A New String Matching Algorithm and its Application in Hand-Written Digits Recognition

Mehrnoosh Bazrafkan
Department of Computer Engineering,
Marvdasht Branch, Islamic Azad University,
Marvdasht, Iran

Ali Broumandnia
Department of Computer Engineering ,Islamic
Azad University-South Tehran Branch

ABSTRACT

In this paper a new algorithm is introduced for syntactic pattern recognition and string matching by using linked listdata structure which later could be used for hand written digits recognition.At first,handwritten digits are changed to string as input pattern by using chain-codethen the achieved string is recognized by using refer algorithm being implemented by linked list. This refer algorithm is able to compute the distance between the chain code strings shown in the implementation. The suggested algorithm reduces time complexity ofleven shtein's algorithm from second-order to linear-orderand in addition is able to decrease the consumption memory and increase accuracy of handwritten digits recognition as well. Our proposed implemented algorithm has 94.8% accuracy over 3000 handwritten digits samples.

Keywords

handwritten digits, chain-code, syntactic pattern recognition, string matching, linked list data structure, dynamic programming, time complexity.

1. INTRODUCTION

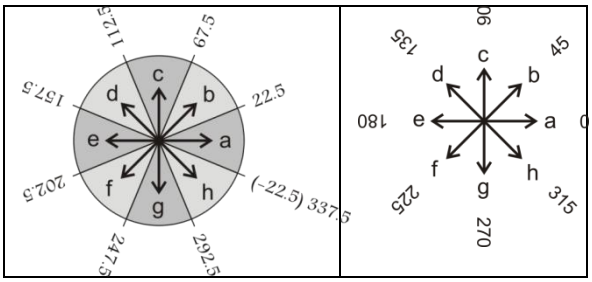

Figure 1: (a)the main 8 directions and their angles ;(b)the range of the main 8 directios.

Some of the pattern recognition algorithms use the data structure string, tree and graph in which, the strings have the most application due to their simplicity. One of the changing approach pattern to string is the use of the chain code, [1].In this method ,the pattern changes to some directions , in which any direction is shown with a letter or a digit , therefore the pattern converts to a string of letters or digits.

Most of the methods used in pattern recognition are based on similarity measure .In these methods ,the similarity between input pattern and sample patterns is computed and the decision for classification of input pattern is made, based on the simplicity rate. One of the most important methods introduced for calculating the similarity measure between two strings is called Leven Shtein method, [2] and most of the other methods for measuring strings similarity are based on this method. Following this paper firstly , the method of changing pattern to chain code is shown[3], and then the new method string matching is introduced by the use of linked list data structure and then ,the results of implementations by the proposed method and its comparing with previous methods are shown and finally ,the conclusion section is introduced .

2. CONVERSION OF PATTERN TO A CHAIN CODE

In conversion of pattern to a chain code, at first some directions (usually 8) would be defined and any of this directions is called with a letter (Fig 1(a)).

For the chain code with 8 directions,a used alphabet set could be shown as follows:

T={a,b,c,d,e,f,g,h} (1)

In the above relation , T is an alphabet set and the letters a to h are different directions(fig 1(a)).Since the number of directions are 8 in this problem and as a complete rotation is 360° degrees then the centrality of directions would respectively be: $0^{\circ}, 45^{\circ}, 90^{0}, \dots, 360^{\circ}$ that their values are derived from the relation 2.

$$\theta_i = \frac{360}{8} i \qquad i = 0,1,\dots,7 \quad (2)$$

In the above relation , θ_i is the angle of i-th direction.

To classify the pattern firstly, the input patterns must be changed to chain code. To accomplish this goal the input pattern should be converted to some discrete vector and after computing the vector angles, each vector is specified with a direction which is closer to its vector

angle. Suppose the vector angle is , to convert the vector to
any direction ,the number of direction(k) is achieved from the relation 3.

$$k = arg\min_i(|\alpha - \theta_i|) \qquad (3)$$

According to the relation 3, the range of any direction is computed as table 1 .These ranges are shown in figure 1(b).

