УДК 621.391

МАТЕМАТИЧЕСКИЕ МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ РАСПОЗНАВАНИЯ ГОВОРЯЩЕГО

Х.М. Ахмад

Кафедра вычислительной техники,
Владимирский государственный университет

Представлена профессором В.Н. Ланцовым и членом редколлегии профессором В.И. Коноваловым

Ключевые слова и фразы: векторное квантование; динамическое искажение времени; кодовая книга; распознавание; речевой сигнал.

Аннотация: Рассматриваются методы выделения наиболее информативных характеристик речевого сигнала, методы распознавания, основанные на использовании алгоритмов динамического искажения времени (**ДИВ**) и векторного квантования (**ВК**). В качестве основного типа систем распознавания говорящего рассматриваются системы идентификации с применением ДИВ и ВК с созданием кодовых книг образцов речевого сигнала.

Автоматическое распознавание речи – важное поле исследований, имеющее большое значение в современной практике.

1. Выделение наиболее информативных характеристик речевого сигнала

Возможность различать голоса разных людей связана как с анатомическим различием вокального тракта, так и с различием манеры разговора. Однако речь одного и того же человека также заметно меняется в зависимости от скорости разговора, эмоционального состояния или же от состояния здоровья. В ходе работы по выбору характеристик для системы верификации было замечено, что отдельные характеристики из полного набора меньше подвержены вариативности произношения одного и того же диктора. Исходя из этого сделано предположение, что для слова, произносимого одним и тем же диктором, можно найти подмножество характеристик, использование которых при верификации не только не увеличит частоту появления ошибок пропуска, но может и уменьшить вероятность появления ошибок отклонения. Кроме того, сокращение множества характеристик даст значительный выигрыш в скорости при последующих вычислениях. Однако при выборе таких характеристик желательно чтобы не только уменьшалось влияние вариативности произношения, но и увеличивалась бы индивидуальность отдельного диктора [10, 11] .

Проблему выбора подмножества характеристик формально можно описать как выбор наилучшего подмножества X, состоящего из n характеристик из множества Y:

$$X = \{x_i\}, \text{ где } i = 1,2,\ldots,n \text{ и } x_i \in Y;$$
$$Y = \{y_i\}, \text{ где } i = 1,2,\ldots,M \text{ и } M > n.$$

Под наилучшим подмножеством понимается комбинация из n характеристик, которая бы максимизировала некоторый функционал H. Так как перебрать все комбинации, состоящие из n характеристик, не представляется возможным, для составления подмножества будет использован следующий алгоритм – последовательный прямо-обратный поиск (Sequential Direct-Backward Search – **SDBS**), описанный ниже.

1.1. Последовательный прямой поиск

Последовательный прямой поиск (**ППП**) – это простая процедура добавления по одной характеристике в текущее множество. В качестве критерия отбора характеристик используется частота появления ошибки пропуска. На каждом шаге выбирается такая характеристика из оставшегося множества, при включении которой в текущее множество значение используемого критерия было бы минимально. Перед началом работы алгоритма текущим множеством X_i является пустое множество: $X_i = \varnothing$.

1.2. Последовательный обратный поиск

Последовательный обратный поиск (**ПОП**) – это нисходящий эквивалент метода последовательного прямого поиска. Начиная с полного множества $X_i = Y$, происходит удаление по одной характеристике, пока не будет удалено $M - n$ характеристик. На каждом шаге алгоритма удаляется та характеристика, при использовании которой ошибка отклонения максимальна.

1.3. SDBS-алгоритм

Пусть имеется множество характеристик Y.

1) Используя метод ППП, добавляется p характеристик ξ_i из множества доступных характеристик $Y - X_k$ в X_k для получения множества X_{k+1}.

Устанавливается $k = k + p$, а $X_{M-k} = X_k$.

2) Используя метод ПОП, удаляется q худших характеристик ξ_i из множества X_{M-k}, формируя при этом X_{M-k+p}. Устанавливается $k = k - p$. Если $k = n,$ то завершаем алгоритм, иначе $X_M = X_{M-k}$ и возвращаемся к шагу 1.

С помощью приведенного SDBS-алгоритма можно определить наиболее информативные признаки, использование которых позволит достичь достаточно высокого уровня распознавания.

2. Оптимизированный алгоритм поиска минимального маршрута для симметричного алгоритма динамического искажения времени в задачах распознавания дикторов

Хорошо изученный подход к распознаванию речи основывается на хранении одного или нескольких эталонов для каждого слова в словаре распознавания. Процесс распознавания, таким образом, состоит из сравнения входящей речи с хранящимися эталонами. Эталон с минимальным расхождением от полученного сигнала и будет распознанным словом. Алгоритм, используемый для поиска минимального расхождения, базируется на динамическом программировании.

