УДК 004.934.11
DOI: 10.24160/1993-6982-2018-5-65-72

Быстрый алгоритм распознавания голосовых команд на основе стационарного распределения скрытой марковской модели

П.А. Парамонов, И.В. Огнев

За последние несколько десятилетий скрытые марковские модели (СММ) стали доминирующей технологией в автоматическом распознавании речи (АРР). Современные решения, основанные на СММ, используют гауссовы смеси для моделирования акустической вариативности речи. Применение в АРР глубоких нейронных сетей для построения акустических моделей позволило превзойти показатели качества, демонстрируемые моделями на основе гауссовых смесей в распознавании с большим словарем. Данные алгоритмы обладают чрезвычайно высокой вычислительной сложностью, что делает невозможным их реализацию в системах голосового управления для устройств с малой вычислительной мощностью.
Настоящая работа посвящена алгоритму распознавания изолированных слов с малой вычислительной сложностью. Описаны все компоненты распознавателя изолированных слов. В качестве признакового описания речевого сигнала использована последовательность квантованных мел-кепстральных коэффициентов (MFCC). Предложен быстрый алгоритм распознавания изолированных слов, основанный на стационарном распределении скрытой марковской модели. Он обладает линейной сложностью относительно длины наблюдаемой последовательности и требует значительно меньше памяти, чем модели на гауссовых смесях или глубоких нейросетях.
Выполнена оценка качества распознавания на корпусе TIMIT и сформированной базе русских слов. Доказано, что предложенный алгоритм слабо уступает по качеству распознавания моделям гауссовых смесей и превосходит по этому показателю самонастраиваемые нейронные сети.

Ключевые слова: скрытые марковские модели, голосовое управление, распознавание образов, алгоритм прямого хода.

Для цитирования: Парамонов П.А., Огнев И.В. Быстрый алгоритм распознавания голосовых команд на основе стационарного распределения скрытой марковской модели // Вестник МЭИ. 2018. № 5. С. 65—72. DOI: 10.24160/1993-6982-2018-5-65-72.

A Fast Voice Command Recognition Algorithm Based on the Hidden Markov Model Stationary Distribution

P.A. Paramonov, I.V. Ognev

Over the last few decades Hidden Markov Models (HMM) have become a dominating technology in automatic speech recognition (ASR) systems. Contemporary HMM-based solutions use Gaussian mixture models (GMM) for modeling acoustic speech variability. ASR algorithms involving acoustic models constructed with the use of deep neural networks (DNN) outperform GMMs in recognizing large-vocabulary speech. However, these algorithms feature extremely high computation complexity, due to which they cannot be applied in voice control systems with moderate computational resources.
An approach to developing an algorithm for recognizing isolated words with low computation complexity is considered. All components of the isolated word recognition engine are described. A sequence of quantized Mel-frequency cepstral coefficients (MFCC) is used as speech signal description features. A fast isolated words recognition algorithm constructed on the basis of a stationary distribution of the Hidden Markov model is described. The proposed algorithm is characterized by a linear complexity with respect to the observed sequence length and requires significantly less memory compared with algorithms on the basis of GMM or DNN models.

The algorithm's recognition performance is evaluated on TIMIT isolated words dataset and the base of Russian words that was set up by the authors. It has been demonstrated that the proposed algorithm shows recognition performance that is only slightly inferior to GMMs and superior to self-adjustment neural networks.

Key words: hidden Markov models, voice control, pattern recognition, forward algorithm.

For citation: Paramonov P.A., Ognev I.V. A Fast Voice Command Recognition Algorithm Based on the Hidden Markov Model Stationary Distribution. MPEI Vestnik. 2018;5:65—72. (in Russian). DOI: 10.24160/1993-6982-2018-5-65-72.

Введение

Задача распознавания речи чрезвычайно актуальна, поскольку ее решение позволит разрабатывать естественные человеко-машинные интерфейсы. Чаще всего под распознаванием речи понимают распознавание слитной речи. В этом направлении за последние годы достигнут существенный прогресс [1 — 4]. Предложенные алгоритмы на основе глубоких нейросетей демонстрируют наилучшие показатели качества распознавания, дикторонезависимость, способность работать с большими словарями, однако высокая вычислительная сложность не позволяет применять их в условиях ограниченных вычислительных ресурсов.

Распознавание голосовых команд является усеченным вариантом распознавания речи. Задача ставится следующим образом. Дан участок t цифрового речевого сигнала $X_t[n]$, содержащий произнесенную голосовую команду. Необходимо определить индекс поступившей команды из словаря, включающего в себя W различных команд. Решение подобной задачи имеет практическую ценность, поскольку модуль голосового управления является одним из ключевых компонентов информационных роботов и других диалоговых интерфейсов.

