ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

Сборник трудов
Института математики СО АН СССР

1966 г. Выпуск 22

ОБ АЛФАВИТЕ ОБЪЕКТОВ РАСПОЗНАВАНИЯ

В.Н. Елкина, Н.Г. Загоруйко

§ I. Введение

Под алфавитом S объектов распознавания, или алфавитом образов, мы понимаем перечень фиксированных областей, на которые разделено выборочное пространство распознающего автомата. В работе [I] отмечалось, что многие распознающие устройства обычно удается расчленить на последовательную цепочку элементарных распознающих автоматов. Каждый элементарный автомат работает со своим выборочным пространством и, следовательно, имеет свой алфавит объектов распознавания. Перечень тех образов, которые отличает друг от друга распознающее устройство, фактически совпадает с алфавитом последнего элементарного автомата этого устройства. В дальнейшем мы будем обозначать через S_0 алфавит распознающего устройства в целом, а через S_1 - алфавит элементарного автомата, входящего в его состав.

В задачах распознавания речи A_0 определяется целевым назначением устройства. Обычно известно, какие элементы (фонемы, слоги) или какой набор устных команд требуется распознавать. Выбор элементов, по которым производится промежуточная перекодировка, то есть элементов S_1, делается, исходя из следующих очевидных требований:

I. Количество (K) элементов S_1 должно быть по возможности малым.

2. Алгоритм выделения этих элементов из слитной речи, (т.е.

алгоритм сегментации (A))должен быть по возможности прос - тым.

3. Алгоритм распознавания (δ) этих элементов должен быть несложным.

4. Элементы S_1 должны обеспечивать распознавание речевой информации, объём (W) которой задан элементами S_0 .

5. Элементы S_1 должны обеспечивать заданную надежность распознавания (P) элементов S_0 .

Различные варианты выбора S_1 можно было бы сравнить между собой по величине

$$R' = f\left(\frac{W_1 P}{KAD}\right), \qquad (I)$$

если бы был известен вид функции f . Однако мы не умеем оце - нивать по одной и той же шкале количество элементов алфавита, сложность алгоритма сегментации, сложность решающего правила и надежность распознавания. Обобщенной оценкой рациональности распознающего устройства может служить отношение (R) количества информации ($\mathcal{I}$), которое это устройство извлекает ($\mathcal{I}$ определяется количеством элементов S_0 (и надежностью их распознавания) к суммарной стоимости (n) затрат (аппаратуры, машинного времени и т.д.) на извлечение этой информации:

$$R = \frac{\mathcal{I}}{n} . \qquad (2)$$

Так что элементы из соотношения (I) могут служить лишь ориентирами, о которых нужно помнить при выборе S_1 . Окончательная оценка того, насколько удачно выбран S_1 , дается соотношением (2).

Элементом S_1 является отрезок речевого сигнала минимальной длительности, относительно которого элементарный автомат принимает окончательное решение. Это могут быть либо отрезки разной длительности по времени (например, решение принимается о каждом отрезке речевого сигнала длительностью в 20 мсек) , либо отрезки, квазистационарные по какой-нибудь системе параметров (например, стационарные участки гласных или согласных), либо отрезки речи между какими-нибудь характерными точками(отрезки между серединами стационарных участков соседних гласных) и т.д. В фонетической терминологии элементами S_1 могут быть отдельные слова и словосочетания, слоги, фонемы и сегменты . (Будем пользоваться этими терминами, отдавая себе отчет в том, что строгого определения некоторых из них, в частности "фонемы", в фонетике нет).

Значения величин, входящих в уравнение (I), для каждого из этих элементов будут различными. Крупные элементы более устойчивы по отношению к искажениям, вносимым аппаратурой и дикторами, в результате чего можно ожидать более высокой надежности их распознавания.Однако количество слов (а для устной речи самостоятельным звуковым образом является каждая слово - форма) очень велико.

Использование только мелких элементов речи (фонем и сег - ментов) представляет собой другую крайность. Алфавит в этом случае будет небольшим, но надежность выделения и опознавания этих элементов в слитной речи высокой быть не может из-за их слабой помехоустойчивости.

При распознавании речевых сообщений без существенного ограничения на словарь, т.е. при наиболее полной имитации функций восприятия речи человеком, в качестве элементов S_1 ,вероятно, окажется целесообразно использовать те же элементы, какие использует и человек.

Результаты последних исследований, проведенных в области психоакустики [2] и экспериментальной фонетики [3], указывают на предпочтительность использования элементов типа открытых слогов. Перекодировка на уровне открытых слогов оказывается наиболее естественной для человека. Вместе с тем имеются основания утверждать, что проблема членения речевого потока на открытые слоги разрешима уже на современном этапе исследований.

Сведения о фонемном составе русской речи имеются в публикации [4] . Некоторые данные по статистике слов приведены в работах [5,6] . Статистическая обработка слогового состава русской речи была проведена в Институте математики СО АН СССР [7, 4] . Таблица частости встречаемости открытых слогов приведена в работе [8][x]. В текстах объемом III000 слогов было обнаружено II39 различных открытых слогов, встретившихся 5 и более раз и составивших 95,44% от общего числа слогов. 386 слогов встретились в тексте 40 и более раз и в сумме составили 85,67%.Первые 40 слогов частотной таблицы составляют 38,14% от общего числа слогов.На рис.I приведен график зависимости между числом наиболее часто встречающихся открытых слогов и покрываемым ими

x) Данные по статистике открытых слогов получены Б.Н. Елкиной и Л.С. Юдиной.

