}_t(A) = 0.3985$
$\widehat{b}_t(B) = 0.6015$

Вычислим вектор начальных вероятностей $\widehat{\pi}$:

$$\pi = P(s_1 = i),\ 1 \le i \le N$$

$$\widehat{\pi}_s = \frac{\sum_{m=1}^{M}\sum_{j}\gamma_1(s,j)}{\sum_{m=1}^{M}\sum_{ij}\gamma_1(i,j)}$$

$$\sum_{m=1}^{M}\sum_{j}\gamma_1(s,j) = F(ABBA)\times(\gamma_1{}^{ABBA}(s,s) + \gamma_1{}^{ABBA}(s,t)) +$$
$$+F(BAB)\times(\gamma_1{}^{BAB}(s,s) + \gamma_1{}^{BAB}(s,t)) =$$
$$= 10\times(0.286 + 0.5401) + 20\times(0.2279 + 0.6397) = 25,613$$

$$\sum_{m=1}^{M}\sum_{ij}\gamma_1(i,j) = F(ABBA)\times(\gamma_1{}^{ABBA}(s,s) + \gamma_1{}^{ABBA}(s,t) +$$
$$+\gamma_1{}^{ABBA}(t,s) + \gamma_1{}^{ABBA}(t,t)) + F(BAB)\times(\gamma_1{}^{BAB}(s,s) +$$
$$+\gamma_1{}^{BAB}(s,t) + \gamma_1{}^{BAB}(t,s) + \gamma_1{}^{BAB}(t,t)) =$$
$$= 10\times(0.286 + 0.5401 + 0.021 + 0.1532) +$$
$$+20\times(0.2279 + 0.6397 + 0.0112 + 0.121) = 29,999$$

$$\widehat{\pi}_s = \frac{25{,}613}{29{,}999} = 0,8538$$
$$\widehat{\pi}_t = 1 - \widehat{\pi}_s = 1 - 0,8538 = 0.1462$$

Полученная модель h_2:

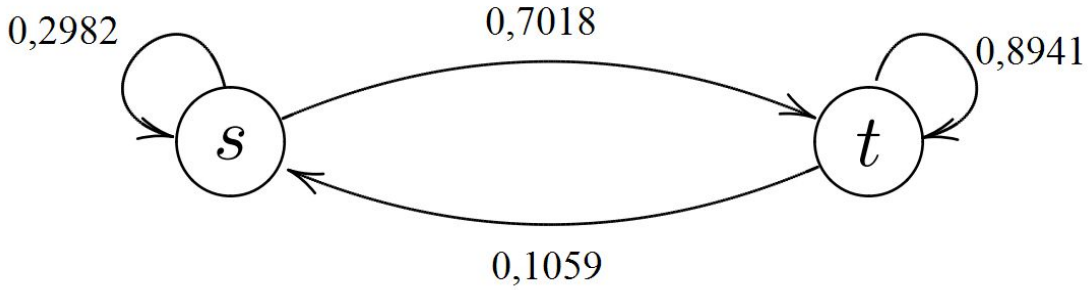

Рисунок 7. Скрытая марковская модель h_2 с уточненными параметрами после первой итерации алгоритма Баума-Уэлша

Вектор начальных вероятностей: $\pi_s = 0.8538,\ \pi_t = 0.1462$.
Вероятности перехода из состояния i в состояние j:
$a_{ss} = 0.2982,\ a_{st} = 0.7018,\ a_{ts} = 0.1059,\ a_{tt} = 0.8941$
Вероятности эмиссии:
$b_s(A) = 0.4063,\ b_s(B) = 0.5937,\ b_t(A) = 0.3985,\ b_t(B) = 0.6015$.
$L(c,\ h_2) = -67.3369$

Уточнение параметров модели производится итеративно.

Задание 7

Вычислите параметры модели h_3 и значение *log-likelihood* для эмпирических данных после второй итерации алгоритма Баума-Уэлша.

Ответы: см. ниже, см. рисунок 8.

