Improved Simulated Annealing, Boltzmann Machine, and Attributed Graph Matching

Ran Chen <ranc@cs.cmu.edu>
09/30/2013
Outline

• Simulated Annealing (SA)
• Improved Simulated Annealing (ISA)
• ISA on Boltzmann Machine (BM)
• Attributed Graph Matching by ISA and Improved BM
• Experiment and Result
Simulated Annealing (SA)

Simulated annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space.

E: Energy
T: Temperature
S: state

Find it!
Simulated Annealing (SA)

Initialization: Generate a random state $s$ as the present solution, and initialize $E(s)$ and $T = T^{(0)}$;

step 1: Randomly make a small perturbation $\Delta s$ to get a new state $s + \Delta s$ with the energy increment $\Delta E = E(s + \Delta s) - E(s)$;

step 2: If $\Delta E < 0$ goto step 3, otherwise, generate a random number $\xi$ by sampling a uniform distribution over $[0,1]$ and if $e^{-\Delta E/T} \leq \xi$, goto step 1;

step 3: $s + \Delta s$ replaces $s$ as the new present solution, and $E := E + \Delta E$;

step 4: Check whether the MS at the present $T$ reached its equilibrium; if not, goto step 1;

step 5: Reduce $T$ into $T' < T$ by some means (e.g., $T := \lambda T$). Check whether the annealing process has terminated (e.g., $T < T_{\text{min}}$); if yes, the present $s$ with its $E(s)$ is taken as the final solution, stop; otherwise, goto step 1.

Problem:
1. How to know whether reach its equilibrium?
2. How to decide $T$?
Improved Simulated Annealing (ISA)

Key Idea:
1. Store the minimum Energy State(s) so far
2. Stay in one minimum Energy State(s) too long, go to next T.
3. Stay one minimum Energy State(s) too long for lots of T, it is done.
Improved Simulated Annealing (ISA)

**S\(_{i,j}\):** the present solution of the j-th iteration at T\(_i\).

**S\(_{i,j}'\):** the best solution so far.

- At a given T\(_i\), if in q successive state s\(_{i,j}'\)=s\(_{i,j+1}'\)=...=s\(_{i,j+q}'\) holds, and q is large enough, then it is equilibrium.
- If in p successive temperatures s\(_{i,ki}'\)=s\(_{i+1,ki+1}'\)=...=s\(_{i+p,ki+p}'\) holds and p is large enough, it is equilibrium.

**Initialization:** Generate a random state s as the present solution, and initialize E(s) and T = T\(_{\text{init}}\); set T\(_{\text{min}}\), j\(_{\text{max}}\), q\(_0\), p\(_0\); Let p = 0, q = 0, j = 0, \(\hat{s} = s\), E = E(s), E\(_{\hat{s}}\) = E, E\(_t\) = E;

step 1: If j > j\(_{\text{max}}\), goto step 6; otherwise, j := j + 1, randomly make a small perturbation \(\Delta s\) to get a new state s + \(\Delta s\) with the energy increment \(\Delta E = E(s + \Delta s) - E(s)\);

step 2: If \(\Delta E > 0\), generate a random number \(\xi\) by sampling a uniform distribution over [0,1]. If \(e^{-\Delta E/T} \leq \xi\), goto step 1;

step 3: s + \(\Delta s\) replaces s as the new present solution, and E := E + \(\Delta E\);

step 4: If \(E < E_{\hat{s}}\) then \(E_{\hat{s}} := E\) and \(\hat{s} := s + \Delta s\), q := 0; Otherwise q := q + 1;

step 5: If q < q\(_0\), goto step 1;

step 6: If E\(_t\) < E\(_{\hat{s}}\) then E\(_t\) := E\(_{\hat{s}}\) and p := 0; Otherwise p := p + 1;

step 7: If p < p\(_0\) and T > T\(_{\text{min}}\), reduce T into T' < T by some means(e.g., T := \(\lambda T\)), q := 0 and k := 0 and goto step 1; Otherwise, the present \(\hat{s}\) with its E(\(\hat{s}\)) is taken as the final solution, stop.
ISA on Boltzmann Machine (BM)

Energy Expression:

\[ E = -0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} c_i c_j + \sum_{i=1}^{n} \theta_i c_i \]

A neuron \( c_i \): 0=>1

\[ \Delta E_i = \sum_{i=1, j \neq i}^{n} w_{ij} c_j - \theta_i \]

Let the neuron take the value 1 with probability

\[ p_i = 1/(1 + \exp(-\Delta E_i/T)) \].
ISA on Boltzmann Machine (BM)

Binary array A: Current global state
Binary array B: Global state of the current minimal energy

B = A when E(A) < E (B)
Attributed Graphs have shown superior adequacy when used for image representation and understanding in computer vision. They have also been successfully used to e.g. to handle optimal task assignment in distributed computer systems.