объемом текста.

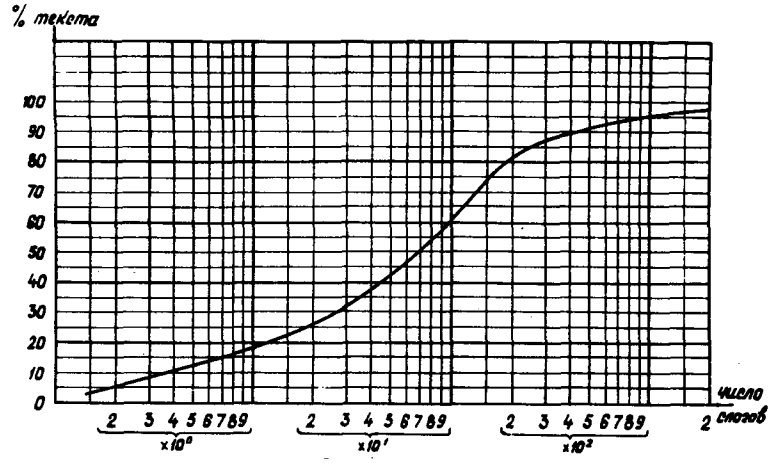

Рис. I. Зависимость объема текста от числа открытых слогов.

По количеству фонем и фонемному составу слоги, встретившиеся в тексте 5 раз и более, распределены следующим образом (в процентном отношении к общему числу слогов, встретившихся 5 и более раз):

Таблица I

Количество фонем в слоге	Фонетический состав слога	Количество различных слогов	Общее количество слогов	Занимаемый объем текста, %
I	Г	I7	I4.075	I8.29
2	СГ	285	56.990	53.80
3	ССГ	656	30.247	28.55
4	СССГ	I67	4.I63	3.93
5	ССССГ	I4	488	0.46

Можно ожидать, что речевой сигнал нетрудно будет членить на элементы, границы между которыми проходят после гласного и перед согласным. Эти элементы типа СГ, СГГ, СССГ и т.д. можно назвать "формальными открытыми слогами". Сведения о формальных открытых слогах русского языка содержатся в работе [7].

§ 2. Алгоритм минимизации элементов алфавита A_I

Во многих задачах распознавания ограниченного набора устных команд оказывается, что количество элементов типа слог обычно превышает количество распознаваемых команд. Однако хорошо известно, что информация о словах, содержащаяся в слогах, избыточна. Вероятно, при заданной надежности распознавания слов можно распознавать лишь часть слогов, входящих в состав этих слов.

Устранение избыточных слогов может быть осуществлено с помощью алгоритма минимизации элементов алфавита элементарного автомата, представленного блок-схемой на рис. 2. Алгоритм состоит из операторов выбора эталонных и сокращения избыточных слогов, вычеркивания слов, однозначно определившихся по эталонным слогам, и других операторов. С помощью этих операторов выбираются в качестве эталонных только те слоги, которые неизбежно должны войти в состав минимального набора слогов.

Если в словаре окажутся группы слов, для которых применение всех приведенных операторов, кроме оператора $(C:\{S\})$, не позволяет однозначно определить минимальный набор эталонных слогов, то к этим группам (тупиковым матрицам $S_{T\Pi}$) применяется оператор ($C:\{S_{T\Pi}\}$) разрыва тупиковых матриц. Процедура выбора эталонных слогов продолжается до тех пор, пока не будет найден минимальный набор слогов, однозначно определяющий весь исходный словарь.

Введем некоторые определения и обозначения.

Исходный словарь S состоит из групп Q_i $(i=1,\dots,r)$, содержащих слова $q_j^{(i)}$ $(j=1,\dots,\ell^{(i)})$ с одинаковым количеством слогов i. Полный набор слогов t_K, встречающихся в исходном словаре S, обозначим через T. Слог t^*, входящий в состав минимального набора слогов, необходимых для распознавания слов данного словаря назовем э т а л о н н ы м. Словарь представлен в виде матрицы, в которой строки соответствуют словам q_j, а столбцы - слогам t_K. На пересечении столбцов и строк стоят числа, равные порядковому номеру слога в данном слове. Множество слогов, входящих в

слово q_j, обозначим ($\mathcal{M}q_j$). Назовём о р и г и н а л ь н ы м слогом t^o слог, содержащийся только в одном из слов, не определившихся по эталонным слогам к данному моменту; к в а з и о р и г и н а л ь н ы м -слог, встретившийся только в группе слов Q_X, выделенной оператором X. Слог th , с помощью которого слово становится определенным однозначно, будем называть о п р е д е л я ю щ и м . Слово считается о п р е д е л е н н ы м, если существует комбинация одного или нескольких эталонных слогов, длины слова (i) и порядкового номера слога в слове, не повторяющаяся для других слов.