Предложен алгоритм распознавания голосовых команд на основе скрытых марковских моделей (СММ), не учитывающий порядок следования звуков.

Общая структура системы распознавания голосовых команд

Система распознавания голосовых команд (рис. 1) состоит из блока выделения признаков (нахождения последовательности MFCC), алгоритма обучения, выполняющего построение акустической модели λ, и алгоритма распознавания. MFCC используются для решения разнообразных задач, например, кластеризации пользователей по голосу или верификации

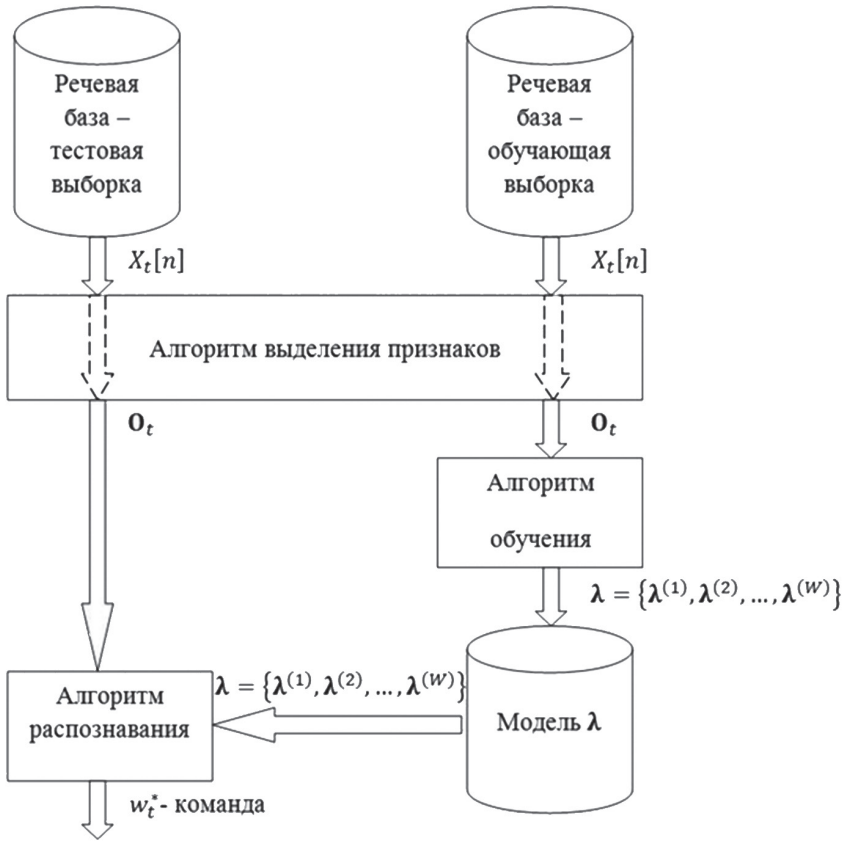


Рис. 1. Структура системы распознавания голосовых команд

пользователей в зашумленном окружении [5, 6]. В цифровом речевом сигнале $X[n]$ выделяют t-й участок $X_t[n]$, содержащий голосовую команду. Выделение признаков выполняется для каждого кратковременного окна i длительностью от 10 до 32 мс, $i = 1, ..., m$. При этом, вначале получают вектор $\mathbf{s}_t^{(i)} \in \mathbb{R}^{p+up}$ из p MFCC-коэффициентов и p их производных u-го порядка, а затем выполняют его квантование. Вектор признаков $s_t^{(i)}$ строится в несколько этапов [5, 7]:

- спектральное выравнивание:

$$\hat{X}_t[n] = X_t[n] - 0{,}95X_t[n-1];$$

- применение преобразования Фурье для перехода в частотную область;
- наложение мел-частотного фильтра;
- вычисление кепстра с помощью дискретного косинусного преобразования;
- расчет производных MFCC-коэффициентов.

Таким образом голосовая команда t описывается последовательностью векторов признаков $\mathbf{S}_t = \left(\mathbf{s}_t^{(1)}, \ldots, \mathbf{s}_t^{(m_t)}\right)$ длины m_t. Затем выполняется векторное квантование V векторов признаков $s_t^{(i)}$, так что каждый вектор признаков преобразуется в целое число $o_t^{(i)} \in [0, K-1]$, где K — количество слов в кодовой книге:

$$V : \mathbf{s}_t^{(i)} \mapsto o_t^{(i)}, \ \mathbf{s}_t^{(i)} \in \mathbb{R}^{p+up}, \ o_t^{(i)} \in \mathbb{Z}.$$

Таким образом, алгоритм выделения признаков возвращает последовательность наблюдаемых значений $\mathbf{O}_t = \left(o_t^{(1)}, \ldots, o_t^{(m_t)}\right)$ для голосовой команды t. Векторное квантование выполняется алгоритмом k-средних.

