Восстановление отсутствующих данных и принцип максимального подобия*

А. Г. Рубцов, М. Г. Садовский, М. Ю. Сенашова
Институт вычислительного моделирования СО РАН, Красноярск, Россия
e-mail: msen@icm.krasn.ru

Data loss recovery procedure is proposed based on the principle of a maximal similarity. Basic constraints of the recovery procedure are discussed. An optimization of computation procedure is presented.

Введение

Восстановление отсутствующих данных — важная прикладная задача в различных областях естествознания и техники [1]. Общие подходы к ней представляют собой одну из фундаментальных проблем вычислительной математики. Результаты восстановления отсутствующих данных зависят как от способа восстановления, так и от характера самих данных. Ранее были предложены подходы к решению этой проблемы, основывающиеся на идее моделирования работы высокопараллельных мелкозернистых вычислительных устройств [2–4]. Настоящая работа продолжает этот цикл исследований; мы изложим некоторые результаты, связанные с восстановлением отсутствующих данных на основе принципа максимального подобия. Данный принцип может быть представлен как экстремальный — принцип минимума условной энтропии (см. разд. 2).

Будем рассматривать проблему восстановления отсутствующих данных в строгой (но достаточно общей) постановке. Под данными будем понимать конечные символьные последовательности с известным алфавитом $\aleph$ (также конечным), в котором записаны изучаемые последовательности. Теоретически, любые данные можно представить в этой форме. Отсутствие части такой последовательности будет рассматриваться как лакуна в данных, при этом ее длина L будет считаться известной, а сама лакуна — связным диапазоном.

Восстановление последовательности символов в лакуне требует развития алгоритмов построения ее возможных заполнений. Каков бы ни был метод построения цепочки, замощающей лакуну, мы будем придерживаться общего подхода, состоящего в том, что замощения следует выполнять из “маленьких фрагментов” тех частей последовательности, которые доступны исследователю. Иными словами, мы будем копировать короткие фрагменты (слова) в имеющихся частях последовательности и комбинировать из них цепочки символов, замощающие лакуну.

В общем случае найдется не одно, а множество различных замощений лакуны; оценка сверху числа замощений дается выражением $|\aleph|^L$, что для алфавитов и характер-

*Работа выполнена при финансовой поддержке гранта Президента РФ для ведущих научных школ № НШ-3431.2008.9.

ных размеров лакун, встречающихся в различных приложениях, представляет собой астрономическую величину. Следует подчеркнуть, что приведенная здесь оценка не эффективна; подход к построению эффективной оценки числа возможных замощений рассмотрен ниже. Если процедура восстановления отсутствующих данных порождает не одно замощение, а несколько, возникает задача выбора наилучшего из них.

Интуитивно, принцип отбора наилучшего заполнения заключается в следующем: заполнять лакуну нужно таким образом, чтобы последовательность, возникающая в результате замощения, была максимально похожа на те части, которые имелись в распоряжении исследователя до восстановления. Такой принцип может быть сформулирован в экстремальной форме: условная энтропия частотного словаря, построенного по имеющимся в распоряжении исследователя фрагментам последовательности, относительно того словаря, который будет построен по всей восстановленной последовательности, должна быть минимальной [5].

Отметим, что сформулированный выше экстремальный принцип выбора наилучшего замощения не гарантирует совпадения отобранного замощения с утраченным фрагментом (если он был известен заранее). Построение оптимального в смысле информационных свойств замощения лакуны и замощения, максимально близкого к (известному) оригиналу, — это две разные задачи, которые следует решать отдельно, а связь между ними требует специальных исследований.

Как видно из верхней оценки числа заполнений, задача построения замощения — весьма ресурсоемка. Кроме того, построение каждого из вариантов замощений не зависит от построения других замощений, что делает возможным использование подходов и методов параллельных вычислений. Естественным представляется использование какого-либо варианта вычислительного устройства с максимально возможным уровнем распараллеливания вычислений. Одним из вариантов такого (идеального) вычислителя может быть устройство, основанное на идее молекулярных вычислений [6–15]. Такой подход имеет достаточно долгую и плодотворную историю, однако алгоритмическая полнота высокопараллельных мелкозернистых вычислительных устройств, основанных на идее квазихимической аналогии, была доказана лишь в работах [16].

Далее в рамках настоящей статьи будут всюду рассматриваться конечные последовательности из конечного алфавита $\aleph$. Пусть N — длина всей восстанавливаемой последовательности:

$$N = N_1 + L + N_2,$$

где N_1 и N_2 — длины известных частей последовательности, а L — длина лакуны, которую необходимо восстановить. *Словом* длиной q назовем любую связную подпоследовательность этой длины, составленную из символов алфавита $\aleph$. Будем называть список $\{w\}$ всех слов длиной q, встречающихся в частях последовательности, доступных исследователю (с указанием их частот f_w), *опорным* частотным словарем W толщиной q. Будем называть частотный словарь $\overline{W}$ (также толщиной q), составленный по той последовательности, которая возникает в результате замощения, *пополненным*. Кроме того, если какой-либо из словарей (W или $\overline{W}$) содержит все слова данной длины q, возможные в алфавите $\aleph$, то будем называть его *полным*.

