Data Loss Reparation Due to Indeterminate Fine-Grained Parallel Computation

Ekaterina O. Gorbunova[1], Yuri V. Kondratenko[2], and Michael G. Sadovsky[3]

[1] Institute of Computational Modelling of SB RAS,
Krasnoyarsk, Russia, 660036, gkat@icm.krasn.ru
[2] Krasnoyarsk State University,
Krasnoyarsk, Russia, 660042
[3] Institute of Biophysics of SB RAS,
Krasnoyarsk, Russia, 660036, uvenal@ktk.ru

Abstract. The new method of a gap recovery in symbol sequences is presented. A covering is combined from the suitable reasonably short strings of the parts of a sequence available for an observation. Two criteria are introduced to choose the best covering. It must yield the maximum of entropy of a frequency dictionary developed over the sequence obtained due to the recovery, if an overlapping combined from the copies of strings from the available parts of the sequence exists. The second criterion identifies the best covering in case when one has to use any string to cover the gap; here the best covering must yield the minimum of specific entropy of the frequency dictionary developed over the available parts of the sequence against that one developed over the entire sequence obtained due to the recovery. Kirdin kinetic machine that is the ideal fine-grained structureless computer has been used to resolve the problem of the reconstruction of a gap in symbol sequence.

1 Introduction

A reconstruction of the lost data seems to be an acute problem both for fundamental and applied sciences. The results of the reconstruction depend significantly on the method to do it and on the information providing the basis for that latter. The up-to-date methods of the reconstruction of lost data implement various additional information concerning the data, or the knowledge concerning the properties of these data. We shall consider the problem of data loss reconstruction using the most strict definitions allowing to avoid such discrepancies.

Finite symbol sequences will be considered as a data source. The absence of a part of sequence would be considered as a data loss. One should distinguish the situation when the length of a gap is known from the situation where this length is unknown. Similarly, the alphabet could be either known, or unknown to a researcher; everywhere below we assume that both the alphabet and the length of a gap are known. Moreover, the finiteness of the alphabet will be assumed.

Simply speaking, the basic principle of the gap fulfillment force to do that in a manner that the sequence resulted from the fulfillment looks mostly similar to

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2658, pp. 794–801, 2003.

the originally available parts of that former. Additionally, the fulfillment must bring the minimum external, additional information concerning the sequence. The principle has two forms:

- a maximum of entropy of the augmented dictionary resulted due to the reconstruction of a sequence, and
- a minimum of specific entropy of the reference frequency dictionary against the augmented one. Let now turn onto the strict formulations and exact statements.

This paper is devoted to the consideration of some preliminary results in data loss reparation obtained due to the first approach (i.e. maximization of the augmented frequency dictioanry).

2 Criteria for a Gap Fulfillment

Consider a sequence of the length N:

$$N = N_1 + N_2 + L.$$

Here N_1 and N_2 are the lengths of available parts of the sequence, and L is the length of the fragment that must be reconstructed. The parts of the sequence known to a researcher are considered to be continuous. A word of the length q is a continuous subsequence (string) of that length. A list of all the words occurred at the available parts of a sequence accompanied with their frequency is called the reference frequency dictionary W_q (of the thickness q); f_ω is the frequency of word ω. The frequency dictionary developed over the entire sequence obtained due to the fulfillment of a gap is called augmented frequency dictionary $\overline{W}_q$. Obviously, the constraint

$$1 \leq q \leq min\{N_1, N_2\}$$

holds true. A word of the length t, $0 \leq t \leq q - 1$ located at the right border (at the left border, respectively) of fragment available to a researcher is called the left basement (the right basement, respectively). To develop a gap recovery means to figure out a string

$$\omega_1,\ \omega_2,\ \omega_3,\ \ldots,\ \omega_{L+2t-q},\ \omega_{L+2t-q+1} \tag{1}$$

