Probabilistic Graphical Models and Evolutionary Computation

Pedro Larrañaga

Computational Intelligence Group
Artificial Intelligence Department
Technical University of Madrid

2010 IEEE World Congress on Computational Intelligence
Barcelona, July 21, 2010
Outline

1. Estimation of Distribution Algorithms
2. Evolutionary Computation for Probabilistic Graphical Models
4. Final Remark
Evolutionary Computation

Techniques

- **Pioneer work** on computer simulation of evolution: Barriceli (1954), Fraser (1957)
- **Evolutionary programming** (Fogel, 1964)
- **Evolution strategies** (Rechenberg, 1973 and Schwefel, 1974)
- **Genetic algorithms** (Holland, 1975)
- **Ant colony optimization** (Dorigo, 1992)
- **Cultural algorithms** (Reynolds, 1994)
- **Particle swarm optimization** (Kennedy and Eberhart, 1995)
- **Estimation of distribution algorithms** (H. Mühlenbein and Paαβ, 1996)
- And more: differential evolution, harmony search, artificial immune systems...
Evolutionary Computation

Techniques

- **Pioneer work** on computer simulation of evolution: Barriceli (1954), Fraser (1957)
- **Evolutionary programming** (Fogel, 1964)
- **Evolution strategies** (Rechenberg, 1973 and Schwefel, 1974)
- **Genetic algorithms** (Holland, 1975)
- **Ant colony optimization** (Dorigo, 1992)
- **Cultural algorithms** (Reynolds, 1994)
- **Particle swarm optimization** (Kennedy and Eberhart, 1995)
- **Estimation of distribution algorithms** (H. Mühlenbein and Paap, 1996)
- And more: differential evolution, harmony search, artificial immune systems ...
EDAs. A Toy Example

\[ \max O(x) = \sum_{i=1}^{6} x_i \]

with \( x_i = 0, 1 \)
EDAs. A Toy Example

\[ \max O(x) = \sum_{i=1}^{6} x_i \]

with \( x_i = 0, 1 \)

\[
\begin{array}{cccccccc}
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & O(x) \\
1 & 1 & 0 & 1 & 0 & 1 & 0 & 3 \\
2 & 0 & 1 & 0 & 0 & 1 & 0 & 2 \\
3 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
4 & 1 & 1 & 1 & 0 & 0 & 1 & 4 \\
5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
6 & 1 & 1 & 0 & 0 & 1 & 1 & 4 \\
7 & 0 & 1 & 1 & 1 & 1 & 1 & 5 \\
8 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
9 & 1 & 1 & 0 & 1 & 0 & 0 & 3 \\
10 & 1 & 0 & 1 & 0 & 0 & 0 & 2 \\
11 & 1 & 0 & 0 & 1 & 1 & 1 & 4 \\
12 & 1 & 1 & 0 & 0 & 0 & 1 & 3 \\
13 & 1 & 0 & 1 & 0 & 0 & 0 & 2 \\
14 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \\
15 & 0 & 1 & 1 & 1 & 1 & 1 & 5 \\
16 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
17 & 1 & 1 & 1 & 1 & 1 & 0 & 5 \\
18 & 0 & 1 & 0 & 1 & 1 & 0 & 3 \\
19 & 1 & 0 & 1 & 1 & 1 & 1 & 5 \\
20 & 1 & 0 & 1 & 1 & 0 & 0 & 3 \\
\end{array}
\]
EDAs. A Toy Example

\[
\text{max } O(x) = \sum_{i=1}^{6} x_i
\]

with \( x_i = 0, 1 \)

<table>
<thead>
<tr>
<th></th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( X_3 )</th>
<th>( X_4 )</th>
<th>( X_5 )</th>
<th>( X_6 )</th>
<th>( O(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
EDA. A Toy Example

Learning the probability distribution from the selected individuals

<table>
<thead>
<tr>
<th></th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>X₄</th>
<th>X₅</th>
<th>X₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Learning the probability distribution from the selected individuals

\[
p(x) = p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5)p(x_6)
\]
EDAs. A Toy Example

Learning the probability distribution from the selected individuals

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$X_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$p(x) = p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5)p(x_6)$

$p(X_1 = 1) = \frac{7}{10}$
Learning the probability distribution from the selected individuals

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$X_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$p(x) = p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5)p(x_6)$

$p(X_1 = 1) = \frac{7}{10}$ \hspace{1cm} $p(X_2 = 1) = \frac{7}{10}$ \hspace{1cm} $p(X_3 = 1) = \frac{6}{10}$

$p(X_4 = 1) = \frac{6}{10}$ \hspace{1cm} $p(X_5 = 1) = \frac{8}{10}$ \hspace{1cm} $p(X_6 = 1) = \frac{7}{10}$
EDAs. A Toy Example

Learning the probability distribution from the selected individuals

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$X_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$p(x) = p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5)p(x_6)$

$p(X_1 = 1) = \frac{7}{10}$

$p(X_2 = 1) = \frac{7}{10}$

$p(X_3 = 1) = \frac{6}{10}$

$p(X_4 = 1) = \frac{6}{10}$

$p(X_5 = 1) = \frac{8}{10}$

$p(X_6 = 1) = \frac{7}{10}$
EDAs. A Toy Example

Obtaining the new population by sampling from the probability distribution

\[ p(X_1 = 1) = \frac{7}{10}; \quad p(X_2 = 1) = \frac{7}{10}; \quad p(X_3 = 1) = \frac{6}{10} \]

\[ p(X_4 = 1) = \frac{6}{10}; \quad p(X_5 = 1) = \frac{8}{10}; \quad p(X_6 = 1) = \frac{7}{10} \]

\[ p(\mathbf{x}) = p(x_1, \ldots, x_6) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5)p(x_6) \]

<table>
<thead>
<tr>
<th>(X_i)</th>
<th>Probability</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_1 = 1)</td>
<td>(\frac{7}{10})</td>
<td>(&gt; 0.23 \rightarrow 1)</td>
</tr>
<tr>
<td>(X_2 = 1)</td>
<td>(\frac{7}{10})</td>
<td>(&gt; 0.65 \rightarrow 1)</td>
</tr>
<tr>
<td>(X_3 = 1)</td>
<td>(\frac{6}{10})</td>
<td>(&lt; 0.89 \rightarrow 0)</td>
</tr>
<tr>
<td>(X_4 = 1)</td>
<td>(\frac{6}{10})</td>
<td>(&gt; 0.12 \rightarrow 1)</td>
</tr>
<tr>
<td>(X_5 = 1)</td>
<td>(\frac{8}{10})</td>
<td>(&gt; 0.48 \rightarrow 1)</td>
</tr>
<tr>
<td>(X_6 = 1)</td>
<td>(\frac{7}{10})</td>
<td>(&gt; 0.54 \rightarrow 1)</td>
</tr>
</tbody>
</table>
EDAs. A Toy Example

Obtaining the new population by sampling from the probability distribution

<table>
<thead>
<tr>
<th></th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
<th>$X_6$</th>
<th>$O(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
Graphical Representation of EDAs

- Initial population of candidate solutions
- Selected candidates
- Bayesian or Gaussian network learning
- Sampling of new candidate solutions
Graphical Representation of EDAs
Probabilistic Models in EDAs

- Probabilistic model: \( p_l(x) = \prod_{i=1}^{n} p_l(x_i) \)
- Structural learning: not necessary