Левой опорой длиной t, $0 \le t \le q-1$, будем называть слово такой длины, расположенное сразу слева от лакуны. Аналогично определяется правая опора. Тем самым, в зависимости от величины t, восстанавливаемая часть имеет длину $L + 2t$ при условии, что длины правой и левой опор совпадают; не всякое сочетание L, q и t оказывается

совместимым. Всюду далее мы будем предполагать, что длины правой и левой опор совпадают; данное предположение не уменьшает общности рассмотрения.

Заполнение лакуны означает построение цепочки

$$w_1, w_2, w_3, \ldots, w_{L+2t-q}, w_{L+2t-q+1} \tag{1}$$

длиной $L + 2t$ из слов длиной q, у которой первые и последние t символов заданы, а для каждой пары соседних слов выполняется условие

$$w_j = i_1\overline{w}, \quad \overline{w}i_q = w_{j+1},$$

т. е. два соседних слова пересекаются по общему подслову длиной $q-1$, а первое (последнее) слово в этой цепочке начинается (заканчивается) левой α_l (правой α_r) опорой.

Многообразие замощений, которые можно получить, комбинируя короткие слова, столь велико, что само по себе построение всех вариантов замощения (или их части, удовлетворяющей тем или иным условиям) — сложная вычислительная задача. Для решения ее представляется целесообразным использование высокопараллельных мелкозернистых вычислительных устройств. Аппаратной реализации таких устройств в настоящее время нет; соответственно, мы будем решать задачу восстановления отсутствующих данных с помощью имитатора такого вычислительного устройства, реализованного на обыкновенной последовательной машине фон-неймановского типа. Перейдем к описанию вычислителя, реализующего построение замощений лакуны.

1. Кинетическая машина Кирдина в задаче восстановления отсутствующих данных

Рассмотрим кратко понятие кинетической машины Кирдина (КМК) [17, 18]. КМК — идеальный мелкозернистый параллельный вычислительный формализм, по степени абстракции сопоставимый с машиной Тьюринга. КМК — алгоритмически полна [17, 19, 20], т. е. любой мыслимый алгоритм представим в ее терминах. Отличительной чертой КМК является мелкозернистый параллелизм. Описание идеального вычислителя начнем с указания объектов, над которыми производятся вычисления; затем опишем собственно сам способ вычисления. Обозначим через $\aleph^*$ множество всех конечных слов или цепочек в алфавите $\aleph$. Обрабатываемой единицей является ансамбль M слов, отождествляемый с функцией F_M, имеющей конечный носитель на $\aleph^*$, которая принимает неотрицательные целые значения $F_M\colon \aleph^* \mapsto N \cup \{0\}$. Значение $F_M(w)$ интерпретируется как число экземпляров слова w в ансамбле M.

КМК обрабатывает ансамбли элементарных событий, причем происходит это недетерминированно и параллельно. Элементарное событие $S\colon M \to M'$ состоит в том, что из ансамбля M изымается ансамбль K^- (это возможно, если для всех слов из ансамбля M выполняется условие $F_{K^-}(w) \le F_M(w)$) и добавляется ансамбль K^+, т. е. $F_{M'} = F_M - F_{K^-} + F_{K^+}$. Ансамбли K^- и K^+ однозначно задаются правилами или командами, которые объединяются в программу. Команды могут быть только трех видов.

Распад. $uvw \to uf + gw$, где u, w — произвольные слова, а слова v, f и g — фиксированные слова из $\aleph^*$.

Синтез. $uk + dw \to usw$, где u, w — произвольные слова, а слова k, d и s — фиксированные слова из $\aleph^*$.

Прямая замена. $uvw \to usw$, где u и w — произвольные слова, а v и s — фиксированные слова из $\aleph^*$.

Неформально КМК можно представить себе как аналог химического реактора, в котором происходят реакции [20]: имеется химический реактор идеального смешения, в котором распределены слова. В реактор добавляются правила-катализаторы; одни из них, взаимодействуя со словами, способствуют их распаду; другие, встречая пару подходящих слов, способствуют их синтезу; наконец, третьи заменяют в словах некоторые подцепочки.

Способ замощения лакуны в символьной последовательности с помощью КМК описывается следующей программой: пусть имеется текст T, по которому требуется составить частотный словарь W_q. Программа для КМК, реализующая этот процесс, состоит из одной команды и выглядит следующим образом:

$$uf^1v^{q-1}g^1w \to uf^1v^{q-1} + v^{q-1}g^1w,$$

где в качестве M нужно взять ансамбль, состоящий из одного слова T. После того как машина остановится, ансамбль M будет содержать все слова длиной q, встречающиеся в исходном тексте, с учетом их кратности.

Программа, реализующая процесс заполнения лакуны в терминах КМК, выглядит следующим образом:

$$\begin{array}{rcl} \alpha_l + \alpha_l v^{q-t} & \to & \alpha_l v^{q-t}*, \\ v^{q-t}\alpha_r + \alpha_r & \to & *v^{q-t}\alpha_r; \end{array} \tag{2}$$

$$\begin{array}{rcl} uv^{q-1}* + v^{q-1}v^1 & \to & uv^{q-1}v^1*, \\ v^1v^{q-1} + *v^{q-1}w & \to & *v^1v^{q-1}w; \end{array} \tag{2а}$$

$$u* + *v \to uv. \tag{2б}$$

Первые две строчки программы осуществляют инициализацию “затравок”, т. е. обеспечивают взаимодействие правой (или левой) опоры длиной t с подходящим словом длиной q. Третья и четвертая строчки осуществляют рост затравок, в ходе которого собственно и строится замощение лакуны. И, наконец, последняя строка склеивает левые и правые части; $\langle * \rangle \notin \aleph$ и используется в программе, чтобы помечать те слова, которые успешно прошли стадию инициации.