of the length $L + 2t$, where ω_i is a word, and

$$\omega_j = i_1\overline{\omega}, \quad \omega_{j+1} = \overline{\omega}i_q.$$

Here $\omega_1 = l_t\alpha$ and $\omega_{L+2t-q+1} = \alpha r_t$ are the first and the last words bearing the basement (the left one, and the right one, respectively), and $\overline{\omega}$ is a word of the length $q - 1$. To fulfill a gap, one shall use the words available at the frequency dictionary of thickness q. There are two options here. The first one is to use the words from the reference dictionary only, and the second one implies a fulfillment with any possible words of the given length, since no fulfillment exists combined from the words from the reference dictionary. To choose the best fulfillment, we propose the extremal principle, for each case.

Consider the first case. If the unique fulfillment with the words from the reference dictionary exists, then the problem is resolved. If a fulfillment is ambiguous, then one must choose the string among all possible entities that yields the maximum of entropy of the augmented frequency dictionary:

$$S = -\sum_{\omega} \tilde{f}_{\omega} \ln \tilde{f}_{\omega}, \tag{2}$$

here $\tilde{f}_{\omega}$ is the frequency of word ω observed at the text obtained due to the gap recovery, i.e. the frequency of word ω at the augmented dictionary $\overline{W}_q$.

Consider now the case, when the fulfillment with the words from the reference dictionary does not exist. It must be done then with any possible words. Here the fulfillment must be chosen yielding the minimum of specific entropy of the reference frequency dictionary W_q against the augmented one $\overline{W}_q$:

$$\overline{S} = \sum_{\omega} f_{\omega} \ln \frac{f_{\omega}}{\tilde{f}_{\omega}}. \tag{3}$$

f_{ω} is the frequency of word ω observed at the reference dictionary, and $\tilde{f}_{\omega}$ is the frequency obtained over the recovered sequence; obviously, $f_{\omega'} = 0$ for some ω', while $\tilde{f}_{\omega'} > 0$ for them. This second criterion is universal, since it is applicable always. The applicability of the first criterion is not guaranteed *á priori*.

The existence of the recovery with the words from the reference dictionary is not guaranteed; moreover, a complete searching for all strings of the length $L + 2t$ composed from the words from the reference dictionary of the thickness q seems to be the only way to do that. An implementation of highly paralleled structureless computation devices is the most efficient way to make up such fulfillment. Kirdin kinetic machine that is the ideal highly paralleled grain computation device, pretends to be the most efficient entity to resolve the problem of data loss reparation [1,2,3].

3 Kirdin Kinetic Machine in the Problem of Gap Reparation

To begin with, let's describe the principle of execution of Kirdin kinetic machine (KKM) in detail. KKM is an ideal fine-grained parallel computer, similar to Turing machine from the point of view of the abstraction [1,2,3]. KKM is algorithmically universal [3,4], i.e. any algorithm could be implemented in the terms of KKM. Besides, it realizes the fine-grained parallelism. Let Ω be an alphabet of symbols, and Ω^* is the set of all finite strings (or words) in that alphabet. An ensemble M of the words from that alphabet is a processed entity; that latter is identified with the function F_M on a finite support from Ω^*. The function takes value in positive integers: $F_M\colon \Omega^* \mapsto N \cup \{0\}$. The value of $F_M(\omega)$ is interpreted as the number of copies of a word ω at the ensemble M.

Properly, KKM operation consists of an assembly of elementary events that take place in parallel and non-deterministic way. An elementary event $S\colon M \mapsto M'$ consists in a removal of the ensemble K^- from the ensemble M, and the addition of the ensemble K^+ to the ensemble M, so that $F_{M'} = F_M - F_{K^-} + F_{K^+}$. The removal is possible, if $F_{K^-}(\omega) \leq F_M(\omega)$ for all the words from the ensemble M. The ensembles K^+ and K^- are defined unambiguously by the commands making a programme. Three types of commands are possible:

1. ***Disintegration***. $uvw \longrightarrow uf + gw$, where u, w are the arbitrary words from Ω^*, while v, f, g are fixed words from Ω^*.
2. ***Synthesis***. $uk + qw \longrightarrow usw$, where u, w are the arbitrary words from Ω^*, while k, q, s are fixed words from Ω^*.
3. ***Mutation***. $uvw \longrightarrow usv$, where u, w are the arbitrary words from Ω^*, while v, s are the fixed ones from Ω^*.