Table1:the 8 main directions and their angles

Range	Centrality Angle	Direction
-22.5 to 22.5	0	a
22.5 to 67.5	45	b
67.5 to112.5	90	c
112.5to157.5	135	d
157.5to202.5	180	e
202.5to247.5	225	f
247.5to292.5	270	g
292.5to337.5	315	h

An example of the conversion pattern (digit 8) to the chain code is shown in figure 2.After converting these pattern's vectors to 8 directions, if these directions are respectively read from the top right corner of the figure 2 ,the string "ehafebb" would be achieved. The full description of the pattern conversion method to the chain code is described in [3].

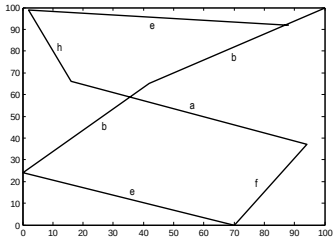

Fig2:An example of a chain code for 8

After converting the problem patterns to the chain code, in the procedure of learning and recognition we would encounter with a string of letters.A simple classification method which could be used with the chain code is called 1NN,[4] ,in which the whole learning patterns and the new pattern are changed to a string and then the distance between the new string pattern and strings learning patterns is computed and the new pattern is put in a classification which has the least distance with its learning pattern. There are many approaches for computing the distance between two strings, that leven shtein is the most important methods. In our proposed paper we introduce a new algorithm which uses linked list and for this reason it has double capability in speed and accuracy for edit operations such as deletion, substitution and insertion comparing to leven shtein algorithm.An example of Persian character recognition using chain code has been in [5].

3. LEVEN SHTEIN METHOD

In this method ,the distance between two strings ,is specified with the least number of edit operations required .[2] For example ,the distance between "kitten" and "sitting" is 3.beacuse the least number of edit operations required for converting a string to another one ,are described in 3 below operations.
1)kitten→ sitten (substitution of k to s)

2)sitten→sittin(substitution of e to i)

3)sittin → sitting (the insertion of g at the end of the string)

The dynamic programming by the bottom-up approach is used for implementing this method,[6].The pseudo code of this method is shown in figure 3.In this algorithm , a matrix d with (n+1)*(m+1)dimensions is used in which n and m are two input strings length.

```
1  begin
2    D(0,0):=0;
3    for i=1 to n do D(i,0) := D(i-1,0) + c(X(i)→ε);
4    for j=1 to m do D(0,j) := D(0,j-1) + c(ε→Y(j));
5    for i=1 to n do
6      for j=1 to m do
7        begin
8          m1 := D(i-1,j-1) + c(X(i)→Y(j));
9          m2 := D(i-1,j) + c(X(i)→ε);
10         m3 := D(i,j-1) + c(ε→Y(j));
11         D(i,j) := min(m1,m2,m3);
12         if m1 = D(i,j) then set pointer from (i,j) to
(i-1,j-1);
13         if m2 = D(i,j) then set pointer from (i,j) to
(i-1,j);
14         if m3 = D(i,j) then set pointer from (i,j) to
(i,j-1);
15       end;
16    d(x,y) := D(n,m);
17  end
```

Figure 3.the leven shtein's algorithm for computing the distance between two strings

This matrix calculates the number of edit operations required for reaching of first string to second string.

In this matrix the required operations numbers for altering the first string to the second string is kept and here every horizontal movement (from a cell to its right side), each vertical movement (form a cell to its bottom side) and every diagonal movement (form a cell to its right side bottom cell) illustrate insertion, deletion and substitution operations, respectively, [2].
After Leven Shtein method, many other techniques for its optimization were introduced such as:

- Damerau-LevenShtein method[7]:

Here the distance between two strings is computed by four operations as deletion, insertion, substitution and symmetry.