Исходным ансамблем для этой программы является некоторое количество копий “затравок” (левая (α_l) и правая (α_r) опоры) и копий словарей, полученных применением к исходному тексту T. В кинетической машине Кирдина элементарные события происходят недетерминированно и параллельно. Тем не менее программа построена так, что вначале в ансамбле просто нет таких слов, к которым могли бы применяться три последние команды. Таким образом, всю программу работы КМК по заполнению лакуны в последовательности можно представить состоящей из трех сменяющих друг друга этапов.

Первый этап (инициализации затравок). Формулы (2) описывают присоединение к левой (α_l) и правой (α_r) опорам слов из W_q.

Второй этап (рост замощений). Формулы (2а) описывают собственно рост замощения. Он начинается, как только в ансамбле будет достаточное количество слов, помеченных символом $\langle * \rangle$; это число зависит от структуры фрагментов, имеющихся в распоряжении исследователя; может случиться так, что опорный словарь W_q не будет

содержать ни одного слова, которое обеспечивало бы инициализацию. Последовательное и многократное применение команд этого этапа позволяет получить слова вида $\alpha_l u*$ и $*v\alpha_r$, длина которых составляет приблизительно $L/2$.

Третий этап. Формула (2б) описывает "склейку" слов вида $\alpha_l u*$ и $*v\alpha_r$. Слова, полученные применением этой команды, будут составлять финальный словарь для данной программы и вышеописанного исходного ансамбля, так как ни одна из команд программы уже не будет к ним применима. Поскольку КМК функционирует недетерминированно и параллельно, в этом финальном ансамбле будут слова разной длины. Самое короткое из них может иметь длину $q+2$.

Теперь нам нужно выбрать из ансамбля слова длиной $L+2t$ и исследовать полученные заполнения лакуны в соответствии с предложенными критериями.

2. Выбор наилучшего заполнения лакуны в последовательности

Если существует цепочка вида (1), составленная из слов опорного словаря, и она единственна, то задача построения заполнения считается решенной. Если существует несколько цепочек вида (1), составленных из слов опорного словаря, то среди всех возможных следует выбрать ту, которая обеспечивает минимум условной энтропии [5]

$$\overline{S} = \sum_w f_w \ln\left(\frac{f_w}{\tilde{f}_w}\right) \tag{3}$$

опорного частотного словаря $W(q)$ относительно пополненного $\overline{W}(q)$, где $\tilde{f}_w$ — частота слов, вычисленная по тексту, полученному в результате замощения лакуны, $\tilde{f}_w \in \overline{W}(q)$. Если замощения лакуны словами из опорного словаря не существует, тогда его следует строить всеми возможными в данном алфавите словами. Очевидно, что такое замощение существует всегда и оно не единственно. Смысл выражения (3) прозрачен — значение $\overline{S}$ равно нулю в том случае, когда частоты слов в опорном словаре $W(q)$ и в пополненном $\overline{W}(q)$ совпадают: $\forall\omega\ f_\omega = \widetilde{f}_\omega$.

Вопрос о существовании замощения по опорному словарю $W(q)$ может быть решен лишь перебором всех возможных символьных цепочек длиной L, составленных из слов длиной q при условии, что первое и последнее из слов имеют фиксированный набор символов в начале (и в конце соответственно), т. е. начинается и заканчивается заранее определенными опорами. Здесь эффективно использование высокопараллельных и кластерных вычислительных алгоритмов и устройств. Один из возможных подходов — применение кинетической машины Кирдина [17, 18].

3. Последовательный имитатор КМК в задаче заполнения лакун в последовательности

Кинетическая машина Кирдина — идеальное вычислительное устройство, обеспечивающее высокий уровень распараллеливания вычислений. Тем не менее, поскольку мы работаем на обычных последовательных машинах фон-неймановского типа (например, на персональных компьютерах — см. в [19] подробнее об этой стороне проблемы), будем строить имитатор КМК, необходимый для решения нашей конкретной задачи, а не всех

алгоритмов, которые могут быть представимы в КМК. Мы делаем это для исключения нерезультативных шагов в работе КМК и создания более оптимальной программы, решающей нашу задачу заполнения лакун.

Если заполнение по опорному словарю возможно[1], то работа имитатора КМК, реализованного на последовательной вычислительной машине, состоит в продолжении левой и правой опор "навстречу" друг другу с тем, чтобы склеить их, как только каждая достигнет длины $[L/2]+q-1+t$. Затем такие половинки следует склеить (с помощью последней команды) и среди всех цепочек, получившихся в результате склеивания, выбрать те, которые удовлетворяют условию (3).

Работа КМК во многом аналогична кинетике химической реакции, протекающей в реакторе: скорость построения замощений и время построения подходящего замощения определяются аналогично соответствующим кинетическим показателям для "химической реакции". Время вычисления цепочки, замощающей лакуну и удовлетворяющей критерию (3), существенно зависит от "концентрации" тех слов, которые могут породить подходящие продолжения опоры. Высокий параллелизм вычислений для заполнения лакуны означает, что мы можем параллельно вычислять любое наперед заданное число продолжений одной и той же опоры.