Graph

\[ G = [(V, V_a), (E, E_a)] \]

Node and node attributes

\[ V = (v_1, v_2, ..., v_n) \]
\[ V_a = (a v_1, a v_2, ..., a v_n) \]

Edge and edge attributes

\[ E = (e_1, e_2, ..., e_p) \]
\[ E_a = (ae_1, ae_2, ..., ae_p) \]
Attributed Graph Matching

Find one-to-one correspondences between nodes of $G$ and $G'$

Use SA: Define $E$, $\Delta E$, $S$, $T$?
Attributed Graph Matching by ISA

\( U_{ij} \): binary \( n \times m \), row \( i \) for node \( v_i \) of \( G \) and column \( j \) for node \( v'_j \) of \( G' \).

(i) For \( n = m \)

\[
E = a \sum_{i=1}^{n} \left( \sum_{j=1}^{n} U_{ij} - 1 \right)^2 + a \sum_{j=1}^{n} \left( \sum_{i=1}^{n} U_{ij} - 1 \right)^2 \\
+ b \sum_{i=1}^{n} \sum_{j=1}^{n} (a v_i - a v'_j)^2 U_{ij} + c \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{n} \sum_{k=1}^{n} (a e_{ij} - a e'_{lk})^2 U_{ij} U_{lk}
\]

(ii) For \( n < m \)

\[
E = a \sum_{i=1}^{n} \left( \sum_{j=1}^{m} U_{ij} - 1 \right)^2 + a \left( \sum_{j=1}^{m} \sum_{i=1}^{n} U_{ij} - n \right)^2 \\
+ b \sum_{i=1}^{n} \sum_{j=1}^{m} (a v_i - a v'_j)^2 U_{ij} + c \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{n} \sum_{k=1}^{m} (a e_{ij} - a e'_{lk})^2 U_{ij} U_{lk}
\]
Attributed Graph Matching by ISA

\( \Delta E \): randomly inverse an element \( U_{ij} \) into \( U_{ij} := 1 - U_{ij} \)

(i) For \( n = m \)

\[
\Delta E(U_{ij}, 0 \to 1) = 2a \sum_{k=1, k \neq j}^{n} U_{ik} + 2a \sum_{k=1, k \neq i}^{n} U_{kj} - 2a
\]

\[+ b(av_i - av_j')^2 + 2c \sum_{r=1, r \neq i}^{n} \sum_{q=1, q \neq j}^{m} (ae_{ri} - ae_{qj}')U_{rq} \tag{6a}\]

and \( \Delta E(U_{ij}, 1 \to 0) = -\Delta E(U_{ij}, 0 \to 1) \)

(ii) For \( n < m \)

\[
\Delta E(U_{ij}, 0 \to 1) = 2a \sum_{k=1, k \neq j}^{n} U_{ik} + 2a \sum_{r=1, r \neq i}^{n} \sum_{q=1, q \neq j}^{m} U_{rq} - 2aN
\]

\[+ b(av_i - av_j')^2 + 2c \sum_{r=1, r \neq i}^{n} \sum_{q=1, q \neq j}^{m} (ae_{ri} - ae_{qj}')U_{rq} \tag{6b}\]

and \( \Delta E(U_{ij}, 1 \to 0) = -\Delta E(U_{ij}, 0 \to 1) \)
Algorithm AGM-ISA