Bivariate EDAs: Mutual Information Maximization for Input Clustering (MIMIC). De Bonet et al. (1997)
- Probabilistic model: \( p_{l^*}(x) = p_l(x_{i_1} \mid x_{i_2}) p_l(x_{i_2} \mid x_{i_3}) \cdots p_l(x_{i_{n-1}} \mid x_{i_n}) p_l(x_{i_n}) \)
- Structural learning: best permutation (factorization closest to the empirical distribution in the sense of Kullback-Leibler divergence)

Multivariate EDAs: Etxeberria and Larrañaga (1999) (EBNA); Pelikan et al. (1999) (BOA); Harik et al. (1999) (EcGA); Mühlenbein and Mahnig (1999) (LFDA)
- Probabilistic model: \( p_l(x) = \prod_{i=1}^{n} p_l(x_i \mid pa_i) \)
- Structural learning: directed acyclic graph

EDAs in continuous domains: Assuming Gaussianity
- Univariate: Larrañaga et al. (2000) (UMDA\(_G\))
- Bivariate: Larrañaga et al. (2000) (MIMIC\(_G\))
- Multivariate: Larrañaga et al. (2000) (EMNA\(_G_{global}\), EMNA\(_G_{ee}\), EGNA\(_G\))
Probabilistic Models in EDAs

- **Probabilistic model:** \( p_{i}(x) = \prod_{i=1}^{n} p_{i}(x_{i}) \)
- **Structural learning:** not necessary

**Bivariate EDAs: Mutual Information Maximization for Input Clustering (MIMIC). De Bonet et al. (1997)**
- **Probabilistic model:**
  \[
p_{i}^{\pi}(x) = p_{i}(x_{i1} \mid x_{i2})p_{i}(x_{i2} \mid x_{i3}) \cdots p_{i}(x_{in-1} \mid x_{in})p_{i}(x_{in})
\]
- **Structural learning:** best permutation (factorization closest to the empirical distribution in the sense of Kullback-Leibler divergence)

**Multivariate EDAs: Etxeberria and Larrañaga (1999) (EBNA); Pelikan et al. (1999) (BOA); Harik et al. (1999) (EcGA); Mühlenbein and Mahnig (1999) (LFDA)**
- **Probabilistic model:**
  \[
p_{i}(x) = \prod_{i=1}^{n} p_{i}(x_{i} \mid pa_{i})
\]
- **Structural learning:** directed acyclic graph

**EDAs in continuous domains: Assuming Gaussianity**
- **Univariate:** Larrañaga et al. (2000) (UMDA\(_{c}^{G}\))
- **Bivariate:** Larrañaga et al. (2000) (MIMIC\(_{c}^{G}\))
- **Multivariate:** Larrañaga et al. (2000) (\(EMNA_{global}^{G}\), \(EMNA_{ee}^{G}\), \(EGNA_{G}\))
Graphical Representation of EDAs

- Initial population of candidate solutions
- Selected candidates
- Bayesian or Gaussian network learning
- Sampling of new candidate solutions

P. Larrañaga
Probabilistic Graphical Models & Evolutionary Computation
Obtaining the new population by sampling with PLS (Henrion, 1988)

Given an ancestral ordering, $\pi$, of the nodes in the directed probabilistic graphical model (Bayesian network or Gaussian Bayesian network):

$$
\text{for } j = 1, 2, \ldots, M \\
\text{for } i = 1, 2, \ldots, n \\
\quad x_{\pi(i)} \leftarrow \text{generate a value from } p(x_{\pi(i)} | \text{pa}_{\pi(i)})
$$
EDAs

Books

- Estimation of Distribution Algorithms
  (2001)

- Hierarchical Bayesian Optimization Algorithm
  (2005)

- Towards a New Evolutionary Computation
  (2006)

Special issues

- International Journal of Approximate Reasoning
  (2002)

- Evolutionary Computation
  (2005)

- IEEE Transactions on Evolutionary Computation
  (2009)

P. Larrañaga

Probabilistic Graphical Models & Evolutionary Computation
Methodological papers

Theoretical papers

EDAs

Application papers

Objectives of the Talk

Double direction synergy

- Genetic Algorithms
- Estimation of Distribution Algorithms
- Evolutionary Programming
- Ant Colony Optimization
- ...

- Bayesian Networks
- Gaussian Bayesian Networks
- Conditional Gaussian Bayesian Networks
- Markov Networks
- ...

Probabilistic Graphical Models & Evolutionary Computation
Objectives of the Talk

Double direction synergy

Evolutionary Computation
- Genetic Algorithms
- Estimation of Distribution Algorithms
- Evolutionary Programming
- Ant Colony Optimization
- ...

Probabilistic Graphical Models
- Bayesian Networks
- Gaussian Bayesian Networks
- Conditional Gaussian Bayesian Networks
- Markov Networks
- ...

Probabilistic Graphical Models & Evolutionary Computation
P. Larrañaga
Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks
   - Learning Gaussian Bayesian Networks from Data
   - Learning Conditional Gaussian Bayesian Networks from Data


4. Final Remark
Directed Probabilistic Graphical Models

Qualitative + quantitative parts

A directed probabilistic graphical model, \( M = (S, \theta^S) \), (Pearl, 1988; Koller and Friedman, 2009) for \( X = (X_1, \ldots, X_n) \) consists of two components:

- A structure \( S \) for \( X \) is a directed acyclic graph (DAG) that represents a set of conditional (in)dependences between triplets of variables
- A set of local probability distributions \( \theta^S = (\theta_1, \ldots, \theta_n) \)

Conditional (in)dependences between triplets of variables

Given three disjoints sets of variables, \( Y, Z, W \), we say that \( Y \) is conditionally independent of \( Z \) given \( W \) if, for any \( y, z, w \), we have \( p(y | z, w) = p(y | w) \)

Factorization of the joint probability distribution

\[
p(x | \theta^S) = \prod_{i=1}^n p(x_i | pa_i^S, \theta_i)
\]
Bayesian Networks

Definition

- $X_i$ discrete variable with $|\Omega_i| = r_i$ for all $i = 1, \ldots, n$
- Local distributions: $p(x_i^k \mid pa_i^j, S_i) = \theta_{x_i^k \mid pa_i^j} \equiv \theta_{ijk}$
- $pa_i^1, S, \ldots, pa_i^{q_i}, S$ denotes the values of $Pa_i^S$ with $q_i = \prod_{X_g \in Pa_i^S} r_g$

Example

Model structure

- Model parameters and local probability distributions
  - $\theta_1 = (\theta_{1-1}, \theta_{1-2})$
  - $\theta_2 = (\theta_{2-1}, \theta_{2-2})$
  - $\theta_3 = (\theta_{311}, \theta_{312})$
  - $\theta_{321}, \theta_{322}$
  - $\theta_{331}, \theta_{332}$
  - $\theta_{341}, \theta_{342}$

Factorization of the joint probability distribution

$$p(x \mid \theta_S) = p(x_1 \mid \theta_1)p(x_2 \mid \theta_2)p(x_3 \mid x_1, x_2, \theta_3)$$
Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks
     - Learning Gaussian Bayesian Networks from Data
     - Learning Conditional Gaussian Bayesian Networks from Data

   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs

4. Final Remark
Inference in Bayesian Networks

Asia network structure
Marginal distributions

Inference in Bayesian Networks

Visit To Asia?
- No Visit: 99%
- Visit: 1%

Smoking?
- Non Smoker: 50%
- Smoker: 50%

Tuberculosis?
- Absent: 99%
- Present: 1%

Lung Cancer?
- Absent: 95%
- Present: 5%

Bronchitis?
- Absent: 55%
- Present: 45%

Tuberculosis or Lung...
- Nothing: 94%
- Cancer OR...: 6%

X-Ray Result
- Normal: 89%
- Abnormal: 11%

Dyspnea?
- Absent: 56%
- Present: 44%
Exact inference: $p(X_i = x_i^k | X_O = x_O)$
Inference in Bayesian Networks

**Exact inference: Triangulation of the moral graph**

Lauritzen and Spiegelhalter algorithm (1988): a) Moral graph; b) Triangulation of the moral graph (NP complete (Wen, 1991)); c) Junction tree; d) Collect and distribute evidence

(a) DAG. (b) Moral graph. **Nodes are eliminated in order:** $X_1$, $X_5$, $X_3$, $X_4$, $X_2$, $X_6$. Let $r_i = i + 1$ ($i = 1, 2, \ldots, 6$).