Алгоритм обучения строит модель речи λ, которая представляет собой набор моделей команд $\boldsymbol{\lambda} = \{\lambda^{(1)}, \lambda^{(2)}, ..., \lambda^{(W)}\}$, где $\lambda^{(W)}$ — модель одного слова из словаря. На этапе обучения строятся модели $\lambda^{(w)}$, $w = 1, ..., W$, каждая из которых представляет собой дискретную скрытую марковскую модель $\lambda^{(w)} = \{\boldsymbol{\pi}^{(w)}, \mathbf{A}^{(w)}, \mathbf{B}^{(w)}\}$ с $N^{(w)}$ состояниями, способную порождать K различных наблюдаемых значений, где $\boldsymbol{\pi}^{(w)} = \{\pi_i^{(w)}\}$, $i = 1, ..., N^{(w)}$ — вектор вероятностей начальных состояний; $\mathbf{A}^{(w)} = \{a_{ij}^{(w)}\}$, $i = 1, ..., N^{(w)}$, $j = 1, ..., N^{(w)}$, $\mathbf{B}^{(w)} = \{b_i^{(w)}(k)\}$, $i = 1, ..., N^{(w)}$, $k = 1, ..., K$ — матрицы вероятностей переходов и выходных вероятностей.

Алгоритм распознавания по последовательности признаков $\mathbf{O}_t$ с использованием модели $\boldsymbol{\lambda}$ возвращает индекс w_t^* произнесенной голосовой команды в словаре. Для этого находится вероятность порождения каждой из моделей рассматриваемой последовательности признаков. Та модель, для которой эта вероятность максимальна, объявляется «победителем» (моделью слова w_t^*):

$$w_t^* = \underset{w=1,\ldots,W}{\operatorname{argmax}} \left[P\left(\mathbf{O}_t \mid \lambda^{(w)}\right) \right].$$

Метод распознавания голосовых команд на основе стационарного распределения скрытой марковской модели

Предлагаемый алгоритм базируется на подходе к определению «победителя», предложенном в [8]. Рассмотрим скрытую марковскую модель $\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ и последовательность наблюдаемых признаков неизвестного слова $\mathbf{O} = (o^{(1)}, ..., o^{(m)})$, опустив индекс модели w и номер произнесенной команды t. Предложенный подход основан на марковском допущении, согласно которому следующее состояние Q_{n+1} определяется только текущим Q_n. Для дискретной марковской цепи, являющейся апериодичной и неприводимой, существует стационарное распределение $\mathbf{P} = \{P_j\}$, $j = 1, ..., N$, где P_j — вероятность пребывания в состоянии j [9]. Пусть $r_{ij}(n)$ — вероятность прихода в состояние j на шаге n при условии, что в начальный момент времени марковский процесс находился в состоянии i, тогда:

$$\lim_{n\to\infty} r_{ij}(n) = P_j.$$

Для нахождения стационарного распределения $\mathbf{P}$ следует решить систему линейных уравнений (balance equations [9]):

$$\begin{cases} \sum_{i=1}^{N} P_i a_{ij} - P_j = 0, \ j = 1, \ldots, N; \\ \sum_{j=1}^{N} P_j = 1. \end{cases}$$

Поскольку скрытая марковская модель обладает теми же свойствами, что и простая марковская цепь, для нее тоже можно найти вектор $\mathbf{P}$, определяющий вероятность обнаружить описываемый моделью процесс в каждом из N состояний. Однако, кроме этого СММ испускает наблюдаемые значения. Если известна вероятность пребывания процесса в состоянии j, то можно также найти вероятность $E(k)$ испускания символа k данной моделью. Она складывается из вероятностей пребывания СММ в состоянии j и испускания из этого состояния символа k:

$$E(k) = \sum_j P_j b_j(k), \quad k = 1, \ldots, K.$$

Предлагаемый алгоритм распознавания основывается на вычислении вероятности порождения скрытой марковской моделью набора наблюдаемых значений $\dot{\mathbf{O}} = \left\{o^{(1)}, \ldots, o^{(m)}\right\}$ без учета их порядка:

$$P\left(\dot{\mathbf{O}} \mid \lambda\right) = \prod_{i=1}^{m} E\left(o^{(i)}\right).$$

Удобно перейти к логарифму $P\left(\dot{\mathbf{O}}\,|\lambda\right)$:

$$\log P\left(\dot{\mathbf{O}}\,|\lambda\right)=\sum_{i=1}^{m}\log E\left(o^{(i)}\right).$$