Informally, KKM looks like a chemical reactor. One has a tank where the words are suspended. Then one provides the suspension with catalysts (that are the commands), and the words interact. The interaction of a word with catalyst may result in a disintegration. A couple of words interacting due to the proper command may yield a new word. Finally, a new word may appear due to the interaction of some word with the command resulting in a substitution of a substring inside a word.

Consider now the method to fulfill a gap in a sequence with a help of KKM in more detail. To do that, we shall provide the programme of a dictionary development, and the programme of the gap reparation, itself. Let a frequency dictionary W_q be developed for the text T. The KKM programme yielding the dictionary consists of a single command:

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

where the ensemble M contains a single word T. Upon a completion of KKM processing, the ensemble M contains all the words of the length q occurred at the original text, with respect to the number of their copies.

The KKM programme covering a gap in a sequence looks as following:

$$\begin{array}{rcl} \alpha_l + \alpha_l v^{q-t} & \longrightarrow & \alpha_l v^{q-t}\star, \\ v^{q-t}\alpha_r + \alpha_r & \longrightarrow & \star v^{q-t}\alpha_r. \end{array} \tag{4}$$

The symbol $\langle\star\rangle$ falls beyond the alphabet Ω and marks the word that has passed the initialization successfully. Metasymbol $\langle\star\rangle$ identifies the words of the length q which start from the left basement (and end with the right basement, respectively). The commands producing a growth of infill itself are the following:

$$\begin{array}{l} uv^{q-1}\star + v^{q-1}v^1 \longrightarrow uv^{q-1}v^1\star, \\ v^1v^{q-1} + \star v^{q-1}w \longrightarrow \star v^1v^{q-1}w, \end{array} \tag{5}$$

and the following is the command that glues two strings into a covering:

$$u\star + \star v \longrightarrow uv. \qquad (6)$$

An original ensemble of this programme consists of several copies of the basements (these are the left (α_l) and the right (α_r) ones), and several copies of the dictionaries obtained due to the execution of the previous programme. KKM runs non-determinally and in parallel way. Nevertheless, the programme is implemented so that initially the ensemble bears no words which could be processed by the commands (5, 6). Thus, a programme recovering a gap in the sequence could be considered to consist of three stages.

The first stage — "Basement initialization" (see Eq.(4)) consists in the connection of words from the dictionary W_q to the left (α_l) and the right (α_r) basement.

The second stage — "Growth" (see Eq.(5)) consists in the connection of words from the dictionary W_q to the primers obtained due to the initialization. It should be said that the number of words marked with $\langle\star\rangle$ depends on the structure of the fragments available for an observation. It might be that the reference dictionary W_q would bear no word which can provide the initialization. Sufficiently long execution of the command (5) yields the occurrence of the words of $\alpha_l u\star$ and $\star v\alpha_r$ types.

The third stage — "Glue". Finally, one must glue up the words of $\alpha_l u\star$ and $\star v\alpha_r$ types due to the last command (6). The strings obtained due to that command make the final ensemble for the given programme, since no one command is applicable to them. Now one can choose the strings of the length $L + 2t$ from the ensemble and select that one satisfying a criterion.

4 Imitator of Kirdin Kinetic Machine for the Gap Recovery Problem

To solve the problem of a gap recovery, we have implemented the consequent imitator of the algorithm for KKM. A number of copies of the left and right basements, as well as a number of copies of the dictionary are the input data for the imitator. To improve the imitator processing, we have modified that latter.

