A comparison between dissimilarity SOM & kernel SOM for clustering the vertices of a graph

Nathalie Villa-Vialaneix\textsuperscript{(1)}  Fabrice Rossi\textsuperscript{(2)}

\textsuperscript{(1)}Institut de Mathématiques de Toulouse, France - nathalie.villa@math.ups-tlse.fr

\textsuperscript{(2)}Projet AxIS, INRIA Rocquencourt, France

September 5\textit{th}, 2007
Various fields:

- Social networks
- Biology: Protein interactions, Neuronal network.
Modelization through large graphs

Various fields:
- Computer science: World Wide Web, P2P network…
- Social networks
- Biology: Protein interactions, Neuronal network,…
- Business, management: Transportation networks, Industry partnerships,…

**Question:** Understanding the structure of these large graphs
- Clustering: building relevant homogeneous groups;
- Graph drawing: giving a global representation of the graph.
Modelization through large graphs

Various fields:
- Computer science: World Wide Web, P2P network…
- Social networks
- Biology: Protein interactions, Neuronal network,…
- Business, management: Transportation networks, Industry partnerships,…

**Question:** Understanding the structure of these large graphs
- Clustering: building relevant homogeneous groups;
- Graph drawing: giving a global representation of the graph.

Here: **Self-Organizing Map for nonvectorial data.**
Table of contents

1. Self-organizing maps for nonvectorial data
   - Kernel SOM (on line)
   - Dissimilarity SOM (batch)
   - Kernel SOM (batch)

2. Kernel for graphs

3. Examples
   - A toy example
   - Social network
Table of contents

1. Self-organizing maps for nonvectorial data
   - Kernel SOM (on line)
   - Dissimilarity SOM (batch)
   - Kernel SOM (batch)

2. Kernel for graphs

3. Examples
   - A toy example
   - Social network

Nathalie Villa & Fabrice Rossi  WSOM - Sept 2007
The data

We are given:

- **Data**: $x_1, x_2, \ldots, x_n$ from a (possibly) non vectorial space $G$;
We are given:

- **Data**: $x_1, x_2, \ldots, x_n$ from a (possibly) non vectorial space $\mathcal{G}$;
- **Kernel**: $k : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$ such that $k$ is:
  - symmetric: $k(x, x') = k(x', x)$;
  - positive definite: $\sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0$ for all $m \in \mathbb{N}$, all $(x_i)_i \in \mathcal{G}$ and all $(\alpha_i)_i \in \mathbb{R}$. 

Main consequence:

$\exists$ Hilbert space $(\mathcal{H}, \langle \cdot, \cdot \rangle)$ and $\phi : \mathcal{G} \rightarrow \mathcal{H}$ such that:

$k(x, x') = \langle \phi(x), \phi(x') \rangle$. 

Nonvectorial data are mapped onto a vectorial space where a SOM can be performed by the use of $k$ as a scalar product.
The data

We are given:

- **Data**: \(x_1, x_2, \ldots, x_n\) from a (possibly) non vectorial space \(G\);
- **Kernel**: \(k : G \times G \rightarrow \mathbb{R}\) such that \(k\) is:
  - symmetric: \(k(x, x') = k(x', x)\);
  - positive definite: \(\sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0\) for all \(m \in \mathbb{N}\), all \((x_i)_i \in G\) and all \((\alpha_i)_i \in \mathbb{R}\).

**Main consequence:**

\(\exists\) Hilbert space \((\mathcal{H}, \langle ., . \rangle)\) and \(\phi : G \rightarrow \mathcal{H}\) such that:

\[k(x, x') = \langle \phi(x), \phi(x') \rangle.\]
The data

We are given:
- **Data**: $x_1, x_2, \ldots, x_n$ from a (possibly) non vectorial space $G$;
- **Kernel**: $k : G \times G \to \mathbb{R}$ such that $k$ is:
  - symmetric: $k(x, x') = k(x', x)$;
  - positive definite: $\sum_{i,j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \geq 0$ for all $m \in \mathbb{N}$, all $(x_i)_i, (x_j)_j \in G$ and all $(\alpha_i)_i \in \mathbb{R}$.