Основным свойством алгоритма, относящегося к методу динамического программирования, является небольшая ресурсоемкость и полиномиальная зависимость требуемых вычислительных затрат от размера входных данных. Это свойство является очень важным с точки зрения практической реализуемости алгоритма, что дает возможность дальнейшего его усложнения (для повышения эффективности) без особых временных затрат разработчика на оптимизацию.

Для того чтобы использовать алгоритм динамического искажения времени (**ДИВ**), необходимо учитывать следующее [2].

Признаки. Информация в каждом сигнале должна иметь одно и то же представление.

Оценки. Должны использоваться одинаковые виды измерения для получения верных результатов. Они делятся на локальные – вычисление разницы между признаком одного сигнала признаком другого, и глобальные – полное вычисление разницы между входящим сигналом и другим сигналом, возможно, другой длины.

Вопрос оптимального выделения признаков не тривиален. Если мы интересуемся распознаванием, то идеальным выделением для нас будет выдача строки слов, без излишнего распознавания. С другой стороны, отделение процесса выделения признаков от процесса распознавания эталонов позволяет нам изолировать процесс распознавания образов.

Для лучшего понимания будем работать с кадрами, на которых основывается процесс выделения признаков. Таким образом, анализ признаков состоит из обработки вектора признаков в регулярных интервалах. Например, если мы выполняем анализ гребенки фильтров (filter bank), то наш вектор признаков может состоять из энергий на каждой усредненной полосе в 20 мс. Для линейного предсказывающего анализа, вектор признаков содержит коэффициенты предсказания или их изменения. В общем, вектор признаков, используемый в распознавании речи, – это mel-частотные кепстральные коэффициенты [9, 11].

Так как вектор признаков может иметь множество элементов, то требуются средства расчета локальной оценки расстояния. Оценка расстояния между двумя векторами признаков рассчитывается с помощью Евклидовой метрики (Euclidean metric). Таким образом, локальное расстояние между вектором X сигнала 1 и вектором Y сигнала 2 рассчитывается по формуле

$$d(x,y)=\sqrt{\sum_i (x_i-y_i)^2}\ . \qquad (1)$$

Хотя Евклидова метрика в вычислительном отношении более дорога, чем другая метрика, она дает больше веса при больших различиях в единственном признаке.

Если требуется обратное прослеживание вдоль маршрута минимального расхождения (а не просто получение оценки в конце этого маршрута), то массив обратного прослеживания должен содержаться вместе с данными в массиве, указывающем на предыдущую точку маршрута.

2.1. Симметричный алгоритм ДИВ

Речь является процессом, изменяющимся во времени. Различные произношения одного и того же слова, в основном, имеют разные длительности, а произношения одного и того же слова с одинаковой длительностью отличаются в середине из-за различных частей слова, произносимых с разной скоростью. Чтобы получить глобальную оценку расхождения между двумя речевыми образцами, представленными как последовательности векторов, должно быть выполнено выравнивание по времени.

Как искать максимальное совпадение (то есть минимальное расхождение) между входным сигналом и эталоном? Можно оценить все возможные варианты, но это чрезвычайно неэффективно, так как количество возможных вариантов растет экспоненциально при увеличении длительности сигнала. Вместо этого наложим ограничения на процесс сравнения и будем использовать их вместе с эффективным алгоритмом. Ограничения, которые мы налагаем, простые и не сильно ограничивающие:

– пути сравнения не могут идти назад во времени;

– каждый кадр входного сигнала должен быть использован при сравнении;

– локальные оценки совпадения объединяются добавлением к данному глобальному расхождению.

Теперь мы можем сказать, что каждый кадр в эталоне и входном сигнале должен быть использован в процессе сравнения. Это означает, что если мы возьмем точку (i, j) во временной матрице (где *i* указывает на кадр входного сигнала, j – на кадр эталона), то предыдущая точка может иметь координаты $(i-1, j-1)$, $(i-1, j)$ или $(i, j-1)$. Ключевая идея динамического программирования заключается в том, что в точке (i, j) мы просто продолжаем самый близкий маршрут сравнения из $(i-1, j-1)$, $(i-1, j)$ или $(i, j-1)$.

Этот алгоритм известен как динамическое программирование. В случае применения в области распознавания речи на базе эталонов, алгоритм называют динамическим искажением времени – Dynamic Time Warping (**DTW**) [4, 6]. Динамическое программирование гарантирует нахождение минимального расхождения в матрице при уменьшении объема вычислений.

