Theoretical and Applied Aspects of the Self-Organizing Maps

Marie Cottrell[1] & Madalina Olteanu[1] & Fabrice Rossi[1] & Nathalie Villa-Vialaneix[2]

[1] Université Paris 1 Panthéon-Sorbonne, Laboratoire SAMM
[2] INRA Toulouse, Unité MIAT

January 6, 2016 - Conference WSOM

The data space

Data belong to a subset $\mathscr{X}$ of an Euclidean space (typically $\mathbb{R}^p$).
For some results, we need to assume that the subset is bounded and convex.
Two different settings from a theoretical point of view :

- ***continuous setting*** : the input space $\mathscr{X}$ is modeled by a probability distribution with density function f ;
- or
- ***discrete setting*** : the input space $\mathscr{X}$ has N data points $x_1, \ldots, x_N$ in $\mathbb{R}^p$ (Here *discrete setting* means a *finite* subset of the input space).

The data can be stored or available on-line.

Neighborhood structure

K units on a regular lattice (string : 1-dim or grid : 2-dim).
If $\mathcal{K} = \{1, \dots, K\}$, neighborhood function h is defined on $\mathcal{K} \times \mathcal{K}$.
If it is time-dependent, it will be denoted by $h(t)$.

- $h_{kk} = 1$, h symmetric
- h_{kl} depends only on the distance $\text{dist}(k, l)$ between units k and l on the lattice and decreases with increasing distance.

Several choices, the most classical : the **step function** with value 1 if the distance between k and l is less than a specific radius (this radius can decrease with time), and 0 otherwise.
Another very classical choice is a **Gaussian-shaped function**

$$h_{kl}(t) = exp\left(-\frac{\text{dist}^2(k, l)}{2\sigma^2(t)}\right),$$

where $\sigma^2(t)$ can decrease over time to reduce the intensity and the scope of the neighborhood relations.

Examples of neighborhood functions

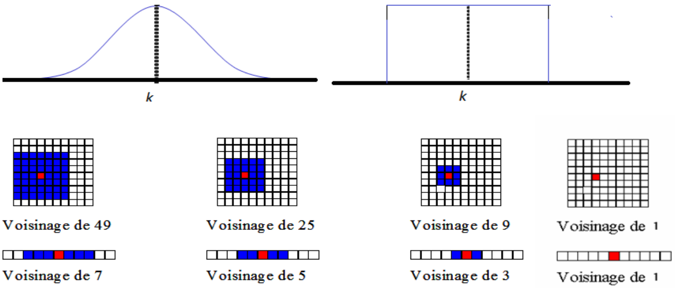


FIGURE – Neighborhood functions

On-line SOM, continuous or discrete settings

[Kohonen, 1982, 1995]

A prototype $m_k \in \mathcal{R}^p$ is attached to each unit k, initial values of the prototypes are chosen at random and denoted by $m(0) = (m_1(0), \dots, m_K(0))$. The SOM algorithm (**on-line stochastic version**) is defined as follows :

1. At time t, a data point x is **randomly drawn** (according to the density function f or in the finite set $\mathcal{X}$),
2. **Best matching unit definition**

$$c^t(x) = \arg \min_{k \in \{1,\dots,K\}} \|x - m_k(t)\|^2, \quad (1)$$

3. **Prototypes update**

$$m_k(t+1) = m_k(t) + \varepsilon(t) h_{kc^t(x)}(t)(x - m_k(t)), \quad (2)$$

where $\varepsilon(t)$ is a learning rate (positive, <1, constant or decreasing).

Clustering

- Cluster C_k : set of inputs closer to m_k than to any other one
- Partition (or Voronoï tesselation) with neighborhood structure between the clusters.

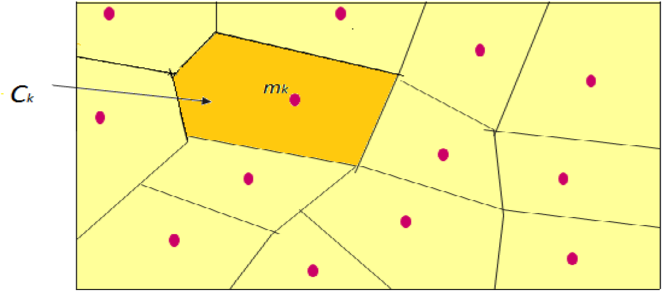


FIGURE – Voronoï tesselation

Data $x \in C_k \iff m_k$ is the winning prototype

SOM properties

- **Quantization property** : The prototypes represent the data space as accurately as possible, as do other quantization algorithms.
- **Self-organization property** : The prototypes preserve the topology of the data : *close inputs* belong to the *same cluster* (as in any clustering algorithm) or to *neighbor clusters*.

To get a better quantization, the learning rate ε decreases with time as well as the scope of the neighborhood function h.

Theoretical concerns

- Algorithm very easy to define and to use.
- Large amount of works and empirical evidences.
- **But** many theoretical properties without complete proof and open problems [Cottrell et al., 1998] and [Fort, 2005].

When t tends to $+\infty$, the $\mathbb{R}^p$-valued stochastic processes $(m_k(t))_{k=1,\dots,K}$ can have : *oscillations*, *explosion* to infinity, *CV in distribution* to an equilibrium process, *CV in distribution or almost sure* to a finite set of points in $\mathbb{R}^p$, etc.

