A Hopfield Neural Network Algorithm for Automatic Name Placement for Point Feature

FAN Hong  TANG Guoqiang  ZHANG Zuxun  DU Daosheng

ABSTRACT  This paper presents a method of adding label to the map especially for the point feature. This method overcomes the shortcoming of traditional methods, e.g., Conflict-Backtracking method. Its kernel algorithm use the hopfield neural network to find the best label position for point feature. The experimental results proves that this algorithm has good permanence and high speed.

KEYWORDS  hopfield neural network; energy function; map name placement

CLC NUMBER  P283.7; P208

1 Experimental data

Our experimental data comes from the 1:25 National Base Map Database and comprises three topographical map sheets that are stored in the vector format of ARC/INFO. The map numbers of the three maps are H-48-{10}, H-48-{13} and H-48-{14}, respectively, each of which is stored into 16 layers that include hydrogen, roads, vegetation and administrative boundary. These data contain both the spatial features in the form of digital map and the attributes information such as map name and the feature code in terms of the national standard encoding of geographic feature by which we can easily discern label attributes such as size, style and spacing, etc.

The three map sheets contain 3,251, 1,651 and 2,734 resident points, respectively. The area of these maps is near Chengdu city, the capital of Sichuan Province in west China and well-known for its dense-population. Because map name placement in a dense map is more difficult than that in a sparse map, these three maps can serve as good representatives of point map.

2 Name placement principle for point feature

Map features can be categorized into points, lines and polygons in terms of their spatial distribution. We find that the following three principles is in common adaptable.

① “Belonging to” principle: label should refer unambiguously to its referent feature.

② “Avoiding off” principle: the label name should make away for important geographic features, but not overlap important features, especially the features or labels of the same color.

③ “Accustomed to” principle: the character position, character order and so on should accord with the reading custom of the readers.

The point name placement algorithm is used to identify point feature depicted by pictorial symbols. According to both the above-mentioned principle and the particularity of the point feature, the principle suitable for point feature is given as
follows.

1) One of the eight ranked positions surrounding a feature symbol with referencing fixed positional weight is used to place a point feature label. Fig. 1 shows the ranked label positions for point feature. Position with higher weights are preferred over positions with lower weights.

![Fig. 1 Ranked label positions for point feature](image)

2) Point label should not overlap the placed label and any point feature. Point label should not overlap the important linear feature of the same color such as railways and major roads, etc. While overlap is unavoidable, efforts should be taken to decrease it. Labels and labels must not overlap each other.

3) The label name and the related features had better be located in the same side of the nearby linear feature, and the label name can not overlap boundary.

3 Solution to the automatic name placement for residence feature

Point feature involves residence, flag point and elevation point etc., among which residence is representative for its largest number and highest density. According to the principle of name placement for point feature mentioned above, the following strategies are taken to automate name placement for residence of topographical map. The total solutions contain the following three steps:

**Step 1:** coarsely choose the candidate position and determine the optimal level for every position.

First of all, eight available positions for every residence are chosen, shown in Fig. 1. Through the evaluation of “readability” and “belonging to” relations of the eight positions, a weight factor is given for every possible position, and the detailed method is depicted as follows.

1. The weight of the candidate position is the function of the primary weight and the secondary weight, that is:

   \[ W = W_{primary} \times W_{secondary} \]

2. Specify a basic weight from 1.0 to 0.93 corresponding to the priority from high to low.

3. Use a rectangle area to represent the area that a label name occupies and judge if the area will overlap the important feature such as railways, major roads etc. Different secondary weight factor is assigned accordingly. For example, if the label rectangle of one feature overlaps with a railway or a major road, according to the extent of overlap, its secondary weight should be assigned to 0.1, 0.2, 0.3 or 0.4, respectively. In order to control the overlap with the minor road, the secondary weight of 0.51, 0.52, 0.53 should be assigned for different extents of overlap.

4. Judge whether the label and the referent feature appear on the same side of the boundary and whether the label overlaps with the boundary. To handle this case, an appropriate secondary factor should be designated.

