Alignment Prediction Algorithm in Context-aware computing

Gennady Kyselov, Alexey Sergeev

National Technical University of Ukraine "Kyiv Polytechnic Institute"
kiselev@cad.kiev.ua

I. INTRODUCTION

Context-aware computing refers to a general class of mobile systems that can sense their physical environment, i.e. their context of use, and adapt their behavior according to human needs [1]. The work of context-aware pervasive system is defined by a context which can be specified.

The context is created on the basis of the data, retrieved from sensors, in mobile phones and other devices of home appliances. Each sensor control has the certain (special) task, which purpose is registration, reproduction and replication of the previous actions of the person. The obtained contextual data is necessary for the subsequent analysis and classification, for the purpose of operative revealing of similar situations in similar conditions.

Each situation has own context, and actions of the person depends on the situation in which she (or he) is located [2].

In this connection the question of competent realization of the algorithms providing a correct mark of a context, and adequate reaction to various actions of the person (situation) is important.

Situation definition, generation or reproduction of actions — demands from system of presence of the templating and a prediction mechanisms.

II. ALIGNMENT PREDICTION

Alignment prediction algorithm can be used in order to find similarities between time series in context.

In addition to the standard alignment approach in the method are used a set of typical context patterns. It represents the rules that guide the prediction process. For example, let $X_1=x_1, x_2, \ldots, x_m$ and $Y_1 = y_1, y_2, \ldots, y_n$ be two time series, where X_1 is a set of context data, and Y_1 is a typical context pattern.

If we align two time series X_1 and Y_1 to each other, we find sub-series in X_1 that has the maximum similarity to sub-series in Y_1. It will allow defining a situation in which there is a person and the possible "sample" actions corresponding to the given context. Matrix W in the size is for this purpose formed by MxN. Matrix W is filled with weight factors according to Needleman–Wunsch algorithm [3]. The filling principle of matrix W, differs from offered in [4].

Forward pass of algorithm:

1. To each element of a column there corresponds a symbol from set of context data, to each element of a line there corresponds a symbol from typical context pattern.

Elements of a matrix (cell) are described by two values: label and weight.

On a matrix, are filled cells, where in line and in column positions are found identical symbols (fig. 1). These cells are painted over.

To remember the painted over cells and do not compare symbols of sequences each time, we will enter a term – "label", that designates coincidence of symbols in two sequences.

If the symbols, that are correspond to a position of an element in a matrix, are equal among themselves, then value of a marker is equal to 1, otherwise value of a marker is equal to 0.

	B	c	a	a	d	b	c
A							
a							
b							
c							
b							
d							
d							

Fig.1 Setting labels, for matrix elements.

2. Are initialized weight for a zero line, and a zero column of a matrix. Values of scales are put down according to algebraic sequence beginning from value of an element in the left top corner of a matrix (fig. 2).

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1						
b	2						
c	3						
b	4						
d	5						
d	6						

Fig.2 Initializing weights of first column and row.

3. Repeat above steps, for remained elements of a matrix [4].

4. In the course of consideration of elements, there can be some cases (fig. 3):

1. Label of matrix element is equal 1 (the element of the first sequence is equal in the given position to an element of the second sequence). In that case value of an element of a matrix is equated to

value of an element in a position (i-1, j-1).

2. Label of matrix element is equal 0 (in this position no matches was found). In this case, the matrix element is equal to the larger of the values of the elements in positions (i-1, j) and (i, j-1).

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1	2					
b	2						
c	3						
b	4						
d	5						
d	6						

Fig. 3 Calculating weight.

5. If the element's label is equal to 1 and conditions (n-i) - (m-j)> =-1 and (n-i) - (m-j) <= 1 are carried out, in this case dimension of a matrix is reduced on (n-i) by quantity of lines and (m-j) to quantity of columns respectively.

6. The element (i, j), becomes "a zero" (initial) element of a matrix and the algorithm comes back to the third step (fig. 4).

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1	2	1	2	3	4	5
b	2		2				
c	3		3				
b	4		4				
d	5		5				
d	6		6				

Fig.4 Changing matrix dimension.