- Is the algorithm **convergent** in distribution or almost surely, when t tends to $+\infty$?
- What happens when ε is **constant** ? when it **decreases** ?
- If a **limit state** exists, is it stable ?
- How to characterize the **organization** ?

Mathematical tools

- ***The Markov Chain theory*** for constant ε and h : to study the convergence and the limit states.
  - If the algorithm converges in distribution, this limit is an **invariant measure** for the Markov Chain ;
  - If there is strong organization, it has to be associated to an **absorbing class**.
- The ***Ordinary Differential Equation method*** (ODE)
  Equation (2) for each $k \in \mathcal{K}$ can be written in a vector form :

$$m(t+1) = m(t) - \varepsilon(t)\Phi(x, m(t)), \tag{3}$$

where Φ is a stochastic term.
Then, the ODE (Ordinary Differential Equation) which describes the **mean behavior** of the process is

$$\frac{dm}{dt} = -\phi(m), \tag{4}$$

where $\phi(m)$ is the expectation of $\Phi(., m)$.

ODE associated to SOM

The k^{th}−component of ϕ is

$$\phi_k(m) = \sum_{j=1}^{K} h_{kj} \int_{C_j} (x - m_k) f(x)\, dx, \tag{5}$$

for the *continuous setting* or

$$\phi_k(m) = \frac{1}{N} \sum_{j=1}^{K} h_{kj} \sum_{x_i \in C_j} (x_i - m_k) = \frac{1}{N} \sum_{i=1}^{N} h_{kc(x_i)} (x_i - m_k), \tag{6}$$

for the *discrete setting*.
Possible limit states are **solutions of equation**

$$\phi(m) = 0.$$

If the zeros of function ϕ are **minima** of a function (*energy* function), one can apply the gradient descent methods.

Mathematical tools and difficulties

- The ***Robbins-Monro algorithm*** theory is used when the learning rate decreases under conditions

$$\sum_t \varepsilon(t) = +\infty \text{ and } \sum_t \varepsilon(t)^2 < +\infty. \quad (7)$$

Some remarks explain why the original SOM algorithm is difficult to study :

- For $p > 1$, it is not possible to define ***any absorbing class*** which could be an organized state ;
- Although $m(t)$ can be written down as a classical stochastic process, [Erwin et al., 1992a, Erwin et al., 1992b] have shown that it does not correspond to an energy function, that it is ***not a gradient descent algorithm*** in the *continuous setting* ;
- Finally, no demonstration takes into account ***the variation of the neighborhood function***. All the existing results are valid for a fixed scope and intensity of the function h.

Simplest case : $p = 1, \mathscr{X} = [0,1]$, string lattice, uniform density, constant ε

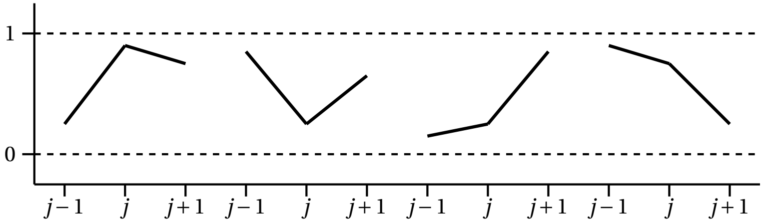
[Cottrell, Fort, 1987]

Theorem

If ε is a constant <1/2 and if the neighborhood are $\{k-1, k, k+1\}$,

- *The number of badly ordered triplets is a **decreasing functional**;*
- *The set of ordered sequences (increasing or decreasing sequences, i.e. organized ones) is an **absorbing class**;*
- *The hitting time of the absorbing class is **almost surely finite**;*
- *The process $m(t)$ converges in distribution to a monotonous stationary distribution which depends on ε.*

Why is the number of non ordered triplets decreasing



Four examples of triplets of prototypes (m_{j-1}, m_j, m_{j+1}). The neighbors of j are $j-1$ and $j+1$. The values of the prototypes are on the y-axis, in $[0,1]$.

On the left, the first two triplets are not ordered. SOM will order them with a strictly positive probability.

At right, the last two triplets are well ordered and SOM will never disorder them.

Simplest case : $p = 1, \mathscr{X} = [0,1]$, string lattice, uniform density, decreasing ε

Theorem

If $\varepsilon \longrightarrow 0$ and satisfies the Robbins-Monro conditions

$$\sum_t \varepsilon(t) = +\infty \text{ and } \sum_t \varepsilon(t)^2 < +\infty. \tag{8}$$

after ordering, *the process $m(t)$ a.s. converges towards a constant monotonous solution of an explicit linear system.*

General one-dimensional case, organization

[Fort, 2005, Cottrell et al., 1998]

Hypothesis on the data distribution and/or the neighborhood function are relaxed. The density is no longer uniform.

Theorem

One assumes that the setting is continuous and that the neighborhood function is strictly decreasing from some distance between the units.