**Step 2:** construct a hopfield network, set the initial value for it and make it run. If necessary, make the network run many times, observe its convergence, select the best results and record it.

**Step 3:** after taking the previous steps, local optimal processing is made so as to resolve the remaining conflicts between the residents labels. Few remained conflicts that have not yet been resolved in the end will be adjusted and revised manually.

The core of the whole algorithm is step 2, that is to use hopfield network to find the best local or global label position for every resident on topographical map.

This hopfield approach runs the network in an iterative way that will converge very fast, avoiding many times backtracking and the nested backtracking of the traditional method, which will, as a result, improve the efficiency of the search algorithm.
4 Algorithm of the hopfield neural network to find the best label position for residence

Hopfield neural network is a one-level of feedback network. Let \( N_1, N_2, \ldots, N_n \) stand for \( n \) neural units, \( W_{ij} \) stands for the connection weight from \( N_i \) to \( N_j \). If we use \( W \) to represent the connection strength between \( n \) nodes, Hopfield is symmetric, then:

\[
W_{ij} = W_{ji}, \quad (i, j \in \{1, 2, \ldots, n\})
\]

For the continuing feedback network, when the network is working, the relation between the input and the output can be represented in terms of the following status equation:

\[
\frac{du_i}{dt} = -\frac{u_i}{t} + \sum_j W_{ij}v_j + I_i
\]

\[
v_i = g(U_i)
\]

where \( g(x) \) is a continuing monotony ascending function with up limit; \( U_i \) represents the status of the neural unit \( i \); \( V_i \) represents the output of the neural unit \( i \); \( I_i \) represents the bias of the neural network namely the stimulus coming from the external world.

If the evolution of the equation takes the asynchronous way, at any time \( t \), only one status of one neural unit will be changed. Suppose \( U_i(t) \) stands for the sum of all the inputs of the neural unit \( i \) at the time \( t \), and \( V_i(t+1) \) stands for the output status this neural unit at the time of \( t+1 \), then:

\[
U_i(t) = \sum_j W_{ij}V_j(t) + I_i
\]

\[
V_i(t+1) = g(U_i(t)) = g(\sum_j W_{ij}V_j(t) + I_i(t))
\]

For the hopfield network, the Lyapunov energy function will have the following form:

\[
E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij}V_iV_j - \sum_{i=1}^{n} V_iI_i
\]

It has been proved that hopfield network is a non-linear motion system. There exists one or more minimum points or balancing points. At some time, after the status of every neural unit is given and the initial network status is set up, the status of the network will change in the direction of energy gradually decreasing in the light of the working equation, and in the end, approach to or reach the balanced status of the network, which is the minimum point of the energy. This is convergence of the energy of the hopfield neural network. In this way, while the energy function converges to a minimum point, the best solution for the problem will be obtained.

The problem of finding the best label position for residence feature can be regarded as a combinatorial optimal problem. If a hypothesized map contains \( m \) residences, and every residence has \( n \) candidate points (for example, \( n = 8, 16, \ldots \)), then there are \( m \times n \) candidate points, and these residences will be listed as a matrix in which \( n \) candidate label positions of one residence will be lined as one row, altogether there are \( m \) row \( \times n \) column, as shown in Table 1. If we match up one candidate label position to one neural unit, thus we can construct a hopfield neural network that comprises \( m \times n \) neural units.

| Candidate label position for residence |
|---------------------------------------|
| Resident point | 1# | 2# | 3# | 4# | 5# | 6# | 7# | 8# |
|----------------|----|----|----|----|----|----|----|----|
| 1#             | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 2#             | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |
| 3#             | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  |
| ...            | ...| ...| ...| ...| ...| ...| ...| ...|
| 2 514#         | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 2 515#         | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  |

In order to define the energy function, we describe the problem as the constraint condition plus the optimal goals as follows.