Группу слов, выделенную оператором X , обозначим $Q_X=\{q_j^{(X)}\}$. Если из группы Q_X исключается некоторое слово $q_\alpha^{(X)}$, то оставшуюся группу будем обозначать ($Q_X-q_\alpha^{(X)}$).

Введем следующие операторы:

1. $S\rightarrow\sum_{i=1}^{n}Q_i$ – разбиение исходного словаря S на группы Q_i, состоящие из слов с равным количеством i слогов;
2. $R:\{Q_i/\ell^{(i)}=1\}$ – поиск группы Q_i, в которой определены все слова, кроме одного.
3. $R:\{Q_i/\ell^{(i)}=0\}$ – поиск группы Q_i , в которой все слова определены.
4. $P:\{Q_i\}$ – поиск оригинальных слогов t^o, входящих в состав слов $q_j^{(i)}\in Q_i$.
5. $P_1:\{Q_F\}$ – поиск неквазиоригинального слога t_{P_1} в группе Q_F .
6. $W^*:\{q_j^{(i)}\}$ – маркировка слова $q_j^{(i)}$ в группе Q_i, определяемого только по количеству слогов.
7. $W:\{q_j\}$ – вычеркивание строки, соответствующей слову q_i
8. $V^*:\{t_K\}$ – маркировка слога, оставляемого в качестве эталонного.
9. $G:\{Q_i\}$ – маркировка группы Q_i , все слова в которой определены.
10. $A:\{S\}$ – поиск слов $q^{(A)}$, определившихся по эталонным слогам.
11. $B:\{S\}$ – поиск слова $q^{(B)}$, состоящего только из оригинальных неэталонных слогов t_K^o .
12. $D:\{S\}$ – поиск слова $q^{(D)}$, состоящего только из оригинальных неэталонных (t_K^o) и одного неоригинального определяющего слова t_n .
13. $E:\{Q_i\}$ – поиск в Q_i группы Q_E слов $q^{(E)}$ с определяющими оригинальными и одинаковыми неопределяю-

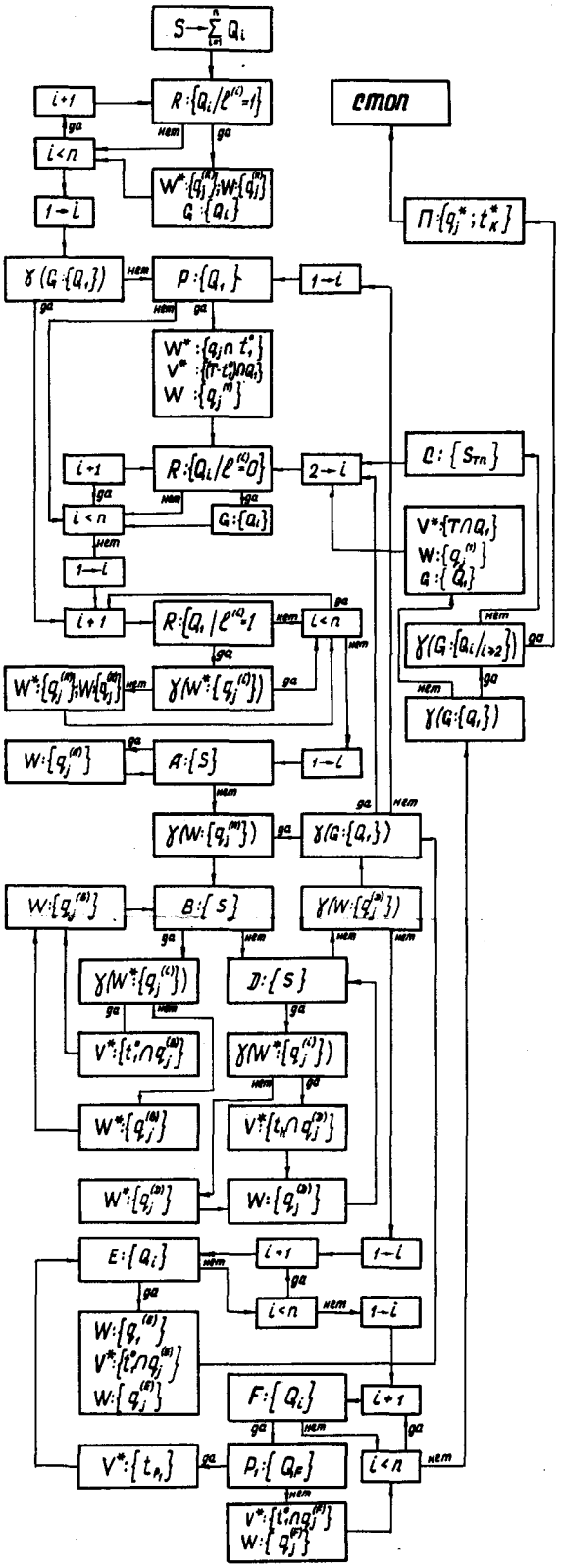

Рис.2. Блок-схема алгоритма минимизации элементов алфавита

щими слогами, причем среди одинаковых есть хотя бы один эталонный слог.

