УДК 681.3.07

М. Н. Фаворская, Н. Д. Торгашин, А. Г. Зотин

ПРОГНОЗИРОВАНИЕ В СИСТЕМАХ РАСПОЗНАВАНИЯ ОБРАЗОВ НА ОСНОВЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ

Показана возможность применения скрытых марковских моделей в различных предметно-ориентированных системах распознавания образов. Анализируются алгоритмы прогнозирования на их основе. Рассмотрены возможные топологии скрытых марковских моделей. Приведен подробный пример распознавания речевых образов конечного тематического словаря.

В теории распознавания образов имеется ряд задач, в которых используется не адаптация вероятностных моделей, а метод сокращения поиска, аналогичный схеме интерпретационного дерева. Основными составляющими такого метода являются следующие допущения:

– имеется последовательность случайных переменных, каждая из которых условно независима от всех других, кроме предшествующей переменной;

– каждая случайная переменная характеризуется измерениями, распределение вероятностей которых зависит от состояния.

Таким элементам соответствует очень эффективная формальная модель, известная как *скрытая марковская модель* (СММ). Эти модели успешно используются в системах распознавания речи. При этом скрытые состояния описывают речевую систему, а под наблюдениями понимаются различные акустические измерения. Обычно СММ соотнесена с каждым словом. Далее эти модели связываются воедино с использованием модели языка, которая задает вероятность появления следующего слова в зависимости от распознанного текущего слова. В результате получается другая (возможно, большая) скрытая марковская модель. Предложение, представленное набором акустических измерений, строится с помощью алгоритма логического вывода, примененного к модели языка. Данную методику можно перенести и на язык жестов.

Человеческие жесты подобны звукам человеческой речи: обычно имеются определенная последовательность событий и измерения, полученные от этих событий, но не определяющие их. Программа чтения языка жестов по видеозаписи некоторого человека должна давать для каждого жеста заключение о внутреннем состоянии. Вывод о нем программа делает, исходя из измерений положения рук, которые, как правило, не бывают точными и выверенными, а зависят (в идеальном случае – весьма сильно) от эмоционального состояния. Жесты следуют друг за другом случайно, но достаточно упорядоченно. При этом некоторые последовательности состояний появляются редко или не появляются никогда. Это означает, что для определения реальной картины можно использовать и измерения, и относительные вероятности различных последовательностей жестов. Данный подход также применим к построению формальных систем жестов. Например, если требуется разработать систему, которая включает телевизор при одном движении руки и выключает его при другом, то данный набор жестов можно рассматривать как сильно ограниченный язык.

Последовательность случайных переменных X_n называется марковской цепью, если выполняется равенство

$$P(X_n = a \mid X_{n-1} = b, X_{n-2} = c, ..., X_0 = x) = \mathrm{P}(X_n = a \mid X_{n-1} = b),$$

и однородной марковской цепью, если данная вероятность не зависит от параметра n. Марковские цепи можно рассматривать как последовательности с небольшой памятью, в которой новое состояние зависит только от предыдущего состояния, а не от всей предыстории. Данное свойство очень полезно при моделировании, поскольку на его основе можно построить множество простых алгоритмов логического вывода. Отметим, что формы записи марковских цепей в дискретных и непрерывных пространствах состояний несколько различны и далее мы будем рассматривать только дискретный случай.

Предположим, что дано дискретное пространство состояний (при этом размерность пространства не существенна). Обозначим элементы пространства как s_i и предположим, что всего имеется k элементов. Также примем, что дана последовательность случайных переменных, принимающих значения из этого конечного пространства состояний, причем переменные формируют однородную марковскую цепь. Далее запишем

$$P(X_n = s_j \mid X_{n-1} = s) = p_{ij},$$

и поскольку цепь не зависит от n, то от n также не зависит p_{ij}. Далее можно записать матрицу $\mathbf{P}$, элемент p_{ij} которой описывает поведение цепи. Такая матрица называется матрицей переходов. Предположим, что X_0 имеет распределение вероятностей $P(X_0 = s_i) = \pi_i$, и запишем $\boldsymbol{\pi}$ как вектор, i-й элемент которого равен π_i. Это означает, что