Тогда распознавание по принципу «победитель забирает все» будет выполняться на основе сравнения «очков» $\log P\left(\dot{\mathbf{O}}|\lambda^{(w)}\right)$, которые получила каждая из W моделей:

$$w^{*}=\underset{w=1,\ldots,W}{\operatorname{argmax}}\left[\log P\left(\dot{\mathbf{O}}\,|\,\lambda^{(w)}\right)\right].$$

Алгоритм обучения

На этапе обучения выполняется построение модели речи $\boldsymbol{\lambda}=\{\lambda^{(1)},\lambda^{(2)},\ldots,\lambda^{(W)}\}$. Каждая модель $\lambda^{(w)}$, $w=1,\ldots,W$ строится независимо от остальных единообразным путем. Сначала следует подобрать тройку параметров ($\boldsymbol{\pi}^{(w)}$, $\mathbf{A}^{(w)}$, $\mathbf{B}^{(w)}$), затем найти стационарное распределение $\mathbf{P}^{(w)}$ и вектор $\mathbf{E}^{(w)}$.

При использовании СММ в распознавания речи вектор $\boldsymbol{\pi}^{(w)}$ обычно не подбирают, а полагают жестко заданным таким образом, чтобы марковский процесс всегда начинался с некоторого стартового состояния. Матрица переходов $\mathbf{A}^{(w)}$ также имеет особый вид: это лево-правая модель (модель Бакиса [10]), в которой возможны только два типа переходов: $Q_n=i$, $Q_{n+1}=i$ и $Q_n=i$, $Q_{n+1}=i+1$ (рис. 2). Однако, для того, чтобы существовало стационарное распределение, необходимо предусмотреть возврат из последнего состояния в первое. В процессе обучения стоит подобрать значения ненулевых элементов матрицы переходов.

Набольшую сложность представляет собой подбор значений матрицы выходных вероятностей $\mathbf{B}^{(w)}$. Для поиска значений матриц $\mathbf{A}^{(w)}$ и $\mathbf{B}^{(w)}$ можно воспользоваться алгоритмом Баума–Велша [10], представляющим собой итерационную процедуру нахождения локального максимума $P\left(\mathbf{O}^{(w)}\,|\,\lambda^{(w)}\right)$ — вероятности порождения обучающей выборки $P\left(\mathbf{O}^{(w)}\,|\,\lambda^{(w)}\right)$ размером T_d данной моделью $\lambda^{(w)}$.

При использовании алгоритма Баума–Велша важным аспектом является выбор начальных значений матриц $\mathbf{A}^{(w)}$ и $\mathbf{B}^{(w)}$. Для матрицы переходов начальные значения ненулевых элементов берут таким образом, чтобы переход из состояния i в состояние $i+1$ был более вероятным, чем переход в состояние i:

$$\begin{cases}a_{ii+1}=1-a_{ii}, & i=1,\ldots,N-1;\\ a_{N1}=1-a_{NN};\\ a_{ii}\le 0{,}5, & i=1,\ldots,N.\end{cases}$$

Для инициализации значений выходных вероятностей матрицы $\mathbf{B}^{(w)}$ возможен простой подход, основанный на частоте появления символа k в обучающей выборке слова w:

$$b_i\left(k\right)=\frac{U_w\left(k\right)}{U_w},\ i=1,\ldots,N,$$

где $U_w(k)$ — число появлений символа k в обучающей выборке слова w; U_w — общее число символов в обучающей выборке слова w.

После инициализации выполняется подбор значений матриц $\mathbf{A}^{(w)}$ и $\mathbf{B}^{(w)}$ по алгоритму Баума–Велша. Следует предусмотреть, чтобы в матрице $\mathbf{B}^{(w)}$ не было нулевых элементов. Если при распознавании встретится наблюдаемое значение, отсутствующее в обучающей выборке, вероятность его порождения будет равна нулю, а логарифм устремится в $-\infty$. Чтобы избежать этой ситуации, часть вероятности распределяется в равных долях между нулевыми элементами матрицы $\mathbf{B}^{(w)}$ так, чтобы оставалось справедливым следующее свойство $\mathbf{B}^{(w)}$:

$$\sum_{k=1}^{K}b_i^{(w)}\left(k\right)=1,\ i=1,\ldots,N^{(w)}.$$

После подбора параметров СММ необходимо найти стационарное распределение и вектор вектор $\mathbf{E}^{(w)}$ вероятностей порождения всех K символов моделью. Стоит отметить, что для последующего использования модели $\lambda^{(w)}$ хранить требуется не тройку ($\boldsymbol{\pi}^{(w)}$, $\mathbf{A}^{(w)}$, $\mathbf{B}^{(w)}$), а только вектор $\mathbf{E}^{(w)}$. Таким образом, предлагаемый