- *The set of ordered sequences (increasing or decreasing sequences, i.e. organized ones) is an* ***absorbing class****;*
- *If ε is constant, the hitting time of the absorbing class is* ***almost surely finite****.*

General one-dimensional case, convergence

Theorem

*One assumes that the setting is continuous, **the density is log-concave**, the neighborhood function is time-independent and strictly decreasing from some distance between the units.*

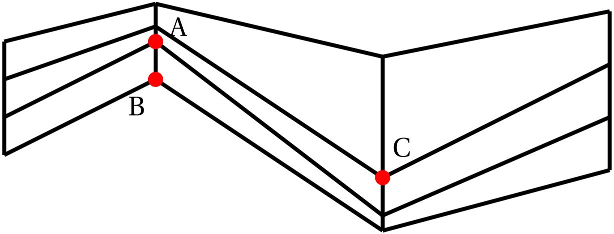
- *If the initial state is ordered, there exists a **unique stable equilibrium** point (denoted by x^*);*
- *If ε **is constant** and the initial disposition is ordered, there exists an invariant distribution which depends on ε and which concentrates on the Dirac measure on x^* when $\varepsilon \longrightarrow 0$;*
- *If $\varepsilon(t)$ satisfies the Robbins-Monro conditions (8) and if the initial state is ordered, then $m(t)$ is **almost surely convergent** towards this unique equilibrium point x^*.*

Two remarks about the general one-dimensional case

- The hypotheses on the density are not very restrictive, but some ***important distributions***, such as the χ^2 or the power distribution, ***do not fulfill*** them.
- Even if the one-dimensional case is more or less well-known, ***nothing is proved*** either about the choice of a ***decreasing function*** for $\varepsilon(t)$ to ensure ordering and convergence simultaneously, or for the case of ***decreasing neighborhood function***.

General multidimensional case, continuous setting

Unfortunately, **it is not possible to find absorbing classes** when the dimension is larger than 1.
For example, in dimension 2, with 8 neighbors, the figure shows the x- and y-coordinates are ordered but that it is possible (with positive probability) **to disorder the prototypes**.



A is a neighbor of C, but B is not a neighbor of C. If C is very often the best matching unit, B is never updated, while A becomes closer and closer to C. Finally, the y coordinate of A becomes smaller than that of B and the **disposition is disordered.**

General multidimensional case, continuous setting

Let p be the data dimension. Assume that h and ε are constant. Sadeghi (2001) proves

Theorem

If the probability density function f is positive on an interval, the algorithm weakly converges to a unique probability distribution which depends on ε.

Assuming $p = 2$ and denoting by F^{++} the set of the prototypes with simultaneously increasing coordinates, these two apparently contradictory results hold.

Theorem

- *for a constant ε and very general hypotheses on the density f, the hitting time of F^{++} is finite with a positive probability (Flanagan, 1996),*
- *but in the 8-neighbor setting, the exit time is also finite with positive probability (Fort & Pages, 1995).*

General multidimensional case, discrete setting

[Ritter et al., 1992]

For the continuous setting, we know that **SOM is not a gradient descent algorithm** (Erwinn).
But the discrete setting is quite different, since the **stochastic process** $m(t)$ **derives from an energy function** (h is not time-dependent).

- It is a very important case, since for the applications, the data are discrete (as for *data mining, data clustering*).
- Then for discrete setting, SOM is a **gradient descent process** associated to

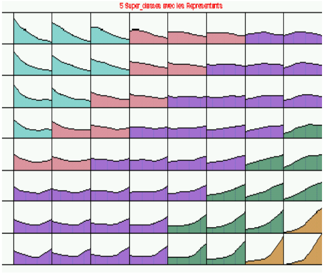
$$E(m) = \frac{1}{2N} \sum_{i=1}^{N} \sum_{k=1}^{K} h_{kc(x_i)} \|m_k - x_i\|^2. \tag{9}$$

Extended distortion

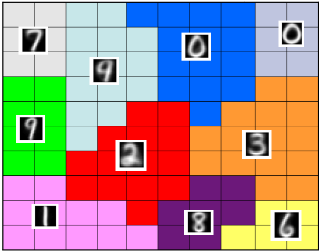
- This result does not ensure the convergence : the *gradient of the energy function is not continuous* on the boundaries of the clusters.
- This energy combines two criteria : **Clustering** criterium and correct **Organisation** criterium.
- In the 0-neighbor setting, SOM reduces to the VQ process, the energy reduces to

$$E(m) = \frac{1}{2N}\sum_{i=1}^{N} \|m_{c(x_i)} - x_i\|^2.$$

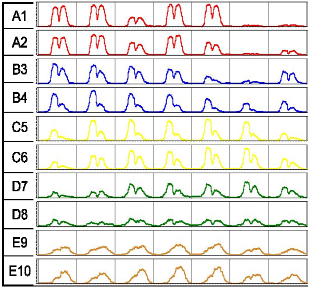
  , the gradient is continuous. But the algorithm converges to one of the local minima,
- This **energy function** is called ***extended distortion***.



7 census of the French Rhône Valley districts from 1936 to 1990), 1783 districts

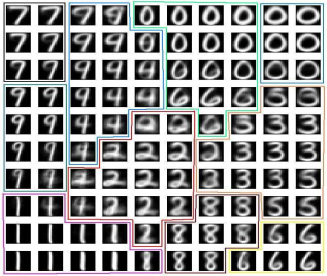


The characters are transformed into 256-dim vectors.