- The primers grow upright, only. As they reach the proper length of $L+2t-q$, a word containing the right basement should be connected to a string.
- To eliminate the inefficient steps, the original frequency dictionary has been modified. The function F_M was replaced with the function $\widetilde{F}_M$: $\Omega^* \mapsto N \cup \{0\}$, $\widetilde{F}_M = F_M(uv^{q-1}v^1\star) + F_M(v^1v^{q-1})$. This modification separates the words which can interact with a primer, due to the commands (5).
- The gap was splitted into several segments by check-points. Some strings resulted as a continuation of primer fail to grow up to the length $L + 2t$. Such truncated strings were twice eliminated during a course of a fulfillment development. Simultaneously, the population of strings that had reached the check-point successfully, were multiplied in number with specific factor.

5 Results

The primary goal of the computation experiments presented here was to learn whether the developed methodology would be fruitful for a gap recovery, and the programme implementing that former would be efficient enough for PC. A detail study of the properties of the recoveries themselves obtained at these experiments falls beyond the scope of this paper. One should bear in mind, that the length of the gap used in the computational experiments is too great to observe any statistically valid data concerning the relative number of the recoveries in comparison to the total number of the possible strings of such length.

To carry out the computational experiments on the gap recovery, we have used the sequence of a complete genome of avian CELO adenovirus, from the four–letter alphabet A, C, G, T. The genome is 43804 symbols in length, its accession number is U46933 in EMBL–bank. Artificial gaps of various length were created within this sequence. The length of these gaps are indicated at the Table 1. All the gaps were located approximately at the center of the sequence. The number of copies of primers was equal to 200, for each experiment. The recovery was executed for the family of the dictionaries of the thickness varied from 2 to 8. The length of a basement was equal to $q - 1$; a continuation of a primer was developed over a dictionary of the thickness $q - 1$, as well.

Table 1. Entropy values observed for various thickness q of dictionaries used to cover a gap; L is the length of the gap; S_1 is the entropy of the frequency dictionary obtained over the original sequence; S_2 is the entropy of the reference frequency dictionary; S_3 is the entropy of the augmented dictionary.

		$L = 2040$		$L = 6120$		$L = 10200$	
q	S_1	S_2	S_3	S_2	S_3	S_2	S_3
2	2.762391	2.762560	2.763216	2.764007	2.764846	2.766264	2.767316
3	4.139174	4.139590	4.140491	4.141834	4.143873	4.145174	4.146399
4	5.510682	5.511079	5.512071	5.513930	5.515805	5.517836	5.518700
5	6.874128	6.874118	6.874410	6.876863	6.876947	6.880571	6.879550
6	8.207368	8.205078	8.202776	8.203433	8.198796	8.202798	8.194168
7	9.408337	9.396622	9.388944	9.376704	9.357722	9.349918	9.318023
8	10.201937	10.171654		10.110157		10.035009	

In our computation experiments, the multiplication factor 2 was used, for the population of strings that had reached the check-points. Here all the coverings have been obtained over the reference dictionary. The computation experiments show that the methodology presented above is quite efficient for reparation of gaps in symbol sequences. The data presented here are obtained over the experiment with a single genetic text. Hence, one is not able to distinguish the effects resulted from the entity under consideration from those ones peculiar to the method itself.

The conditions of our computation experiments allowed 4800 strings which might be the coverings, for each frequency dictionary thickness (see Table 1). It is evident, some of them fail to be complement to the right basement. The number of the available coverings goes down almost exponentially, as the thickness of the dictionary grows up. The reference dictionary of the thickness 8 failed to produce a recovery. Meanwhile, this result may follow from the low number of copies of basement used to carry out the experiment.