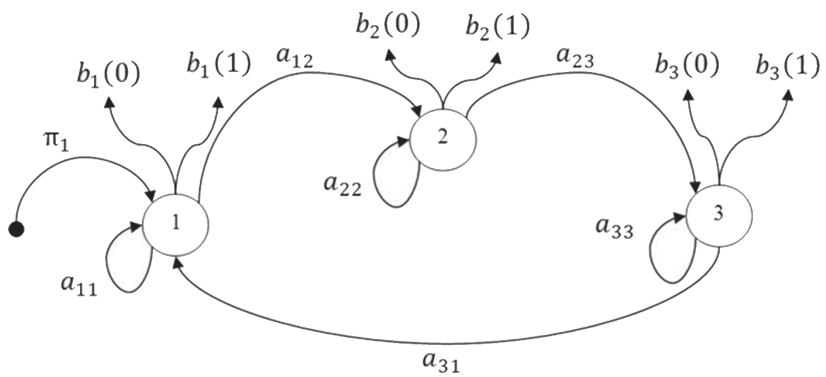


Рис. 2. Структура скрытой марковской модели одного слова (индекс слова w опущен)

подход к распознаванию голосовых команд позволяет экономить память, выделяемую для хранения модели речи $\boldsymbol{\lambda}$.

Алгоритм распознавания на основе стационарного распределения

Структурная схема предлагаемого в работе алгоритма распознавания приведена на рис. 3. На вход алгоритма поступает модель речи $\boldsymbol{\lambda} = \{\lambda^{(1)}, \lambda^{(2)}, ..., \lambda^{(W)}\}$, $\lambda^{(w)} = (\mathbf{E}^{(w)})$, $w = 1, ..., W$, а также набор наблюдаемых

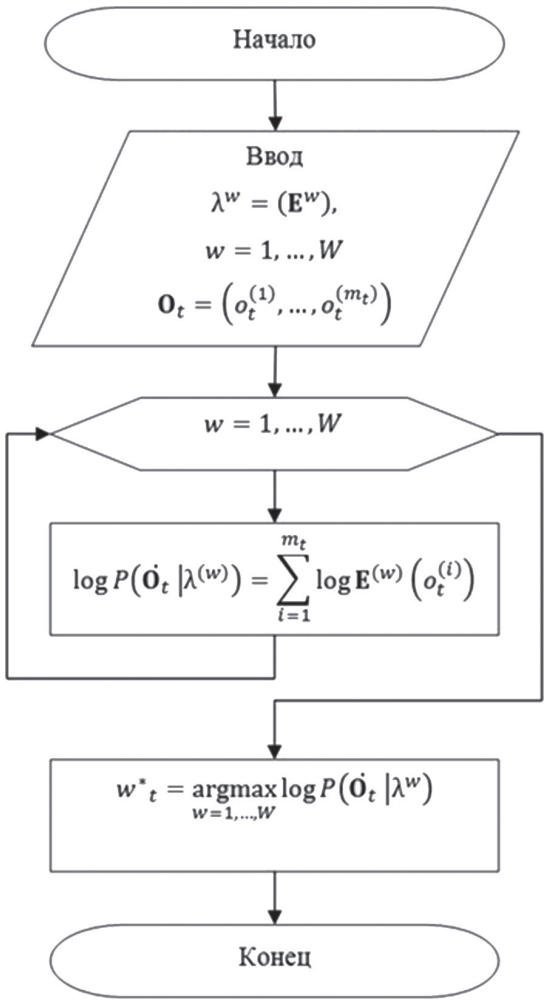


Рис. 3. Схема алгоритма распознавания изолированных слов на основе стационарного распределения скрытой марковской модели

значений неизвестной команды $\mathbf{O}_t = \left(o_t^{(1)}, \ldots, o_t^{(m_t)}\right)$. Для каждой из W моделей находятся «очки ее соответствия» неизвестному слову, и модель, которая набрала наибольшее количество «очков», объявляется победителем.

Сложность предлагаемого алгоритма линейна относительно m_t длины цепочки $\mathbf{O}_t$, а ответ — искомое слово w_t^*, будет получен через $O(Wm_t)$ шагов. Потребление памяти растет пропорционально $O(WK)$, где K — размер кодовой книги, так как для каждой модели $\lambda^{(w)}$ требуется хранить только вектор $\mathbf{E}^{(w)}$. При параллельном вычислении $\lg P\left(\dot{\mathbf{O}}_t \,|\lambda^{(w)}\right)$ для всех слов ответ будет получен через $O(m_t)$ шагов, тогда как алгоритму прямого хода потребуется $O(m_t N^{(w)})$ шагов для вычисления $P\left(\mathbf{O}_t \,|\, \lambda^{(w)}\right)$.