(c) Eliminate $X_1$: $C_1 = \{X_1, X_2, X_3, X_4\}$, added edges: $\{X_2, X_3\}, \{X_3, X_4\}$. (d) Eliminate $X_5$: $C_2 = \{X_4, X_5\}$.

(e) Eliminate $X_3$: $C_3 = \emptyset$. (f) Eliminate $X_4$: $C_4 = \{X_2, X_4, X_6\}$, added edge: $\{X_2, X_6\}$. (g) Eliminate $X_2$: $C_5 = \emptyset$.

(h) Eliminate $X_6$: $C_6 = \emptyset$. (i) **Total weight of the triangulated graph:** $\log_2(2 \cdot 3 \cdot 4 \cdot 5 + 5 \cdot 6 + 3 \cdot 5 \cdot 7) = \log_2 255$
Double direction synergy

Evolutionary Computation
- Genetic Algorithms
- Estimation of Distribution Algorithms
- Evolutionary Programming
- Ant Colony Optimization
- ...

Probabilistic Graphical Models
- Bayesian Networks
- Gaussian Bayesian Networks
- Conditional Gaussian Bayesian Networks
- Markov Networks
- ...

P. Larrañaga
Probabilistic Graphical Models & Evolutionary Computation
Genetic algorithms for the triangulation of the moral graph

- Larrañaga et al. (1997). *Statistics and Computing*
  - Optimal node elimination sequence (absolute order) resembles the traveling salesman problem (relative order)
  - Path representation: 8 crossover operators (PMX, CX, OX1, OX2, POS, ER, VR, AP) and 6 mutation operators (DM, EM, ISM, SIM, IVM, SM)
  - Rank-based selection, steady state GA
  - The best pair between combinations of crossover and mutation operators was: CX (cycle crossover) and ISM (insertion mutation)

- Wang et al. (2006). *Congress on Evolutionary Computation*
  - Path representation
  - Adaptive genetic algorithm: adjusts the probabilities of crossover and mutation operators by itself, and adjusts the pressure of selection according to the evolution of the population

- Dong et al. (2010). *Conference on Knowledge Discovery and Data Mining*
  - Path representation
  - New crossover operator (order reversing crossover) and a two-fold mutation operator based on ISM and minimum filling weight
**EDAs for the triangulation of the moral graph**


- **The standard simulation** of $n$ random variables, each with $n$ possible values, does not guarantee that the output sequence is a permutation.

- **EDAs (UMDA and MIMIC)** in a discrete domain: $n - 1$ random variables $Z_1, ..., Z_{n-1}$, each with $|\Omega_i| = n - i + 1$ possible values.

<table>
<thead>
<tr>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>$Z_3$</th>
<th>Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1-2-3-4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1-2-4-3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1-3-2-4</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1-3-4-2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1-4-2-3</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1-4-3-2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2-1-3-4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2-4-3-1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3-1-2-4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>4-1-2-3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4-3-2-1</td>
</tr>
</tbody>
</table>

- **EDAs (UMDA and MIMIC)** in a continuous domain: the individual (0.5, 1.6, 0.2, 0.1) will be associated with the ordering 3-4-2-1, but so will the individual (0.3, 2.0, 0.2, 0.0) (redundancy).
Inference in Bayesian Networks

**ACO for the triangulation of the moral graph**

Gámez and Puerta (2002). *Pattern Recognition Letters*

- **Graph to be considered**: a complete graph (always possible to reach a node $i$ from node $j$ for every pair of nodes $(i, j)$)
- **Reduction of the search space cardinality** by first eliminating simplicial nodes (the node and its adjacent nodes form a clique)
- For ant $k$ located at node $i$, the probability of choosing node $j$ as the following node to visit is:

$$p_k(i, j) = \begin{cases} 
(\tau_{ij})^{\alpha}(\nu_{ij})^{\beta} & \text{if } j \in J_k(i) \\
0 & \text{otherwise}
\end{cases}$$

- $\tau_{ij}$ pheromone amount in the direct path between $i$ and $j$
- $\nu_{ij}$ knowledge about the problem (deterministic heuristic approaches)
- $\alpha$ and $\beta$ control parameters
- $J_k(i)$ set of nodes for which there is a direct path from node $i$ and not yet visited by ant $k$

- Classic rules for adaptation of pheromones
Inference in Bayesian Networks

**Total abductive inference (Most Probable Explanation (MPE))**

Which is the most probable configuration for the unobserved variables, $X_U = X \setminus X_O$, given the observed evidence, $X_O = x_O$?

$$x_U^* = \arg \max_{x_U} p(x_U | X_O = x_O)$$

**Partial abductive inference (Maximum a Posteriori (MAP))**

Given a subset of the unobserved variables, $X_E \subseteq X_U$, called the explanation set, which is the most probable configuration for the variables in $X_E$ given the observed evidence, $X_O = x_O$?

$$x_E^* = \arg \max_{x_E} p(x_E | X_O = x_O) = \arg \max_{x_E} \sum_{x_R} p(x_E, x_R | X_O)$$

where variables $X_R = X_U \setminus X_E$
Genetic algorithms for total abductive inference (MPE)


- An individual of the GA represents a possible solution, is an array of dimension $r = |X_U|$, where:
  - Each position codifies the value of an unobserved variable
  - If $X_i$ is in the $i$-th position of the array, then this position can take $r_i = |\Omega_{X_i}|$ values

- Fitness function to evaluate each individual:
  $$p(x_u|x_o) \propto p(x_u, x_o) = \prod_{i=1}^{n} p(x_i | pa_i^S, \theta_i)$$

- GA characteristics: 2-point crossover, 1-bit mutation, steady state

- Seeding of the initial population using the a priori probabilities and propagation of evidence (observed values)
Inference in Bayesian Networks

**Genetic algorithms for total abductive inference (MPE)**

  - Genetic algorithm with a crossover operator that takes advantage of the topology of the Bayesian network
  - Genetic algorithm based on probabilistic crowding niching
  - Genetic algorithm whose operators are guided by reinforcement learning
- Guo et al. (2004). *Lecture Notes in Computer Science, Vol. 3339*
  - Ant system algorithm (all times probabilistic decision is carried out)
  - Ants must visit all nodes in the topological order
  - The decision on which value to select is based on the conditional probability table, observed evidence and previously visited parents
- Sriwachirawat and Auwatanamonkol (2006). *IEEE Conference on Cybernetics and Intelligent Systems*
  - Genetic algorithm based on a niching method for the k-MPE problem
Inference in Bayesian Networks