Работа с имитаторами КМК на последовательной машине фон-неймановского типа делает актуальной задачу повышения эффективности его работы; она очень сильно падает с уменьшением "концентрации" применимых слов. Для этого последовательный имитатор КМК был модифицирован. Поскольку имитатор КМК должен до определенной степени отражать работу параллельного вычислительного устройства, постольку число экземпляров затравок (левых опор, для определенности) бралось достаточно большим, и в наших вычислительных экспериментах это число составляло 10^3–10^5 копий. Вычисления со всеми этими затравками проводились последовательно.

Если цепочка вида (1) заканчивается словом ω_1 (длины $q-1$), то ее продолжение определяется тем, какие именно слова из частотного словаря (опорного или пополненного — не важно) могут прореагировать с этим словом. Если в носителе частотного словаря $W(q)$ есть несколько слов, которые могут провзаимодействовать в силу (2а), то выбор конкретного слова, вступающего в реакцию (2а) со словом ω_1, определяется случайно, пропорционально относительной частоте каждого из возможных реагентоспособных слов-продолжений. Последовательный имитатор КМК лишь моделирует параллельную работу кинетической машины Кирдина; для повышения эффективности построения заполнений лакун в символьной последовательности последовательный имитатор КМК был модифицирован; всего было внесено три модификации.

Первая модификация. Рост затравок происходил только в одном направлении — слева направо (для определенности). Как только цепочка достигала длины $L+2t$, осуществлялась проверка того, оканчивается ли она правой опорой. Если правая опора в нее входила, то эта цепочка считалась одним из возможных заполнений, в противном случае она исключалась из рассмотрения.

Вторая модификация. Модифицировался словарь, по которому строилось заполнение (1). Для реализации этапов, соответствовавших работе команд (2)–(2б), исходный словарь заменялся модифицированным, который содержал только те слова, которые имели начала, соответствовавшие затравке. Модификация частотного словаря означает построение на ансамбле M новой функции $\overline{F}_M\colon \Omega^* \to N \cup \{0\}$, такой, что

[1] Оно возможно всегда, когда опорный словарь совпадает с полным.

$\overline{F}_M = F_M(uv^{q-1}*) + F_M(*v^{q-1}v^1)$ для всех v^1, v^{q-1}, для которых выполняются команды подпрограмм (2) и (2а), и $\overline{F}_M = 0$ для всех остальных слов. Здесь верхний индекс указывает длину слов. Поскольку в общем случае у одной опоры существует несколько продолжений, постольку из всех имеющихся продолжений случайным образом выбиралось одно (для данной затравки) с вероятностью, пропорциональной доле этого продолжения. Для реализации этапов, соответствовавших работе команд (2а) КМК, исходный словарь заменялся на словарь, содержавший только те слова, которые имели начала, соответствовавшие слову, прошедшему инициализацию.

Третья модификация. Периодически проводилась селекция всех слов, являвшихся продолжениями опор, построенных в силу команд (2а) КМК. Среди продолжений слова $uv^{q-1}*$ могут быть такие, которые сами уже не имеют никаких продолжений среди слов из используемого в текущем вычислительном эксперименте частотного словаря. Поэтому возможна ситуация, в которой для некоторых слов команда (2а) не выполняется никогда. С точки зрения повышения эффективности работы последовательного имитатора КМК, такие "тупиковые" слова следует удалить. С другой стороны, удаление "тупиковых" слов на каждом шаге времени существенно понижает эффективность работы имитатора: приходится сравнивать большое количество слов. Соответственно, селекция (удаление "тупиковых" слов) проводилась не постоянно, а дважды за время роста продолжений. Для этого вся лакуна, требующая заполнения, разбивалась на три интервала равной длины. Понятно, что некоторые затравки давали такие продолжения, которые обрывались на длине, меньшей длины ее первого фрагмента (см. описание работы КМК выше). По достижении остальными словами этой пороговой длины (равной трети длины лакуны) "тупиковые" слова из всего множества слов, с которыми работал имитатор КМК, удалялись. Затем те слова, которые достигли этой критической длины, удваивались (либо, в общем случае, их число увеличивалось в k раз) и процедура построения заполнения в силу команд (2а) продолжалась до тех пор, пока эти слова не достигали следующей длины, на которой проводилась селекция. По достижении этой длины (составляющей две трети от длины лакуны) оставшиеся слова опять "размножались".

4. Матричное представление частотных словарей

Задачу поиска замощений, эффективно восстанавливающих отсутствующие данные, помогает решить специальное представление частотного словаря. Всякий частотный словарь представляет собой (упорядоченный) список слов длиной q. Такой список можно однозначно преобразовать в матрицу $\mathfrak{A}$ порядка $|\aleph|^{q-1} \times |\aleph|^{q-1}$, в которой строки и столбцы помечены словами ω' и ω'' длиной q каждое. Тогда любое слово ω из словаря $W(q)$, начинающееся последними $q-1$ символами слова ω' и заканчивающееся подсловом ω'', соответствует элементу матрицы, находящемуся на пересечении соответствующих строки и столбца; а сам этот элемент является частотой слова ω.