Распознавание изолированных слов. Анализ качества распознавания

Предлагаемый алгоритм распознавания был реализован программно. Эксперименты проходили на двух принципиально различных речевых базах: TIMIT и базе русских слов (БРС), сформированной авторами (табл. 1). Эксперимент на БРС моделирует условия работы системы голосового управления персонального устройства: только один диктор, обучающих примеров мало, различных голосовых команд много. Эксперимент на TIMIT нацелен на моделирование голосового управление в «публичной» системе, например, автоответчике: множество различных не повторяющихся дикторов, большое количество обучающих примеров, небольшое количество различных команд.

В качестве признакового описания объекта выступили квантованные MFCC-коэффициенты. Вектор MFCC $\mathbf{s}_t^{(i)}$ t-й команды включал 39 коэффициентов: первая компонента $\mathbf{s}_t^{(i)}$ — логарифм энергии кратковременного окна. Далее следовали 12 коэффициентов MFCC, затем 13 производных первого порядка и 13 производных второго порядка. Длина кратковременного окна составила 25 мс, смещение — 10 мс. Такой набор параметров признакового описания речи полностью повторяет эксперимент, описанный в [11], что позволяет сравнить производительность предлагаемого алгорит-

Таблица 1

Описание используемых речевых баз

Показатель	БРС (обучающая выборка/тестовая выборка)	TIMIT (обучающая выборка/тестовая выборка)
Количество слов (классов)	100/100	10/10
Количество примеров на класс	20/20	462/168
Количество дикторов	1/1	462/168
Общее количество примеров	2000/2000	4620/1680

ма и двух других подходов к распознаванию изолированных слов, данные по которым приведены в [11].

Распознавание голосовых команд является примером задачи классификации, поэтому важнейшей метрикой качества для оценки результата распознавания является общая точность (Accuracy) ρ — отношение количества правильных ответов к общему числу ответов T_l:

$$\rho = \sum_{t=1}^{T_l} \frac{\mathbb{I}\left(w^*\left(\mathbf{O}_t\right) = y\left(\mathbf{O}_t\right)\right)}{T_l},$$

где $\mathbf{O}_t$ — признаковое описание t-й команды; $w^*(\mathbf{O}_t)$ — предсказанный класс объекта $\mathbf{O}_t$; $y(\mathbf{O}_t)$ — истинный класс объекта $\mathbf{O}_t$; $\mathbb{I}(e) = \begin{cases} 1, & \text{если } e \text{ истинно;} \\ 0, & \text{если } e \text{ ложно} \end{cases}$ — индикаторная функция.

Исследовано влияние на точность:

- размера кодовой книги K (количества кластеров алгоритма k-средних);
- количества состояний $N^{(w)}$ моделей $\lambda^{(w)}$, $w = 1, ..., W$.

На рисунке 4 представлены зависимости точности распознавания от размера кодовой книги при различном количестве состояний для баз TIMIT и БРС. Для TIMIT наилучшая точность достигается при $K = 4096$ и $N^{(w)} = 6$, для БРС — при $K = 1024$ и $N^{(w)} = 3$, поэтому дальнейшие исследования проходили с этими значениями параметров (табл. 2).

Помимо общей точности распознавания другими важными метриками качества, традиционными для задач классификации, являются точность (Precision), полнота (Recall) и F_1-метрика [12]. Они вычисляются на основе матрицы ошибок (Confusion Matrix) $\mathbf{C} = \{c_{ij}\}$, где элемент c_{ij} — количество предсказаний того, что объект принадлежит классу i в то время, как его истинный класс — j. Диагональные элементы матрицы ошибок представляют собой количество верных предсказаний, все остальные элементы — количество ошибок.