$$P(X_1 = s_j) = \sum_{i=1}^{k} P(X_1 = s_j \mid X_0 = s_i)\, P(X_0 = s_i) =$$

$$= \sum_{i=1}^{k} P(X_1 = s_j \mid X_0 = s_i)\, \pi_i = \sum_{i=1}^{k} p_{ij}\, \pi_i,$$

так что распределение вероятностей для состояния X_1 записывается как $\mathbf{P}^T\boldsymbol{\pi}$. Следуя аналогичным рассуждениям можно показать, что распределение вероятностей для состояния X_n описывается выражением $(\mathbf{P}^T)^n\boldsymbol{\pi}$. Для всех марковских цепей существует по крайней мере одно такое распределение $\boldsymbol{\pi}^s$, что $\boldsymbol{\pi}^s = \mathbf{P}^T\boldsymbol{\pi}^s$. Оно называется стационарным распределением цепи.

Доказано, что марковские цепи позволяют создавать довольно простые информативные схемы. Так, например, можно изобразить взвешенный направленный граф с узлом для каждого состояния и весовым коэффициентом на каждом ребре, который указывает вероятность перехода между состояниями. В простой марковской цепи (рис. 1) вероятность перехода из состояния 1 в состояние 2 равна p, для перехода из состояния 1 в состояние 1 эта вероятность составляет $(1 - p)$ и т. д. Данную цепь можно

описать матрицей переходов. Стационарное распределение цепи равно $q/(p+q), p/(p+q)$. Если значение p мало, а значение q близко к единице, то цепь будет находиться в основном в состоянии 1. Если обе величины: p и q – малы, то цепь длительное время будет находиться в одном из состояний, перемещаться в другое состояние и там проводить значительный интервал времени.

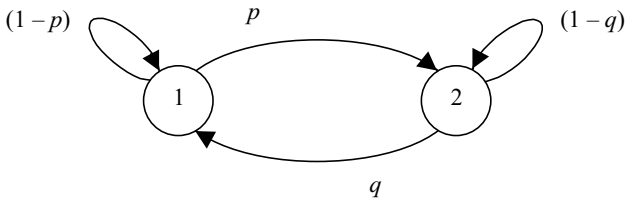


Рис. 1. Простая марковская цепь из двух состояний

При наблюдении случайной переменной X_n логический вывод прост, поскольку известно, в каком состоянии находится цепь. Однако такая модель не является адекватной моделью наблюдаемого объекта. Лучшую модель можно получить, приняв, что для каждого элемента последовательности наблюдается иная случайная переменная, распределение вероятности которой зависит от состояния цепи, а именно: от некоторой переменной Y_n, причем распределение вероятностей в i-й точке определяется как

$$P(Y_n \mid X_n = s_i) = q_i(Y_n).$$

Эти элементы можно представить в виде матрицы $\mathbf{\Omega}$. Таким образом, для задания скрытой марковской модели требуется обеспечить процесс перехода между состояниями, связи между состоянием и распределением вероятности переменной Y_n, а также знать исходное распределение состояний. Это означает, что модель записывается как $(\mathbf{P}, \mathbf{\Omega}, \boldsymbol{\pi})$. Кроме того, предполагается, что пространство состояний имеет k элементов.

Рассмотрим, каким образом производятся вычисления с использованием СММ. Имеются две подзадачи этого процесса:

– логический вывод, когда требуется определить, какой исходный набор состояний привел к наблюдаемому результату. Это позволяет сделать заключение, какую песню поет певец или какие действия совершают движущиеся объекты;

– подбор, когда требуется выбрать СММ, хорошо представляющую последовательность предыдущих наблюдений.