I4. $F:\{Q_i\}$ – поиск группы Q_r слов, у которых все определяющие слоги оригинальны, а неопределяющие слоги t_r одинаковы, причем среди одинаковых слогов все, или все, кроме одного, квазиоригинальны.

I5. $\gamma(X)$ – проверка, применялся ли оператор X.

I6. $C:\{S_{TП}\}$ – разрыв тупиковой матрицы.

I7. $П:\{X\}$ – печать массива X.

Проиллюстрируем на примерах работу отдельных операторов.

Так как мы имеет информацию о количестве слогов в слове, то одно слово $q_j^{(i)*}$ в каждой группе Q_i может быть распознано только по этому параметру. Очевидно, что для распознавания односложных слов (группы Q_1) необходимо оставить в наборе эталонных все слоги $t_k^*=(T\cap Q_1)$, кроме слога ($T\cap q_j^{(i)*}$), соответствующего слову, определяемому только по количеству слогов. В качестве этого слога (t_1^o) необходимо выбрать любой из оригинальных слогов группы ($T\cap Q_1$). Ясно, что исключение из эталонов оригинального слога дает оптимальное решение, так как неоригинальный слог, оставленный в качестве эталонного, может позволить определить и другие слова словаря. Для обработки группы Q_1 используются операторы $P:\{Q_1\}$, $W^*:\{q_j^*\cap t_1^o\}$, $W:\{Q_1\}$, $V^*:\{t_k^*\}$, $G:\{Q_1\}$. Если оригинального слога при первом применении оператора $P:\{Q_1\}$ не обнаруживается, то группа Q_1 оставляется для рассмотрения после анализа других групп. Если и после обработки остальной части словаря ($S-Q_1$) в группе Q_1 оригинальный слог не будет обнаружен, то в качестве эталонных остаются все слоги ($T\cap Q_1$).

После введения эталонных слогов, выделенных одним из операторов, применяется оператор $A:\{S\}$, позволяющий исключить из дальнейшего рассмотрения слова, однозначно определившиеся по всем введенным к данному моменту эталонным слогам. Вычеркивание строк $q^{(A)}$, найденных оператором $A:\{S\}$, может привести к появлению новых оригинальных слогов, что существенно упрощает процедуру минимизации количества слогов.

Слова $q^{(B)}$, целиком состоящие из оригинальных неэталонных слогов, могут быть определены по одному из этих слогов или по количеству слогов в слове. Если в рассматриваемой группе Q_i ещё не было выделено слово $q_j^{(i)*}$, то одно из слов $q^{(B)}$, най-

денных оператором $B:\{Q_i\}$, выделяется в качестве $q_j^{(i)*}$, а для распознавания остальных слов выбираются в качестве эталонных по одному слогу из каждого слова $(q_B - q_j^{(i)})^*$. Очевидно, что и этот блок алгоритма позволяет выделить минимальный набор эталонных слогов.

Если слово состоит только из оригинальных неэталонных слогов и одного неоригинального, но определяющего слога t_R, то ясно, что в качестве эталонного для распознавания этого слова необходимо оставить слог t_R, так как он, в отличие от оригинальных, может определить и другие слова. Отсюда следует оптимальность оператора $D:\{S\}$.

Оптимальность оператора $E:\{Q_i\}$ вытекает из следующего: если в группе Q_E, состоящей из слов, у которых все определяющие слоги оригинальны, а неопределяющие слоги t_F одинаковы, среди одинаковых есть хотя бы один эталонный, то дополнительно в качестве эталонных слогов нужно оставить по одному оригинальному слогу для всех слов, кроме одного слова группы Q_E.

Таблица 2

Слог / Слово	на	пря	же	ни	е	я	ю	Количество слогов (i)
Напряжение	I	2	③	4	5			⑤
Напряжения	I	2	③	4		⑤		⑤
Напряжению	I	2	③	4			⑤	⑤

В примере, приведенном в таблице 2, группу Q_E составляют слова "напряжение, напряжения, напряжению", причем слог "же" был ранее выделен в качестве эталонного. Тогда для определения слов этой группы необходимо и достаточно включить в набор эталонных любые два из трех оригинальных слогов "е", "я", "ю".

Оператор F выделяет группу Q_F слов, в которых все определяющие слоги оригинальны, а неопределяющие одинаковы, но в отличие от слов группы Q_E, среди одинаковых слов нет ни одного эталонного и все они или все, кроме одного, квазиоригинальны. Если такая группа Q_F найдена и в ней все одинаковые слоги квазиоригинальны, то в качестве эталонных следует оставить по одному оригинальному слогу в каждом слове. Если среди одинаковых окажется один неквазиоригинальный слог t_{p_i}, то его следует включить в эталонные и применить к этой группе оператор E.

Таблица 3

Слог / Слово	зна	чи	те	льно	мо	сть	сло	Количество слогов
значительно	I	②	3	4				④
значимость	I	②			③	4		④
число		①					2	②

В табл. 3 группу Q_F составляют слова "значительно", значимость". Так как слог "чи" неквазиоригинальный, то, оставив его в списке эталонов, сделаем группу Q_F эквивалентной группе Q_E. Добавление к эталонам любого из оригинальных слогов (например, "мо") однозначно определяет слова этой группы. Так как для распознавания n слов из групп Q_F необходимо иметь не менее n эталонных слогов, то включение в эти n слогов неквазиоригинального дает дополнительно возможность определить и слова, не входящие в группу Q_F (в приведенном примере слово "число").