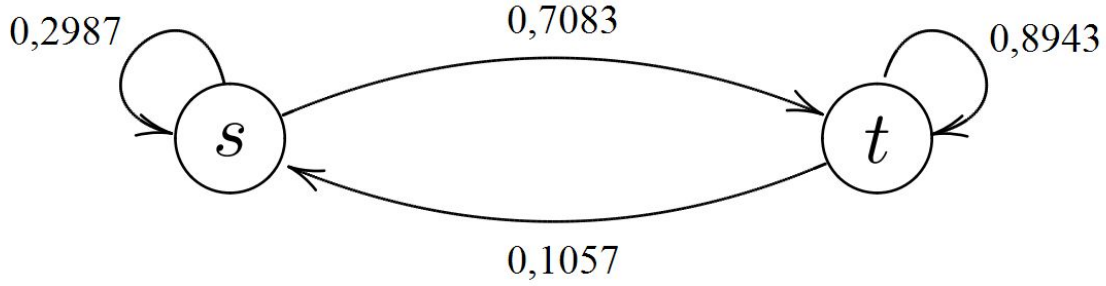

Рисунок 8. Параметры скрытой Марковской модели h_3 после второй итерации алгоритма Баума-Уэлша

Вектор начальных вероятностей модели h_3 : $\pi_s = 0.8537,\ \pi_t = 0.1463$.

Вероятности перехода из состояния i в состояние j :

$a_{ss} = 0.2987,\ a_{st} = 0.7013,\ a_{ts} = 0.1057,\ a_{tt} = 0.8943$

Вероятности эмиссии:

$b_s(A) = 0.3838,\ b_s(B) = 0.6162,\ b_t(A) = 0.4037,\ b_t(B) = 0.5963$.

$L(c,\ h_3) = -\ 67.2702$

Лабораторная работа № 3. Частеречная разметка с помощью скрытых марковских моделей

Задание. Обучить морфологический анализатор присваивать входному тексту метки частей речи, используя алгоритм Баума-Уэлша из библиотеки автоматической обработки естественного языка NLTK[9].

Отчет. Демонстрация работы алгоритма на произвольном предложении.

Указания по выполнению лабораторной работы.

В классе HiddenMarkovModelTrainer реализованы алгоритм обучения с учителем с помощью метода максимального правдоподобия (train_supervised) и без учителя – алгоритм Баума-Уэлша (train_unsupervised).

Пример обучения HMM-тэггера на Брауновском корпусе в hmm_trainer.py:

demo_pos – обучение скрытой марковской модели и оценка качества;

demo_pos_bw – алгоритм Баума-Уэлша для частеречной разметки;

demo_bw – алгоритм Баума-Уэлша.

Необходимо подготовить корпус для обучения тэггера (обучающая выборка) и его оценки (тестовая выборка) и, используя один из примеров обучения HMM-тэггера, обучить модель на своем корпусе, затем оценить качество работы на тестовой выборке.

Можно использовать любой корпус, подходящий по формату входных данных: можно сформировать корпус самостоятельно для русского/английского языков или использовать уже существующий (кроме того, что используется в примере). В зависимости от сложности подготовительной работы при сдаче задания будет применяться повышающий коэффициент.

[9] http://www.nltk.org/_modules/nltk/tag/hmm.html.

Литература

1. Карпов, В.Э. Классическая теория компиляторов [Текст]: учеб. пособие / В.Э. Карпов; М-во образования РФ, Моск. гос. ин-т электрон. и математики (техн. ун-т). – М.: МГИЭМ, 2002. – 78 с.
2. Романовский И. В. Дискретный анализ. — СПб.: Невский Диалект; БХВ-Петербург, 2003. — 338 с.
3. Huang, Xu, Acero, A., Hon, Hs. Spoken Language Processing: A Guide to Theory, Algorithm, and System Development (1st ed.). — 2001. — 960 с.
4. Kempe A. Finite State Transducers Approximating Hidden Markov Models. — 1997. — URI: http://www.aclweb.org/anthology/P97-1059 (дата обращения 07.04.2018)
5. Jurafsky, D., Martin, J.H. Language Modeling with N-grams // Speech and Language Processing. Draft of August 7, 2017. — URI: https://web.stanford.edu/~jurafsky/slp3/4.pdf (дата обращения 07.04.2018).
6. Захаров В.П., Хохлова М.В. Анализ эффективности статистических методов выявления коллокаций в текстах на русском языке // Труды конференции “Диалог”, 2009.
7. Rabiner L. R. A tutorial on hidden Markov models and selected applications in speech recognition //Proceedings of the IEEE. – 1989. – Т. 77. – №. 2. – С. 257-286.