Одним из наиболее эффективных подходов к решению подзадачи логического вывода является построение решетчатой модели и определение лучшего пути с помощью динамического программирования или алгоритма Витерби [1].

Пусть дан набор из N измерений Y_i, которые предположительно являются выходом скрытой марковской модели. Такие измерения можно поместить в структуру, называемую решеткой. Под решеткой понимается взвешенный направленный граф, состоящий из N копий пространства состояний, которые расположены столбцами, при этом каждому измерению соответствует столбец (рис. 2). С узлом, представляющим состояние X_i, в столбце, соответствующем Y_j, соотносится весовой коэффициент $\log q_i(Y_j)$. Элементы различных столбцов обрабатываются следующим образом. Рассмотрим столбец, соответствующий Y_j. Элемент в данном столбце, представляющий состояние X_k, объединяется с элементом в столбце, соответствующем Y_{j+1} и представляющем состояние X_l, если p_{kl} не равно нулю. Ребро – это возможный переход между данными состояниями. Весовой коэффициент этого ребра равен $\log p_{kl}$.

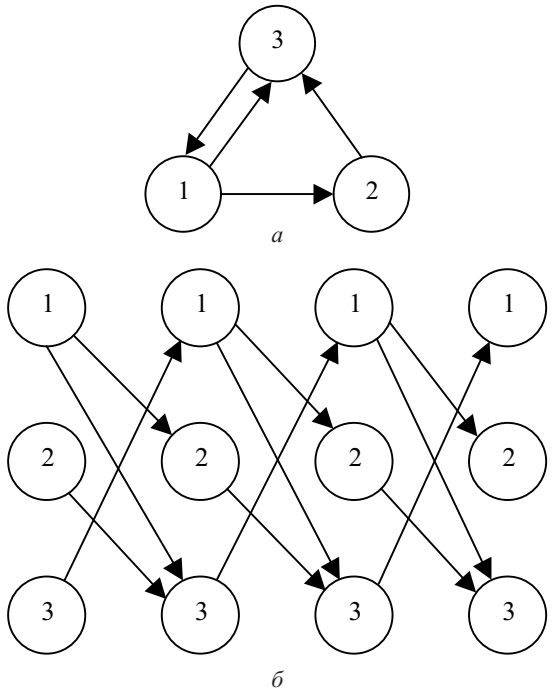


Рис. 2. Построение решетчатой модели: *а* – простая модель переходов между состояниями; б – решетка, соответствующая данной модели (весовые коэффициенты не обозначены)

Решетка обладает следующим свойством: каждый направленный путь через решетку представляет возможную последовательность состояний. Поскольку каждая вершина решетки умножается на весовой коэффициент, равный логарифму вероятности ухода из вершины, а каждое ребро учитывается с весовым коэффициентом, равным логарифму вероятности перехода, то правдоподобие последовательности состояний можно получить, определив путь, соответствующий данной последовательности, и просуммировав весовые коэффициенты ребер и вершин вдоль пути. В результате к такой структуре применимы методы динамического программирования, а также алгоритм Витерби для получения эффективного поиска максимально правдоподобного пути.

Работа алгоритма по методу динамического программирования начинается с последнего столбца решетки. Изначально известен логарифм правдоподобия пути с одним состоянием, оканчивающимся в каждой вершине, поскольку данное значение является весовым коэффициентом этой вершины. Затем рассмотрим путь с двумя состояниями, который начинается в предпоследнем столбце решетки. Далее легко получить наилучший путь, выходящий из вершины в этом столбце. Перейдем теперь к вершине. Нам известен весовой коэффициент каждого ребра, покидающего вершину, и весовой коэффициент

вершины в конце ребра. Таким образом, можно выбрать сегмент пути с наибольшим значением суммы, и такое ребро будет наилучшим ребром, выходящим из данной вершины. Для каждой вершины ее весовой коэффициент складывается со значением наилучшего сегмента пути, покидающего вершину весовой коэффициент ребра плюс весовой коэффициент конечного узла. Такая сумма будет наилучшим значением, которое можно получить при достижении данной вершины (назовем это значение значением вершины).