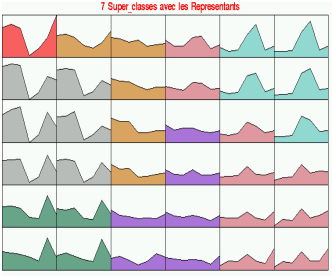
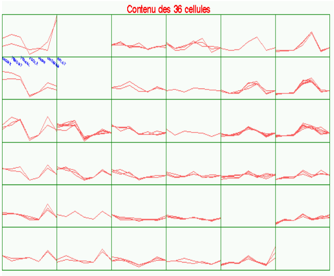
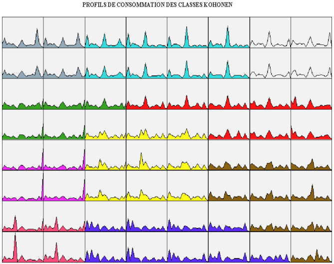
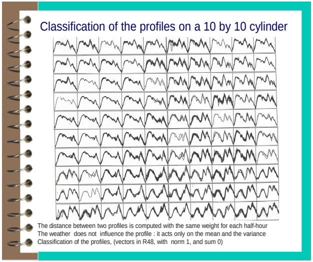


During a week, at each quarter-hour 1 if he works, 0 if not

So the dimension 4×24×7=672



10 super-classes. digit 5 is missing, it belongs to the same class than digit 3.

<table>
<tr>
<td>
</td>
<td>96 ciountries in 1996
7 economic indexes
Super-classes</td>
<td>
</td>
<td>Clustering of all the data
At left poor countries
At right rich ones</td>
</tr>
<tr>
<td>
</td>
<td>Consumptions of Canadian households according to 20 categories</td>
<td>
</td>
<td>Electricity consumptions observed each half an hour over 24 hours</td>
</tr>
</table>

Batch SOM

[Kohonen, 1995]

The possible limit states are solutions of the ODE equation $\phi(m) = 0$, so it is natural to compute its solutions.

For the **continuous setting**

$$m_k^* = \frac{\sum_{j=1}^K h_{kj} \int_{C_j} x f(x)\, dx}{\sum_{j=1}^K h_{kj} \int_{C_j} f(x)\, dx}.$$

In the **discrete setting**, the analogous is

$$m_k^* = \frac{\sum_{j=1}^K h_{kj} \sum_{x_i \in C_j} x_i}{\sum_{j=1}^K h_{kj} |C_j|} = \frac{\sum_{i=1}^N h_{kc(x_i)} x_i}{\sum_{i=1}^N h_{kc(x_i)}}.$$

The limit prototypes m_k^* are **the weighted means** of all the inputs which belong to the cluster C_k or to its neighboring clusters. The weights are given by the neighborhood function h.

Definition of the Batch SOM

Random initial values of the prototypes;

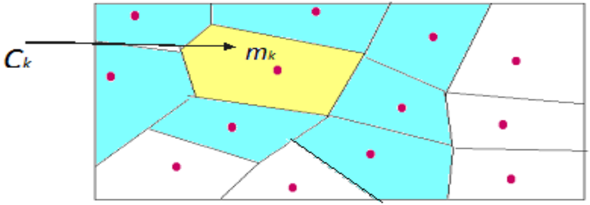
- **Construction of all the clusters (nearest neighbors)**;
- **Update of all the prototypes**

$$m_k(t+1) = \frac{\sum_{j=1}^{K} h_{kj}(t) \int_{C_{j(m_k(t))}} x f(x) dx}{\sum_{j=1}^{K} h_{kj}(t) \int_{C_{j(m_k(t))}} f(x) dx} \tag{10}$$

for the continuous setting, and

$$m_k(t+1) = \frac{\sum_{i=1}^{N} h_{kc^t(x_i)}(t) x_i}{\sum_{i=1}^{N} h_{kc^t(x_i)}(t)} \tag{11}$$

for the discrete case.



Theoretical properties of Batch SOM

- Batch SOM is **quasi-Newtonian algorithm** associated to the **extended distortion** and converges to a local minimum of it.
- In the 0-neighbor setting, Batch SOM reduces to Forgy process (***k*-means**), which converges towards a local minimum of the distortion.

Relations between these 4 clustering algorithms

	Stochastic	Deterministic
0 neighbor	VQ, SCL	Forgy, moving centers
With neighbors	**SOM**	Batch SOM

- SOM and Batch SOM preserve the data topology : **neighbor data belong to the same cluster or to neighbor clusters** ;
- Hence the **visualization** properties of the **Kohonen maps**, while 0-neigbor algorithms (Forgy and VQ) have not ;
- SOM depends very little on the initialization, while Batch SOM is very **sensitive** ;
- Batch SOM is **deterministic** and often preferred for industrial applications.

Hard assignment in the Heskes's rule

[Heskes, 1999]

- In the continuous setting, *the on-line SOM is not a gradient algorithm*, and in the discrete setting, the *the gradient of the energy function is not continuous*
- To overcome these problems, Heskes **modifies the rule for computing the best matching unit** (BMU) in the on-line version of the SOM
- Equation (1) becomes

$$c^t(x) = \arg\min_{k\in\{1,\dots,K\}} \sum_{j=1}^{K} h_{kj}(t)\|x - m_k(t)\|^2 \tag{12}$$

- Then, this modified SOM is a gradient descent process of the *energy function*

$$E(m) = \frac{1}{2}\sum_{j=1}^{K}\sum_{k=1}^{K} h_{kj}(t)\int_{x\in C_j(m)} \|x - m_k(t)\|^2 f(x)\,dx \tag{13}$$

.

Comparison of both rules

The regularity properties of the energy function and of its gradient are summarized as discussed in Heskes, 1999.