A correspondence between the number of the recoveries obtained by KKM and the entire possible number of them was evaluated due to the following computation experiment. The original sequence has been replaced with the surrogate one that is a realization of a random process with the same probability distribution of the isolated symbols, as at the original entity. Totally, three realizations have been studied. Also, the number of primers in various series of the experiment increased from 750 up to 45000; all the other parameters (such as the gap length, its location inside the sequence, multiplication factor etc.) were the same. The number of proper fulfillments grows up, as the number of primers increases. This fact allows to assume the further growth of number of proper fulfillments following the growth of the number of primers. Thus, one hardly could expect that the best recovery has been obtained, in our computation experiments.

6 Discussion

The efficiency of KKM implementation to the problem of the lost data reconstruction is evident. Nevertheless, still there are some problems to be discussed. **The problem of a test object** is the basic one. A random non-correlated sequence is the standard test object. There could be other types of test objects. For example, a one-dimensional fractal with a relevant number of elementary cells could be a good test object. Probably, the symbol sequences generated with clear and unambiguous rules could be the test objects; one can consider expansion of Lioville numbers, or transcendental numbers.

The existence of the exact recovery is another problem. It might be observed for some specific sequences, and peculiar lengths of words used to make up a recovery. Anyway, the answer on this question will provide researchers with important knowledge concerning the basic properties of the method presented here. Since the exact recovery is impossible, in general, then one needs to compare the versions of the recovery obtained due to the method presented above, and the original sequence (see, e.g., [6]). The comparison of an original entity and that one obtained with the method presented above allows to clarify some peculiarities of the methodology.

A study of relationship between the entities obtained under the different extremal principles is of great importance. The question arises whether these two principles may produce the same covering or not, and if not, is a researcher able to evaluate the difference between them. That question is still waiting for the student.

A relation between the dictionary used to develop a covering and that one to evaluate the best entity among several is another substantial subject to be considered. Everywhere above we have searched for the best covering through the calculation of the entropy for the frequency dictionary of the same thickness, as that one used to develop the covering. There is no constraint to use the dictionaries of different thickness to develop a string covering a gap, and to find the best one [5]. We believe, this matter requires special, careful and comprehensive investigation; further discussion of that subject falls beyond the scope of our paper.

Finally, a number of questions concerning the improvement and/or modification of the imitator of KKM are to be studied carefully. For example, a development of self-training algorithms for a choice of the points of multiplication of primers that are selected from the entire pool of growing strings, as well, as the factor of that multiplication is the important question. We used the version with two point of replications; the factor of this replication was 2, 3 and 4. This choice was the matter of experience. A development of a proper modification of a frequency dictionary can also contribute significantly the progress in Kirdin kinetic machine applications.

Acknowlegdement. We are grateful to Prof. Alexander N.Gorban for his permanent and encouraging support of our efforts and this work. This was partially granted by Krasnoyarsk Regional Science Foundation (grant 11F0114C).

References

1. Kirdin A.N. Ideal ensemble model of parallel computations. In: "Neural informatics and its applications". Krasnoyarsk, KGTU, 1997. p.101.
2. Gorban A.N., Gorbunova K.O., Wunsch D.C. Liquid Brain: Kinetic Model of Structureless Parallelism. // Advances in Modelling & Analisis. AMSE, v.**5**, No.5, 2000.
3. Gorban A.N., Gorbunova K.O., Wunsch D.C. Liquid Brain: The Proof of Algorithmic Universality of Quasichemical Model of Fine-Grained Parallelism. Neural Network World, **4**/2001, p.391–412.
4. Katya O. Gorbunova. Kinetic Model of Parallel Data Processing // (Lecture notes in computer science; Vol. 1662) Parallel computing technologies: 5^{th} International Conference; Rroc./ PaCT-99, St.Petersburg, Russia, September 6–10, 1999. Victor Malyshkin (ed.). Springer, 1999, P.55–59.
5. Sadovsky M.G. Information capacity of symbol sequences / Open Systems & Information Dynamincs, 2002, **v.9**, pp. 231–247.
6. Sadovsky M.G. Comparison of symbol sequences: no editing, no alignment / Open Systems & Information Dynamincs, 2002, **v.8**, pp. 123–132.