CSIT Conference 2019, Yerevan, Armenia, September 23-27

Segmentation of String to Match a Fuzzy Pattern

Armen Kostanyan
American University of Armenia
Yerevan, Armenia
e-mail: armko@aua.am

Arevik Harmandayan
American University of Armenia
Yerevan, Armenia
e-mail: arevik.harmandayan@gmail.com

ABSTRACT

The problem of segmentation of a given string to match a fuzzy pattern is considered in this paper. The fuzzy pattern is defined as a sequence of fuzzy properties. It is assumed that each string can match a fuzzy property in some measure. Being increasing, decreasing, or oscillating are examples of fuzzy properties of a numerical sequence. The problem we consider is how to split the given string (sequence) into substrings (contiguous subsequences) to match the pattern as well as possible. This problem can be considered in the frame of the fuzzy clustering problem that has many applications in such areas as image processing, bioinformatics, etc. It can also be viewed as a special case of the fuzzy string matching problem.

In this paper, we propose the optimal solution to the fuzzy segmentation problem and consider its application to the decomposition of a given function.

Keywords

Fuzzy clustering, string matching, fuzzy logic.

1. INTRODUCTION

The problem of data clustering and pattern recognition is widely used in bioinformatics and image processing [1][2]. The latter can be considered as a problem of grouping elements around certain points of consolidation, which are determined in the process of evaluation [3]. Due to the difficulties in the precise description of the cluster in many applications, modern approaches use fuzzy clusters [4][5][6].

On the other hand, there are a number of approximate string matching algorithms that solve the string matching problem with a given accuracy [7]. A specific case of approximate string matching is fuzzy string matching [8], when pattern is considered as a sequence of values of a linguistic variable.

The problem we solve in this paper combines aspects of the fuzzy clustering and fuzzy string matching problems. More precisely, we consider a fuzzy version of Bellman's string segmentation problem [9], where the problem of splitting a given string into k parts with elements similar to each other was considered. By contrast, we assume that segmentation must be done in accordance with a sequence of fuzzy properties in order to best match it. (It is assumed that the resulting segments do not necessarily have the same length.)

After providing preliminaries, we formulate the fuzzy segmentation problem and give a dynamic programming method of polynomial complexity to solve it. As an application of the proposed algorithm, the problem of fuzzy decomposition of a function is considered. Finally, the conclusion summarizes the obtained results.

2. PRELIMINARIES

Suppose $(L, \vee, \wedge, 0, 1)$ is a lattice with the least element 0 and the greatest element 1. We also assume that a binary operation $\otimes$ of accumulation is defined on L such that
$(L, \otimes, 1)$ is a commutative monoid and for all $a, b, c \in L$,

$$a \leq b \implies a \otimes c \leq b \otimes c.$$

Let us call the elements of the set L measures.

The fuzzy subset A of the universal set U [10] is defined by the membership function $\mu_A : U \to L$ that associates with each element x of U a measure $\mu_A(x)$, called the degree of membership of x in A. A fuzzy subset A in U can be represented as an additive form

$$A = \sum_{x \in U} x/\mu_A(x).$$

We say that an element x definitely belongs to A, if $\mu_A(x) = 1$, and definitely does not belong to A, if $\mu_A(x) = 0$. On the contrary, if $0 < \mu_A(x) < 1$, we say that x belongs to A with degree $\mu_A(x)$.

3. THE FUZZY SEGMENTATION PROBLEM

Let Σ be an alphabet of symbols. Suppose that we are given a sequence $T[1..n]$ of symbols from Σ of length n, called *text*.

Let Σ^* be the set of all finite length strings in Σ. Let us define a *fuzzy segmentation symbol* (or, briefly, a *segmentation symbol*) as a fuzzy subset of Σ^* that allows measuring strings in the alphabet Σ by elements from L. For a given segmentation symbol α and a string $x \in \Sigma^*$, we say that x matches α with the grade $\mu_\alpha(x)$.

Suppose that we are given a sequence $P[1..m]$ of segmentation symbols of length m, called a *fuzzy segmentation pattern* (or, briefly, a *segmentation pattern*). Given the text $T[1..n]$ and the segmentation pattern $P[1..m]$, we define the fuzzy segmentation problem (or, in short, the *segmentation problem*) as decomposition

$$T[1..n] = T[1..j_1]T[j_1 + 1..j_2]\ldots T[j_{m-1} + 1, j_m]$$

$$(1 \leq j_1 < j_2 < \ldots < j_{m-1} < j_m = n),$$

of the string $T[1..n]$ into m substrings

$$t_1 = T[1..j_1], t_2 = T[j_1{+}1..j_2], \ldots, t_m = T[j_{m-1}{+}1, j_m = n])$$

that maximizes the value

$$\mu_P(t_1, ..., t_m) = \bigotimes_{i=1}^{m} \mu_{P[i]}(t_i).$$

Denote

$$\mu_P(T) = \max\{\mu_P(t_1, ..., t_m) | \text{ for all decompositions of } T \text{ into substrings } t_1, ..., t_m\}.$$