Полнота для класса j — это вероятность верной классификации при условии, что истинным классом $\mathbf{O}_t$ является j:

$$R_j = P\left(w^*\left(\mathbf{O}_t\right) = j \mid y\left(\mathbf{O}_t\right) = j\right) = \frac{c_{jj}}{\sum_{i=1}^{W} c_{ij}}.$$

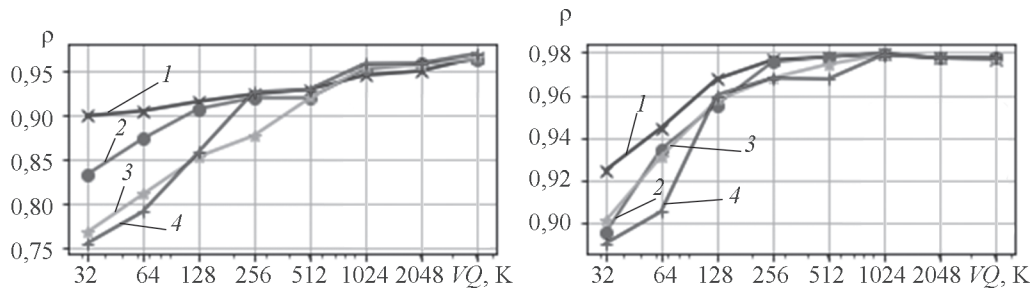
Точность для класса i — это вероятность того, что истинным классом $\mathbf{O}_t$ является i при условии, что был предсказан класс i:

$$P_i = P\left(y\left(\mathbf{O}_t\right) = i \mid w^*\left(\mathbf{O}_t\right) = i\right) = \frac{c_{ii}}{\sum_{j=1}^{W} c_{ij}}.$$

Метрика $F_1^{(i)}$ представляет собой взвешенное среднее метрик R_i и P_i:

$$F_1^{(i)} = 2\frac{P_i R_i}{P_i + R_i}.$$

Идеальный классификатор дает $R_i = P_i = F_1^{(i)} = 1$ для $i = 1, ..., W$. В табл. 2 приведены средние показатели точности P, полноты R и F_1 для алгоритма прямого хода и предложенного алгоритма, полученные на БРС и TIMIT, а также общее время, затраченное на распозна-



а *б*

Рис. 4. Зависимость общей точности распознавания ρ от размера кодовой книги VQ при различном количестве состояний $N^W = 3$ (*1*); 4 (*2*); 5 (*3*); 6 (*4*):

а — TIMIT; *б* — БРС

Таблица 2

Результаты распознавания русских слов на БРС и TIMIT

База	Алгоритм распознавания	R	P	F_1	Общее время, мкс
БРС	Прямой ход	0,9800	0,9816	0,9787	13935277
	Стационарное распределение	0,9795	0,9812	0,9782	452055
TIMIT	Прямой ход	0,9702	0,9703	0,9700	84512
	Стационарное распределение	0,9702	0,9704	0,9700	5000

вание всей тестовой выборки (аппаратная платформа: Intel Core i3-2100 3.10 ГГц, 8 Гб DDR3 ОЗУ; программная платформа: Windows 7 (64-битная версия), Python 3.6, numpy 1.14.0, scipy 1.0.0.). Кроме того, выполнено сравнение предложенного алгоритма с двумя другими подходами — самонастраиваемыми нейронными сетями (SANN) [11] и непрерывными СММ с гауссовыми смесями (HMM–GMM) [11] на основе метрики общей точности ρ (табл. 3). Алгоритм на основе стационарного распределения превосходит SANN по качеству распознавания, однако уступает HMM–GMM примерно на 2,68%.

Таблица 3

Сравнение результатов распознавания изолированных слов на базе TIMIT различными алгоритмами

Алгоритм распознавания	Общая точность, %
Прямой ход	97,02
Стационарное распределение	97,02
HMM-GMM [11]	99,70
SANN [11]	92,02

Заключение

Предложен алгоритм распознавания голосовых команд на основе стационарного распределения скрытой марковской модели, опускающий порядок следования звуков речи. Он отличается низкой вычислительной сложностью (растет линейно относительно длины наблюдаемой последовательности), простотой модели, а также высокой точностью распознавания, лишь на 2,68% уступая подходу на основе гауссовых смесей и превосходя по этому показателю качества алгоритм на основе самонастраиваемых нейронных сетей.

Литература

1. **Dahl G.E., Yu D., Deng L., Acero A.** Context-dependent Pre-trained Deep Neural Networks for Large-vocabulary Speech Recognition // IEEE Trans. Audio, Speech and Language Proc. 2012. V. 20. No. 1. Pp. 30—42.

2. **Hinton G. e. a.** Deep Neural Networks for Acoustic Modeling in Speech Recognition // IEEE Signal Proc. 2012. V. 29. No. 6. Pp. 82—97.

3. **Mohamed A., Dahl G.E., Hinton G.** Acoustic Modeling Using Deep Belief Networks // IEEE Trans. Audio, Speech and Language Proc. 2012. V. 20. No. 1. Pp. 14—22.

4. **Hinton G., Bengio Y., Le Cun Y.** Deep Learning // Nature. 2015. V. 521. Pp. 436—444.