**Genetic algorithms for partial abductive inference (MAP)**

De Campos et al. (1999). *Pattern Recognition Letters*

- Each individual is an array of integers of length $|X_E|$
- The $i$-th position can take as many values as different states has the variable in position $i$ in $X_E$
- The search space is (by far) smaller than in MPE ... but the problem is (by far) much harder to solve: the cost of evaluating individuals is now higher
- Reduce the computational burden of the evaluation function:
  - Precomputation of the junction tree: leaf cliques without explanation variables are recursively marginalized out
  - Consistent marginalization: avoid realoading junction tree potentials each time a new individual is going to be evaluated
- The GA is based on the generational modGA by Michalewich
EDAs for partial abductive inference (MAP and $k$-MAP)


- Same individual representation and evaluation function as for GAs
- Experiments with ALARM: 37 nodes, 46 arcs, $|X_0| = 5$, and $|X_E| = 20$
- $\%\text{mass}^k$ denotes the percentage of probability mass obtained for the $k$-MAP with respect to the probability provided by the exact algorithm

<table>
<thead>
<tr>
<th></th>
<th>$%\text{mass}^1$</th>
<th>$%\text{mass}^{10}$</th>
<th>$%\text{mass}^{25}$</th>
<th>$%\text{mass}^{50}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMDA</td>
<td>88.07</td>
<td>81.32</td>
<td>77.62</td>
<td>74.54</td>
</tr>
<tr>
<td>MIMIC</td>
<td>87.14</td>
<td>81.06</td>
<td>76.67</td>
<td>73.26</td>
</tr>
<tr>
<td>EBNA</td>
<td>97.76</td>
<td>96.79</td>
<td><strong>95.19</strong></td>
<td><strong>93.80</strong></td>
</tr>
<tr>
<td>GA</td>
<td><strong>99.37</strong></td>
<td><strong>97.36</strong></td>
<td>93.00</td>
<td>87.55</td>
</tr>
</tbody>
</table>

$\%\text{mass}^k$ denotes the percentage of probability mass obtained for the $k$-MAP with respect to the probability provided by the exact algorithm.
Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks
  - Learning Gaussian Bayesian Networks from Data
  - Learning Conditional Gaussian Bayesian Networks from Data

   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs

4. Final Remark
Learning Bayesian Networks

Learning structure and parameters

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$X_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- $p(X_1=0) = 0.60$
- $p(X_2=0|X_1=0) = 0.10, p(X_2=0|X_1=0) = 0.70$
- $p(X_3=0|X_1=0) = 0.40, p(X_3=0|X_1=0) = 0.90$
- $p(X_4=0|X_2=0, X_3=0) = 0.60, p(X_3=0|X_2=0, X_3=1) = 0.20$
- $p(X_4=0|X_2=0, X_3=0) = 0.60, p(X_3=0|X_2=0, X_3=1) = 0.20$
- $p(X_5=0|X_3=0) = 0.60, p(X_5=0|X_3=0) = 0.60$
Learning Bayesian Networks

Learning parameters

Given a data set of cases $D = \{x^{(1)}, \ldots, x^{(N)}\}$ drawn at random from a joint probability distribution $p(x_1, \ldots, x_n)$

- **Maximum likelihood estimation:**
  $$\hat{\theta}_{ijk} = p(X_i = x_i^k | Pa_i = pa^j_i) = \frac{N_{ijk}}{N_{ij}}$$

- **Bayesian estimation:**
  - It is assumed a prior knowledge expressed by means of a prior joint distribution over the parameters:
    $$p(\theta_{i1}, \theta_{i2}, \ldots, \theta_{ir_i}) \sim Dir(\theta_{i1}, \ldots, \theta_{ir_i}; \alpha_1, \ldots, \alpha_{ri}) = \frac{\Gamma(\sum_{w=1}^{r_i} \alpha_w)}{\prod_{w=1}^{r_i} \Gamma(\alpha_w)} \theta_{i1}^{\alpha_1-1} \cdots \theta_{ir_i}^{\alpha_{ri}-1}$$
  - For a multinomial distribution, if the prior is $Dir(\theta_{i1}, \ldots, \theta_{ir_i}; \alpha_1, \ldots, \alpha_{ri})$, then the posterior is $Dir(\theta_{i1}, \ldots, \theta_{ir_i}; \alpha_1 + N_{ij1}, \ldots, \alpha_r + N_{ijr_i})$
  - $$\hat{\theta}_{ijk} = p(X_i = x_i^k | Pa_i = pa^j_i) = \frac{N_{ijk} + \alpha_k}{N_{ij} + \sum_{w=1}^{r_i} \alpha_w}, \text{ where } \sum_{w=1}^{r_i} \alpha_w \text{ is called the equivalent sample size (the virtually observed sample)}$$
Learning Bayesian Networks

Learning structures

Finding the best network according to some criterion even with the constraint that each node has no more than $K$ parents is NP-hard (Chickering et al., 1994)

- Based on detecting conditional independencies
  - First: carry out a study of the dependence and independence relationships between the variables by means of statistical tests
  - Second: try to find the structure (or structures) that represents the most (or all) of these relationships

- Based on score + search
  - They try to find the structure that best “fits” the data
  - They need:
    - A score (metric or evaluation function) in order to measure the fitness of each candidate structure
    - A search method (heuristic) to explore in an intelligent manner the space of possible solutions
    - Several types of spaces can be considered
Learning Bayesian Networks

Learning structures (score + search)

Score: Penalized log-likelihood

- Log-likelihood of the data: $\log p(D|S, \hat{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}}$

- Penalizing the complexity: $\sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \text{dim}(S) \text{pen}(N)$
  - $\text{dim}(S) = \sum_{i=1}^{n} q_i (r_i - 1)$ model dimension
  - $\text{pen}(N)$ non negative penalization function
    - $\text{pen}(N) = 1$: Akaike’s information criterion (AIC)
    - $\text{pen}(N) = \frac{1}{2} \log N$: Bayesian information criterion (BIC) or the minimum description length (MDL) criterion
Learning Bayesian Networks

Learning structures (score + search)

Score: Bayesian scores

\[ \hat{S} = \arg \max_S p(S|D) \equiv \arg \max_S p(D|S)p(S) \]

where \( p(D|S) \) denotes the marginal likelihood and \( p(S) \) the prior distribution over structures. If \( p(S) \) is uniform, \( \hat{S} = \arg \max_S p(D|S) \)

- **K2 score**: Assuming that \( p(\theta|S) \) is uniform, it is possible to obtain a closed formula for \( p(D|S) \) (Cooper and Herskovits, 1992):

\[
p(D|S) = \prod_{i=1}^{n} \left( \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \right) \prod_{k=1}^{r_i} N_{ijk}!
\]

- **BDe score**: Assuming that \( p(\theta|S) \) follows a Dirichlet distribution, it is possible to obtain a closed formula for \( p(D|S) \) (Heckerman et al., 1995):

\[
p(D|S) = \prod_{i=1}^{n} \left( \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right)
\]

This score is called **Bayesian Dirichlet equivalence metric** because it verifies the score equivalence property (two DAGs representing the same set of conditional independencies score the same)
Learning Bayesian Networks

Learning structures (score + search)

Search: Space of DAGs

- Cardinality of the search space (Robinson, 1977):
  \[ S(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^i (n-i) S(n-i) \]
  \[ S(0) = 1; \quad S(1) = 1 \]