Example: Let us choose the set L of measures to be the segment [0, 1] of ordered reals with multiplication as an accumulation. Suppose that Σ={0, 1} and the segmentation symbols α_0 and α_1 are defined over Σ^* as follows: for a string $x \in \Sigma^*$, the values $\mu_{\alpha_0}(x)$ and $\mu_{\alpha_1}(x)$ are the relative numbers of 0's and 1's in x, correspondingly. Then, for

$$T = 101110001101 \text{ and } P = \alpha_1\alpha_0\alpha_1,$$

the solution to the segmentation problem is the decomposition $T = t_1t_2t_3$, where

$$t_1 = 10111, t_2 = 000, t_3 = 1101$$

with $\mu_p(T) = (4/5) \cdot (3/3) \cdot (3/4) = 3/5$.

4. SOLUTION TO SEGMENTATION PROBLEM

The segmentation problem can be solved using the following recurrent equation:

$$\sigma_P(T) = \begin{cases} 1, \text{ if } m = 0 \text{ or } n = 0 \\ \mu_{P[m]}(T[1..n]), \text{ if } m = 1, n > 0 \\ \max\limits_{1 \le k \le n} \left(\sigma_{P[1..m-1]}(T[1..k-1]) \otimes \mu_{P[m]}(T[k..n])\right), \\ \qquad\qquad \text{if } m > 0, n > 0. \end{cases} \quad (1)$$

Direct programming of this equation leads to an overlap of sub-problems, so we will use the dynamic programming approach to get a better result. For $0 \le i \le m$, $0 \le j \le n$, denote

$$s[i, j] = \sigma_{P[1..i]}(T[1..j]) \in L.$$

The recurrent equation (1) yields

$$s[i, j] = \begin{cases} 1, \text{ if } i = 0 \text{ or } j = 0 \\ \mu_{P[i]}(T[1..j]), \text{ if } i = 1, j > 0 \\ \max\limits_{1 \le k \le j} \left(s[i-1, k-1] \otimes \mu_{P[i]}(T[k..j])\right), \\ \qquad\qquad \text{if } i > 0, j > 0. \end{cases} \quad (2)$$

To obtain the optimal segmentation, let us also maintain the integer value b[i, j], $2 \le i \le m$, $1 \le j \le n$, such that $b[i, j]$ denotes k that maximizes the measure

$$s[i-1, k-1] \otimes \mu_{P[i]}(T[k..j])$$

in formula (2). The following procedure represents the memoization phase (i.e., the calculation of matrices s and b):

Memoization(P, T)
1 *m = P.length, n = T.length*
2 let $b[2..m, 1..n]$ and $s[0..m, 0..n]$ be new tables
3 **for** $i = 0$ **to** m
4 $s[i, 0] = 1$
5 **for** $j = 1$ **to** n
6 $s[0, j] = 1$
7 $s[1, j] = \mu_{P[1]}(T[1..j])$
8 **for** $i = 2$ **to** m
9 **for** $j = 1$ **to** n
10 $s[i, j] = 0$
11 **for** $k = j$ **downto** 1
12 $r = s[i-1, k-1] \otimes \mu_{P[i]}(T[k..j])$
13 **if** $r > s[i, j]$
14 $s[i, j] = r$
15 $b[i, j] = k$
16 **return** s and b

The $s[m, n]$ component represents the optimal value for the segmentation problem. The optimal segmentation of the input string can be constructed on the basis of table b in the following recursive way:

Print-Optimal-Segmentation(*b, i, j*)
1 **if** $i == 0$ or $j == 0$
2 **return**
3 **if** i==1
4 **Print**(1, j)
5 **else**
6 **Print-Optimal-Segmentation**(b, i-1, $b[i, j]$-1)
7 **Print**($b[i, j]$, j)

The initial call to this procedure is **Print-Optimal-Segmentation(*b, m, n*)**.

5. ANALYSIS

Three nested loops in lines 8, 9 and 11 lines in the **Memoization** procedure run m, n and at most n times, correspondingly. Under a reasonable assumption that the value $\mu_\alpha(x[k..j])$ can be obtained from the value $\mu_\alpha(x[k + 1..j])$ in constant time, we obtain $O(mn^2)$ time complexity for the **Memoization** procedure. The **Print-Optimal-Segmentation** procedure obviously has $O(n)$ time complexity. Thus, the considered solution to the segmentation problem has $O(mn^2)$ time complexity. The procedure requires $O(mn)$ space to store the L-value table s and the integer-value table b.