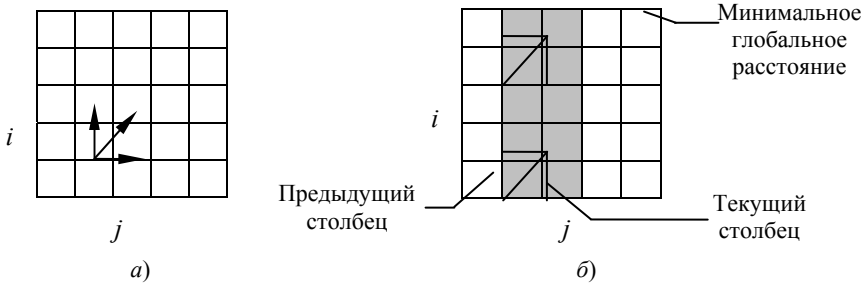
Алгоритм динамического программирования работает в манере временного синхронизирования: каждый столбец временной матрицы считается последовательно (эквивалентно обработке входного сигнала по кадрам). Таким образом, для эталона длины *n* максимальное число маршрутов в любое время будет равно *n*.

Если $D(i, j)$ является глобальным расхождением от точки (i, j), $d(i, j)$ – локальным, то

$$D(i, j) = \min\left[D(i-1, j-1), D(i-1, j), D(i, j-1)\right] + d(i, j). \qquad (2)$$

Начальное условие $D(1,1) = d(1,1)$. Таким образом, у нас есть основа для эффективного рекурсивного алгоритма обработки $D(i, j)$. Конечное глобальное расстояние $D(n, N)$ дает нам общую оценку сравнения эталона с входным сигналом. Входной сигнал распознается как слово, соответствующее эталону с минимальным отклонением. Отметим, что N будет различно для каждого эталона.

Для пояснения, предполагаем, что столбцы и строки временной матрицы нумеруются от нуля. Это означает, что направления, в которых маршрут сравнения может проходить от точки (i, j) временной матрицы, будут только такие, как на рис. 1. Ячейки (i, j) и $(i, 0)$ имеют различные возможные родительские ячейки. Маршрут к $(i, 0)$ может начинаться только из точки $(i-1, 0)$. Тем не менее, маршрут к точке (i, j) может начинаться из трех точек (рис. 1, *б*).



а) *б*)

Рис. 1. Три возможных направления, в которых может пролегать маршрут сравнения от (i, j) в симметрическом алгоритме ДИВ

В вычислительном отношении формула (2) может быть рекурсивно запрограммирована. Тем не менее, если язык программирования не оптимизирован под рекурсии, этот метод может быть медленным даже для относительно малых размеров эталонов. Другой метод, быстрый и требующий меньше памяти, использует два вложенных цикла. Этот метод нуждается только в двух массивах, которые содержат смежные столбцы временной матрицы.

2.2. Алгоритм поиска глобального наименьшего маршрута

В данной работе предлагается следующий оптимизированный алгоритм поиска глобального наименьшего маршрута.

1. Вычислить нулевой столбец, начиная с самой нижней ячейки. Глобальный маршрут к этой ячейке равен локальному. Тогда глобальный маршрут для каждой последующей ячейки равен локальному маршруту для этой ячейки плюс глобальный маршрут до ячейки под ней. Это называется predCol (предшествующий столбец).

2. Вычислить глобальный маршрут к первой ячейке следующего столбца (curCol), сложив локальный маршрут с глобальным маршрутом к самой нижней ячейке предыдущего столбца.

3. Вычислить глобальный маршрут для оставшихся ячеек текущего столбца. Например, для точки (i, j) – локальная дистанция до точки (i, j) плюс минимум глобального маршрута из $(i-1, j)$, $(i-1, j-1)$ или $(i, j-1)$.

4. Текущий столбец становится предыдущим, и все начинается со второго шага до тех пор, пока не будут обсчитаны все столбцы.

5. Глобальный маршрут – это значение, сохраненное в самой верхней ячейке последнего столбца.

Ниже показан псевдокод для этого процесса.

```
calculate first column (predCol)
for i = 1 to number of input feature vectors
{
      curCol[0] = local cost at (i,0) + global cost at (i – 1,0);
      for j = 1 to number of template feature vectors
      {
          curCol[j] = local cost at (i, j) + minimum of global cost
                    at (i – 1, j), (i, j – 1) or (i – 1, j – 1);
      }
predCol = curCol;
}
minimum global cost is value in curCol[number of template feature vectors].
```

Для распознавания берется входящий сигнал, и вышеприведенный процесс повторяется для каждого эталона. Файл эталонов, который дает самый короткий глобальный маршрут, является текстовой оценкой.