	Discrete setting	Continuous setting
Kohonen rule	Energy : discontinuous (but finite on V) Gradient : discontinuous (infinite on V)	Energy : continuous Gradient : discontinuous
Heskes rule	Energy : continuous Gradient : discontinuous (finite on V)	**Energy : continuous** **Gradient : continuous**

Soft Topographic Mapping (STM)

[Heskes, 1999, Graepel, 1998]

- The energy function in the discrete SOM can be written :

$$E(m,c) = \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik} \sum_{j=1}^{K} h_{kj}(t)\|m_j(t) - x_i\|^2$$

  where c_{ik} is equal to 1 iif x_i belongs to cluster k.
- **Crisp assignment is smoothed** by considering $c_{ik} \geq 0$ such that $\sum_{k=1}^{K} c_{ik} = 1$, so that c_{ik} is the $\mathbb{P}(x_i \in C_k)$.
- **Deterministic annealing scheme** to avoid the local minima : the energy function is transformed into a "*free energy*" cost function,

$$F(m,c,\beta) = E(m,c) - \frac{1}{\beta}S(c) \ ,$$

  where β is the **annealing parameter**.

Soft Topographic Mapping (STM)

- For fixed β and h, the minimization of the free energy leads to iterating over two steps

$$\mathbb{P}(x_i \in C_k) = \frac{\exp(-\beta e_{ik})}{\sum_{j=1}^K \exp(-\beta e_{ij})}, \tag{14}$$

where $e_{ik} = \frac{1}{2}\sum_{j=1}^K h_{jk}(t)\|x_i - m_j(t)\|^2$

$$m_k(t+1) = \frac{\sum_{i=1}^N x_i \sum_{j=1}^K h_{jk}(t)\mathbb{P}(x_i \in C_j)}{\sum_{i=1}^N \sum_{j=1}^K h_{jk}(t)\mathbb{P}(x_i \in C_j)} \tag{15}$$

- If $\beta \approx 0$, there is *only one global minimum* computed by gradient descent or EM schemes
- When $\beta \to +\infty$, the *free energy* tends to be $E(m,c)$
- *Deterministic annealing* minimizes the free energy, starting from a small β, to finally get (with increasing β) an approximation of **the global minimum of** $E(m,c)$
- When $\beta \to +\infty$, the **classical batch SOM is retrieved, and most of the local minima are avoided**

Comparison between the SOM-inspired probabilistic models

Consider a **mixture of K Gaussian distributions**, centered on the prototypes, with equal covariance matrix

- In *Regularized EM*, [Heskes, 2001], the constraint is enforced by a **regularization term on the data space** distribution
- In *Variational EM*, [Verbeek et al., 2005] the constraint is induced at **the latent variable level** (via approximating $p(Z|X,\Theta)$) by a smooth distribution)
- In *Generative Topographic Mapping*, the constraint is induced **on the data space** distribution, because the centers of the Gaussian distributions are obtained by mapping a fixed grid to the data space via a nonlinear smooth mapping

All the probabilistic variants enable **missing data analysis** and easy extensions to **non numerical data**

Extensions for non numerical data

Until this point : the data were described by **numerical variables**

SOM algorithm may be adapted to :

- survey data (variables are **qualitative**, they are answers to questions with multiple choices) ;
- data described by a **dissimilarity matrix** or a **kernel** (observations are known by their pairwise relations : graphs, qualitative time series, ...)

Contingency Tables, KORRESP algorithm

Cottrell et Letrémy, 1993, 2005

The data : a contingency table (two qualitative variables) $\mathbf{T} = (t_{ij})$ with p rows and q columns.

- ***Scaling*** of the rows and of the columns as in Factorial Correspondence Analysis
  The **scaled contingency table** denoted by T^{sc} :

$$t^{sc}_{i,j} = \frac{t_{i,j}}{\sqrt{t_{i.}t_{.j}}}.$$

- ***Definition*** of an extended data table X by associating to each row the most probable column and to each column the most probable row
- ***Simultaneous classification*** of the rows and of the columns onto a Kohonen map, by using the extended data table X as input for the SOM algorithm

Contingency table, KORRESP Algorithm

$$
\mathbf{X} = \begin{array}{ccc}
\text{columns} & \text{rows} & \\
\textit{scaled row} & \textit{most probable column} & \text{rows} \\
\textit{most probable row} & \textit{scaled column} & \text{columns}
\end{array}
$$

- **Assignment** uses the scaled rows or columns
- **Prototypes update** uses the extended rows or columns
- **Alternating draw** of a row or of a column

Generalization to general survey data

Three kinds of data

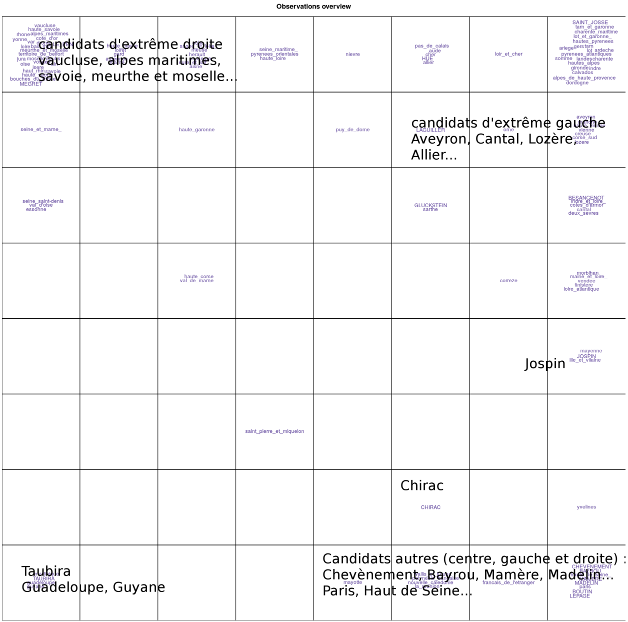
- **Simple contingency table** crossing two questions
- **Burt Table**, i.e. full contingency table for any number of questions
- **Complete disjunctive table** that contains the answers of all the individuals