5. **Вагин В.Н., Ганишев В.А.** Кластеризация пользователей по голосу с помощью улучшенных самоорганизующихся растущих нейронных сетей // Программные продукты и системы. 2015. № 3. С. 136—141.

6. **Mohammadi M., Sadegh Mohammadi H.R.** Study of Speech Features Robustness for Speaker Verification Application in Noisy Environments // Proc. 8th Intern. Symp. Telecommunications. 2016. Pp. 489—493.

7. **Molau S., Pitz M., Schlüter R., Ney H.** Computing Mel-frequency Cepstral Coefficients on the Power Spectrum // IEEE Intern. Conf. Acoustics, Speech and Signal Proc. 2001. V. 1. Pp. 73—76.

8. **Paramonov P., Sutula N.** Simplified Scoring Methods for HMM-based Speech Recognition // Soft Computing. 2016. V. 20. Pp. 3455—3460.

9. **Bertsekas D., Tsitsiklis J.** Introduction to Probability. Belmont: Athena Sci., 2008.

10. **Рабинер Л.Р.** Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: обзор // Труды ин-та инженеров по электротехнике и радиоэлектронике. 1989. Т. 77. № 2. С. 86—120.

11. **Ting H.-N., Yong B.-F., Mirhassani S.M.** Self-adjustable Neural Network for Speech Recognition // Engineering Appl. Artificial Intelligence. 2013. V. 26. Pp. 2022—2027.

12. **Murphy K.P.** Machine Learning: a Probabilistic Perspective. Cambridge, Massachusetts, London: MIT Press, 2012.

References

1. **Dahl G.E., Yu D., Deng L., Acero A.** Context-dependent Pre-trained Deep Neural Networks for Large-vocabulary Speech Recognition. IEEE Trans. Audio, Speech and Language Proc. 2012;20;1:30—42.

2. **Hinton G. e. a.** Deep Neural Networks for Acoustic Modeling in Speech Recognition. IEEE Signal Proc. 2012;29;6:82—97.

3. **Mohamed A., Dahl G.E., Hinton G.** Acoustic Modeling Using Deep Belief Networks. IEEE Trans. Audio, Speech and Language Proc. 2012;20;1:14—22.

4. **Hinton G., Bengio Y., Le Cun Y.** Deep Learning. Nature. 2015;521:436—444.

5. **Vagin V.N., Ganishev V.A.** Klasterizatsiya Pol'zovateley po Golosu s Pomoshch'yu Uluchshennykh Samoorganizuyushchikhsya Rastushchikh Neyronnykh Setey. Programmnye Produkty i Sistemy. 2015;3:136—141. (in Russian).

6. **Mohammadi M., Sadegh Mohammadi H.R.** Study of Speech Features Robustness for Speaker Verification Application in Noisy Environments. Proc. 8th Intern. Symp. Telecommunications. 2016:489—493.

7. **Molau S., Pitz M., Schlüter R., Ney H.** Computing Mel-frequency Cepstral Coefficients on the Power Spectrum. IEEE Intern. Conf. Acoustics, Speech and Signal Proc. 2001;1:73—76.

8. **Paramonov P., Sutula N.** Simplified Scoring Methods for HMM-based Speech Recognition. Soft Computing. 2016;20:3455—3460.

9. **Bertsekas D., Tsitsiklis J.** Introduction to Probability. Belmont: Athena Sci., 2008.

10. **Rabiner L.R.** Skrytye Markovskie Modeli i Ikh Primenenie v Izbrannykh Prilozheniyakh pri Raspoznavanii Rechi: Obzor. Trudy In-ta Inzhenerov po Elektrotekhnike i Radioelektronike. 1989;77;2:86—120. (in Russian).

11. **Ting H.-N., Yong B.-F., Mirhassani S.M.** Self-adjustable Neural Network for Speech Recognition. Engineering Appl. Artificial Intelligence. 2013;26:2022—2027.

12. **Murphy K.P.** Machine Learning: a Probabilistic Perspective. Cambridge, Massachusetts, London: MIT Press, 2012.

Сведения об авторах

Парамонов Павел Александрович — старший преподаватель кафедры вычислительной техники НИУ «МЭИ», e-mail: ParamonovPA@mpei.ru

Огнев Иван Васильевич — доктор технических наук, профессор кафедры вычислительной техники НИУ «МЭИ», e-mail: OgnevIV@mpei.ru

Information about authors

Paramonov Pavel A. — Senior Lecturer of Computer Engineering Dept., NRU MPEI, e-mail: ParamonovPA@mpei.ru

Ognev Ivan V. — Dr.Sci. (Techn.), Professor of Computer Engineering Dept., NRU MPEI, e-mail: OgnevIV@mpei.ru

Статья поступила в редакцию 15.03.2018