Очевидно, что число ненулевых элементов в каждой строке в $\mathfrak{A}$ не превышает $|\aleph|$. Соответственно, вся матрица $\mathfrak{A}$ является весьма разреженной: число ненулевых элементов в ней растет с толщиной q словаря лишь линейно, в то время как общее число элементов в матрице $\mathfrak{A}$ растет пропорционально квадрату толщины словаря. Заменой в матрице $\mathfrak{A}$ элементов в каждой строке таким образом, чтобы их сумма по строке стала равной единице, а соотношение элементов сохранилось, получаем матричное представление $\mathcal{A}$ модифицированного частотного словаря. Наконец введем еще одно представление ча-

стотного словаря. Заменив в матрице $\mathfrak{A}$ все ненулевые элементы на единицу, а нулевые оставив неизменными, получаем матрицу $\mathbb{A}$, которую будем называть индикаторной матрицей.

Матричное представление частотного словаря позволяет пролить свет на решение вопроса о существовании (и построении) замощения из слов заданного частотного словаря. Действительно, построение замощения (1) в силу процедуры (2)–(2б) эквивалентно возведению матрицы $\mathfrak{A}$ в степень $L+t$, где L и t — длина лакуны и длина опоры соответственно. Пусть $t = q-1$; тогда вопрос о существовании замощения из словаря $W(q)$ сводится к вопросу о существовании ненулевого элемента в матрице $\mathfrak{A}^{L+t}$, стоящего на пересечении строки и столбца, соответствующих опорам α_l и α_r. Если $t < q-1$, то вопрос о существовании замощения из словаря $W(q)$ сводится к вопросу о существовании хотя бы одного ненулевого элемента в пересечении полос строк и столбцов шириной $z = \aleph^{q-t}$, где q и t соответственно толщина словаря $W(q)$ и длина опоры; напомним, $t < q-1$. При этом элементы матрицы $\mathfrak{A}^{L+t}$, попавшие в нужную полосу (быть может, единичной ширины), являются вероятностями формирования требуемого замощения лакуны словами из словаря, соответствующего матрице $\mathfrak{A}$.

Кроме вопроса о существовании замощения из словаря $W(q)$, соответствующего матрице $\mathfrak{A}$, важным является вопрос о числе таких замощений, удовлетворяющих граничным условиям (т. е. опирающимся на заданные опоры). Ответ на этот вопрос может быть получен с помощью индикаторной матрицы $\mathbb{A}$. Действительно, в силу построения самой этой матрицы, ее произведение с самой собой даст возможное число продолжений исходной затравки на два символа “вперед”. Вообще, элементами матрицы $\mathbb{A}^P$ при некотором натуральном P будут только натуральные числа; соответственно, элементы матрицы $\mathbb{A}^{L+t}$ будут соответствовать числу замощений, которые могут начаться с заданной опоры (затравки) и заканчиваться также заданной опорой.

Итак, подведем краткий итог этого раздела. Для всякого частотного словаря возможны три (различных) матричных представления. Первое полностью эквивалентно самому частотному словарю, второе соответствует марковскому процессу порядка $q-1$, который реализует гипотезу о наиболее вероятном продолжении слов этой длины, третье представление эквивалентно задаче определения числа маршрутов на графе, соответствующем матрице. Это третье представление позволяет вычислить число замощений, возможных для заданного частотного словаря $W(q)$.

5. Результаты

В работе представлены предварительные результаты восстановления отсутствующих данных, проведенного в силу принципа максимального подобия. Данный принцип имеет форму экстремального — минимума условной энтропии (3). Изложение результатов начнем с описания тест-объектов, которые были использованы в вычислительных экспериментах. Выбор тест-объекта не является простым и очевидным. В вычислительных экспериментах, результаты которых изложены в настоящей статье, использовались символьные последовательности различной природы: искусственно сгенерированные последовательности из двухбуквенного алфавита $\aleph = \{0, 1\}$, генетические тексты (использовались последовательности полностью расшифрованных геномов различных организмов [21]) и последовательности из естественных языков.

Изучение символьных последовательностей из естественных языков наталкивается на ряд трудностей. Во-первых, тексты для исследования должны быть нормативными —

записанными в полном соответствии с (академической) грамматикой соответствующего языка. Во-вторых, сравниваемые тексты должны описывать один и тот же сюжет. Наконец, есть естественное третье ограничение: сравниваться должны тексты, записанные в языках с алфавитной системой письма. Указанным ограничениям удовлетворяют тексты различных международных документов, которые переводятся на различные языки мира. Мы использовали текст Всеобщей декларации прав человека [22].

Двоичные последовательности, использовавшиеся как тест-объекты, были получены с помощью простых и прозрачных правил. Первая последовательность была составлена из записанных подряд, без пробелов, натуральных чисел в двоичной системе счисления. Еще один вариант двоичной последовательности был получен путем записи подряд, без пробелов, натуральных чисел в троичной системе счисления, после чего из записи вычеркивались все цифры 2. Длина последовательностей составляла не менее 6000 символов.

Во всех последовательностях искусственно создавались лакуны длиной от 50 до 200 символов. Затем с помощью имитатора КМК строился опорный частотный словарь $W(q)$ для различных значений q и по нему строились замощения. Результаты построения замощений для двоичных последовательностей представлены в таблице.

