

A Structural Matching Algorithm Using Generalized Deterministic Annealing

Laura Davlea

Institute for Research in Electronics Lascar Catargi 22 Iasi 6600 , Romania
E-mail: ldavlea@mail.dntis.ro

Abstract: We present a practical implementation of a structural matching algorithm that uses the generalized deterministic annealing theory. The problem is formulated as follows: given a set of model points and object points, find a matching algorithm that brings the two sets of points into correspondence. An “energy” term represents the distance between the two sets of points. This energy has many local minima and the purpose is to escape from these local minima and to find the global minimum using the simulated annealing theory. We use a windowed implementation and a suitable definition of the energy function that reduces the computational effort of this annealing schedule without decreasing the solution quality.

Keywords: generalized deterministic annealing, structural matching, unary and binary features, graph matching, local Markov chain.

1 Theoretical Framework

The problem of finding a correspondence between object and model features is an important part of object recognition. It can be formulated as one of graph matching and there are many approaches to solve it such as: probabilistic relaxation, neural networks.

The neural networks methods are not restricted to looking for right isomorphism. They use a “energy”, or distance measure between graphs, that is minimized by finding a correct correspondence between vertex labels of model graph with respect to object graph.

Our algorithm applies the generalized deterministic annealing framework, formulated in [1], to the problem of structural matching between 2D model and object features. The problem is to find a correspondence between points of interest (high curvature or zero crossing points). If we have N object points and K model points, the solution space has cardinality $|K^N|$. It is a complicated combinatorial optimization problem. These problems have often multiple conflicting constraints, and have typically numerous local minima in the energy function.

The method of simulated annealing allows the solution to escape from local minima. Thermal noise is added to neural network by simulating high temperatures, then temperature is slowly reduced so that the amount of thermal noise decreases.

Generalized deterministic annealing utilizes N local Markov chains of K states. These are K states neurons, representing the state probability density of local Markov

chains. We have N variables that have K possible states. For each variable , we assigna K-state neuron. We define a matrix D(i,j,T) whose lines computes the distribution for each local Markov chain. We note each line of the matrix with Dn,

n=0.. N-1.. Generalized deterministic annealing (GDA), iteratively computes the distribution of each local Markov chain. For each temperature , after a number of iterations, these distributions converge to a state of thermal equilibrium (stationary distribution), characterized by the uniform convergence to within a ε- ball. As the temperature is lowered, the distributions become singular, all of the N local Markov chains are frozen, and the solution is complete. In simulated annealing (SA) theory , the probability of a transition from solution i to solution j is computed by :

$$P(i,j,T)=G(i,j,T)A(i,j,T) \quad i \neq j$$

$$P(i,i,T) = 1 - \sum_{l \in N(i)} P(i,l,T) \tag{1}$$

N(i) is the solution neighborhood, G(i,j) is the generation function and A(i,j,T) is the acceptance function . We can assume a uniform generation function

$$G(i,j) = \frac{1}{|N(i)|}, \quad |N(i)| \text{ is the cardinality of the solution neighborhood. In our case}$$

G(I,j)=1/K. The acceptance function can be sigmoidal :

$$A(i,j,T) = \frac{1}{1 + \exp(E(j) - E(i)/T)}, \tag{2}$$

E(i) is the energy with respect to solution i. The distribution D_n^t of each local Markov chain is updated using the formula

$$D_n^{t+1} = D_n^t P_n(T) \tag{3}$$

The matrix P_n is constructed from the SA transition probabilities defined by (1). The acceptance function is defined as:

$$A_n = \frac{1}{1 + \exp((E_n(j) - E_n(i))/T)} \tag{4}$$

for neuron n, and temperature T.

E_n(i) is the local energy function with respect to neuron n in the state i, I=0..(K-1)

Using the expression (1) ,(2) and (3), it can be demonstrated [1] that the distributions can be updated using the formula:

$$D_n^{t+1}(j) = \frac{1}{K} \sum_0^{K-1} A_n(i,j,T)(D_n^t(i) + D_n^t(j)) \tag{5}$$

The local energy function depend on the state of the neighboring neurons. We estimate the state of the j neuron as:

$$j_n = \{j \mid D_n^t(j) > D_n^t(i), i = 0..K - 1\}$$

In GDA the convergence of the neurons is demonstrated and the final value of the temperature , where the elements of $D_n^t(j)$ have converged to zero or one is deterministic. The final temperature is:

$$T_f \leq \frac{\Delta E_{\min}}{\ln \frac{K - K\varepsilon - 1}{K\varepsilon}} \tag{6}$$

ΔE_{\min} is the minimum change on the local energy associated with one update, ε is the radius of the ball to within D_n^t converges. The starting temperature is defined for the condition that all solutions have equal probability, and the next update remain close to initial solution. Using the expression

$A_{\max} = [1 + \exp(-\Delta E_{\max}/T)]^{-1}$, it can be demonstrated that the starting temperature for the annealing schedule must satisfy the condition:

$$T_0 > \frac{\Delta E_{\max}}{\ln \frac{K - 1 + K^2\varepsilon}{K - 1 - K^2\varepsilon}} \tag{7}$$

As a conclusion ,for high temperatures GDA can “climb” and escape local minima , and for low temperature , GDA settles into the best locally minima.