KORRESP deals with all these kinds of tables, viewed as "contingency tables".

The **scaling step** allows us to use the Euclidean distance instead of the χ^2 distance and to take into account the weighting as in FCA.

After convergence, rows and columns items are simultaneously classified as in FCA, but on **only one map**.

Example : Presidential elections 2002



Non numerical data, Median SOM

[Kohohen and Somervuo, 1998]

The data : are described by a symmetric (dis)similarity matrix $\mathbf{D} = (\delta(x_i, x_j))_{i,j=1,\ldots,N}$, in a discrete setting. Observations (x_i) do not necessarily belong to a vector space.

Median SOM : optimal prototypes are restricted to the data points instead of $\mathcal{X}$.

Discrete optimization scheme, **in batch mode** :

1. **Assignment** of *all* data to their best matching units : $c(x_i) = \arg\min_k \delta(x_i, m_k(t))$;
2. **Update** of all the prototypes within the dataset $m_k(t) = \arg\min_{x_i} \sum_{j=1}^{N} h_{c(x_j)k}(t)\delta(x_i, x_j)$.

- The algorithm explores a finite set so it **is convergent** to a local minimum of the energy function.
- Strong limitations
  - Restriction of the prototypes to the dataset ;
  - Large computational cost.

Dissimilarity data,"relational" SOM

[Hammer and Hasenfuss, 2010, Olteanu and Villa-Vialaneix, 2015a, Rossi, 2014]

If the data are described by a (dis)similarity matrix $\mathbf{D} = (\delta(x_i, x_j))_{i,j=1,\ldots,n}$, [Goldfarb, 1984] shows that the context is **pseudo-Euclidean** :

Theorem

There exist two Euclidean spaces $\mathcal{E}_1$ *and* $\mathcal{E}_2$ *and* $\psi_1 : \{x_i\} \rightarrow \mathcal{E}_1$, $\psi_2 : \{x_i\} \rightarrow \mathcal{E}_2$ *such that*

$$\delta(x_i, x_j) = \|\psi_1(x_i) - \psi_1(x_j)\|^2_{\mathcal{E}_1} - \|\psi_2(x_i) - \psi_2(x_j)\|^2_{\mathcal{E}_2}.$$

Dissimilarity data, "relational" SOM

Principle : to use the data representation in $\mathcal{E} = \mathcal{E}_1 \otimes \mathcal{E}_2$, where $\psi(x) = (\psi_1(x), \psi_2(x))$.

- **The prototypes** are expressed as convex combinations of the $(\psi(x_i))$:

$$m_k(t) = \sum_{i=1}^{N} \gamma_{ki}^t \psi(x_i)$$

where $\gamma_{ki}^t \geq 0$ and $\sum_i \gamma_{ki}^t = 1$

- **The distance** : $\|\psi(x_i) - m_k(t)\|_{\mathcal{E}}^2$ can be expressed with $\mathbf{D}$ and the γ

$$\left(\mathbf{D}\gamma_k^t\right)_i - \frac{1}{2}(\gamma_k^t)^T \mathbf{D}\gamma_k^t$$

For the on-line framework,

- **The prototypes update** concerns the coordinates (γ_k) only :

$$\gamma_k^{t+1} = \gamma_k^t + \varepsilon(t) h_{kc^t(x_i)}(t) \left(\mathbf{1}_i - \gamma_k^t\right) \qquad (16)$$

where x_i is the current observation and $\mathbf{1}_{il} = 1$ iif $l = i$

Dissimilarity data, "Relational Batch SOM

In the batch framework, the prototypes update concerns the coordinates (γ_k) only :

$$m_k(t+1) = \sum_{i=1}^{N} \frac{h_{kc^t(x_i)}(t)}{\sum_{j=1}^{N} h_{kc^t(x_j)}(t)} \psi(x_i) \Leftrightarrow \gamma_{ki}^{t+1} = \frac{h_{kc^t(x_i)}(t)}{\sum_{j=1}^{N} h_{kc^t(x_j)}(t)} \quad (17)$$

For the γ, the updating step is identical to the ***original SOM*** or to the ***original Batch SOM algorithm***.

If the dissimilarities are in fact given by Euclidean distances between data points in $\mathbb{R}^p$, the relational SOM is ***strictly equivalent*** to the original SOM.