Непосредственная интерпретация результатов, полученных на двоичных последовательностях, затруднена. Для того чтобы облегчить восприятие работы предложенного способа восстановления утерянных данных, мы провели серию вычислительных экспериментов на текстах из естественных языков. Из оригинального текста Декларации были удалены пробелы и знаки препинания. Длина получившейся символьной последовательности составила около 7500 символов, длина лакуны — 100 символов. Использовались словари толщиной $3 \le q \le 8$ символов. Для словарей толщиной $q > 8$ не было получено заполнений, совпавших с правой опорой. Число затравок в каждом вычислительном эксперименте для любой толщины словаря бралось равным 100000.

Ниже приведены варианты замощений лакуны для словарей толщиной $q = 4$, $q = 6$, $q = 8$ и собственно текст, который был удален из Декларации для получения лаку-

Значения условной энтропии для наилучшего заполнения, вычисленной по пополненному словарю

	Виды тестовых последовательностей			
q	I	II	III	IV
3	$2.11001 \cdot 10^{-7}$	$2.11731 \cdot 10^{-7}$	$1.08111 \cdot 10^{-6}$	0.002501334
4	$8.63964 \cdot 10^{-7}$	$1.15992 \cdot 10^{-7}$	$7.28976 \cdot 10^{-6}$	0.004114464
5	$8.80001 \cdot 10^{-6}$	$9.15474 \cdot 10^{-6}$	$3.55110 \cdot 10^{-5}$	0.005274570
6	$3.25038 \cdot 10^{-5}$	$2.83748 \cdot 10^{-5}$	0.000127584	0.006469274
7	$8.54041 \cdot 10^{-5}$	$9.11896 \cdot 10^{-5}$	0.000383415	0.007102911
8	0.000250881	0.000225592	0.000727322	0.006994414
9	0.000533898	0.000531811		
10	0.00102765	0.00104568		
11	0.0018486	0.00176425		
12	0.0025176	0.00244363		
13	0.0034176	0.00342195		

Примечание. q — толщина словаря; I — двоичные последовательности; II — двоичные последовательности, построенные на основе троичных; III — генетические тексты; IV — Всеобщая декларация прав человека (на русском языке).

ны; правая, а также левая опоры не показаны. После удаления знаков препинания и пробелов исходный (удаленный) текст выглядел следующим образом:

егоимуществакаждыйчеловекимеетправонасвободумыслисовестииирелигии этоправовключаетсвободуменятьсвоюрел

При восстановлении удаленного текста по словарям $W(4)$, $W(6)$ и $W(8)$, соответственно, были получены следующие цепочки:

— *имладатьсяниисовявляющеедолжностиправонасоциальности идругиминациюкаждыйчеловекимеетправонасторойирел*

— *егогражданствоиработающийимеетправонаравнуюзащитой каждыйчеловекимеетправоприниматьучаствоватьсвоюрел*

— *егогражданстваилитакихпосягательствкаждыйчеловекимеетправо наобеспечивающихсвободуисповедоватьсвоюрел*

Хорошо видно, что длина слов (цепочек) в шесть символов (и более) обеспечивает восстановление утерянного фрагмента, который поддается интерпретации.

О числе замощений

Представление частотного словаря $W(q)$ в матричном виде позволяет эффективно определить число замощений, которые могут быть получены из слов такого словаря, а также вероятность построения требуемого замощения. Эти данные позволяют оценить необходимое число затравок для получения замощений и требуемые для вычислений ресурсы. Напомним, что матричное представление словаря представляет собой матрицу $\mathfrak{A}$ порядка $|\aleph|^{q-1} \times |\aleph|^{q-1}$, — для сколько-нибудь значительной толщины словаря и, тем более, реальных алфавитов это представляет собой необозримый объект.

	aaaa	*aaac*	*aaag*	...	caaa	...	gaaa	...	taaa	...
aaaa	15089	15089	15089	...	15206	...	13913	...	15206	...
aaac	15089	15089	15089	...	15206	...	13913	...	15206	...
aaag	14278	14278	14278	...	14389	...	13165	...	14389	...
aaat	14958	14958	14958	...	15074	...	13792	...	15074	...
aaca	15089	15089	15089	...	15206	...	13913	...	15206	...
aacc	15089	15089	15089	...	15206	...	13913	...	15206	...
aacg	14278	14278	14278	...	14389	...	13165	...	14389	...
aact	14958	14958	14958	...	15074	...	13792	...	15074	...
aaga	15089	15089	15089	...	15206	...	13913	...	15206	...
aagc	11463	11463	11463	...	11552	...	10569	...	11552	...
aagg	14711	14711	14711	...	14825	...	13564	...	14825	...
aagt	14958	14958	14958	...	15074	...	13792	...	15074	...
aata	15089	15089	15089	...	15206	...	13913	...	15206	...
aatc	14140	14140	14140	...	14250	...	13038	...	14250	...
aatg	14711	14711	14711	...	14825	...	13564	...	14825	...
aatt	14958	14958	14958	...	15074	...	13792	...	15074	...
acaa	15089	15089	15089	...	15206	...	13913	...	15206	...
acac	15089	15089	15089	...	15206	...	13913	...	15206	...
acag	14278	14278	14278	...	14389	...	13165	...	14389	...
acat	14958	14958	14958	...	15074	...	13792	...	15074	...
...	...	...	...	...	...	...	...	...	...	...