- Search algorithms:
  - **K2 algorithm** (Cooper and Herskovits, 1992):
    - A total ordering between the nodes and an upper bound is set on the number of parents for any node are assumed
    - At each step K2 incrementally adds the parent whose addition provides the best value for
      \[ g(X_i, Pa_i) = \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \]
    - K2 stops when adding a single parent to any node cannot increase \( g(X_i, Pa_i) \)
  - **B algorithm** (Buntine, 1991): insert, delete and invert an arc
  - **Tabu search** (Bouckaert, 1995)
  - **Simulated annealing** (Heckerman et al., 1995)
**Genetic algorithms** for learning BNs in the **space of DAGs**

Larrañaga et al. (1996). *IEEE Transactions Pattern Analysis and Machine Intelligence*

An individual of the GA corresponds to a Bayesian network structure, and can be represented by an $n \times n$ connectivity matrix, with elements $c_{ij} = 1$ iff $X_j$ is a parent of $X_i$, and $c_{ij} = 0$ otherwise. As a string: $c_{11} c_{21} \ldots c_{n1} c_{12} c_{22} \ldots c_{n2} \ldots c_{1n} c_{2n} \ldots c_{nn}$

- **Assuming a total ordering** between the variables: Connectivity matrices are lower triangular matrices. The genetic operators are closed operators under DAG conditions.

- **Without assuming an ordering** between the variables: The genetic operators are not closed operators regarding the DAG conditions. To ensure the closeness a repair operator is introduced.

(a) The parent structures: 001001000 and 000000110

Offspring: 001001110 and 000000000

(b) Crossover point: sixth bit

(c) Mutation is not a closed operator

P. Larrañaga

Probabilistic Graphical Models & Evolutionary Computation
## Genetic algorithms for learning BNs in the space of DAGs

  GAs hybridized with a local search strategy

- **Etxeberria et al. (1997). Pattern Recognition Letters**
  Fuse-DAG algorithm to guarantee the closeness of the crossover operator

- **Myers et al. (1999). Uncertainty in Artificial Intelligence**
  Data sets with incomplete data. Evolve the missing data and the BN structure at the same time

- **Cotta and Muruzábal (2002). Lecture Notes in Computer Science, Vol. 2439**
  Hybrid operators using phenotypic information

- **Van Dijk et al. (2003). GECCO**
  Design principles that serve to guide the construction of selecto-recombinative GAs

- **Wong et al. (2004). Decision Support Systems**
  Hybrid algorithm: first conditional independence tests to reduce the search space; second GAs in this reduced space

- **Kim et al. (2005). Lecture Notes in Computer Science, 3488**
  A Bayesian network structure is encoded by a variable ordering list together with a connectivity matrix

  Chi-squared test to break the cycles

- **Barriére et al. (2009). GECCO**
  Evolve a population of independence models (each of them is a set of conditional independence statements satisfied in a dataset)
Learning Bayesian Networks

**Evolutionary programming** for learning BNs in the space of DAGs

Wong et al. (1999). *IEEE Transactions on Pattern Analysis and Machine Intelligence*

- The probabilities of executing 1, 2, 3, 4, 5, or 6 number of mutations are 0.2, 0.2, 0.2, 0.2, 0.1, and 0.1 respectively.
- Four types of mutations:
  - **Simple mutation**: randomly selects an edge. If this edge is present in the network it is deleted, otherwise it is added.
  - **Reversion mutation**: randomly selects an edge, and modifies its direction.
  - **Move mutation**: modifies the parent set of a node.
  - **Knowledge-guided mutation**: similar to the simple mutation operator, but it considers the MDL metric of all possible edges and determines which edge should be removed or inserted.

MDL metric for the ALARM network with 500 cases

P. Larrañaga

Probabilistic Graphical Models & Evolutionary Computation
**EDAs for learning BNs in the space of DAGs**

Blanco et al. (2003). *International Journal of Intelligent Systems*

- **Individual representation**: connectivity matrix (Bayesian network structure)
- **Univariate EDAs**: UMDA and PBIL (Baluja, 1994)
- **When no total ordering between the variables**: simple repair operator (randomly delete cycles)

The original *Alarm* network structure (left) has 37 nodes and 46 arcs. The network structure learned by UMDA (right) and total ordering with the BIC penalized log-likelihood has 45 arcs. Notice that the arc from node 12 to node 32 is missing.
ACOs for learning BNs in the space of DAGs

De Campos et al. (2002). *International Journal of Approximate Reasoning*

- **Representation of the problem**: a graph where the states are DAGs with \( n \) nodes. A state \( G_h \) will be a graph with exactly \( h \) arcs and no directed cycles. Ant incremental construction until the ant decides to stop the construction phase.

- **Heuristic information**: include in the graph the arc producing the greatest increase in the metric (K2 metric).

- **Pheromone updating rules**:
  - **Global**: only the ant which constructed the best solution reinforces the level of pheromone.
  - **Local**: if an ant carries out the transition from node \( i \) to node \( j \), then the pheromone level of the corresponding arc is changed.

- **Pheromone transition rule**: the next arc to be included in the current graph by an ant is selected using a stochastic decision rule that takes into account the improvement in the metric and the current pheromone deposited at each arc.

- **Local optimizer**: hill climbing with arc addition, arc deletion and arc reversal operators.
Markov equivalent DAGs

Two DAGs represent the same conditional independencies iff they have the same edges (arcs without direction) and the same head to head patterns. Each PDAG represents an equivalent class of DAGs. BDe as score

Evolutionary computation for learning BNs in the space of equivalence classes

- Cotta and Muruzábal (2004). *Second European Workshop on PGMs* Evolutionary programming
- Daly and Shen (2009). *Journal of Artificial Intelligence Research* Ant colony optimization
Learning Bayesian Networks

Genetic algorithms for learning BNs in the space of orderings

Larrañaga et al. (1996). *IEEE Transactions on Systems, Man, and Cybernetics*

- Some algorithms (as K2) assume a total ordering between the variables
- The question is what is the optimal ordering for this type of algorithms?
- Cardinality of the search space: $n!$
- Path representation: 8 crossover operators (PMX, CX, OX1, OX2, POS, ER, VR, AP) and 6 mutation operators (DM, EM, ISM, SIM, IVM, SM)
- K2 score as fitness function
- The best pair: CX (cycle crossover) and SM (scramble mutation)
**Evolutionary computation** for learning BNs in the *space of orderings*

  Genetic algorithms: variants of the standard crossover and mutation operators

- Hsu et al. (2002). *GECCO*
  Genetic algorithm: order crossover (OX) and exchange mutation operator (EM). **K2 greedy search** for structure learning. A *wrapper fitness function* based on inferential loss (exact algorithm or by sampling)

- De Campos et al. (2002). *Mathware and Soft Computing*
  Ant colony optimization

- Romero et al. (2004). *International Journal of Pattern Recognition*
  Two types of **EDAs** (UMDA and MIMIC) with discrete and continuous encoding

- Lee et al. (2006). *Lecture Notes in Computer Science, Vol. 3971*
  Individual of the **GA** as a pair of chromosomes: ordering among the nodes + the *child set* for each variable

- Kabli et al. (2007). *GECCO*
  **chainGA**: evaluates a chain structure of the given ordering using the K2 metric and evolves the orderings with standard crossover and mutation operators. The K2 greedy search algorithm is run on the best ordering found with this strategy

- Pinto et al. (2009). *IEEE Transactions on Evolutionary Computation*
  Ant colony optimization. Integration of ACO with the max-min hill climbing algorithm