На основании вышеизложенного рассматривается эффективность использования алгоритма ДИВ в области распознавания речи созданием системы идентификации спикера в среде программного продукта Matlab.

В созданной системе для идентификации спикера из входного речевого сигнала извлекаются следующие различающие признаки: форманты (F); mel-частотные коэффициенты (**MFC**); уровни переходов через нуль (Z-C); E – уровень

энергии показанного в окне сегмента, где P – определенный порог энергии, $S1$ – пользователь на стадии обучения и $S2$ – пользователь на стадии тестирования, (рис. 2, 3).

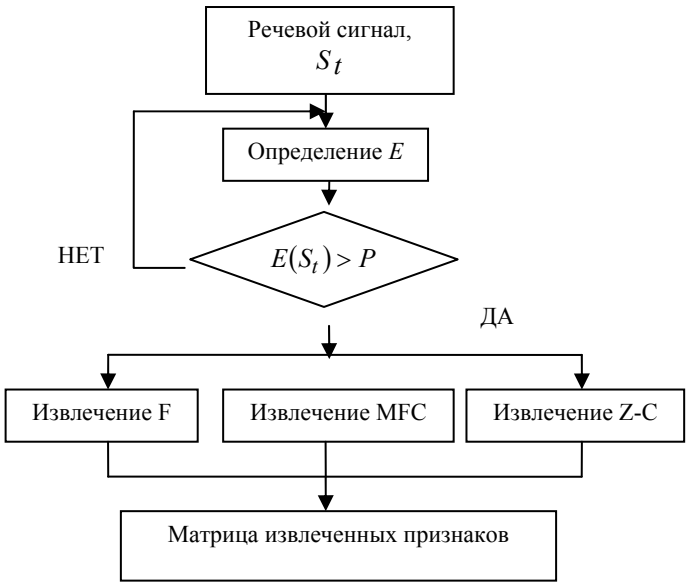


Рис. 2. Блок-схема обработки речевых признаков

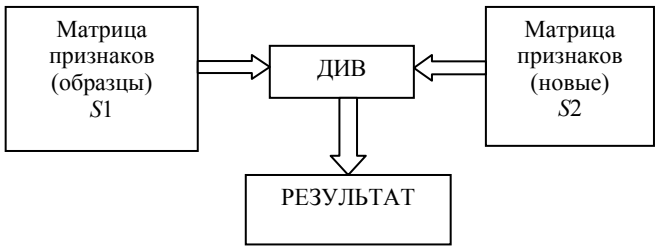


Рис. 3. Блок-схема сравнения

Реализация данной системы подтверждает, что для распознавания речи динамическое программирование требует мало оперативной памяти, единственное хранение, требующееся для поиска (в отличие от эталонов), – массив, который содержит единственный столбец временной матрицы.

Система была проверена с тремя пользователями. Образцы были созданы с использованием одного образца пользователя, и тестирование было сделано с различным набором образцов. Все пользователи были правильно идентифицированы со 100 %-й точностью и незарегистрированные пользователи были отклонены.

3. Система идентификации говорящего методом создания кодовых книг образцов речи

Идентификация спикера существует в сфере распознавания спикера, которая охватывает и идентификацию, и верификацию спикеров. Верификация спикера – предмет выяснения: действительно ли пользователь тот, кем он себя заявляет.

Фундаментальное предположение, сделанное в любой из этих систем, – это то, что есть измеримые особенности (признаки, поддающиеся количественному определению) голоса каждого человека, которые являются уникальными среди людей и поэтому могут быть измерены. Особенности (признаки), которыми мы пользуемся в этой системе, являются корневыми в частотной области.

В этой работе рассмотрен процесс идентификации спикеров путем сравнения образцов их голоса с базой данных спикеров. Система состоит из двух фаз. В первой фазе создается «кодовая книга» спикеров, для того чтобы характеризовать их вокальные особенности, используя обучающее высказывание (фраза). Во второй фазе сравнивается образец голоса спикера с кодовой книгой, для того чтобы определить идентичность (личность) спикера.

3.1. Речевой сигнал

Модель речи, которая является полезной в идентификации спикера, имеет следующий вид: $X(t) = h(t)p(t)$.

Таким образом, речевой сигнал $X(t)$ является сверткой фильтра $h(t)$ и некоторого сигнала $p(t)$.