1. Constraint condition: every residence can choose only one label position.

2. The most optimal goal: the number of overlapping between two label rectangle area is the smallest.

According to the above constraint condition and the most optimal goal, we can give out the energy function of the network as follows.

\[
E = \frac{B}{2} \sum_i \left( \sum_j V_j \right)^2 + \frac{A}{2} \sum_i \sum_j \sum_k \sum_l D(i,j,k,l)V_iV_l
\]

Of the above equation, the first item represents the constraint condition, and only one la-
bel position is allowed for one resident feature. When this condition is met, the first item will be equal to 0. The second item is optimal goal, \( D(i,j,k,l) \) is specified as:

\[
D(i,j,k,l) =
\begin{cases} 
1, & \text{when } V_o, V_w \text{ overlap with each other} \\
0, & \text{when } V_o, V_w \text{ do not overlap with each other}
\end{cases}
\]

Thus the second item is the multiplied number of the overlap between every two label rectangles. If the choice of label position is the most optimal, \( E_2 \) can reach the minimum value; and if the choice of label position is more optimal, \( E_2 \) can reach the smaller value.

By comparing the energy function with the standard energy function, the connecting weight between unit \( i \) and unit \( j \) can be determined as

\[
T_{o,w} = -AD(i,j,k,l) - B\delta_{o,w}
\]

(1)

\[
\delta_{o,w} =
\begin{cases} 
1, & \text{when } ij = kl \\
0, & \text{when } ij \neq kl
\end{cases}
\]

(2)

Because \( W_{o,w} \propto T_{o,w} \), let \( W_{o,w} = T_{o,w} \), then the running equation will be written as

\[
du_o \over dt = -u_o + \sum (AD(i,j,k,l) - B\delta_{o,w})V_o + B
\]

(3)

\[
V_o = g(u_o)
\]

(4)

Here we adopt sigmoid function as the I/O function of the neural unit, and \( ij \) is the subscript which stands for the neural unit of \( j \)th label position of the \( i \)th resident.

The detailed calculation procedure and the iterative steps are listed below.

1. The initial value is set up in the light of the above guideline;
2. Calculate the output of every neural unit \( V_o(t_o) \) according to \( V_o = g(u_o) \);
3. Put \( V_o(t_o) \) into Eq. (3), calculate \( du_o \over dt \) \( t = t_o \);
4. According to equation \( u_o(t_o + \Delta t) = u_o(t_o) + {du_o \over dt} \bigg|_{t = t_o} \Delta t \), calculate \( u_o(t_o + \Delta t) \) of the next time point.
5. Return to step 2.

5 Experimental results and conclusions

The experiment of running the equation shows that while an appropriate parameter and an appropriate constant of the time \( \tau \), are specified, the network will converge normally to a satisfactory end status.

The original spatial data can not be directly used in our algorithm. These map data must take preprocessing before being available. The preprocessing operations consist of the symbolization of map feature, the conversion of vector feature to raster format, the coding of feature and the overlay of the raster maps. Besides, raster spatial information will be written into different binary files in terms of their feature type. Meanwhile, the attribute information of the name will be read out of the source file and be written into a binary file to be easily used.

The experimental results show that the neural network method can be used to solve the combinatorial optimal problem including finding the best label position. Because the network can converge swiftly, therefore, the complex problem with "combinatorial explosion" danger can be converted to a simple problem, which will largely improve the efficiency of algorithm.

The problems of name placement differ significantly among the three categories. The approach to the placement of names for points is relatively easier than linear and areal features. In the future, we will do further research to make our algorithm more powerful to solve more problems and suit more cases.

REFERENCES

1. Hiersch S A (1982) An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5-17
2. Zoraster S (1986) Interger programming applied to the map label placement Problem. *Cartographica*, 23(3):16-27
3. Mower J E (1993) Automated feature and name placement on parallel computers. *Cartography and Geographic Information Systems*, 20(2):9-82
4. Zoraster S (1991) Expert systems and the map label placement problem. *Cartographica*, 28(1):1-9