Initialization: Set an \( N \times M \) Matrix \( [U_{ij}] \) by randomly deciding each of its elements to be 1 or 0, and let another \( N \times M \) Matrix \( [U'_{ij}] \) take the same values as \( [U_{ij}] \); Initialize \( T^{(0)}, T_{\min}, k_{\max}, q_{\max}, p_{\max} \); Let \( p = 0, q = 0, k = 0, \hat{s} = s, E = E(s), E_{\hat{s}} = E, E_t = E \);

\begin{align*}
\text{step 1 :} & \quad k := k + 1; \text{ Choose with equal probabilities an integer } i \text{ among } [1,2,\ldots,N] \text{ and an integer } j \text{ among } [1,2,\ldots,M]; \text{ Compute } \Delta E \text{ by Eq.}(6); \\
\text{step 2 :} & \quad \text{If } \Delta E < 0 \text{ goto step 4, otherwise, generate a random number } \xi \text{ by sampling a uniform distribution over } [0,1]; \text{ If } \exp[-\Delta E/T] > \xi \text{ goto step 4}; \\
\text{step 3 :} & \quad \text{If } k > k_{\max}, \text{ goto step 7, otherwise goto step 1}; \\
\text{step 4 :} & \quad U_{ij} := 1 - U_{ij} \text{ and } E := E + \Delta E; \\
\text{step 5 :} & \quad \text{If } E < E_{\hat{s}} \text{ then } E_{\hat{s}} := E \text{ and } [U'_{ij}] := [U_{ij}] \text{ and } q := 0; \text{ Otherwise } q := q + 1; \\
\text{step 6 :} & \quad \text{If } q < q_{\max}, \text{ goto step 1}; \\
\text{step 7 :} & \quad \text{If } E_t < E_{\hat{s}} \text{ then } E_t := E_{\hat{s}} \text{ and } p := 0; \text{ Otherwise } p := p + 1; \\
\text{step 8 :} & \quad \text{If } p < p_{\max} \text{ and } T > T_{\min} \text{ then } T := \lambda T \quad (0 < \lambda < 1) \text{and } q := 0 \text{ and } k := 0 \text{ and goto step 1}; \text{ Otherwise, the present } [U'_{ij}] \text{ is taken as the final solution, and use Eq.}(5) \text{ to calculate its } E \text{ and stop.} \end{align*}
Attributed Graph Matching by BM

(i) For $n = m$

\[
E = -0.5 \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{l=1}^{n} \sum_{k=1}^{n} w_{ij,lk} U_{ij} U_{lk} - \sum_{i=1}^{n} \sum_{j=1}^{n} U_{ij} \theta_{ij},
\]

\[
w_{ij,lk} = -2a(\delta_{il} + \delta_{jk}) - 2c(1 - \delta_{il})(1 - \delta_{jk})(a e_{ij} - a e'_{lk})^2,
\]

\[
\theta_{ii} = -4a + b(a v_i - a v'_j)^2
\]

where $\delta_{ij}$ is the Kronecker delta;

(ii) For $n < m$

\[
E = -0.5 \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{n} \sum_{k=1}^{m} w_{ij,lk} U_{ij} U_{lk} - \sum_{i=1}^{n} \sum_{j=1}^{m} U_{ij} \theta_{ij},
\]

\[
w_{ij,lk} = -2a(\delta_{il} + 1) - 2c(1 - \delta_{il})(1 - \delta_{jk})(a e_{ij} - a e'_{lk})^2,
\]

\[
\theta_{ij} = -2a(n + 1) + b(a v_i - a v'_j)^2.
\]

$U_{ij}, U_{lk} = \text{neurons } s_i, s_j$

$w_{ij,lk} = \text{connections } w_{ij}$

$\theta_{ij} = \theta_i$
# Experiment and Result

<table>
<thead>
<tr>
<th></th>
<th>$T$</th>
<th>$T_{\text{min}}$</th>
<th>$k_{\text{max}}$</th>
<th>$\lambda$</th>
<th>iteration</th>
<th>final $E$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SA</strong></td>
<td>2000</td>
<td>500</td>
<td>500</td>
<td>0.95</td>
<td>14028</td>
<td>3010.25</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>10</td>
<td>500</td>
<td>0.95</td>
<td>52105</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>1</td>
<td>500</td>
<td>0.95</td>
<td>74649</td>
<td>1025.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$T$</th>
<th>$T_{\text{min}}$</th>
<th>$k_{\text{max}}$</th>
<th>$\lambda$</th>
<th>$q$</th>
<th>$p$</th>
<th>$i$</th>
<th>final $E$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ISA</strong></td>
<td>2000</td>
<td>500</td>
<td>500</td>
<td>0.95</td>
<td>infinite</td>
<td>infinite</td>
<td>14028</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>500</td>
<td>500</td>
<td>0.95</td>
<td>250</td>
<td>15</td>
<td>7384</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Thanks for Listening!
Any Question?