Main consequence:

\[ k(x, x') = \langle \phi(x), \phi(x') \rangle. \]

Nonvectorial data are mapped onto a vectorial space where a SOM can be performed by the use of $k$ as a scalar product.
Practical computation:

- **Prototype** of neuron $i$ is in $\mathcal{H}$ and is of the form

\[
p_i = \sum_{j=1}^{n} \gamma_{ji} \phi(x_j);
\]
kernel SOM [Lau et al., 2006]: prototypes

Practical computation:

- **Prototype** of neuron $i$ is in $\mathcal{H}$ and is of the form

$$p_i = \sum_{j=1}^{n} \gamma_{ji} \phi(x_j);$$

- $\phi$ is **implicit** as $\forall i, j = 1, \ldots, n$,

$$\|\phi(x_i) - \phi(x_j)\|^2 = k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j);$$
**assignment step:** for $x_l$,

$$
\arg\min_{j=1,...,M} \left( \sum_{i=1}^{n} \gamma_{ij} k(x_l, x_i) - \sum_{i,i'=1}^{n} \gamma_{ij} \gamma_{i'j} k(x_i, x_{i'}) \right)
$$
kernel SOM: practical computation (online)

- **assignment step**: for $x_l$,

$$\arg \min_{j=1,...,M} \left( \sum_{i=1}^{n} \gamma_{ij} k(x_l, x_i) - \sum_{i,i'=1}^{n} \gamma_{ij} \gamma_{i'j} k(x_i, x_{i'}) \right)$$

- **representation step**: $p_i^l = \sum_{j=1}^{n} \gamma_{ji}^l \phi(x_j)$:

$$\gamma_{ji}^l = \gamma_{ji}^{l-1} + \alpha(l) h(f^l(x_l), j) \left( \delta_{il} - \gamma_{ji}^{l-1} \right)$$
Another assumptions for the data set

We are given:

- **Data**: $x_1, x_2, \ldots, x_n$ from a (possibly) non vectorial space $G$;
- **A dissimilarity measure**: $\delta : G \times G \rightarrow \mathbb{R}$ such that:
  - $\delta$ is symmetric: $\delta(x, x') = \delta(x', x)$;
  - $\delta$ is positive: $\delta(x, x') \geq 0$;
  - $\delta(x, x) = 0$. 

Nathalie Villa & Fabrice Rossi  
WSOM - Sept 2007
Another assumptions for the data set

We are given:

- **Data**: \( x_1, x_2, \ldots, x_n \) from a (possibly) non vectorial space \( \mathcal{G} \);
- **A dissimilarity measure**: \( \delta : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R} \) such that:
  - \( \delta \) is symmetric: \( \delta(x, x') = \delta(x', x) \);
  - \( \delta \) is positive: \( \delta(x, x') \geq 0 \);
  - \( \delta(x, x) = 0 \).

Adaptation for SOM:

- The **prototypes** are one of the elements of the data set \( (x_i)_{i=1,...,n} \);
Another assumptions for the data set

We are given:

- **Data**: \(x_1, x_2, \ldots, x_n\) from a (possibly) non vectorial space \( \mathcal{G} \);
- **A dissimilarity measure**: \(\delta : \mathcal{G} \times \mathcal{G} \to \mathbb{R}\) such that:
  - \(\delta\) is symmetric: \(\delta(x, x') = \delta(x', x)\);
  - \(\delta\) is positive: \(\delta(x, x') \geq 0\);
  - \(\delta(x, x) = 0\).

Adaptation for SOM:

- The **prototypes** are one of the elements of the data set \((x_i)_{i=1,\ldots,n}\);
- \(\delta\) is used **instead of the usual distance**.
Dissimilarity SOM: practical computation

- **assignment step**: for $x_i$,
  \[
  \arg \min_{j=1,...,M} \delta(x_i, p_j^{l-1})
  \]

- **representation step**:
  \[
  p_j^l = \arg \min_{x \in (x_i')_{i'=1,...,n}} \sum_{i=1}^{n} h(f^l(x_i), j) \delta(x_i, x)
  \]
Link with kernel SOM

A dissimilarity built from a kernel $k$:

$$
\delta_{\text{mean}}(x, x') = \|\phi(x) - \phi(x')\|^2 \\
= k(x, x) + k(x', x') - 2k(x, x').
$$
A dissimilarity built from a kernel $k$:

$$
\delta_{\text{mean}}(x, x') = \|\phi(x) - \phi(x')\|^2 \\
= k(x, x) + k(x', x') - 2k(x, x').
$$

