Self-Organizing Maps for clustering and visualization of bipartite graphs

Nathalie Villa-Vialaneix

en collaboration avec Madalina Olteanu

nathalie.villa@toulouse.inra.fr
http://www.nathalievilla.org

Journées de Statistique de la SFdS
Rennes, 5 juin 2014
1. a short review of Self-Organizing Maps for non vectorial data
2. SOM for bipartite graphs
3. Application
1. a short review of Self-Organizing Maps for non vectorial data

2. SOM for bipartite graphs

3. Application
Aim: Project the data $x \in \mathbb{R}^d$ on a square 2-dimensional grid made of $U$ units $\{1, \ldots, U\}$:
- clustering
- non-linear projection (& visualization) that preserves topology
- generalizes $k$-means
Basics on SOM [Kohonen, 2001]

- \((x_i)_{i=1,\ldots,n} \subset \mathbb{R}^d\) are affected to a unit \(C(x_i) \in \{1, \ldots, U\}\)
- the grid is equipped with a “distance” between units: \(d(u, u')\)
- and observations affected to close units are close in \(\mathbb{R}^d\)
- every unit \(u\) corresponds to a prototype, \(p_u(x)\) in \(\mathbb{R}^d\)
Iterative learning (affectation step): $x_i$ is picked at random within $(x_k)_k$ and affected to best matching unit:
\[
C(x_i) = \arg \min_u \|x_i - p_u\|^2
\]
Iterative learning (representation step): all prototypes in neighboring units are updated with a gradient descent like step minimizing $\mathcal{E} = \sum_{i=1}^{n} \sum_{u=1}^{U} H^t(d(C(x_i), u))||x_i - p_u||^2$:

$$p_u^{t+1} \leftarrow p_u^t + \mu(t)H^t(d(C(x_i), u))(x_i - p_u^t)$$
Extensions to non vectorial data

KORRESP [Cottrell et al., 1993]

Data: contingency table $T = (n_{ij})_{ij}$ with $p$ rows and $q$ columns transformed into a numeric dataset $X$:

$$X = \begin{pmatrix}
\text{columns} & \text{rows} \\
\text{row profile} & \text{rows} \\
\text{column profile} & \text{columns}
\end{pmatrix}$$

with

$$\forall i = 1, \ldots, p \text{ and } \forall j = 1, \ldots, q, x_{ij} = \frac{n_{ij}}{n_i} \times \sqrt{\frac{n}{n_j}}$$
Extensions to non vectorial data

KORRESP [Cottrell et al., 1993]

Data: contingency table \( T = (n_{ij})_{ij} \) with \( p \) rows and \( q \) columns transformed into a numeric dataset \( X \):

\[
X = \begin{pmatrix}
\text{columns} \\
\text{augmented row profile} \\
\text{rows} \\
\text{augmented column profile} \\
\text{columns}
\end{pmatrix}
\]

with

\[
\forall i = 1, \ldots, p \text{ and } \forall j = q + 1, \ldots, q + p, \quad x_{ij} = x_{k(i)+p,j+q} \quad \text{with} \\
k(i) = \arg \max_{k=1,\ldots,q} x_{ik}
\]
**Data:** contingency table $\mathbf{T} = (n_{ij})_{ij}$ with $p$ rows and $q$ columns

transformed into a numeric dataset $\mathbf{X}$:

<table>
<thead>
<tr>
<th>columns</th>
<th>rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>row profile</td>
<td>augmented row profile</td>
</tr>
</tbody>
</table>

$$\mathbf{X} =$$

<table>
<thead>
<tr>
<th>columns</th>
<th>rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>augmented column profile</td>
<td>column profile</td>
</tr>
</tbody>
</table>

- **affectation** uses reduced profile
- **representation** uses augmented profile
- alternatively process row profiles and column profiles

This method is implemented in the R package **SOMbrero**.
Data: described by a dissimilarity matrix \( D = (\delta(x_i, x_j))_{i,j=1,...,n} \) 
\(((x_i)_i \text{ non necessarily vectorial})\)
Data: described by a dissimilarity matrix $D = (\delta(x_i, x_j))_{i,j=1,\ldots,n}$
($\langle x_i \rangle_i$ non necessarily vectorial)