- Wu et al. (2010). *CEC*
  Ant colony optimization. ACO versions of ChainGA and K2GA algorithms
Learning Bayesian Classifiers

Bayesian network classifiers

\[ c^* = \arg \max_c p(c | x_1, \ldots, x_n) = \arg \max_c p(c)p(x_1, \ldots, x_n | c) \]

Depending on the factorization of \( p(x_1, \ldots, x_n | c) \) we obtain:

- **Naive Bayes** (Minsky, 1961): \( c^* = \arg \max_c p(c) \prod_{i=1}^{n} p(x_i | c) \)
- **Semi naive Bayes** (Pazzani, 1997):
  \[ c^* = \arg \max_c p(c) \prod_{i=1}^{r} p(x_i | c) \prod_{l,t=1}^{m} p(x_l, x_t | c) \prod_{k,h,f=1}^{w} p(x_k, x_h, x_f | c) \]
- **Tree augmented naive Bayes** (TAN) (Friedman et al., 1997):
  \[ c^* = \arg \max_c p(c) \prod_{i=1}^{n} p(x_i | x_{j(i)}, c) \]
Evolutionary computation for feature subset selection for naive Bayes

**Individual:** $n$ dimensional binary array. Fitness function: classification accuracy (wrapper)

- **Inza et al.** (2000). *Artificial Intelligence*  
  Compare the behavior of genetic algorithms and estimation of distribution algorithms on real and simulated datasets

- **Inza et al.** (2001). *International Journal of Approximate Reasoning*  
  Compare genetic algorithms and sequential algorithms

- **Inza et al.** (2001). *Artificial Intelligence in Medicine*  
  Genetic algorithms in FSS for naive Bayes to predict the survival of cirrhotic patients

- **Liu et al.** (2001). *IEEE Transactions on Systems, Man, and Cybernetics*  
  Genetic algorithms in FSS for naive Bayes in a rainfall prediction problem

  Estimation of distribution algorithms in gene selection for cancer classification
EDAs for a supervised wrapper multidimensional discretization in naive Bayes

Flores et al. (2007). *Intelligent Data Analysis*

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-17</td>
<td>29</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

**Original data set**

<table>
<thead>
<tr>
<th>$d_1$</th>
<th>$d_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-12</td>
</tr>
<tr>
<td>4</td>
<td>-10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Individuals of the EDA (multidimensional discretization)**

<table>
<thead>
<tr>
<th>$t^1_1$</th>
<th>$t^1_2$</th>
<th>$t^2_1$</th>
<th>$t^2_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-12</td>
<td>-6</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>-10</td>
<td>-3</td>
<td>9</td>
</tr>
</tbody>
</table>

**Discretized data sets**

<table>
<thead>
<tr>
<th>$X^d_1$</th>
<th>$X^d_2$</th>
<th>$X^d_3$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

**Bayesian Network**

P. Larrañaga

Probabilistic Graphical Models & Evolutionary Computation
EDAs for a supervised wrapper multidimensional discretization in *naive Bayes*

Flores et al. (2007). *Intelligent Data Analysis*

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-17</td>
<td>29</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

Original data set

<table>
<thead>
<tr>
<th>$d_1$</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>-12</td>
<td>-6</td>
<td>12</td>
<td>87.13</td>
</tr>
<tr>
<td>4</td>
<td>-10</td>
<td>-3</td>
<td>9</td>
<td>71.14</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Individuals of the EDA (multidimensional discretization)

<table>
<thead>
<tr>
<th>$X_1^{d_1}$</th>
<th>$X_2^{d_1}$</th>
<th>$X_3^{d_1}$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$X_1^{d_2}$</td>
<td>$X_2^{d_2}$</td>
<td>$X_3^{d_2}$</td>
<td>$C$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Discretized data sets

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1^*$</td>
<td>$X_2^*$</td>
<td>$X_3^*$</td>
<td>$C$</td>
</tr>
<tr>
<td>Accuracy 87.13%</td>
<td>Accuracy 71.14%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
EDAs for semi naive Bayes

Robles et al. (2004). *Artificial Intelligence in Medicine*

From the greedy modeling process of semi naive Bayes (Pazzani, 1997) to an EDA based approach, where individuals will have \( n \) variables each one with an integer value in \( \{0, 1, 2, \ldots, n\} \) representing the (super)node each variable belongs to.

\[
\begin{align*}
C &\rightarrow X_1, X_3 \\
&\rightarrow (1,0,1,2) \\
C &\rightarrow X_1, X_2, X_3, X_4 \\
&\rightarrow (1,1,1,1)
\end{align*}
\]
Learning Bayesian Classifiers

**Evolutionary computation for TAN**

  Genetic algorithms to evolve Prüfer numbers for encoding trees of $n$ nodes (a sequence of length $n - 2$ on an alphabet of $n$ symbols). Fitness function: AIC or BIC

- Dong et al. (2010). *International Journal of Innovative Computing, Information and Control*  
  Genetic algorithms with ad hoc operators to evolve TAN structures. Fitness function: likelihood
Learning Bayesian Classifiers

Markov blanket

In a Bayesian network, any variable is conditionally independent on the rest given its Markov blanket (its parents, its children, and its children’s parents).

Evolutionary computation for Markov blanket

- Sierra and Larrañaga (1998). *Artificial Intelligence in Medicine*
  Each individual of the genetic algorithm encodes a Markov blanket of the class variable: two alternatives depending on whether all predictor variables belong to the Markov blanket.

- Kline et al. (2005). *Annals of Emergency Medicine*
  Genetic algorithm for searching for the best Markov blanket in predicting venous thromboembolism.

- Zhu et al. (2007). *Pattern Recognition*
  Genetic algorithms for Markov blanket in selecting genes in several microarray datasets. Good compromise among classification accuracy, number of selected genes, computational cost, and robustness.
Learning Bayesian Networks for Clustering

Bayesian networks for clustering

\[
p(c, x \mid \theta^S) = p(c \mid \theta_c) \prod_{i=1}^n p(x_i \mid c, pa_i^S, \theta_c, \theta_i)
\]

The value of variable \( C \) is unknown, it is estimated with the EM algorithm

**EDAs for clustering**


- **EDAs (UMDA)** to search for the optimal structure of dependency between the predictor variables
- **Each individual** in the UMDA represents a connectivity matrix \( C \) with elements \( c_{ij} \), such that:
  (i) \( c_{ij} = 1 \) if \( X_j \in Pa_i \)
  (ii) \( c_{ij} = 0 \) otherwise
- **Every solution** in the search space can be represented by an \( m \)-dimensional individual \( z = (z_1, \ldots, z_m) \), where \( m = (n^2 - n)/2 \), consisting only of the elements of \( C \) above the diagonal
Learning Dynamic Bayesian Networks

**Evolutionary computation** for learning *dynamic* Bayesian networks

- Tucker et al. (2001). *International Journal of Intelligent Systems*
  Genetic algorithms to evolve dynamic Bayesian network structures. Initial population with evolutionary programming
- Tucker et al. (2003). *GECCO*
  Genetic algorithms for spatial time series
- Jia et al. (2003). *International Conference on Machine Learning and Cybernetics*
  Inmune evolutionary algorithms for learning dynamic Bayesian networks
- Ross and Zuviria (2007). *Applied Intelligence*
  Multi-objective genetic algorithms to evolve dynamic Bayesian networks with two objectives: maximizing likelihood and minimizing complexity
Learning Gaussian Bayesian Networks