Consequence: By generalizing dissimilarity SOM to the case where the prototype of neuron $i$ is of the form

$$
p_j = \sum_{i=1}^{n} \gamma_{ji} \phi(x_i);
$$

we can derive a **batch version** of kernel SOM from dissimilarity SOM.
From dissimilarity SOM to batch kernel SOM

**Assignment step**

for \( x_i \),

\[
\text{arg} \min_{j = 1, \ldots, M} \delta(x_i, p_j^{l-1})
\]

**Representation step**

\[
p_j^l = \text{arg} \min_{x \in (x_i')_{i' = 1, \ldots, n}} \sum_{i=1}^n h(f^l(x_i), j) \delta(x_i, x)
\]
From dissimilarity SOM to batch kernel SOM

**Assignment step**

for $x_i$,

$$\text{arg min}_{j=1,...,M} \left\| x_i - \sum_{i=1}^{n} \gamma_{ji} \phi(x_i) \right\|$$

**Representation step**

$$\gamma'_j = \text{arg min}_{\gamma \in \mathbb{R}^n} \sum_{i=1}^{n} h(f_l(x_i), j) \left\| x_i - \sum_{l'=1}^{n} \gamma_{l'} \phi(x_{i'}) \right\|^2$$
From dissimilarity SOM to batch kernel SOM

Assignment step

for $x_i$,

$$\arg \min_{j=1,...,M} \sum_{u,u'=1}^{n} \gamma_{ju} \gamma_{ju'} k(x_u, x_{u'}) - 2 \sum_{u=1}^{n} \gamma_{ju} k(x_u, x_i)$$

Representation step

$$\gamma_{ji}^l = \frac{h(f^l(x_i), j))}{\sum_{i'=1}^{n} h(f^l(x_{i'}, j))}$$
Table of contents

1. Self-organizing maps for nonvectorial data
   - Kernel SOM (on line)
   - Dissimilarity SOM (batch)
   - Kernel SOM (batch)

2. Kernel for graphs

3. Examples
   - A toy example
   - Social network
Laplacian and diffusion kernel

Definitions

For a graph with vertices $V = \{x_1, \ldots, x_n\}$ having positive weights $(w_{i,j})_{i,j=1,\ldots,n}$ such that, for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and $d_i = \sum_{j=1}^n w_{i,j}$, 

- **Laplacian**: $L = (L_{i,j})_{i,j=1,\ldots,n}$ where
  
  $$L_{i,j} = \begin{cases} 
  -w_{i,j} & \text{if } i \neq j \\
  d_i & \text{if } i = j 
  \end{cases}.$$
Laplacian and diffusion kernel

Definitions

For a graph with vertices \( V = \{x_1, \ldots, x_n\} \) having positive weights \((w_{i,j})_{i,j=1,\ldots,n}\) such that, for all \(i, j = 1, \ldots, n\), \(w_{i,j} = w_{j,i}\) and \(d_i = \sum_{j=1}^{n} w_{i,j}\),

- **Laplacian**: \( L = (L_{i,j})_{i,j=1,\ldots,n} \) where

\[
L_{i,j} = \begin{cases} 
-w_{i,j} & \text{if } i \neq j \\
 d_i & \text{if } i = j 
\end{cases}
\]

- **Regularization: the diffusion matrix**: for \(\beta > 0\), \(K^\beta = e^{-\beta L}\).
For a graph with vertices $V = \{x_1, \ldots, x_n\}$ having positive weights $(w_{i,j})_{i,j=1,\ldots,n}$ such that, for all $i, j = 1, \ldots, n$, $w_{i,j} = w_{j,i}$ and $d_i = \sum_{j=1}^{n} w_{i,j}$,

- **Laplacian:** $L = (L_{i,j})_{i,j=1,\ldots,n}$ where

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } i \neq j \\ d_i & \text{if } i = j \end{cases};$$

- **Regularization:** the diffusion matrix: for $\beta > 0$, $K^\beta = e^{-\beta L}$.

$\Rightarrow$

$$k^\beta : V \times V \rightarrow \mathbb{R}$$

$$(x_i, x_j) \rightarrow K_{i,j}^\beta$$

is the **diffusion kernel** (or heat kernel).
Properties

1. **Diffusion on the graph**: $k^\beta(x_i, x_j) \approx$ quantity of energy accumulated in $x_j$ after a given time if energy is injected in $x_i$ at time 0 and if diffusion is done along the edges. 
   