Оператор C разрыва тупиковой матрицы $S_{T\Pi}$ выбирает наиболее информативный слог $t_{j_{max}} \in (T \cap S_{T\Pi})$. Наиболее информативным слогом $t_{j_{max}}$ является слог, введение которого в список эталонов позволяет в результате применения к матрице $S_{T\Pi}$ всех предыдущих операторов максимально уменьшить энтропию H матрицы $S_{T\Pi}$. Исходная энтропия H_0 матрицы $S_{T\Pi}$ определяется формулой:

$$H_0 = -\log_2 \frac{1}{n_{T\Pi}},$$

где $n_{T\Pi}$ – число слов исходной тупиковой матрицы $S_{T\Pi}$.

Введение в число эталонных слога $t_j \in (T \cap S_{T\Pi})$ может определить некоторые слова из $S_{T\Pi}$, что в итоге позволяет расщепить эту матрицу на N более мелких тупиковых матриц. Если в каждой из этих матриц S_N будет n_i слов, то энтропия станет равной

$$H = -\sum_{i=1}^{N} \frac{1}{n_i} \log_2 \frac{1}{n_i}.$$

Разность $\mathcal{I} = H_o - H$ может служить мерой информативности слога $t_{\mathcal{I}}$. Оценка $\mathcal{I}$ поочередно для всех слогов $t \in (T \cap S_{Tn})$ и выбор слога $t_{\mathcal{I}_{max}}$, дающего $\mathcal{I}_{max}$, позволяет решить задачу расщепления матрицы S_{Tn} наиболее эффективным способом. Разумеется, наилучшим (но не всегда достижимым) вариантом является выбор слога, информативность которого $\mathcal{I} = H_o$.

Описанный алгоритм позволяет выбрать для заданного исходного словаря S минимальный набор эталонных слогов, однозначно определяющих все слова $q_j \subset S$. Так, в результате применения алгоритма для 40 слов из текстов по радиоэлектронике, состоящих из 49 формально открытых слогов, были выбраны 20 эталонных слогов; для 100 слов, состоящих из 134 слогов, было выбрано 36 эталонных слогов. Для словаря, состоящего из 200 слов, в качестве эталонных потребовалось оставить лишь 56 из 280 разных слогов.

Алгоритм сокращения количества элементов алфавита пригоден для рассмотрения элементов любого типа. С его помощью можно узнать, какого количества и каких именно фонем или слогов достаточно для распознавания данного списка слов, каких дифференциальных признаков (или групп фонем) достаточно для распознавания данного набора слогов и т.д.

§ 3. Выбор формальных элементов алфавита (алгоритмы таксономии)

При распознавании речи, как и в ряде других задач, часто возникают задачи таксономии. Эти задачи (задачи распознавания "4-го" типа [1]) можно сформулировать следующим образом: множество Q реализаций q_i ($i = 1, \ldots, L$), заданных в пространстве признаков X, с помощью решающих функций δ (т.е. по определенному критерию "близости", "сходства") нужно разделить на такое количество и таких элементов алфавита S, чтобы потери, например, потери информации при перекодировке, не превышали заданной величины R.

Существует два подхода к выбору формальных элементов алфавита S:

а) элементы алфавита удовлетворяют любым формально заданным критериям, в том числе и "неестественным" с точки зрения человека, но удобным, например, для реализации на ЭВМ [9, 10, 11, 12].

б) элементы алфавита представляют собой подмножества, выделенные по критериям, "естественным" для человека [12].

Рассмотрим несколько алгоритмов, предложенных для решения задачи таксономии.

✓ Р.Е. Боннером [9] был предложен метод "масок", состоящий в следующем. Исходные реализации представлены двоичными словами, каждый двоичный разряд которых соответствует наличию (I) или отсутствию (0) определенного параметра. Пространство параметров предварительно выбирается человеком. Разделение исходного множества Q на непересекающиеся подмножества S_i производится на основании взаимного "сходства" между реализациями. В качестве критерия "сходства" устанавливается число совпадающих признаков, т.е. некоторый порог T. Берется первая попавшаяся реализация и запоминается как маска I. Следующая реализация сравнивается с маской I. Если в результате сравнения число совпавших разрядов $r \geq T$, то слова считаются "похожими". Второе слово отбрасывается. Если $r < T$, то новое слово запоминается как маска II и т.д. В примере, приведенном в таблице 4, T задано равным 3.