One of its applications is error founding in typing words in which it is possible that a typist because of the speed of typing might substitute two letters by mistake. For example in the word "recieve" and "receive".

- Hirschberg Algorithm,[8]:

This algorithm is similar to Leven Shtein algorithm but for every operations a different cost is determined.

- Cyclic string matching method:

Here, displacement of the letters in strings as a loop, isignored. For example the strings "abcdefg" and "efgabcd" have zero distance.

4. THE SUGGESTIVE METHOD

In this paper ,a new method for string matching is introduced based on Leven shtein method in which the linked list data structure is used illustrated in fig 5.In this optimized and completed method the capability of for all editing operations for matching two strings is existed .
Thealgorithm's time complexity is O(min(n,m)) or O(n) generally. In some cases as below the time complexity approaches from T(n) to T(n/2).
At the first case, inthe twoanalyzing strings, the two successive characters from the above strings would match together.
At the second there would be a state in which on the contrary to the mismatching of the two characters in the analyzed strings, the other successive character would match together. In this case, only the substitution editing operation is required. At the third state in addition to the two analyzed characters in the two strings don't match together, the second string successive character wouldn't match with the present character compared as well. So if the occurrence of the above three cases would be more in the problem then T(n) gets closer to T(n/2).
In this paper, the cost of insertion and deletion operations would be equal to one and the cost of substitution operation would be related to input symbol according to the below table (2) and then the act of classification on the input patterns is done.

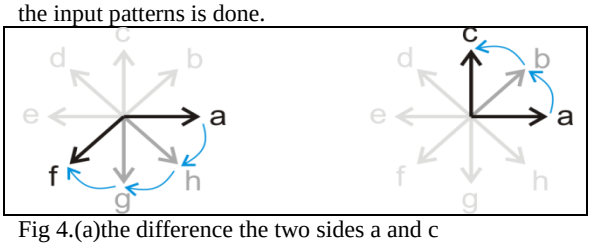

Fig 4.(a)the difference the two sides a and c
(b)the difference the two sides a and f

Table2.Some cost of substitution operation

α	β	$c(\alpha \to \beta)$
a	*a*	*0*
a	*b*	*1*
a	*c*	*2*
a	*d*	*3*
a	*e*	*4*
a	*f*	*3*
a	*g*	*2*
a	*h*	*1*
b	*a*	*1*
b	*b*	*0*
b	*c*	*1*
b	*d*	*2*
b	*e*	*3*
b	*f*	*4*
b	*g*	*3*
b	*h*	*2*

```
1  node  *h , *p , *q , *k ;
2  h = first x ; // h is linked list of first string //
3   p = q = first y ; // p and q are linked lists of second
string //
4     ce=0;
5  while not eof(h) and not eof(q)
6  { if data(h) is equal data(q)
7       if data(next(h)) is equal data(next(q))
8       {  h=h→next→next;
9          p=q→next;
10          q=p→next;  }
11     if data(next(h)) is not equal data(next(q))
12     { h=h→next;
13       p=q;
14       q=p→next;  }
15   if data(h) is not equal data(q)
16   {  if eof(h) or eof(q)
17       Break;
18     if data(next(h)) is equal data(next(q))
19       { m=c((data(q))→ (data(p)));
20         ce+=m;
21         q→data=h→data;
22         h=h→next→next;
23         p=q→next;
24         q=p→next;  }
25    if data(h) is equal data(next(q))
26     {  h=h→next;
27       p→next=q→next;
28        // delete;
29          ce+=1;
30        p=p→next;
31        q=p→next;  }
32      if data(next(h)) is equal data(q)
33      {  k=new node; // k is a pointer of new node //
```

```
34          k→data=h→data;
35          k→next=q;
36          p→next=k;
37          h=h→next→next;
38         p=q;
39          q=p→next;
40           // Insert;
41             Ce+=1;     }
42       if data(next(h)) is not equal data(q)
43              {delete two nodes from the point of p for
second string
44             ce+=2;
45          q= q→next→next;
46           p→next=q;
47           q=p→next;}
48  p → next = h ; // add  rest of h to the end of q //
49   p=h;
50 while(p!=null)
51 ce+=1;
52 p= p → next;//insert in end of string q//
```