В наших вычислительных экспериментах мы использовали матричное представление для оценки числа (и вероятности) построения замощений, удовлетворяющих граничным условиям, для символьных последовательностей различной природы. Для всех изученных последовательностей (двоичных, четырехбуквенных и последовательностей из естественных языков) такое представление оказалось весьма эффективным способом получения указанных оценок. Из-за большого объема матричного представления словаря мы привели пример лишь одного из них, полученного для генетического текста. Напомним, что опорами в данном случае являлись слова длиной в три: *aac* — левая опора и *aaa* — правая.

В примере показан лишь небольшой фрагмент матрицы, полученной возведением в степень (равную длине лакуны) индикаторной матрицы $\mathbb{A}$. В строках приведены те слова, которыми начинаются замощения, в столбцах — те, которыми они заканчиваются. В боксы взяты те слова, которые соответствуют левым (строки) и правым (столбцы) опорам; длина опоры составляла три символа. Несмотря на то, что для изученной последовательности опорный словарь не был полон, число замощений, удовлетворяющих таким граничным условиям, весьма велико.

Следует подчеркнуть, что число возможных замощений, удовлетворяющих граничным условиям, еще не говорит о том, что последовательный имитатор КМК сможет породить такое (и тем более — наилучшее) замощение. Ниже приведен фрагмент матрицы частот, показывающий те частоты, с которыми будут возникать замощения, отличающиеся только первым символом справа (соответственно, слева) от левой (соответственно, правой) опоры. Хорошо видно, что для данной последовательности существуют замощения, которые продолжают левую опору (либо стыкуются с правой опорой) любой буквой. Очевидно, что такое богатство замощений не является общим случаем и определяется структурой частотного словаря (которая, в свою очередь, определяется структурой изучаемой последовательности).

	aaaa	*caaa*	*gaaa*	*taaa*
aaca	0.0328002902542269	0.0137625953859325	0.00684852939361589	0.0180010041574707
aacc	0.0327980643940794	0.0137697935627674	0.00684637367694756	0.0180035516235888
aacg	0.0327889705125126	0.0137562379298597	0.00683589956965863	0.0179913272051014
aact	0.0327767710829256	0.0137629890086235	0.00683752554494568	0.0179945072580299

Подчеркнем, что обе матрицы, фрагменты которых представлены выше, имеют порядок 251×251; иными словами, опорный частотный словарь $W(5)$ лишь немного — на пять слов — отличается от полного словаря $W^*(5)$. Приведенные во втором фрагменте (во фрагменте матрицы $\mathfrak{A}$) частоты являются суммой частот **всех** замощений, которые можно построить из словаря $W(5)$, а не какого-то отдельного замощения.

6. Обсуждение

В работе рассмотрены некоторые предварительные результаты в проблеме восстановления утерянных данных. Данные были представлены в форме символьной последовательности; такое представление не ограничивает общности рассмотрения, хотя и не всегда позволяет перейти к решению непосредственных прикладных задач. Восстановление понимается как построение цепочки символов, которая была бы в наибольшей степени похожа на фрагменты последовательности, имеющиеся в распоряжении исследователя. Такой подход означает, что построение замощения должно делаться из ко-

пий подпоследовательностей малой длины, комбинация которых позволяет "накрыть" лакуну. При этом такое замощение должно удовлетворять тем или иным "граничным" условиям: например, комбинации слов (сравнительно коротких подпоследовательностей), покрывающие лакуну, должны начинаться и заканчиваться вполне определенными комбинациями символов.

В целом такое замощение не единственно. В этом случае проблема выбора наилучшего решается с помощью экстремального принципа — принципа максимального подобия: наилучшим замощением считается такое, которое обеспечивает минимум условной энтропии опорного частотного словаря относительно пополненного. Первый словарь строится по имеющимся в распоряжении исследователя фрагментам последовательности, а второй — по той последовательности, которая получается в результате построения замощения.

Построение таких замощений представляет собой достаточно сложную и ресурсоемкую вычислительную задачу: заранее не очевиден алгоритм построения наилучшего замощения, позволяющий отказаться от перебора всех комбинаций слов. Приведенные результаты показывают, что применение высокопараллельных мелкозернистых вычислительных устройств (и даже их имитаторов) дает хорошие результаты в поиске соответствующих замощений. Выбор наилучшего из них зависит от критерия и может быть осуществлен многими способами. Возможно, наиболее универсален подход, опирающийся на идею сравнения распределений. Такое сравнение в свою очередь может быть произведено многими способами; принцип минимума условной энтропии (3) представляется наиболее универсальным и обладающим наибольшей областью применимости.

Матричное представление частотного словаря — весьма эффективный прием для разрешения вопроса о существовании замощения из слов заданного частотного словаря и о числе таких замощений, удовлетворяющих граничным условиям. Ответ на этот вопрос сводится к проверке того, что элементы матриц $\mathfrak{A}^L$ и $\mathbb{A}^L$ не равны нулю. Заметим, что обе матрицы $\mathfrak{A}$ и $\mathbb{A}$ являются неотрицательными [23,24]; для таких матриц хорошо известна теорема Перрона, гарантирующая положительность матрицы $\mathfrak{A}^L$ (либо $\mathbb{A}^L$), при определенных условиях накладываемых на матрицы $\mathfrak{A}$ и $\mathbb{A}$.