Здесь $h(t)$ – импульсная характеристика элементов, например, зубов, носовой впадины, губ, и т.д., получившаяся на пути, идущем от легких; $p(t)$ – возбуждение, которое мы именуем как высота тона речи. Согласно свойствам преобразования Фурье (**FFT**), такая модель в частотной области примет вид

$$X(j\omega) = H(j\omega)P(j\omega),$$

где ω – круговая частота [9]. Если $h(t)$ является импульсной характеристикой в модели речи, то здесь $H(j\omega)$ является передаточной (частотной) характеристикой в частотном домине (области) преобразования Фурье.

3.2. Кепстральный анализ

Слово «кепстр» – от лат. *cepstrum* – означает косинус-преобразование Фурье логарифма спектра мощности, которое математически обозначается

$$c(n) = \text{ifft}(\log|\text{fft}(x(n))|),$$

где $x(n)$ – выбранный речевой сигнал, и $c(n)$ – сигнал в кепстральном домене [9]. Из формул:

$$c(n) = \text{ifft}(\log(\text{fft}(h(t)p(t))));$$

$$c(n) = \text{ifft}(\log(\text{fft}(H(j\omega)P(j\omega))));$$

$$c(n) = \text{ifft}(\log(H(j\omega)) + \text{ifft}(\log(P(j\omega)))),$$

видно, что, начиная с логарифма, хотя нелинейно, в основном только затухает каждый спектр; и начиная с $P(j\omega)$ – последовательности импульсов (это (см. FFT) то же самое, что и сигнал непосредственно), можем все еще восстановить исходную последовательность импульсов. Здесь ищем периодические пики (острый выступ) в сигнале и высоту тона спикера. Хотя $H(j\omega)$ не является последовательностью импульсов, мы все еще можем использовать эту информацию. Она

всегда располагается около более низкой части кепстра; будем использовать первые двенадцать кепстральных коэффициентов как общие в данной области.

Для спикеров частота тона F_p может принимать значения в интервале 80…300 Гц, таким образом, мы в состоянии снизить часть кепстра, где мы ищем тон. В кепстре, который, в основном, является временной областью, мы ищем последовательность импульсов, отделенных периодом тона, то есть $1/F_p$. Частота тона определятся по формуле

$$F_p = F_x / n,$$

где F_x – частота дискретизации; n – число образцов между пиками и 0-м коэффициентом.

В данной работе при проведении кепстрального анализа для создания кодовых книг особое внимание уделяется вычислению параметров, краткое описание которых приводится ниже [5, 9].

1. Коэффициенты кепстральной mel-частоты (Mel-Frequency Cepstral Coefficients (**MCC**)).

Для расчета этих коэффициентов, применяется mel-гребенка фильтров со следующими важными моментами:

– фильтры, располагаемые однородно в mel-шкале, переводятся логарифмически в Герц-шкалу;

– фильтры треугольной формы подчеркивают среднюю частоту ω_i и диапазон (интервал) к следующей средней частоте;

– область под каждым фильтром является постоянной и иногда масштабируется в сумме к 1;

– фильтры распределяют однородно через mel-частотное пространство;

– полученные mel-коэффициенты преобразуют к Герц-шкале, чтобы получить частоты ω_i в линейной шкале:

$$\text{mel}\,2\,\text{Hz}(\text{mel}) = 700\left(\mathrm{e}^{\frac{\text{mel}}{1125}} - 1\right),$$

где $\text{mel}(f) = 2595\log_{10}(1 + f/700)$; m – заданное число гребенки фильтров; f – рабочая частота.

Фильтры создаются следующим образом. Пусть $f[m]$ – элемент разрешения по частоте, связанный со средней частотой ω_m:

$$H_m[k] = \begin{cases} 0, & k \le f[m-1]; \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, & f[m-1] \le k \le f[m]; \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, & f[m] \le k \le f[m+1]; \\ 0, & k > f[m+1]. \end{cases}$$

Применив mel-гребенку фильтров

$$S[m] = \log\left[\sum_{k=0}^{N-1} |X(k)|^2 H_m[k]\right], \qquad 0 < m \le M,$$

получим mel-кепстр применением дискретного преобразования Фурье (**ДПФ**)

$$c[n] = \mathrm{dct}(S[m]) = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n(m-1/2)}{M}\right), \qquad 0 \le n < M.$$

В результате, как правило, 12 коэффициентов сохраняются, и, по крайней мере, 24 mel-фильтра используются.

2. Дельта-кепстральные коэффициенты (Delta-Cepstral Coefficients – **DCC**).