   $\beta \approx$ intensity of diffusion;
Properties

1. **Diffusion on the graph**: $k^\beta(x_i, x_j) \approx$ quantity of energy accumulated in $x_j$ after a given time if energy is injected in $x_i$ at time 0 and if diffusion is done along the edges. 
   $\beta \approx$ intensity of diffusion;

2. **Regularization operator**: for $u \in \mathbb{R}^n \sim V$, $u^T K^\beta u$ is higher for vectors $u$ that vary a lot over “close” vertices of the graph. 
   $\beta \approx$ intensity of regularization (for small $\beta$, direct neighbors are more important);
Self-organizing maps for nonvectorial data
- Kernel SOM (on line)
- Dissimilarity SOM (batch)
- Kernel SOM (batch)

Kernel for graphs

Examples
- A toy example
- Social network
Simulated data

Random generation

50 non-weighted graphs having:

- 5 cliques \((C_i)_{i=1,\ldots,5}\) with \((n_i)_{i=1,\ldots,5}\) vertices where \(n_i \sim \mathcal{P}(50)\);
- for \(i = 1, \ldots, 5\), \(l_i\) links between \(C_i\) and the other vertices \(l_i \sim \mathcal{U}(\{1, \ldots, 100n_i\})\).
Simulated data

Random generation

50 non-weighted graphs having:
1. 5 cliques \((C_i)_{i=1,\ldots,5}\) with \((n_i)_{i=1,\ldots,5}\) vertices where \(n_i \sim \mathcal{P}(50)\);
2. for \(i = 1, \ldots, 5\), \(l_i\) links between \(C_i\) and the other vertices \(l_i \sim \mathcal{U}([1, \ldots, 100n_i])\).

Example:

Number of vertices: 237  
Total of degrees: 12 924  
Diameter: 2  
Density \(\approx 0.46\)
## Compared results ($\beta = 0.1$)

<table>
<thead>
<tr>
<th></th>
<th>k-SOM (online)</th>
<th>d-SOM</th>
<th>k-SOM (batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean energy</td>
<td>0.10</td>
<td>0.13</td>
<td>0.10</td>
</tr>
<tr>
<td>Mean nb of classes</td>
<td>8.04</td>
<td>9</td>
<td>6.56</td>
</tr>
<tr>
<td>Mean % of good classif.</td>
<td>79.84</td>
<td>77.89</td>
<td>94.34</td>
</tr>
<tr>
<td>Time of computation</td>
<td>260</td>
<td>80</td>
<td>20</td>
</tr>
</tbody>
</table>
Compared results ($\beta = 0.1$)

<table>
<thead>
<tr>
<th></th>
<th>k-SOM (online)</th>
<th>d-SOM</th>
<th>k-SOM (batch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean energy</td>
<td>0.10</td>
<td>0.13</td>
<td>0.10</td>
</tr>
<tr>
<td>Mean nb of classes</td>
<td>8.04</td>
<td>9</td>
<td>6.56</td>
</tr>
<tr>
<td>Mean % of good classif.</td>
<td>79.84</td>
<td>77.89</td>
<td>94.34</td>
</tr>
<tr>
<td>Time of computation (min)</td>
<td>260</td>
<td>80</td>
<td>20</td>
</tr>
</tbody>
</table>

Nathalie Villa & Fabrice Rossi  
WSOM - Sept 2007
A medieval social network [Villa et al., 2007]

Graph built from a large corpus of medieval contracts

From a corpus of 1000 agrarian contracts made in South West of France (1250-1350), we built a weighted graph:

- **vertices**: peasants found in the contracts;
- **edges**: number of contracts where two peasants are mentionned together.

Example:
- Number of vertices: 515
- Number of edges: 4193
- Total of weights: 40 329
- Diameter: 10
- Density: 2.2%
A medieval social network [Villa et al., 2007]

Graph built from a large corpus of medieval contracts

From a corpus of 1000 agrarian contracts made in South West of France (1250-1350), we built a weighted graph:

- **vertices**: peasants found in the contracts;
- **edges**: number of contracts where two peasants are mentionned together.

Example:

Number of vertices: 515
Number of edges: 4193
Total of weights: 40 329
Diameter: 10
Density: 2.2%
Results on a $7 \times 7$ rectangular map

Nathalie Villa & Fabrice Rossi

WSOM - Sept 2007
Results on a $7 \times 7$ rectangular map