Для решения задачи о существовании заполнения лакуны словами из (опорного) словаря следует исследовать матрицу $\mathfrak{A}^L$ на положительность; точнее, условие превращения матрицы $\mathbb{A}$ в строго положительную нас не интересует. Нас интересуют условия (выраженные в терминах каких-либо свойств матрицы $\mathbb{A}$, например, в терминах ее собственных значений), при которых вполне определенные элементы матрицы $\mathfrak{A}^L$ становятся положительными. Однако детальное исследование этого вопроса выходит за рамки настоящей статьи.

Сформулируем также еще одну интересную проблему, возникающую в связи с задачей восстановления утерянных данных — это проблема устойчивости восстановления при малом шевелении длины лакуны. Действительно, построение замощения (1) из заданного частотного словаря $W(q)$, удовлетворяющего граничным условиям, возможно далеко не всегда. Тем не менее разрешим при построении замощения (1) малые изменения длины исходной лакуны. Если в результате таких малых изменений длины (в сторону уменьшения или увеличения — для нас неважно) замощение может быть построено наверняка, будем называть такую ситуацию *ситуацией устойчивой восстанавливаемости*. Более детальные обсуждения этой проблемы выходят за рамки данной работы.

Список литературы

[1] Gorban A.N., Rossiev D.A., Wunsch II D.C. Neural Network Modelling of Data with Gaps // Радиоэлектроника. Информатика. Управление. 2000. № 1. С. 47–55.

[2] Неменчинская Е.О., Кондратенко Ю.В., Садовский М.Г. Предварительные результаты в проблеме восстановления отсутствующихных данных с помощью кинетической машины Кирдина // Вычисл. технологии. 2004. Т. 9, № 1. С. 42–57.

[3] Nemenchinskaya E.O., Kondratenko Yu.V., Sadovsky M.G. Entropy based approach to data loss reparation through the indeterminate fine-grained parallel computation // Open Systems & Information Dynamics. 2004. Vol. 11, N 2. P. 161–175.

[4] Gorbunova E.O., Kondratenko Yu.V., Sadovsky M.G. Data loss reparation due to indeterminate fine-grained parallel computation. ICCS 2003, LNCS 2658. Berlin; Heidelberg: Springer-Verlag: 2003. P. 794–801.

[5] Горбань А.Н. Обход равновесия. Новосибирск: Наука, 1984.

[6] Alba E., Tomassini M. Parallelism and Evolutionary Algorithms // IEEE Trans. on Evol. Computation. 2002. Vol. 6, N 5. P. 434–462.

[7] Abelson H., Allen D., Coore D. et al. Amorphous computing // Communications of the ACM. 2000. Vol. 43, N 5. P. 74–82.

[8] Mitchell M., Crutchfield J., Hraber P. Evolving cellular automata to perform computations: Mechanisms and impediments // Physica D. 1994. Vol. 75. P. 361–391.

[9] Adleman L.M. Molecular computation of solutions to combinatorial problems // Science. 1994. Vol. 266. P. 1021–1024.

[10] Guarnieri F., Fliss M., Bancroft C. Making DNA Add // Science. 1996. Vol. 273. P. 220–223.

[11] Benenson Ya., Adar R., Paz-Elizur T. et al. DNA molecule provides a computing machine with both data and fuel // PNAS. 2003. Vol. 100, N 5. P. 2191–2196.

[12] Aoki T., Kameyama M., Higuchi T. Interconnection-free biomolecular computing // Computer. 1992. Vol. 25. P. 41–50.

[13] Banzhaf W., Dittrich P., Eller B. Selforganization in a system of binary strings with topological interactions // Physica D. 1999. Vol. 125. P. 85–104.

[14] Banzhaf W. The "molecular" traveling salesman // Biol. Cybern. 1990. Vol. 64. P. 7–14.

[15] Calude C.S., Paun G. Computing With Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing. Taylor & Francis, 2001. 309 p.

[16] Gorban A.N., Gorbunova E.O., Wunsch D.C. Liquid brain: kinetic model of structureless parallelism // Advances in Modelling & Analysis, AMSE. 2000. Vol. 5. P. 37–45.

[17] Горбунова Е.О. Формально-кинетическая модель бесструктурного мелкозернистого параллелизма // Сиб. журн. вычисл. математики. 1999. Т. 2, № 3. С. 239–256.

[18] Кирдин А.Н. Модель идеального ансамбля для параллельных вычислений // Нейроинформатика и ее приложения. Красноярск: Изд-во КГТУ, 1997.

[19] Gorban A.N., Gorbunova E.O., Wunsch D.C. Liquid Brain: The proof of algorithmic universality of quasichemical model of fine-grained parallelism // Neural Network World. 2001. Vol. 4. P. 391–412.

[20] Яблонский Г.С., Быков В.И., Горбань А.Н. Кинетические модели каталитических реакций. Новосибирск: Наука, 1983.

[21] http://www.ebi.ac.uk/genomes

[22] http://www.un.org

[23] Курош А.Г. Курс высшей алгебры. М.: Наука, 1971.

[24] Кострикин А.И. Введение в алгебру. М.: Наука, 1977.

Поступила в редакцию 8 декабря 2006 г.