Все методы, которые мы обсудили, не показывают, как сигнал изменяется, поэтому применяются производные кепстров, известные как динамические особенности, как попытка захватить информацию, связанную с развитием сигнала. Отметим следующие моменты:

– первая производная вычисляется в соответствии с кривой вектора особенностей, которые происходят между интервалом в N мс вокруг текущего вектора особенностей;

– типично, N в пределах 40…50 мс:

$$\underbrace{\begin{bmatrix} c_1 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}\begin{bmatrix} c_1 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}\begin{bmatrix} c_1 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}\begin{bmatrix} c_1 \\ c_1 \\ \vdots \\ c_d \end{bmatrix}}_{N\,ms};$$

– эта производная известна как дельта-кепстр и обычно обозначается греческим символом Δ;

– с 10 мс – движением по фазе, мы могли определить первую производную 40 мс следующим образом:

$$\Delta c_k = c_{k+2} - c_{k-2};$$

– следовательно, дельта-кепстры будут приложены к вектору особенности:

$$x = \left(c_1\, c_2 \ldots c_N\ \Delta c_1\ \Delta c_2 \ldots \Delta c_N\right).$$

3. Дельта-дельта-кепстральные коэффициенты (Delta-Delta-Cepstral Coefficients – **DDCC**).

Дельта-дельта-кепстр, или коэффициенты ускорения, являются приближениями вторых производных:

$$\Delta\Delta c_k = \Delta c_{k+1} - \Delta c_{k-1},$$

и могут быть также приближенными к вектору особенностей.

3.3. Векторное квантование

Векторное квантование (**ВК**) – процесс взятия большого множества векторов признаков и создания меньшего множества векторов признаков, которые представляют центроидное (простое суммирование) распределение, то есть точки, расположенные так, чтобы минимизировать среднее расстояние к каждой другой точке. Мы используем векторное квантование, так как было бы непрактично хра-

нить (запоминать) каждый отдельный вектор признаков, который мы генерируем от обучающего высказывания. В то время как алгоритм ВК действительно требует времени для вычисления, он экономит время в течение фазы тестирования, и поэтому это компромисс, с которым мы можем смириться.

Для векторного квантования в созданной системе применяется известный алгоритм кластеризации LBG, предложенный Линде и др. [2, 3] в 1980 г. как усовершенствование метода Ллойда. Они развивали результаты Ллойда от моно-к случаям *k*-размеров. По этой причине их алгоритм известен как обобщенный Ллойд-алгоритм (Generalized Lloyd Algorithm – **GLA**) или **LBG** (от инициалов авторов).

В целом, алгоритм LBG – конечная последовательность шагов, в которых в каждом шаге будет произведен новый квантователь, с полным искажением, меньшим или равным, предыдущему. Данный алгоритм имеет две основные фазы: инициализация кодовой книги и ее оптимизация [3].

После квантования спикера в его/ее кодовой книге необходимо измерить сходство/несходство между двумя пользователями. Для этого применяется простая мера – Евклидово кодовое расстояние [1].

3.4. Коэффициенты весового расстояния

С целью увеличения вероятности выбора истинного спикера, для вычисления весовых расстояний в данной системе используется алгоритм, суть которого заключается в следующем.

Рассматривается база данных кодовых книг спикера (**ККС**) $C_1,\ldots,C_N$. Кодовые книги (**КК**) будут окончательно обработаны к назначенным весам для кодовых векторов, и результатом процесса будет множество взвешенных кодовых книг

$$\left(C_i, W_i\right),\ i=1,\ldots,N,$$

где $W_i+\{w(c_{i1}),\ldots,(c_{ik})\}$ – веса, назначенные для *i*-й кодовой книги [1, 8]. Таким образом, весовой подход не увеличивает вычислительную нагрузку процесса соответствия, поскольку это может быть сделано в обучающей фазе путем создания базы данных спикера. Веса будут вычислены с использованием следующего кода.

```
PROCEDURE ComputeWeights (S:SET OF CODEBOOKS) RETURNS WEIGHTS
FOR EACH C_j IN S DO                % loop over all codebook
    FOR EACH c_i IN C_i DO          % loop over code vectors
        Sum := 0;
        FOR EACH C_K , K ≠ i, IN S DO        % Find nearest code vector_
            d_min := DistanceToNeareset (c_i , C_K ); %_from all other codebooks
            Sum := sum + 1/d_min ;
        ENDFOR
        W(C_ij) := 1/sum;
    ENDFOR
ENDFOR
```

3.5. Архитектура системы обучения

Архитектура системы обучения состоит из двух основных частей (рис. 4).