Таблица 4

№	Реализации	Номер маски	Маски
I	00011	I	00011
2	01011		
3	10111		
4	11001	2	11001
5	11100		
6	01110	3	01110
7	01010		

Эти маски будут использованы как основа для формирования "знака" для каждой реализации из Q. Все исходные реализации сравниваются со всеми масками, и формируется "знак" реализации - двоичное слово, в j-ом разряде которого стоит I при совпадении (с порогом T) с j-ой маской и 0 - при несовпадении. Все реализации с одним "знаком" помещаются в одну категорию, и этот знак служит "именем" категории. Так, например, сравнение образца "а" с масками I,2,3, дает следующий результат:

Таблица 5

образец "а"	маски	$r \geq T$	знак "а"
IIIOO	OOOII	нет	OII -
	IIOOI	да	
	OIIIO	да	

Достоинство этого алгоритма - в простоте его реализации. Недостаток - в том, что результат классификации существенно зависит от порядка поступления реализаций. Так, если те же реализации, что и в таблице 4, будут следовать в порядке (2,I,7, 3,5,4,6), то получаем маски (OIOII, IOIII, IIIOO) и для реализации IIIOO будет получен знак OOI.

Г.С. Себестианом [IO] предложен следующий алгоритм построения "совокупности обучающих изображений". Задается некоторый порог T. Вводится первое изображение, и оно принимается за среднее значение для некоторого класса. В памяти хранится среднее значение m_1 и число M_1 изображений в классе S_1. Следующее входное изображение $\bar{q}$ сопоставляется с m_1. Если $d(m_1,\bar{q})<T$, то оно включается в класс S_1, и характеристики класса пересчитываются:

$$m_1' = \frac{M_1 m_1 + \bar{q}}{M_1+1}\ ;\quad M_1' = M_1+1.$$

Если $d(m_1,\bar{q})>T$, то $\bar{q}$ считается средним значением m_2 следующего класса. Для всех изображений, лежащих внутри сферы радиуса T, в памяти сохраняется только одно среднее изображение и число изображений внутри этой области (примерно, пропорциональное априорной вероятности подкласса).

Алгоритм, предложенный Себестианом, легко реализуем. Однако следует отметить, что результат классификации входных сигналов зависит от порядка введения информации. Так, легко заметить, что если последовательно будут введены точки I и 2 (рис.3), то они будут отнесены к различным множествам. В то же время при случайном начальном выборе точки 3 точки I и 2 будут отнесены к одной области. К недостатку постепенной классификации методом Себестиана следует отнести и возможность по-

тери некоторых точек.

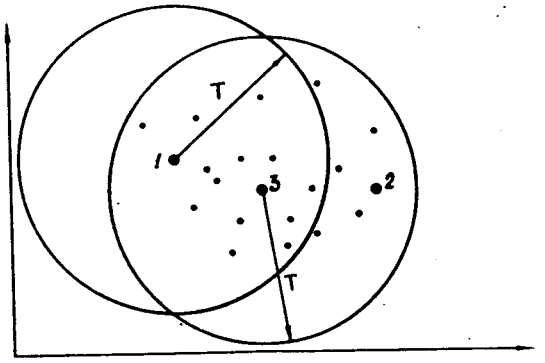

Рис. 3. Влияние порядка просмотра точек на результат таксономии.

Рассмотрим случай одномерного равномерного распределения. Пусть Δq - расстояние между соседними точками, так что $q_i = q_0 + j\Delta q$. Пусть также задан порог $T = N\Delta q$. Координата текущего центра тяжести

$$q_n' = \frac{\sum_{j=1}^{n} q_j}{n} = \frac{\sum_{j=1}^{n}(q_0 + j\Delta q)}{n} = q_0 + \frac{n-1}{2}\Delta q.$$

Найдем условие, при котором $q_n - q_0 > T$

$$q - q_0 = \frac{n-1}{2}\Delta q > N\Delta q\ ;\quad \frac{n-1}{2} > N\ ;\quad n > 2N+1,$$

то есть, начальные точки станут "выпадать" из области $d(q,q_n)<T$ как только она охватит число точек $n > 2N+1$.

Более интересной, чем предыдущие, является работа Шлезингера М.И. [II]. Её достоинством можно считать более четкую постановку задачи и введение такого критерия оценки качества решения, полученного алгоритмом, как функция потерь P. Эта функция потерь отличается от обычно вводимой в теории решений тем, что она задана не на множестве заранее известных элементов алфавита, а на исходном множестве реализаций. При этом задача таксономии может ставиться как задача нахождения такого разбиения множества Q на множество элементов S_i, при котором минимизируются средние потери P.

В этой работе также выражена тенденция разработать алгоритм, результаты которого не зависели бы от порядка осмотра точек. Однако конструктивный алгоритм, который описан автором, также зависит от порядка осмотра точек, как и приведенные выше алгоритмы.