Поскольку известно наилучшее значение, которое можно получить при достижении каждой вершины во втором от конца столбце, то можно вычислить наилучшее значение, получаемое при достижении каждой вершины в третьем от конца столбце. Затем в каждой вершине третьего от конца столбца проверяются ребра, каждое из которых входит в вершину, с известным значением. Выберем ребро с наибольшим значением величины (вес ребра плюс значение вершины), прибавим это значение к весовому коэффициенту начальной вершины в третьем от конца столбце и получим значение начальной вершины. Данный процесс можно повторять до получения значения каждой вершины первого столбца, при этом наибольшее значение является максимально правдоподобным. Таким образом получается путь с максимальным значением функции правдоподобия. При вычислении значения вершины удаляются все ребра, покидающие вершину, кроме лучшего ребра. После достижения первого столбца далее просто необходимо идти по пути от узла с наилучшим значением.

Вернемся к постановке задачи логического вывода. Пусть имеется ряд наблюдений $\{Y_0, Y_1, ..., Y_n\}$ и требуется получить последовательность $n+1$ состояний $\boldsymbol{S} = \{S_0, S_1, ..., S_n\}$, максимизирующих величину

$$P(\boldsymbol{S} \mid \{Y_0, Y_1, ..., Y_n\}, (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi})),$$

что равносильно максимизации совместного распределения

$$P(\boldsymbol{S}, \{Y_0, Y_1, ..., Y_n\}, (\mathbf{P}, \boldsymbol{\Omega}, \mathbf{p})).$$

Для решения поставленной задачи, как правило, применяется алгоритм Витерби, в котором проверяется $(n+1)$-й путь от состояния S_0 до состояния S_n. Всего существует k^{n+1} таких путей, поскольку в каждом состоянии можно выбирать любой элемент пути (предполагается, что в матрице $\mathbf{P}$ отсутствуют нули и в большинстве случаев количество путей составляет $O(k^{n+1})$). Однако проверять каждый путь здесь не требуется.

Опишем общий подход к реализации данного алгоритма. Предположим, что для каждого возможного состояния s_l известно значение совместной вероятности для лучших путей из n элементов, оканчивающихся в $S_{n-1} = s_l$. Таким образом, путь, максимизирующий по значениям мест соединений путь в $(n+1)$-й элемент, должен включать один из упомянутых выше путей, удлиненный на один шаг.

Нахождение пути с максимальным значением совместной вероятности можно сформулировать как задачу с индукцией. Предположим, что для каждого значения j состояния S_{t-1} известно значение фрагмента наилучшего пути, оканчивающегося в $S_{t-1} = j$, которое запишем как

$$\delta_{t-1}(j) = \max_{S_0, S_1, ..., S_{n-2}} P(\{S_0, S_1, \ldots, S_{t-1} = j\},$$
$$\{Y_0, Y_1, \ldots, Y_{t-1}\} \mid (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi})).$$

Теперь получим

$$\delta_t(j) = (\max_i \delta_{t-1}(i)\, P_{ij}\,)\, q_j\,(Y_t)$$

При этом должны быть известно не только максимальное значение, но и путь, приводящий к этому значению. Определим переменную

$$\psi_t(j) = (\arg\max \delta_{t-1}(i)\, P_{ij}\,),$$

– лучший путь, оканчивающийся в $S_t = j$. Таким образом, индуктивный алгоритм получения наилучшего пути можно определить следующим образом. Пусть известен наилучший путь в каждое состояние для $(t-1)$-го измерения. Для каждого состояния при t-м измерении вычисляется наилучшее состояние при $(t-1)$-м измерении. Поскольку известен лучший путь из текущей точки, то можно предположить, что имеется лучший путь в каждое состояние и для t-го измерения. При этом считается, что лучший путь к каждому доступному состоянию на первом шаге известен.