Первая часть состоит из обработки каждого образца входного голоса человека с целью уплотнения (то есть уменьшения объема) и суммирования характеристики их вокальных трактов. Вторая часть включает в себя сбор данных каждого человека вместе в легкоманипулируемый сигнал – трехмерную матрицу.

3.6. Архитектура системы тестирования

Система тестирования отражает архитектуру системы обучения. Сначала анализируется сигнал, затем сравнивается с данными, сохраненными в кодовой книге (рис. 5).

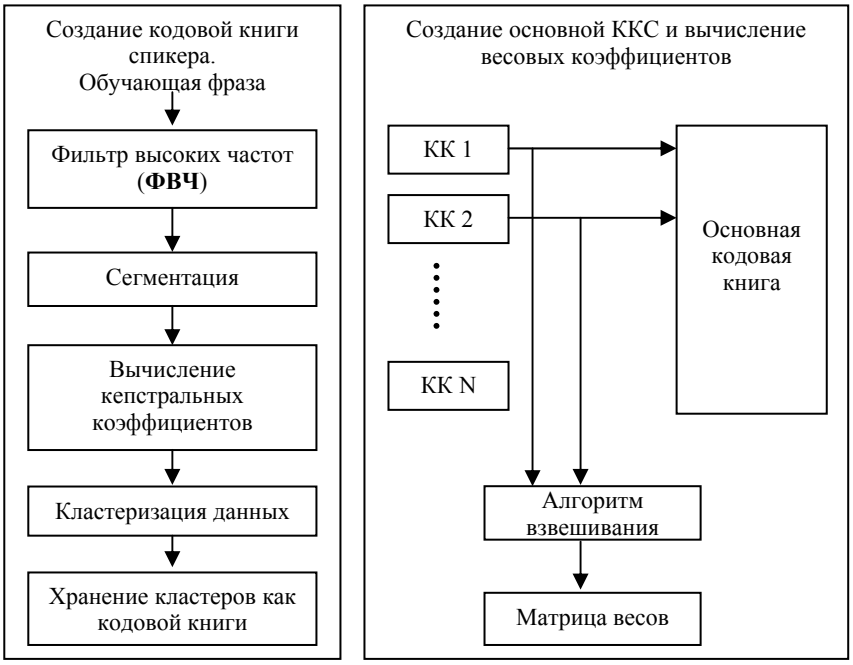


Рис. 4. Структурная схема подсистемы обучения

Алгоритмы тестирования

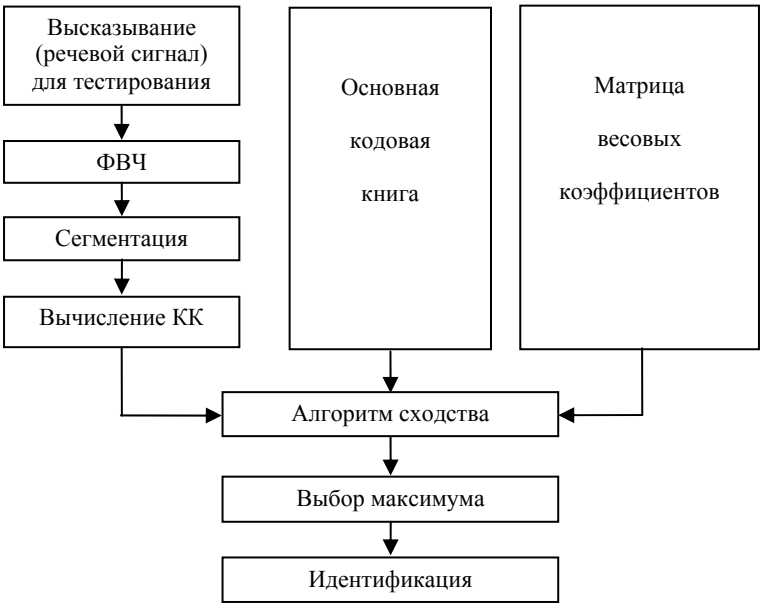


Рис. 5. Структурная схема подсистемы тестирования

Выводы

1. Mel-шкала переводит регулярные (правильные) частоты в масштаб, который является более соответствующим речи, так как человеческое ухо чувствует звук нелинейным способом.

2. DMCC и DDMCC говорят о том, как быстро голос говорящего изменяется и о чем-то, подобном ускорению тона, соответственно.