Adaptations of the SOM algorithm:

- **prototypes**: expressed as (symbolic) convex combination of
  \((x_i)_i\): $p_u \sim \sum_{i=1}^{n} \gamma_{ui} x_i$, $\gamma_{ui} \geq 0$ and $\sum_{i} \gamma_{ui} = 1$
Extensions to non vectorial data 2

Relational SOM [Olteanu and Villa-Vialaneix, 2014]

Data: described by a dissimilarity matrix $D = (\delta(x_i, x_j))_{i,j=1,\ldots,n}$
($(x_i)$: non necessarily vectorial)

Adaptations of the SOM algorithm:

- **prototypes**: expressed as (symbolic) convex combination of
  
  $(x_i)$: $p_u \sim \sum_{i=1}^{n} \gamma_{ui} x_i$, $\gamma_{ui} \geq 0$ and $\sum_i \gamma_{ui} = 1$

- **distance computation**: $\|x_i - p_u\|^2$ replaced by

  $$(D \gamma_u)_i - \frac{1}{2} \gamma_u^T D \gamma_u$$

  in reference to a pseudo-Euclidean framework [Goldfarb, 1984]
Extensions to non vectorial data 2

Relational SOM [Olteanu and Villa-Vialaneix, 2014]

Data: described by a dissimilarity matrix $D = (\delta(x_i, x_j))_{i,j=1,...,n}$
$((x_i)_i$ non necessarily vectorial)

Adaptations of the SOM algorithm:

- **prototypes**: expressed as (symbolic) convex combination of $(x_i)_i$: $p_u \sim \sum_{i=1}^{n} \gamma_{ui} x_i$, $\gamma_{ui} \geq 0$ and $\sum_i \gamma_{ui} = 1$

- **distance computation**: $\|x_i - p_u\|^2$ replaced by

$$
(D\gamma_u)_i - \frac{1}{2} \gamma_u^T D \gamma_u
$$

in reference to a pseudo-Euclidean framework [Goldfarb, 1984]

- **representation**: replaced by an update of $\gamma_u$:

$$
\gamma_u^{t+1} \leftarrow \gamma_u^t + \mu(t)H^t(d(C(x_i), u))(1_i - \gamma_u^t)
$$

with $1_{il} = 1$ if $l = i$ and 0 otherwise.
Outline

1. a short review of Self-Organizing Maps for non vectorial data
2. SOM for bipartite graphs
3. Application
a bipartite graph is a graph $G = (V, E, W, C)$ such that:

- the vertices $V = \{x_1, \ldots, x_n\}$ are split into two sets (labeled with $C_i = 0$ or $C_i = 1$);
A bipartite graph is a graph $G = (V, E, W, C)$ such that:

- the vertices $V = \{x_1, \ldots, x_n\}$ are split into two sets (labeled with $C_i = 0$ or $C_i = 1$);
- the edges $E \subset V \times V$ are such that if $(x_i, x_j) \in E \Rightarrow$ either ($C_i = 1$ and $C_j = 0$) or ($C_i = 0$ and $C_j = 1$);
- the edges are eventually weighted by $W_{ij}$ (st: $W_{ii} = 0$, $W_{ij} = W_{ji} \geq 0$).
A bipartite graph is a graph $\mathcal{G} = (V, E, W, C)$ such that:

- the vertices $V = \{x_1, \ldots, x_n\}$ are split into two sets (labeled with $C_i = 0$ or $C_i = 1$);
- the edges $E \subset V \times V$ are such that if $(x_i, x_j) \in E \Rightarrow$ either $(C_i = 1$ and $C_j = 0)$ or $(C_i = 0$ and $C_j = 1)$;
- the edges are eventually weighted by $W_{ij}$ (st: $W_{ii} = 0$, $W_{ij} = W_{ji} \geq 0$).