Остановимся на задаче выбора скрытой марковской модели, наилучшим образом представляющей набор данных. Для этого можно использовать версию алгоритма ожидания–максимизации (ОМ-алгоритм). В этом алгоритме предполагается, что дана скрытая марковская модель $(\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi})$. Требуется использовать эту модель для конкретного набора данных, чтобы получить и оценить новый набор значений указанных параметров $(\mathbf{P}^*, \boldsymbol{\Omega}^*, \boldsymbol{\pi}^*)$. Имеются два варианта решения [2]: $P(\boldsymbol{Y} \mid (\mathbf{P}^*, \boldsymbol{\Omega}^*, \boldsymbol{\pi}^*)) > P(\boldsymbol{Y} \mid (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi}))$ или $(\mathbf{P}^*, \boldsymbol{\Omega}^*, \boldsymbol{\pi}^*) = (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi})$, т. е. данная итерация гарантированно сходится к локальному максимуму величины $P(\boldsymbol{Y} \mid (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi}))$. Обновленные значения параметров модели имеют следующий вид:

– p^*_{ij} вычисляется как отношение ожидаемого числа переходов из состояния s_i в состояние s_j к ожидаемому числу переходов из состояния s_i;

– $q^*_j(k)$ определяется как отношение ожидаемого времени нахождения в состоянии s_i и наблюдения $\boldsymbol{Y} = y_k$ к ожидаемому времени нахождения в состоянии s_j;

– π^*_i равно ожидаемой частоте нахождения в состоянии s_i в момент 0.

Требуется вычислить данные выражения. В частности, необходимо определить величину

$$P(X_t = s_i, X_{t+1} = s_j \mid \boldsymbol{Y}, (\mathbf{P}, \boldsymbol{\Omega}, \boldsymbol{\pi})\,),$$

которую обозначим как $\xi_t(i, j)$. Если известно значение $\xi_t(i, j)$, то получаем, что ожидаемое числа переходов из состояния s_i в состояние s_j равно

$$\sum_{t=0}^{n} \xi_t(i, j).$$

Ожидаемое число попаданий в состояние s_i равно ожидаемому числу переходов из состояния s_j:

$$\sum_{t=0}^{n} \sum_{j=1}^{k} \xi_t(i, j).$$

Ожидаемая частота нахождения в состоянии s_i в момент 0 определяется как

$$\sum_{j=1}^{k} \xi_0(i, j),$$

а ожидаемое время нахождения в состоянии s_i и наблюдения $(\boldsymbol{Y} = y_k)$ – как

$$\sum_{t=0}^{n}\sum_{j=1}^{k}\xi_i(i,\,j)\,\delta(Y_t, y_k),$$

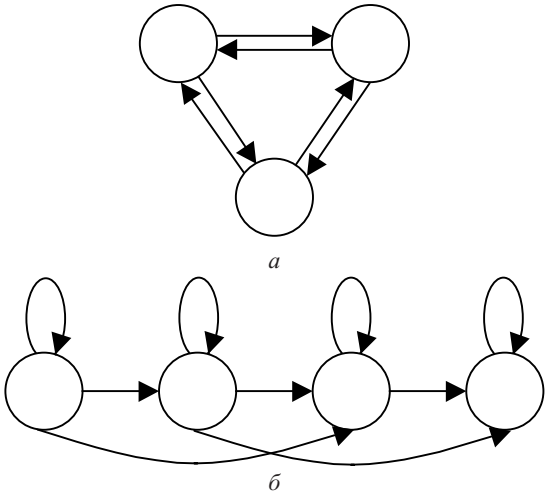
где выражение $\delta(u, v)$ равно единице, если аргументы равны, и равно нулю в противном случае. Для оценки выражения $\xi_t(i,j)$ требуется определить две промежуточных переменных: прямую переменную

$$\alpha_t(j) = P(Y_0, Y_1, ..., Y_t, X_t = s_j \mid (\mathbf{P}, \mathbf{\Omega}, \boldsymbol{\pi}))$$