Fig 5.the suggestive algorithm for string matching

5. THE HAND-WRITTEN DIGITS RECOGNITION

The data base used in this section is prepared in the University of Turkey in 1998.this data base consist of 7494 training samples and 3498 test samples. The related data given to each sample is made of a display page sensitive to pressure (an instrument similar to a light pen).Here the writer takes a pen and writes his favorite digits on a display page.Six sample of the existed digits in this data base are shown in figure 6.

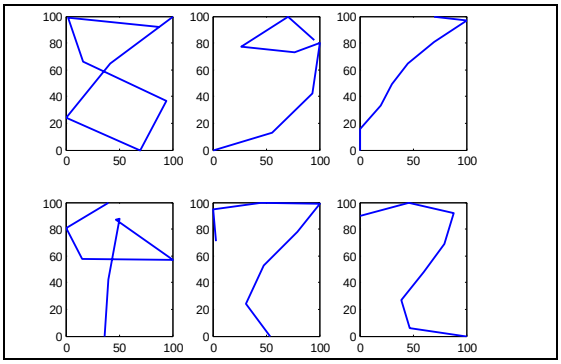

Fig6. A few samples of the digits in the data base

The nearest neighbor method is used for classification data base .The results of these implementations are shown in the table 3 as shown the suggestive method compared to the traditional method has reduced the error remarkably .In table 3 ,the concept of the variable editing costs is the distance chain code directions.

Table3.The results of data base handwritten digits classification

approach	(%) Recognition rate
Leven shtein	87.02
Leven shtein with variable cost	90.98
The proposed algorithm	94.8

6. CONCLUSION

In this paper,a new method for changing pattern to chain code and a new algorithm for string matching based on leven shtein algorithm for computing the distance between chain code patterns are introduced .From the most important advantages of this algorithm is the high recognition rate and as the result the good accuracy of this method.This algorithm with the linear time complexity compare to Leven Shtein method with second order has less time complexity so this method takes less time for execution .This property is suitable for every system with any hardware specification. Also due to the use of linked list data structure the memory loss is reduced to zero. The results for handwritten digits recognition showed that the most errors of classification is related to digit 7 with 52% of the errors.Also,in the cases which digit 7 is misclassified in more than 77% ,this digit is placed in one of the classes of 1 or 2 by mistake.

7. REFERENCES

[1] H. Freeman, *On the encoding of* arbitrary geometric configurations, IEEE Trans, Electron, Compute, EC-10, 260-268, 1961.

[2] Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversalsPDF. Soviet Physics Doklady 10: 707–10, 1966.

[3] Gonzalez and Woods ,Digital Image Processing Third Edition, Prentice Hall ,2008

[4] Sergios Theodoridis, Konstantinos Koutroumbas, *Pattern Recognition fourth edition*. Academic Press is an imprint of Elsevier,2009.

[5] H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar, *Multi-Font Farsi/Arabic Isolated Character Recognition Using chain code s*, World Academy of Science, Engineering and Technology 43, 2008.

[6] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. , *Introduction to Algorithms (2nd ed.)*, MIT Press & McGraw-Hill, ISBN 0-262-03293-7 . pp. 327–328, 2001.

[7] Gonzalo Navarro. *A guided tour to approximate string matching*. ACM Computing Surveys (CSUR) archive, 33(1), pp. 31-88, 2001.

[8] D. S. Hirschberg. *A linear space algorithm for computing maximal common subsequences*. Comm. A.C.M. 18(6) p341-343, 1975.