Examples of such data:

- very frequently used in recommendation systems (persons liking pages in facebook, persons buying objects...)
- authorship networks (persons and articles)
- affiliation networks (persons and firms)
- ...
Most frequent approaches are based on projected graphs, $G^0$ and $G^1$ st:

- $V^0 = \{x_i \in G : C_i = 0\}$
- $(x_i, x_j) \in E^0 \iff \begin{cases} x_i, x_j \in V^0 \\ \exists x_k \notin V^0 : \{(x_i, x_k) \in E \text{ and } (x_k, x_j) \in E\} \end{cases}$
- $W_{ij}^0 = \{x_k \notin V^0 : (x_i, x_k) \in E \text{ and } (x_k, x_j) \in E\}$

But not useful to understand the relations between the two types of nodes...
Most frequent approaches are based on projected graphs, $\mathcal{G}^0$ and $\mathcal{G}^1$ st:

- $V^0 = \{ x_i \in \mathcal{G} : C_i = 0 \}$
- $(x_i, x_j) \in E^0 \iff \begin{cases} x_i, x_j \in V^0 \\ \exists x_k \notin V^0 : (x_i, x_k) \in E \text{ and } (x_k, x_j) \in E \end{cases}$
- $W^0_{ij} = \{ x_k \notin V^0 : (x_i, x_k) \in E \text{ and } (x_k, x_j) \in E \}$

But not useful to understand the relations between the two types of nodes...
Similarities in the projected graph.

simulated bipartite graph:
- nodes (labeled either 0 or 1) belong to 2 (densely connected) groups;
- edges are generated independently with a given probability: nodes within the same groups have a high probability to be connected and nodes between two groups have a low probability to be connected.

Bipartite graph:
Simulated bipartite graph:

- nodes (labeled either 0 or 1) belong to 2 (densely connected) groups;
- edges are generated independently with a given probability: nodes within the same groups have a high probability to be connected and nodes between two groups have a low probability to be connected.

Projected graphs:
Similarities in the projected graph:

simulated bipartite graph:

- Length of shortest paths
- (based on) number of common neighbors
bipartite graphs are:

- dissimilarity data;
- and data describing relations between two sets of objects (like contingency tables)
bipartite graphs are:

- dissimilarity data;
- and data describing relations between two sets of objects (like contingency tables)

⇒ adapt KORRESP: alternatively process nodes of type 0 and 1

pick a node at random and affect it to the closest prototype (dissimilarity SOM based on a bipartite dissimilarity)
bipartite graphs are:

- dissimilarity data;
- and data describing relations between two sets of objects (like contingency tables)

⇒ adapt KORRESP: **alternatively** process nodes of type 0 and 1

1. **pick a node at random and affect it to the closest prototype**
   - (dissimilarity SOM based on a bipartite dissimilarity)

2. **\( \rho_u = \left( \sum_{i: c_i=0} \gamma_{ui}^0 x_i, \sum_{i: c_i=1} \gamma_{ui}^1 x_i \right) \) is updated in two steps:**

   - **when processing a node of type 0** standard dissimilarity update for \( (\gamma_{ui}^0)_i: c_i=0 \)
   - **when processing a node of type 0**

   \[
   \gamma_{ui}^{1,t+1} \leftarrow \frac{\gamma_{ui}^{1,t} + \mu(t)H^t(d(C(x_i), u))\left(\sum_{k \in \mathcal{N}(x_i)} 1_k - \gamma_{ui}^{1,t}\right)}{1 + \mu(t)H^t(d(C(x_i), u))(d_{i0}^0 - 1)}
   \]
a short review of Self-Organizing Maps for non vectorial data

SOM for bipartite graphs

Application
100 randomly generated graphs clustered into 3 groups

- nodes (labeled either 0 or 1) belong to 3 (densely connected) groups with ~ 50 nodes each;
- edges are generated independently with a given probability (high intra-group probability and low inter-group probability)
100 randomly generated graphs clustered into 3 groups

- nodes (labeled either 0 or 1) belong to 3 (densely connected) groups with \( \sim 50 \) nodes each;
- edges are generated independently with a given probability (high intra-group probability and low inter-group probability)

Compared dissimilarities:
- shortest path length;
- dissimilarity based on the number of common neighbors.
the two distances seem to be approximately equivalent
**Results**

Clusters are well organized on the map...