и переменную обратного пути (обратную переменную)

$$\beta_t(j) = P(\{Y_{t+1}, Y_{t+2}, ..., Y_n\} \mid X_t = s_j, (\mathbf{P}, \mathbf{\Omega}, \boldsymbol{\pi})).$$

Приведенные выше алгоритмы не учитывают топологии графа, на основе которого построена модель. Наиболее известны две разновидности топологии СММ (рис. 3): полный граф (рис. 3, *а*) и лево-правая модель, или модель Бакиса (рис. 3, *б*).



а

б

Рис. 3. Примеры топологии скрытых марковских моделей: *а* – полносвязная или эргодическая модель; *б* – лево-правая модель с четырьмя состояниями при условии $j > i + 2$

Граф, каждая вершина которого связана со всеми другими вершинами в обоих направлениях, называется полным графом. Недостатком использования полных графов является необходимость оценки большого числа параметров.

В практических приложениях обычно используют другие варианты. Так, лево-правая модель имеет следующее свойство: из состояния i модель может переходить только в состояние $j > i$. Это означает, что для $j < i$ вероятность перехода p_{ij} равна нулю. Более того, $\pi_1 = 1$ и для всех $i \neq 1$ вероятность p_i равна нулю. Это говорит о том, что матрица переходов $\mathbf{P}$ является верхней треугольной матрицей. И лево-правая модель с n состояниями после перехода в n-е состояние будет в нем оставаться.

Следует отметить, что в данной модели фиксируется порядок, в котором могут происходить события. Это может быть удобным, если скрытые марковские модели используются для кодирования различной информации о движении или речевой информации.

Общая лево-правая модель предполагает, что большое число событий может пропускаться, т. е. состояние процесса может развиваться свободно. Однако на практике маловероятно, что будет пропущено не одно-два измерения или одно-два состояния, а большой набор состояний. Обычно принято использовать дополнительное условие, что для $j > i + \delta, p_{ij} = 0$. Здесь в качестве δ обычно выбирается небольшое значение.

Исследования показали, что ограничения топологии лево-правой модели не влияют на работу алгоритмов: если модель конкретной топологии имеет нули на определенных местах матрицы переходов, то новая оценка матрицы переходов будет иметь нули в тех же положениях.

В качестве примера рассмотрим математическую модель словаря распознавания звуковых образов слов «Да» и «Нет» (рис. 4). Он включает два уровня, представляющих различные аспекты речи человека: акустический и лингвистический.

Акустический уровень системы распознавания речи содержит множество скрытых марковских моделей отдельных слов. Скрытыми состояниями системы являются отдельные звуки речи (фонемы или их составные части – аллофоны), а в качестве наблюдений выступают элементы специальной кодовой книги. Эти элементы вычисляются при помощи метода векторного квантования на-