В работе [12] рассмотрен алгоритм, в котором в качестве решающих функций использованы гиперсферы. Алгоритм состоит в следующем. Задается радиус R гиперсферы. Центр ($\bar{C}^{(1)}$) этой гиперсферы совмещается с любой точкой исходного множества Q реализаций. Определяются точки $q_i^{(1)}$, для которых $d=|\bar{q}^{(1)}-\bar{C}^{(1)}|\leq R$. Затем центр $\bar{C}^{(1)}$ гиперсферы смещается в центр тяжести $\bar{C}^{(2)}$ точек $q_i^{(1)}$,попавших на предыдущем шаге в гиперсферу. Вновь определяются точки $q_i^{(2)}$, для которых, как и на предыдущем шаге, $d=|\bar{q}_i^{(2)}-\bar{C}^{(2)}|\leq R$. Процедура повторяется до тех пор, пока не перестанут изменяться координаты центра тяжести $\bar{C}^{(j)}$. Очевидно, при этом гиперсфера остановится в области локального или главного экстремума плотности точек исходного пространства. Гиперсфера с центром в точке $\bar{C}^{(j)}$ и представляет собой формальный элемент алфавита S_1 . И только после остановки гиперсферы (в отличие от алгоритма Сэбестиана) точки $q_i \in S_1$ из дальнейшего рассмотрения исключаются. Точки же, через которые гиперсфера проходила на своем пути к конечному положению, но которые к моменту останова"выпали" из неё, сохраняются. Затем центр следующей гиперсферы совмещается с любой из точек множества $Q-S_1$, и процедура выделения элементов S_n повторяется до тех пор, пока все исходное множество Q реализацией q_i не будет разделено на элементы S_n $(n=1,\ldots,K)$. В итоге получены K элементов S_n, представленных центрами гиперсфер радиуса R . Очевидно,что K есть некоторая функция радиуса R . При заданном K по аналогии с критерием P, предложенным в [11], эквивалентом функции потерь может служить суммарной объем V_Σ K гиперсфер. Легко видеть, что минимум суммарного объёма достигается при минимальных значениях R, дающих разбиение множества Q на K элементов в S_n . Выбор R для заданного K производится методом последовательных приближений.

Применение данного алгоритма к некоторым задачам (из области палеонтологии и минералогии) показало, что формальные элементы S , выделенные алгоритмом, оказались приемлемыми с точки зрения специалистов для практического применения. Результат разбиения в этих задачах не зависел от порядка ввода

исходных данных. Однако в случае равномерного распределения точек в выборочном пространстве разбиение с помощью этого алгоритма, как и с помощью описанных ранее алгоритмов, будет зависеть от порядка ввода точек. Но даже в этом случае данный алгоритм не приводит к потере точек, как это имеет место в случае алгоритма Себестиана.

На материалах тех же задач делалась попытка установить формальный критерий выбора количества элементов (K), наиболее естественного для заданного множества Q. Оказалось,что при многократном применении алгоритма для разных значений R, наибольшее число раз получалось одно и то же значение K . В частности, в задаче классификации трилобитов в 13-мерном пространстве качественных признаков кранидиев при последовательном уменьшении R $(R_{i+1}=R_i-\Delta R)$ были получены следующие значения K : 1, 4, 5, 5, 5, 9, 27 ...

По-видимому, многократное повторение одного и того же K для нескольких последовательных значений R и резкое увеличение K на следующих шагах может служить некоторым основанием для выбора количества элементов алфавита. В приведенном выше примере исходный материал был разделен на 5 формальных элементов алфавита, что совпало с разбиением, произведенным специалистом-палеонтологом.

Так как в описанном алгоритме разделение множества точек осуществляется с помощью фигур строго определенной формы (гиперсферы), то это обстоятельство ограничивает область его применения. Было бы желательно иметь возможность выделять в заданном пространстве признаков области произвольной формы, если точки, лежащие в этих областях, удовлетворяют некоторым критериям "близости".

Была предпринята попытка установить, какими критериями "близости" пользуется человек при разбиении множества точек на некоторые подмножества. Испытуемым предъявлялись изображения точек на плоскости и предлагалось разделить это множество точек на K групп (задавалось конкретное значение K или указывались для него допустимые пределы). "Естественным" разбиением считалось то, которое осуществлялось большинством испытуемых В результате этих экспериментов выяснилось , что при разбиении человек учитывает ,кроме расстояния между точками, еще и степень однородности распределения этих точек, характер их локализации. Обычно в первую очередь человек выделяет группу наиболее близко расположенных друг к другу точек. Было установле-

но, что "естественное" разбиение может быть осуществлено следующим образом. В качестве критериев "близости" используется евклидово расстояние ρ между точками и функция[x)]

$$P_i = \sum_{j=1}^{L} \frac{1}{\rho_{ij}}, \quad i=1,\dots,L, \quad i \neq j.$$

Будем называть эту функцию P – потенциальной функцией.

На первом шаге алгоритма для каждой точки q_i вычисляются потенциальная функция P_i и евклидово расстояние до ближайшей точки ($\rho_{i\,min}$). Находится точка q_i с $P_{i\,max}$. Определяются точки q_j, удовлетворяющие условию "близости": "близкими" считаются точки, для которых

$$\Delta P_{ij} \leqslant \alpha; \quad \rho_{ij} \leqslant \beta \rho_{i\,min}, \text{ где } \Delta P_{ij} = \frac{|P_i - P_j|}{P_i + P_j}. \quad (3)$$

Точки q_j, удовлетворяющие этому условию, относятся к тому же подмножеству S_1, что и точка q_i, для этих точек вычисляется $\Delta P_{ij}, \rho_{ij}$. Точка q_i, сыгравшая роль центра "кристаллизации", из дальнейшего рассмотрения исключается. Среди оставшихся точек, отнесенных к этому времени к подмножеству S_1, вновь определяются точка q_j с $P_{j\,max}$ и для точек q_e, еще не объединенных в S_1, проверяется условие (I).