### Clusters

<table>
<thead>
<tr>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Class 1" /></td>
<td><img src="image2" alt="Class 2" /></td>
<td><img src="image3" alt="Class 3" /></td>
<td><img src="image4" alt="Class 4" /></td>
<td><img src="image5" alt="Class 5" /></td>
</tr>
<tr>
<td>Class 6</td>
<td>Class 7</td>
<td>Class 8</td>
<td>Class 9</td>
<td>Class 10</td>
</tr>
<tr>
<td><img src="image6" alt="Class 6" /></td>
<td><img src="image7" alt="Class 7" /></td>
<td><img src="image8" alt="Class 8" /></td>
<td><img src="image9" alt="Class 9" /></td>
<td><img src="image10" alt="Class 10" /></td>
</tr>
<tr>
<td>Class 11</td>
<td>Class 12</td>
<td>Class 13</td>
<td>Class 14</td>
<td>Class 15</td>
</tr>
<tr>
<td><img src="image11" alt="Class 11" /></td>
<td><img src="image12" alt="Class 12" /></td>
<td><img src="image13" alt="Class 13" /></td>
<td><img src="image14" alt="Class 14" /></td>
<td><img src="image15" alt="Class 15" /></td>
</tr>
<tr>
<td>Class 16</td>
<td>Class 17</td>
<td>Class 18</td>
<td>Class 19</td>
<td>Class 20</td>
</tr>
<tr>
<td><img src="image16" alt="Class 16" /></td>
<td><img src="image17" alt="Class 17" /></td>
<td><img src="image18" alt="Class 18" /></td>
<td><img src="image19" alt="Class 19" /></td>
<td><img src="image20" alt="Class 20" /></td>
</tr>
<tr>
<td>Class 21</td>
<td>Class 22</td>
<td>Class 23</td>
<td>Class 24</td>
<td>Class 25</td>
</tr>
<tr>
<td><img src="image21" alt="Class 21" /></td>
<td><img src="image22" alt="Class 22" /></td>
<td><img src="image23" alt="Class 23" /></td>
<td><img src="image24" alt="Class 24" /></td>
<td><img src="image25" alt="Class 25" /></td>
</tr>
</tbody>
</table>
... and clusters contain the two types of nodes

Class 1  
Class 2  
Class 3  
Class 4  
Class 5  
Class 6  
Class 7  
Class 8  
Class 9  
Class 10  
Class 11  
Class 12  
Class 13  
Class 14  
Class 15  
Class 16  
Class 17  
Class 18  
Class 19  
Class 20  
Class 21  
Class 22  
Class 23  
Class 24  
Class 25
Real-world bipartite graph

CAC 40

- **nodes**: CAC 40 firms (40) and board members (838)
- **edges**: membership (975; i.e., probability that an edge exists between a firm and a board member $\sim 2.9\%$)
Real-world bipartite graph

CAC 40

- **nodes**: CAC 40 firms (40) and board members (838)
- **edges**: membership (975; *i.e.*, probability that an edge exists between a firm and a board member $\sim 2.9\%$)

- firms have from 15 to 45 board members (most of them have less than 30 board members)
- most board members are involved in only one firm (more than 700), a few board members are involved in up to 6 firms
Firm maps

- **interesting facts**: central position of Total, position at borders of Arcelor Mittal, EADS, PPR, close positions of Renaud and PSA...

- **issues to work on**: distant positions of Suez Environnement and of GDF Suez, position at border of AXA...
• methods to cluster and display bipartite graphs;
• work in progress to produce a map of CAC40 firms and board members;
• **in development**: integration of additional information regarding board members personal information (e.g., studies...) to improve the map
Thank you for your attention...

... questions?
Analyzing a contingency table with Kohonen maps: a factorial correspondence analysis.  

A unified approach to pattern recognition.  
*Pattern Recognition*, 17(5):575–582.


On-line relational and multiple relational SOM.  
*Neurocomputing*.  
Forthcoming.