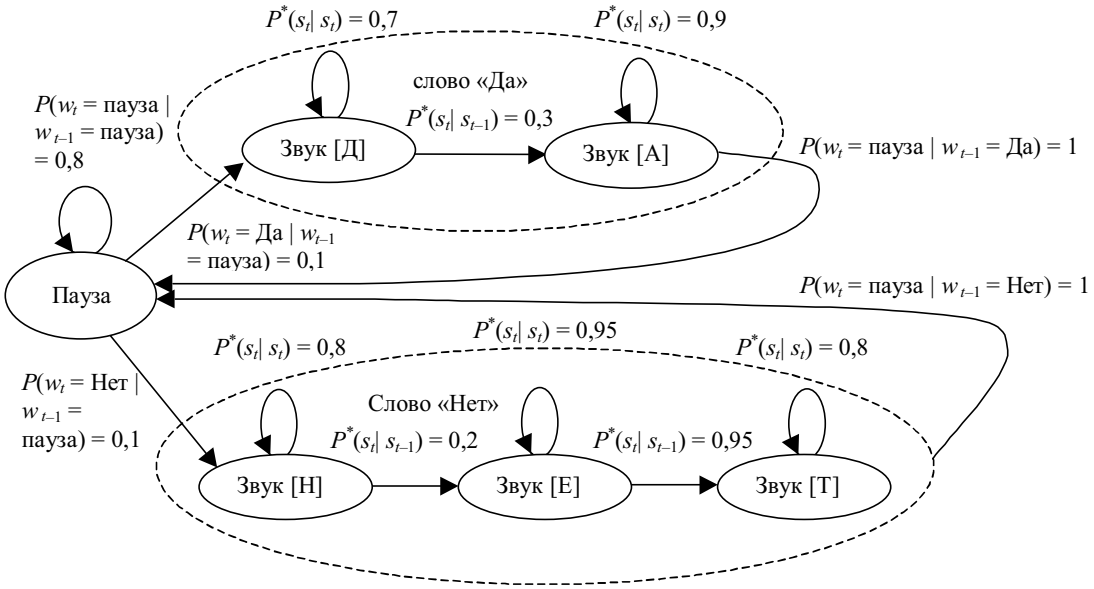


Рис. 4. Двухуровневая математическая модель словаря распознавания (для слов «Да» и «Нет»)

бора акустических измерений, получаемого на этапе анализа звукового сигнала.

Лингвистический уровень представляет собой обобщенную модель словаря распознавания. Состояния системы на данном уровне представлены отдельными словами. Для определения параметров модели требуется задать множество исходных вероятностей выбора начального состояния и условные вероятности переходов между состояниями. Обычно эту информацию получают путем статистического анализа достаточно большого по объему текста на языке распознавания.

При практической реализации системы распознавания речи нецелесообразно производить построение, определять параметры и хранить в памяти отдельные СММ для каждого распознаваемого системой слова, что приводит к неоправданному увеличению требований к занимаемой памяти и снижению производительности, которая является критичной для обеспечения возможности работы в реальном масштабе времени. Более рациональным является разбиение слова распознаваемого словаря на некоторые составные части, для каждой из которых строится СММ на акустическом уровне. В процессе работы алгоритма распознавания построение моделей отдельных слов производится путем динамического объединения моделей составных частей, входящих в их состав. В то же время на лингвистическом уровне модель словаря должна в полной мере определять вероятности произнесения и вероятности взаимных переходов для всех распознаваемых слов. На этом уровне модель применяется без каких-либо модификаций, а необходимая производительность достигается за счет применения специализированных алгоритмов, учитывающих разряженность матрицы вероятностей модели.

Таким образом, самым важным преимуществом СММ является простота логического вывода, которая вытекает из сильных структурных ограничений модели. В связи с этим важной темой исследований в сфере распознавания образов является поиск моделей, обладающих следующими свойствами: получением кадра с хорошей разрешающей способностью, возможностью использования относительно простых алгоритмов логического вывода, возможностью комбинации нескольких моделей для получения новых моделей.

Библиографический список

1. Volger, C. Parallel hidden markov models for American sign language recognition / C. Volger, D. Metaxas // Proceedings. Seventh International Conference on Computer Vision. 1999. P. 116–122.

2. Форсайт, Д. А. Компьютерное зрение. Современный подход : пер. с англ. / Д. А. Форсайт, Ж. Понс. М. : Вильямс, 2004. 928 с.

M. N. Favorskaya, N. D. Torgashin, A. G. Zotin

THE PREDICTION IN SYSTEMS OF PATTERN RECOGNITION BASED ON HIDDEN MARKOV MODELS

The using of hidden markov models in variable subject-oriented systems of pattern recognition is represented in this paper. The algorithms of prediction based on these models are analyzed. The possible topologies of hidden markov models are considered. The example of recognition of sound patterns in restrict dictionary based on this model is discussed.