6. FUZZY DECOMPOSITION OF A FUNCTION

There are a number of applications (such as, statistical analysis, data mining, etc.), in which it is necessary to split a given function into consecutive parts so that they would best match some predefined list of properties. Let us consider one particular case of this problem. Suppose that a function f is given in the *point-value form*, i.e., as a sequence

$$f : (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \quad (3)$$

of points on the plane, where $y_i = f(i), 1 \le i \le n$. Consider the problem of decomposition of the function f based on a given sequence of the *increasing, decreasing* and *oscillating* properties. To be precise, given a sequence of points (3),let us first define the segmentation symbols *inc, dec* and *osc* using the following measure functions:

• $\mu_{inc}(f)$ is the relative part of the domain of f, where it is increasing, i.e.,

$$\mu_{inc}(f) = \frac{1}{x_n - x_1} \sum_{\{1 \le i \le n | y_{i+1} \ge y_i\}} (x_{i+1} - x_i);$$

• $\mu_{dec}(f)$ is the relative part of the domain of f, where it is decreasing, i.e.,

$$\mu_{dec}(f) = \frac{1}{x_n - x_1} \sum_{\{1 \le i \le n | y_{i+1} \le y_i\}} (x_{i+1} - x_i);$$

• $\mu_{osc}(f)$ is the relative number of the inflection points of function f, i.e.,

$$\mu_{osc}(f) = \frac{|\{2 \le i \le n-1 | y_{i-1} < y_i > y_{i+1}\}|}{n-2}.$$

For example, if

$$f : (1, 1), (3, 5), (5, 7), (9, 3), (11, 5),$$

then

$$\mu_{inc}(f) = (2+2+2)/10 = 0.6; \quad \mu_{dec}(f) = 4/10 = 0.4;$$

$$\mu_{osc}(f) = 2/3 \approx 0.67.$$

Let us define the set of measures L to be the segment [0, 1] of real numbers with the operation of calculating the *minimum* as an accumulation. Given the point-value form (3) of a function f and the pattern $P[1..m]$ representing the sequence of properties, define the problem of decomposition of f as the problem of splitting the sequence (3) into m parts:

$$f = f_1 f_2 ... f_m$$

so that each next part begins at the point where the previous one ends, in a way that the value

$$\min\{\mu_{P[1]}(f_1), \mu_{P[2]}(f_2), ..., \mu_{P[m]}(f_m)\}$$

is maximized.
The diagram in **Figure 1** demonstrates the result of applying the proposed algorithm to the depicted function and the pattern P=*inc.osc.dec.*

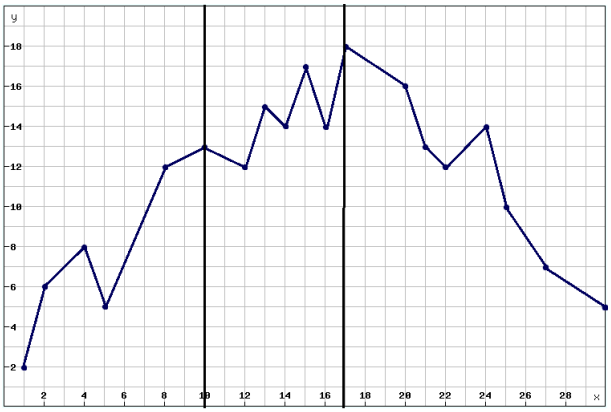

Figure 1: f is *fuzzy increasing* in [1..10], *fuzzy oscillating* in [10..17] and *fuzzy decreasing* in [17, 30].

7. CONCLUSION

The problem of segmentation of a given string according to a fuzzy pattern, which has applications in image processing and bioinformatics, is considered in this paper. A polynomial algorithm using the dynamic programming approach is suggested to solve this problem. The problem of decomposition of a given function based on a given list of properties is considered as an application of the proposed algorithm.

8. ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education and Science of the Republic of Armenia, project N 18T-1B341.

REFERENCES

[1] Wang ZhenZhou, "Image segmentation by combining the global and local properties", *Expert Systems with Applications Volume 87* , pp. 30-40, 2017.

[2] L. Liao, K. Li, K. Li, Q. Tian, and C. Yang, "Automatic density clustering with multiple kernels for high-dimension bioinformatics data", *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* , Kansas City, MO, pp. 2105-2112, 2017.

[3] Estivill-Castro, Vladimir, "Why so many clustering algorithms – A Position Paper", *ACM SIGKDD Explorations Newsletter*, pp. 65–75, 2002.

[4] Bezdek, James C, " *Pattern Recognition with Fuzzy Objective Function Algorithms*", 1981.

[5] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data", *IEEE Transactions on Medical Imaging, vol. 21, no. 3*, pp. 193-199, March 2002.

[6] Dembélé, Doulaye amp; Kastner, Philippe, "Fuzzy C-Means Method for Clustering Microarray Data", *Bioinformatics (Oxford, England)*, pp. 973-80, 2003.

[7] Baeza-Yates, R.; Navarro, G., "A faster algorithm for approximate string matching". In Dan Hirchsberg; Gene Myers (eds.). *Combinatorial Pattern Matching* (CPM'96), LNCS 1075. Irvine, CA. pp. 1–23, 1996.

[8] A. Kostanyan, "Fuzzy String Matching with Finite Automata", in *Proceedings on 2017 IEEE Conference CSIT-2017*, Yerevan, Armenia, pp. 25-29. IEEE Press, USA (2018).

[9] R. Bellman, "On the approximation of curves by line segments using dynamic programming", *Communications of the ACM, Volume 4 Issue 6*, pp. 286, 1961.

[10] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences, vol. 8*, pp. 199-249, 1975.