7. If the label is not equal 1, and the condition (n-i) - (m-j)> =-1 and (n-i) - (m-j) <= 1 is not carried out, putting down of scales proceeds without changing of dimension of a matrix (fig. 5).

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1	2	1	2	3	4	5
b	2		2	3	4	3	6
c	3		3	4	5	6	3
b	4		4	5	6	5	6
d	5		5			6	7
d	6		6			7	8

Fig. 5 Next steps.

Feature of the given realization consists in minimizing arithmetic addition operations, and comparison of elements, by gradual narrowing of dimension of a matrix.

The conditions (n-i) - (m-j)> =-1 and (n-i) - (m-j) <= 1 provide uniform filling of a matrix with scales, concerning its main diagonal.

Return pass of algorithm (finding of coincidence) always begins with the bottom right corner, as it contains the greatest value of weight (fig. 6):

1. Value of a label of an element is estimated.
2. If the label is equal 1, the element position (i, j) is remembered (as coincidence), and search moves to an element with a position (i-1, j-1).
3. If the label is equal 0, values of elements (i-1, j) and (i, j-1) are compared. The element about value of weight, on 1 smaller from the flowing gets out. The priority in a choice, in case of identical values of scales, is given to an element with position (i-1, j).
4. As an ending condition, can be taken an element of a zero line of a matrix, or zero cell (fig. 7).

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1	2	1	2	3	4	5
b	2		2	3	4	3	6
c	3		3	4	5	6	3
b	4		4	5	6	5	6
d	5		5			6	7
d	6		6			7	8

Fig.6 Back trace.

	B	c	a	a	d	b	c
A	0	1	2	3	4	5	6
a	1	2	1	2	3	4	5
b	2		2	3	4	3	6
c	3		3	4	5	6	3
b	4		4	5	6	5	6
d	5		5			6	7
d	6		6			7	8

Fig.7 Result: was found two similar elements.

III. TESTING

In a kind of that the algorithm belongs to the class of algorithms of dynamic programming, its basic advantage consists in speed of work for a finding of the answer, as he demands only by two passes.

Testing indispensable condition is presence of group of templates (the sequences brought in advance) which characterize one of possible conditions of system.

One of the simplest examples of the usage is recognition of time of day, by a camera placed on a shop's facade. In case of advantage of dark tones, it is possible to tell with confidence that in the street is a night-time. In opposite case there is a day in the street.

However, how to be in a twilight case or when in the street cloudy weather, after all then we will have values of comparisons close to a night-time, but nevertheless not such as in ideal value of a template.

For such cases, we introduce lower and upper limits of possible deviations from the pattern states.

In testing, it is necessary to take into account the hardware capabilities of the sensor; these include the size of RAM. So in the case of long sequences, they should be separated into smaller, more calculated groups.

IV. CONCLUSION

A local alignment based context prediction method that suits ubiquitous environments due to a high tolerance for slight fluctuations in user behavior and because it is computationally not expensive.

Proposed in this paper, algorithm implementation could accelerate the running time, by calculating only the necessary elements of the matrix, while avoiding unnecessary comparing operations the values of the weights of the previous items (located below and left).

The condition of the mandatory approach matches to a diagonal matrix, ensures their uniform distribution over the entire set of input sequences.

REFERENCES

[1] Context-aware pervasive systems, A Wikipedia project, url: http://en.wikipedia.org/wiki/Context-aware_pervasive_systems

[2] Context Prediction by Alignment Methods. Stephan Sigg, Sandra Haseloff, Klaus David. Chair for communication Technology University of Kassel. D-34121 Kassel, Germany

[3] A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins, S. B. Needleman and C. D. Wunsch, Department of Biochemistry, Northwestern University, V. A. Research Hospital , Chicago, Ill. 60611, U.S.A , 21 July 1969, pp. 4-6

[4] Sequence Alignment Needleman-Wunsch, Rolf Backofen, Lehrstuhl fur Bioinformatik, Institut fur Informatik, 2010