Gaussian Bayesian networks

Model structure

\[ X_1 \leftarrow X_3 \rightarrow X_2 \]

Model parameters and local probability density functions

\[ \theta_1 = (m_1, -, v_1) \quad f(x_1 \mid \theta_1) \sim \mathcal{N}(x_1; m_1, v_1) \]
\[ \theta_2 = (m_2, -, v_2) \quad f(x_2 \mid \theta_2) \sim \mathcal{N}(x_2; m_2, v_2) \]
\[ \theta_3 = (m_3, b_3, v_3) \quad f(x_3 \mid x_1, x_2, \theta_3) \sim \mathcal{N}(x_3; m_3 + b_{13}(x_1 - m_1) + b_{23}(x_2 - m_2), v_3) \]
\[ b_3 = (b_{13}, b_{23})^t \]

Factorization of the joint density

\[ f(x \mid \theta^S) = f(x_1 \mid \theta_1)f(x_2 \mid \theta_2)f(x_3 \mid x_1, x_2, \theta_3) \]

Evolutionary computation for Gaussian Bayesian networks

To the best of our knowledge there are no applications of evolutionary computation
Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks
   - Learning Gaussian Bayesian Networks from Data
   - Learning Conditional Gaussian Bayesian Networks from Data

   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs

4. Final Remark
Conditional Gaussian Bayesian Networks

\( X = (X_1, \ldots, X_n) \) is said to follow a conditional Gaussian distribution:

(i) \( X = Y \cup Z \) and \( Y \cap Z = \emptyset \);

(ii) \( f(z \mid y) \sim \mathcal{N}(z; \mu(y), \Sigma(y)) \) for all \( y = y \) such that \( p(y) > 0 \)

\[
f(x \mid \theta^S) = \prod_{i=1}^{f} p(x_i \mid \text{pa}_i^S, \theta_i) \prod_{i=f+1}^{n} f(x_i \mid \text{pa}_i^S, \theta_i)
\]

\[
\begin{align*}
\theta_1 &= (\theta_1^{-1}, \theta_1^{-2}) & p(x_1 \mid \theta_1, s^h) \\
\theta_2 &= (\theta_2^{-1}, \theta_2^{-2}) & p(x_2 \mid \theta_2, s^h) \\
\theta_3 &= (\theta_3^{-1}, \theta_3^{-2}, \theta_3^{-3}, \theta_3^{-4}) \\
\theta_4 &= (\theta_4^{-1}, \theta_4^{-2}, \theta_4^{-3}, \theta_4^{-4}) \\
\theta_0 &= (m_0, b_{04}, v_0^{-1}) \\
\theta_4 &= (m_4, b_{43}, v_4^{-1}) \quad f(x_4 \mid x_1^1, x_2^1, x_3^1, \theta_4, s^h) \sim \mathcal{N}(x_4; m_4^1 + b_{43}^1(x_4 - m_3^4), v_4^1)
\end{align*}
\]

\[
f(x \mid \theta, s^h) = p(x_1 \mid \theta_1, s^h)p(x_2 \mid \theta_2, s^h)f(x_3 \mid x_1, x_2, \theta_3, s^h)f(x_4 \mid x_1, x_2, x_3, \theta_4, s^h)
\]

**Evolutionary computation for conditional Gaussian Bayesian networks**

Mascherini and Stefanini (2005). *Intern. Conf. Computational Intelligence for Modelling, Control, and Automation Genetic algorithms* for the optimal conditional Gaussian Bayesian network. Invalid structures are corrected by deleting inadmissible arcs at random.
Outline

1. Estimation of Distribution Algorithms
2. Evolutionary Computation for Probabilistic Graphical Models
   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs
4. Final Remark
Graphical Representation of EDAs

Initial population of candidate solutions → Selected candidates → Bayesian or Gaussian network learning → Sampling of new candidate solutions
Graphical Representation of EDAs
PGMs in EDAs

Evolutionary Computation
- Genetic Algorithms
- Estimation of Distribution Algorithms
- Evolutionary Programming
- Ant Colony Optimization
- ...

Probabilistic Graphical Models
- Bayesian Networks
- Gaussian Bayesian Networks
- Conditional Gaussian Bayesian Networks
- Markov Networks
- ...

P. Larrañaga
Probabilistic Graphical Models & Evolutionary Computation
Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks
   - Learning Gaussian Bayesian Networks from Data
   - Learning Conditional Gaussian Bayesian Networks from Data

   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs

4. Final Remark
Bayesian Networks for EDAs

EBNA, BOA, LFDA

- **EBNA (Estimation of Bayesian Networks Algorithm)** (Etxeberria and Larrañaga (1999). *II Symposium on Artificial Intelligence*)
  - Detecting conditional independencies: EBNA$_{PC}$
  - Score: penalized likelihood (EBNA$_{BIC}$ and EBNA$_{K2}$)
  - Search: greedy search starting from the previous generation

- **BOA (Bayesian Optimization Algorithm)** (Pelikan et al. (1999). *GECCO*)
  - Score: marginal likelihood
  - Search: greedy search starting from scratch at each generation

- **LFDA (Learning Factorized Distribution Algorithm)** (Mühlenbein and Mahnig (1999). *Evolutionary Computation*)
  - Score: BIC
  - Search: greedy search starting from scratch at each generation
Symmetrical multimodal optimization problems. Graphs for the $P_{\text{grid}16}$ problem (left) and the $P_{\text{cat}28}$ problem (right). Dashed lines indicate the optimal cuts.

- Pelikan and Goldberg (2000). *PPSN VI*
  Cluster the set of selected individuals at each generation with *K*-means.
  \[ p_l(x) = \sum_{i=1}^{k} \pi_{l,i} p_{l,i}(x) \]
  where $\pi_{l,i}$ denotes the weight of the $i^{th}$ mixture component, and $p_{l,i}(x)$ is the probability distribution of the $i^{th}$ cluster.

  Mixture model, where each mixture is a Bayesian network. Its corresponding weight is obtained with the EM algorithm.
Markov Networks for EDAs

Markov networks

- Additively decomposed function iff: \( O(x) = O_1(x_{s_1}) + O_2(x_{s_2}) + \ldots + O_l(x_{s_l}) \)
- MAXSAT:
  \[
  O(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = (x_1 \lor x_2 \lor x_3 \lor \overline{x_4}) + (x_4 \lor \overline{x_5}) + (x_5 \lor \overline{x_6} \lor \overline{x_7})
  \]
  \[
  = O_1(x_1, x_2, x_3, x_4) + O_2(x_4, x_5) + O_3(x_5, x_6, x_7)
  \]
- The structure of the Markov network is maintained fixed over the generations

\[
p(x) = \psi_1(x_1, x_2, x_3, x_4)\psi_2(x_4, x_5)\psi_3(x_5, x_6, x_7)
\]

Markov networks for EDAs

- Santana (2005). Evolutionary Computation (MN-EDA)
EDAs based on other Discrete PGMs

Other *discrete PGMs for EDAs*

- Harik et al. (1999). *IEEE Transaction on Evolutionary Computation*
  Extended compact genetic algorithm
  \[ p_l(x) = \prod_{c=1}^{C} p_l(x_c) \]

- Gámez et al. (2007). *Lecture Notes in Computer Science, Vol. 4527*
  Dependency networks as models are learned from pairwise interactions. Gibbs sampling to generate new solutions