Particular case of Kernel SOM

[Aronszajn, 1950, Villa and Rossi, 2007, Mac Donald and Fyfe, 2000]

- The data can be described by a kernel matrix $\mathbf{K} = (K(x_i, x_j))_{i,j=1,\ldots,N}$
- A kernel $\mathbf{K}$ is a *particular case* of symmetric similarity measure, positive semi-defined and satisfying

$$\forall\, M > 0,\ \forall\, (x_i)_{i=1,\ldots,M} \in \mathcal{X},\ \forall\, (\alpha_i)_{i=1,\ldots,M}, \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \geq 0.$$

- Observe that a *kernel matrix* $\mathbf{K}$ is an Euclidean distance matrix, but a *dissimilarity matrix* $\mathbf{D}$ may not necessarily be transformed into a kernel matrix

For *Kernel data*, [Aronszajn, 1950] proves

Theorem

There exists a Hilbert space $\mathcal{H}$, also called feature space, and a mapping $\psi : \mathcal{X} \to \mathcal{H}$, called feature map, such that $K(x_i, x_j) = \langle \psi(x_i), \psi(x_j) \rangle_{\mathcal{H}}$ (dot product in $\mathcal{H}$)

Particular case of Kernel SOM

- **The prototypes** are expressed as convex combinations of the $(\psi(x_i))$:

$$m_k(t) = \sum_{i=1}^{N} \gamma_{ki}^t \psi(x_i)$$

where $\gamma_{ki}^t \geq 0$ and $\sum_i \gamma_{ki}^t = 1$

- **The distance** :

$$\|\psi(x_i) - m_k(t)\|^2 = \left(\gamma_k^t\right)^T \mathbf{K} \gamma_k^t - 2\mathbf{K}_i \gamma_k^t + \mathbf{K}_{ii} \ ,$$

where $\mathbf{K}_i$ is the ith row of $\mathbf{K}$ and $\left(\gamma_k^t\right)^T = \left(\gamma_{k,1}^t, \ldots, \gamma_{k,N}^t\right)$

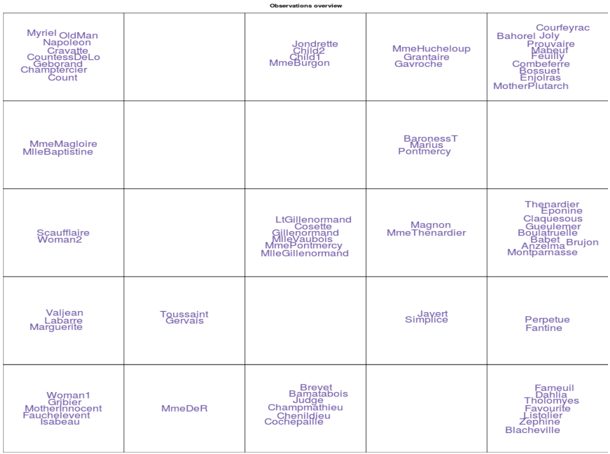
- The **prototypes updates** are the same as before, acting only on the γ
- If the dissimilarity is the squared distance induced by the kernel, kernel SOM and relational SOM are **strictly equivalent**
- **Fully equivalent to the original SOM algorithm** in the feature Euclidean space, and suffer the same theoretical limitations

The characters in "Les misérables"

The graph : graph of co-occurrences (in a same chapter) of the 77 characters in the Victor Hugo's novel "Les misérables"

The characters in "Les misérables"

Dissimilarity : length of the shortest path between two vertices

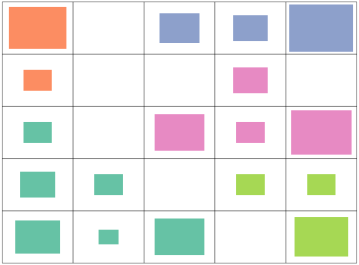


The characters in "Les misérables"

The size of the clusters is proportional to the number of characters.

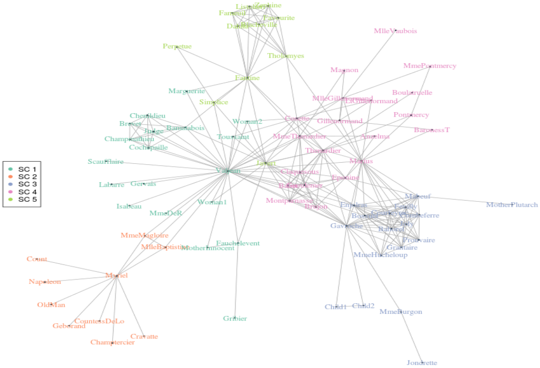
Principle : [Olteanu and Villa-Vialaneix, 2015b]

- Relational SOM
- Hierarchical clustering of the prototypes to build **“super-classes”**



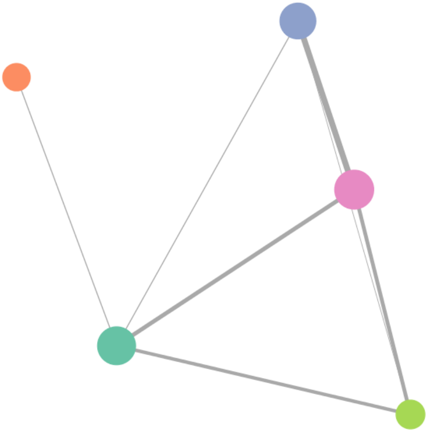
The characters in "Les misérables"

The initial graph is colored



The characters in "Les misérables"

Graph projection : each super-class is represented by a circle with a radius proportional to the number of vertices it contains. The width of the edges is proportional to the number of connections between two super-classes