3. Использование алгоритма кластеризации LBG для векторного квантования позволяет практично хранить (запоминать) каждый отдельный вектор признаков, который мы генерируем от обучающего высказывания. В то время как алгоритм ВК действительно требует времени для вычисления, он экономит время в течение фазы тестирования.

4. LBG, в основном, отражает большую важность уникальных (однозначных) кодовых слов. Это – очень важная часть системы, так как качества, которые мы ищем в речевых сигналах, являются очень тонкими. Аналогично, алгоритм также уменьшает важность подобных ключевых слов, так как они не приводят ни к какой информации различения.

5. Путем вычисления вышеприведенных параметров речевого сигнала было идентифицировано 96 % тестируемых пользователей.

Список литературы

1. Akhmad Kh.M. Codebook modeling in speaker verification/identification task solution. 8-th international conference on pattern recognition and image analysis: new information technologies. Yoshkar-Ola, the Russian Federation, 2007, Proceedings -2. – P. 223–227.

2. C.S. Myers and L.R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal, 60(7):1389–1409, September 1981.

3. G. Patane and M. Russo. The enhanced LBG algorithm. IEEE Transactions on Neural Networks, vol. 14, no. 9, pp. 1219–1237, November 2001.

4. Kruskall, J. & M. Liberman. The Symmetric Time Warping Problem: From Continuous to Discrete. In Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, pp. 125–161, Addison-Wesley Publishing Co., Reading, Massachusetts, 1983.

5. Marie Roch. Cepstral processing. San Diego State University. http://www-rohan.sdsu.edu/~mroch/cs682/slides/06Cepstral.pdf.

6. Sakoe, H. & S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-26, 1978.

7. T. Kinnunen, P. Fränti: «Speaker discriminative weighting method for VQ-based speaker identification», Proc. 3rd International Conference on audio-and video-based biometric person authentication (AVBPA), pp. 150–156, Halmstad, Sweden, 2001.

8. T. Kinnunen, T. Kilpeläinen, P. Fränti: «Comparison of clustering algorithms in speaker identification», Proc. IASTED Int. Conf. Signal Processing and Communications (SPC 2000), pp. 222–227, Marbella, Spain, 2000.

9. Ахмад, Х.М. Параметрическое представление речевого сигнала для задачи распознавания спикера. Применение mel-частотных кепстральных коэффициентов / Х.М. Ахмад // Математические методы в технике и технологиях – ММТТ-20: сб. тр. XX Междунар. науч. конф. в 10 т. / под общ. ред. В.С. Балакирева. – Ярославль, 2007. – Т. 6. – С. 66–68.

10. Рабинер, Л.Р. Цифровая обработка речевых сигналов / Л.Р. Рабинер, Р.В. Шафер // М. : Радио и связь, 1981. – 496 с.

11. Рабинер, Л.Р. Теория и применение цифровой обработки сигналов / Л.Р. Рабинер, Б. Гоулд. – М. : Мир, 1978. – 848 с.

Decision-Making Mathematical Models for Tasks of Speaker's Recognition

Kh.M. Akhmad

Department of Computing, Vladimir State University

Key words and phrases: codebook; dynamic distortion of time; speech signal; vector quantization.

Abstract: Methods of identifying the most informative characteristics of speech signal, as well as methods of recognition based on the application of algorithms for dynamic time distortion (**DTD**) and vector quantization (**VQ**) are considered. As the main type of speech recognition systems we examine the identification systems based on DTD and VQ aimed at compiling codebooks of speech signal samples.

Matematische Modelle der Beschlüssenfassung in den Aufgaben der Sprechererkennung

Zusammenfassung: Es werden die Methoden der Absonderung der am meisten informativen Charakteristiken des Sprechsignales, die Methoden der Erkennung, die auf der Nutzung der Algorithmen der dynamischen Entstellung der Zeit (**DEZ**) und der Vektorquantisierung (**VQ**) gegründet ist, betrachtet. Als der Haupttyp der Systeme der Sprechererkennung werden die Systeme der Identifizierung mit der Anwendung von

DEZ und VQ mit der Schaffung der Kodebücher der Muster des Sprechsignales betrachtet.

Modèles mathématiques de la prise de solution dans les problèmes du décodage de la personne qui parle

Résumé: Sont examinées les méthodes de la déduction des caractéristiques les plus informatives du signale de la parole, les méthodes du décodage fondées sur l'emploi des algorithmes du décalage dynamique du temps (**DDT**) et de la quantification vectorielle (**QV**). En qualité du type essentiel des systèmes du décodage de la personne qui parle sont examinés les systèmes. d'identification avec l'utilisation des DDT et QV avec la création des livres du codage des échantillons des signaux de la parole.