- Yang et al. (2010). *GECCO*
  L1-Regularization for learning Bayesian networks

- Santana et al. (2010). *Evolutionary Computation*
  Affinity algorithm for grouping variables
Bayesian classifiers for EDAs

Outline

1. Estimation of Distribution Algorithms

2. Evolutionary Computation for Probabilistic Graphical Models
   - Inference in Bayesian Networks
     - Triangulation of the Moral Graph
     - Total Abductive Inference
     - Partial Abductive Inference
   - Learning Bayesian Networks from Data
     - Learning the Joint Probability Distribution
     - Learning Bayesian Classifiers
     - Learning Bayesian Networks for Clustering
     - Learning Dynamic Bayesian Networks

   - Multivariate Discrete EDAs
   - Multivariate Continuous EDAs

4. Final Remark
EDAs based on Multivariate Normal Densities

Multivariate normal density

![Multivariate Normal Density](image)

\[ f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu)^t \Sigma^{-1} (x - \mu) \right] \]

- \( x \), \( n \) dimensional column vector
- \( \mu \), \( n \) dimensional mean vector
- \( \Sigma \), \( n \times n \) variance-covariance matrix

Estimation of Multivariate Normal Algorithm (EMNA\textsubscript{global})

![EMNA Global](image)

Larrañaga et al. (2000). \textit{GECCO}

- \textbf{Structure} of EMNA\textsubscript{global} in all generations
Multivariate Discrete EDAs  Multivariate Continuous EDAs

EDAs based on Sparse Normal Multivariate Densities

Estimation of Multivariate Normal Algorithm by Edge Exclusion (EMNA\textsubscript{ee})

Larrañaga et al. (2000). GECCO

- Based on detecting independencies between pairs of variables
- The learning is carried out by means of \( \binom{n}{2} \) tests for arc exclusion
  - \( X_i \) and \( X_j \) are independent iff the following null hypothesis is accepted (Smith and Whittaker, 1998)
    
    \[
    \begin{cases}
    H_0 : w_{ij} = 0 & \text{null hypothesis} \\
    H_A : w_{ij} \neq 0 & \text{alternative hypothesis}
    \end{cases}
    \]

  with \( w_{ij} \) elements of the precision matrix \( W = \Sigma^{-1} \)
- Likelihood ratio test:
  
  \[
  T_{lik} = -n \log(1 - r_{ij|rest}^2) \quad \text{with} \quad r_{ij|rest} = -\hat{w}_{ij}(\hat{w}_{ii}\hat{w}_{jj})^{-1/2}
  \]
Gaussian Bayesian networks

The local density functions follow a linear regression model:

\[ f(x_i \mid \text{pa}_i^S, \theta_i) \equiv \mathcal{N}(x_i; m_i + \sum_{x_j \in \text{pa}_i} b_{ji}(x_j - m_j), v_i) \]

- \( b_{ji} \): strength of the relationship between \( X_j \) and \( X_i \) (\( b_{ji} = 0 \) iff there is not an arc from \( X_j \) to \( X_i \))
- \( v_i \): variance of \( X_i \) conditioned to \( \text{Pa}_i \)
- \( \theta_i = (m_i, b_i, v_i) \): local parameters, \( b_i = (b_{1i}, \ldots, b_{i-1i})^t \)

Estimation of Gaussian Network Algorithm (EGNA\textsubscript{BIC})

Larrañaga et al. (2000). GECCO

- Score: penalized likelihood (BIC)
- Search: greedy
  - First generation: a disconnected graph
  - The rest of generations: start with the model obtained in the previous one
EDAs based on other Continuous PGMs

**Other continuous PGMs for EDAs**

- Tsutsui et al. (2001). *GECCO*. Marginal histogram models
- Pošik (2009). *GECCO*. Cauchy univariate densities
- Li et al. (2006). *International Conference on Pattern Recognition*. Boosting Gaussian mixtures for estimating the density
- Wang et al. (2009). *CEC*. Copulas for learning the density of the selected individuals
- Cuesta-Infante et al. (2010). *CEC*. Copulas for learning the density of the selected individuals
Outline

1. Estimation of Distribution Algorithms
2. Evolutionary Computation for Probabilistic Graphical Models
4. Final Remark
Evolutionary Metro
Evolutionary Metro

Inference Area
- Triangulation
- MPE
- MAP

Learning Area
- Bayesian Networks
- jpd
- DAGs
- Eq-class
- Order

ACOs
EDAs
GAs
EP
Evolutionary Metro

Inference Area
- Triangulation
- MPE

Learning Area
- Bayesian Networks
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB

Sup. classif.
- GAs
- EDAs
- ACOs
- EP

Evolutionary Computation

Probabilistic Graphical Models & Evolutionary Computation
Evolutionary Metro

Inference Area
- Triangulation
- MPE
- MAP

Learning Area
- Bayesian Networks
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB
- Sup. classif.
- Cluster

Evolutionary Computation

Probabilistic Graphical Models & Evolutionary Computation

P. Larrañaga
Evolutionary Metro

Inference Area
- Triangulation
- MPE
- MAP

Learning Area
- Bayesian Networks
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB
- Sup. classif.
- Cluster
- Dyn

Evolutionary Computation
- EDAs
- EC for PGMs
- PGMs for EDAs
- Probabilistic Graphical Models & Evolutionary Computation
Evolutionary Metro

Inference Area

- Triangulation
- MPE
- MAP
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB
- Sup. classif.
- Cluster
- Dyn
- DAGs

Learning Area

- Bayesian Networks
- ACOs
- EP
- GAs
- PGMs
- EDAs
- EDAs for PGMs
- PGMs for EDAs

Final Remark

Evolutionary Metro

CGBNs
**Evolutionary Metro**

**Inference Area**
- Triangulation
- MPE
- MAP
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB
- Sup. classif.

**Learning Area**
- Bayesian Networks
- EBNA
- BOA
- LFDA
- EBCOA
- ACOs
- EP
- EDAs

**Combinatorial Optimization Area**
- EBCOA
- MN-EDA
- LFDA
- BOA
- EBNA

**Evolutionary**
- Metro

**CGBNs**

**Probabilistic Graphical Models & Evolutionary Computation**
Many Thanks to

- R. Armañanzas
- E. Bengoetxea
- C. Bielza
- R. Blanco
- A. Cuesta-Infante
- J. I. Hidalgo
- I. Inza
- H. Karshenas
- P. L. López-Cruz
- J. A. Lozano
- T. Miquélez
- J. M. Peña
- V. Robles
- T. Romero
- R. Santana
Evolutionary Metro

Inference Area
- Triangulation
- GAs
- EDAs
- ACOs
- EP
- MPE
- MAP
- DAGs
- Eq-class
- Order
- jpd
- nB
- semi-nB
- TAN
- MB
- Sup. classif.
- Cluster
- Dyn
- DAGs

Learning Area
- Bayesian Networks
- EBNA
- BOA
- LFDA
- MN-EDA
- EBCOA

Continuous Area
- EMNA
- ee
- EGNA
- BIC

Combinatorial Optimization Area
- EBCOA
- MN-EDA
- LFDA
- BOA
- EBNA

Probabilistic Graphical Models & Evolutionary Computation

P. Larrañaga
PROBABILISTIC GRAPHICAL MODELS AND EVOLUTIONARY COMPUTATION

Pedro Larrañaga

Computational Intelligence Group
Artificial Intelligence Department
Technical University of Madrid

2010 IEEE World Congress on Computational Intelligence
Barcelona, July 21, 2010