Заключение

В учебно-методическом пособии рассмотрены средства вероятностного языкового моделирования: *n*-граммы и скрытые марковские модели. *N*-граммы и скрытые марковские модели широко применяются в задачах интеллектуального анализа текстов при разработке алгоритмов автоматического извлечения терминологии, алгоритмов частеречной разметки, построении языковых моделей, извлечении формальных грамматик, извлечении моделей процессов из неструктурированных и структурированных данных и т.д.

N-граммы и скрытые марковские модели к настоящему времени вытесняются более эффективными методами анализа текста, однако, важно уметь их применять к поставленным задачам и использовать в качестве базового тестового уровня при оценке качества новых моделей и методов.

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

К середине 70-х годов в ЛИТМО активно развиваются работы по созданию систем автоматизации проектирования. Резко возрастает потребность в кадрах, способных эффективно применять вычислительную технику в различных областях наук и производства. В связи с этим в 1976 году из кафедры Вычислительной техники выделяется кафедра Прикладной математики, на которую возлагается задача по подготовке специалистов в области программирования и методов вычислений. Кафедру возглавляет д.т.н., проф. О.Ф. Немолочнов, работающий в области систем автоматизации проектирования ЭВМ. На кафедре работают: д.т.н., проф. Я.М. Цейтлин, специализирующийся в области машинного эксперимента.

Научное направление кафедры состояло в разработке регулярных методов проектирования тестов для логических схем ЭВМ на основе аппарата исчисления кубических комплексов. В результате выполнения ряда НИР были созданы САПР тестов для плат бортовых ЦВМ, стендовое оборудование для контроля, диагностики и наладки цифровых схем в виде двухслойных и многослойных плат ЭВМ. Сотрудничество с академическими организациями в участии в ежегодных школах-семинарах по технической диагностике, проводимых под руководством члена-корреспондента АН СССР Пархоменко П.П. (ИПУ г. Москва) отраслевыми НИИ (НИЦЭВТ г. Москва), конструкторскими бюро (КБЭ г. Харьков) и промышленными предприятиями (НПО ВТ г. Минск) позволило интегрировать научные исследования с последующим внедрением результатов в производственные технологии и учебный процесс в единое целое.

В результате научных исследований были подготовлены кадры высшей квалификации: были подготовлены и защищены 5 докторских и несколько десятков кандидатских диссертаций. После защиты сотрудники НИЧ нашего института, как правило, переходили на преподавательскую работу. В частности, по кафедре ИПМ защитили диссертации и стали преподавателями: Шипилов П.А., Голованевский Г.Л., Блохин В.Н., Усвятский А.Е., Звягин В.Ф.,

Голыничев В.Н., Щупак Ю.А., Кукушкин Б.А., Раков С.В., Слоев Б.А., Павловская Т.А., Денисова Э.В. и ряд других.

Кафедра занимается разработкой теоретических основ программирования в области моделирования, верификации, тестирования и диагностики вычислительных процессов программ. Вычислительные процессы, порождаемые программами, представляются и описываются графо-аналитическими моделями (ГАМ) в виде множества вершин и дуг связи между ними. ГАМ строится на основе концептуальной двухконтурной итерационно-рекурсивной модели (IRM), позволяющей описывать как ациклические, так и циклические вычислительные процессы.

Математическое описание ГАМ строится в виде кубических покрытий с использованием алгебро-топологического аппарата исчисления кубических комплексов. При построении кубических покрытий вычислительный процесс декомпозируется на множество параллельных структур с любым уровнем вложенности их друг в друга. Переход от программ к вычислительным процессам позволяет решать задачи проектирования программного продукта через верификацию в общем виде, т.е. без учета конкретных особенностей языков программирования, операционных систем и процессоров, которые могут быть как реальными, так и виртуальными.

Разрабатываемые методы являются детерминированными и являются составной частью любой технологии проектирования программного продукта. На основе разрабатываемых методов кафедра планирует создание учебно-исследовательской системы (УИС) в виде САПР, позволяющей унифицировать лабораторный практикум путем создания единой базы знаний и базы данных, повысить объективность и качество оценки знаний студентов, повысить производительность работы преподавателей.

В настоящее время кафедрой руководит доцент, к.т.н. Д.И.Муромцев. Список проектов и научных направлений опубликован на сайте кафедры http://iam.ifmo.ru/ru/listnaprav/show_all_naprav.htm.