2 Implementation

Our object model uses points of high curvature, or zero curvature crossing, as points of interest to be matched. The points are identified using three unary features and two binary features. The unary features are:

- U1 curvature of interesting points
- U2 summation of lengths adjacent to points
- U3 the absolute value of the variation of curvature on the segments adjacent to interesting points.

The binary features are:

- B1 length of the line connecting two points
- B2 length along the contour between two interesting points.

The energy function encodes the compatibility between model and object features. For the curvature we have used the following formula :

$$k(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{(x(u)^2 + y(u)^2)^{3/2}} \tag{8}$$

u is the length parameter along the contour.

The local energy function is defined by the formula:

$$E(i, j) = \sum_{k2=0}^1 \sum_{k1=0}^{KN1} WB(k2) f(Bk2obj(j, k1) - Bk2mod(i, k1)) + \sum_{k3=0}^2 WU(k3) f(Uk3obj(i) - Uk3mod(j)) \quad (9)$$

$E(i, j)$ is the local energy of point i that has the label j .

$K2=0,1$ addresses the binary features

$K3=0,1,2$ addresses the unary features

$WBk2, WUk3$ are weights representing the strength of each attribute.

f is the compatibility function and is expressed as:

$f = \tanh(c|u|)$, c is a constant.

$$\sum_{k2=0}^1 WB(k2) + \sum_{k3=0}^2 WU(k3) = 1 \quad (10)$$

The weight coefficients sum is equal to 1, and we take all these weights=0.2

$Bk2$ and $Uk3$ are object and model unary and binary features.

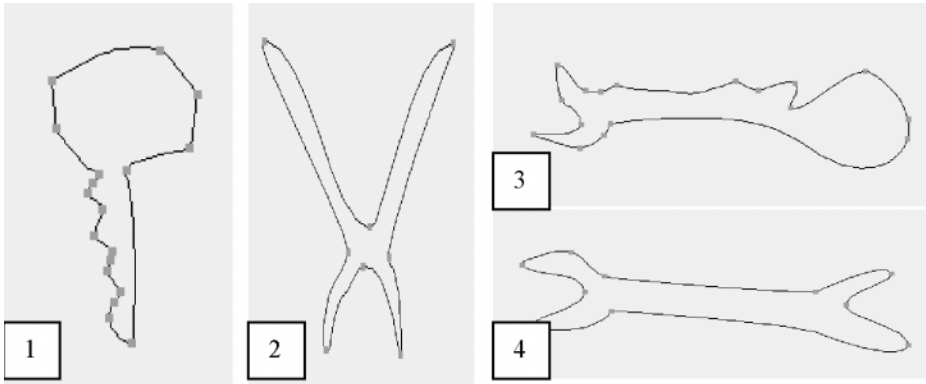
The purpose of the algorithm is to find the best solution without searching the entire solution set, and has the following steps:

- 1) Set the temperature $T=T_0$ and $Dn(i, j, T)=1/K$
- 2) Use (4) to update $Dn(i, j, T)$
- 3) If $|Dn^{t+1}(i, j, T) - Dn^t(i, j, T)| < \epsilon$ for all i, j , then set $T=0.9T$, else return to 2
- 4) If all $Dn^t(i, j, T)$ have converged to 0 or 1 then all the neuron have reached saturation and the annealing schedule stops, else return to 2.

The local neighborhood of a point in the formula of the energy function is defined by taking into consideration only a few points around the interesting point in the formula of the local energy. $KN1$ represent the number of these points, and we have taken for our models $KN1=7$. This local structure reduces the execution time and has a good behaviour to partial object occlusion.

A windowed implementation can reduce the wasteful execution time. For every object point we consider a window of model points that have the same sign of curvature. Taking into account a standard deviation of 10% of corruptive noise for our experiments, we have used a few points window size ($Kw=10$).

Another approach for time reduction can be a suitable choice of ϵ . The expression (7) shows that when $\epsilon \rightarrow 0$, T_0 is very large and the annealing schedule is very long. We have tested this algorithm using object models like fig. 1-4.



Some experimental results are presented in the table 1.

Table 1

Object model	Nr. Of iterations	ϵ	Windowed implementation time reduction
1	137	0.01	45%
2	162	0.01	56%
3	141	0.01	61%
4	152	0.01	52%

3 Conclusion

The algorithm utilizes the major benefit of the simulated annealing algorithms , that is the possibility to escape from local minima of the energy function. A windowed implementation , a suitable choice of decreasing temperature and a suitable expression for the energy function can provide an important reduction time in the case of applications with a large number of variable

References

- 1.S. Acton. A. Bovik, "Generalized deterministic annealing" . IEEE Transactions on Neural Networks May 1996
- 2.S. Gold, "A graduate assignment algorithm for graph matching", IEEE Transactions on Pattern Anal. And Machine Intell. April 1996
- 3.A. Rangarajan, "A Lagrangian Relaxation Network for Graph Matching, IEEE Transactions on Neural Networks, Nov 1996