Точки, удовлетворяющие этим условиям, присоединяются к S_1. Точка q_j из дальнейшего рассмотрения исключается, и процедура повторяется до тех пор, пока на ($t+1$)-ом шаге не выполнится условие прекращения "процесса кристаллизации" $S_1^{(t+1)} = S_1^{(t)}$, т.е. в исходном множестве Q больше нет точек, удовлетворяющих условиям [3] для множества S_1. Среди точек множества $Q - S_1$ ищется точка с максимальным значением потенциальной функции, и с этой точки начинается процедура выделения множества S_2. Критерием окончания работы алгоритма является условие $\sum_{i=1}^{\kappa} S_i = Q$. В результате исходное множество Q разбивается на κ элементов алфавита S_i. Ясно, что κ – есть функция α и β. Если κ задано однозначно, то α и β могут быть найдены методом последовательных приближений. Если κ не задано, то многократное применение алгоритма для различных α и β даст несколько значений κ. Если для ряда значений α и β получено одинаковое κ и при этом на разных шагах (ℓ, m) $S_i^{\ell} \neq S_i^{m}$, то критерием для выбора наиболее предпочтительного разбиения исходного множества Q на κ элементов алфавита может служить минимум функции потерь:

$$P = \sum_{1}^{m} R_m, \text{ где } \quad R_m = \sum_{q_i \in S_m} \Delta P_{ij}\, \rho_{ij}.$$

Вопрос о формальных критериях выоора значения κ, наиоолее "естественного" для исходного множества Q, требует дальнейшего исследования.

Преимуществом данного алгоритма перед алгоритмами, использующими конкретные гиперповерхности (гиперсферы, гиперкубы и т.д.), является возможность выделения элементов алфавита S произвольной конфигурации. Так как выбор "центров" кристаллизации" осуществляется однозначно, то результат работы алгоритма, в отличие от алгоритмов, описанных выше, не зависит от порядка просмотра точек.

В заключение следует подчеркнуть, что результат деления множества Q на формальные элементы S определяется пространством признаков, которое задает специалист. Он же решает, на каком варианте разделения множества наиболее целесообразно остановиться. Поэтому приведенные алгоритмы могут рассматриваться только в качестве вспомогательного инструмента при решении специалистом задач таксономии.

Исключение составляют, как указывалось ранее, те случаи, когда алгоритмы таксономии применяются для промежуточной перекодировки, где могут быть использованы элементы S, "неестественные" с точки зрения специалиста. Так, при распознавании устных слов можно в качестве элементов перекодировки использовать наряду с фонемами и слогами отрезки речевого сигнала фиксированной длины (например, отрезки 20-миллисекундной длительности) или отрезки, граница между которыми проходит по серединам стационарных участков фонем и т.д.

x) Тип функции P был выбран в результате экспериментальной проверки серии функций вида $P_i = \sum_{j=1}^{n} \frac{1}{\rho_{ij}^{\kappa/2}}$, где $\kappa = 1,\dots,6$.

Л И Т Е Р А Т У Р А

I. Н.Г. Загоруйко. Классификаця задач распознавания . Данный сборник. стр. 3-I9.

2. Л.А. Чистович. Речь. Артикуляция и восприятие. М.-Л. "Наука", I965.

3. Л.Р. Зиндер. Фонетическая характеристика русских ударных слогов (Отчет каф. фонетики ЛГУ, I965).

4. В.Н. Елкина, Л.С. Юдина. Статистика слогов русской речи.Вычислительные системы, Новосибирск, I964. вып. I0, стр. 58.

5. Josselson H. The Russian word count and frequency analysis of gram. categories of stand. lit.Russian.Detr.I953.

6. П.М. Алексеев, В.М. Калинин, Е.А. Чернядьева. О статистических закономерностях в современных русских и английских текстах по электронике. - Вопросы радиоэлектроники, I963, серия УШ, вып.3, стр.76.

7. В.Н. Елкина, Л.С. Юдина. Статистика открытых слогов русской речи.-Вычислительные системы.Новосибирск,I964, вып. I4, стр. 55.

8. В.Н. Елкина, Н.Г. Загоруйко. Алфавит объектов распознавания. В сб. Распознавание слуховых образов, Новосибирск, "Наука", I966 (в печати)

9. Bonner R.E. A "Logical Pattern" Recognition Program. IBM J.Res. and Dev., I962, vol.6, VII, N 3, p.353.

I0. Г.С. Себестиан. Процессы принятия решений при распознавании образов. Пер. с англ. Киев, "Техника", I965.

II. М.И. Шлезингер. О самопроизвольном различении образов.-Читающие автоматы, Киев, "Техника", I965.

I2. Е.А. Елкин, В.Н. Елкина, Н.Г. Загоруйко. О применении методики распознавания образов к решению задач палеонтологии. - Доклад на Всесоюзном совещании по применению математики в геологии. Новосибирск, декабрь, I965.

Поступила в редакцию
3.УI. I966.