Example 2 : Professional trajectories

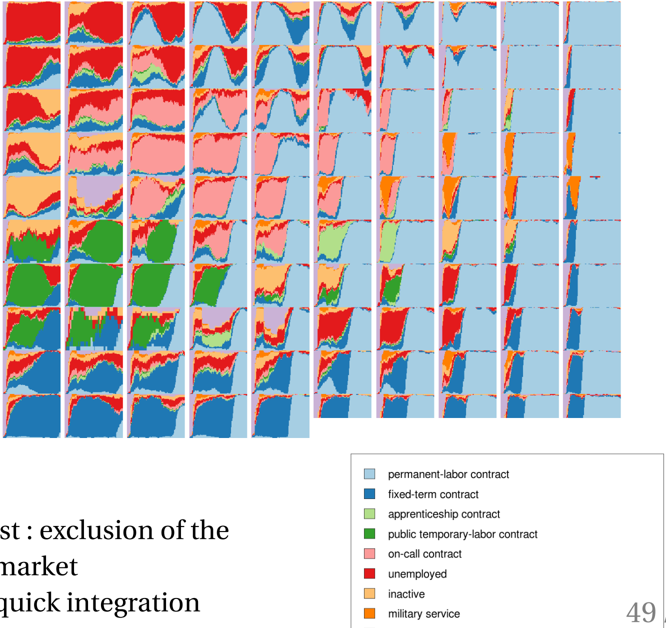
The data : “Generation 98” à 7 ans - 2005, CEREQ, Centre Maurice Halbwachs (CMH). 16 040 young people leaving initial training in 1998 are observed during 94 months. Each month, the nature of their activity is recorded (non-fixed term contracts, fixed term contracts, training program, unemployment, public contract,...)

Example 2 : Professional trajectories

The data : “Generation 98” à 7 ans - 2005, CEREQ, Centre Maurice Halbwachs (CMH). 16 040 young people leaving initial training in 1998 are observed during 94 months. Each month, the nature of their activity is recorded (non-fixed term contracts, fixed term contracts, training program, unemployment, public contract,...)

Dissimilarity between recorded sequences : **Edit Distance**, also called **Optimal Distance**.
See [Olteanu and Villa-Vialaneix, 2015a] for details

Example 2 : Professional trajectories



Up west : exclusion of the labor market

East : quick integration

Stochasticity of the results

Finding : several runs of the on-line SOM algorithm provide different resulting maps, even with the same initialization.

Three tracks :

Improve the stability as in the following papers [Petrakieva and Fyfe, 2003, Saavedra et al., 2007, Vrusias et al., 2007, Baruque and Corchado, 2011, Mariette et al., 2014, Mariette and Villa-Vialaneix, 2016]

or

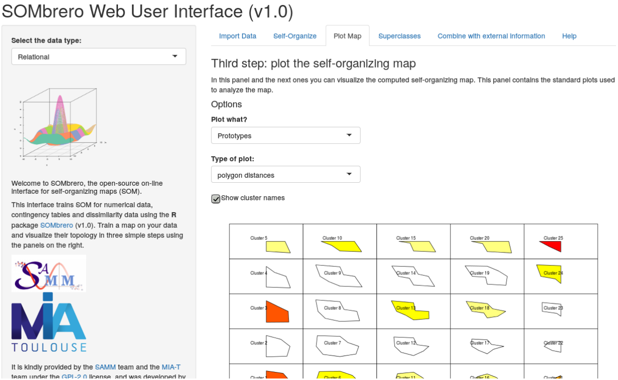
Use this stochasticity to qualify the reliability of the results with stability index [de Bodt et al., 2002]

or

Distinguish stable pairs and fickle pairs of data points to improve the interpretation and the visualization as in [Bourgeois et al., 2015] for medieval text mining

SOM in practice...

- Batch SOM for numerical data or relational data is implemented in **yasomi**
  `http://yasomi.r-forge.r-project.org`
- KORRESP and on-line SOM for numerical data or relational data are implemented in **SOMbrero** (CRAN)



Conclusion

SOM and Batch SOM are *Clustering* algorithms with **very interesting properties**

- *The complexity is linear* with respect to the number of data, well adapted to ***Big Data*** context ;
- Nice properties of *visualization* of the data and of the clusters ;
- Easy use with missing data, and *estimation of these missing data* ;
- Interesting *initialization* and *acceleration* of 0-neighbor algorithms.

The **relational version** provides an interesting alternative for non numerical data, but its complexity is increased and its interpretability decreased (representation of the results, prototypes interpretations).

Thank you for your attention

Some extra slides

SOM and regularized EM

[Heskes, 2001]

- Consider a **mixture of K Gaussian distributions**, centered on the prototypes, with covariance matrix $\frac{1}{\beta}\mathbf{I}$
- Maximizing the likelihood is equivalent to **minimize the VQ distortion**, so there is not any topology preservation
- A regularization term **penalizes** prototypes that do not respect the prior structure
- Applying the EM principle to the regularized (log)likelihood leads to an algorithm that **resembles** the batch SOM one.
- But the final algorithm is **significantly different** from the batch SOM :
  - Crisp assignments are replaced by probabilistic ones
  - The neighborhood function is fixed
- Increasing β progressively implies to **reduce the neighborhood** function during the EM algorithm, but this might have consequences that **remain untested**

SOM and variational EM

[VerBeek et al., 2005]

- As before, let us assume a **mixture model** (e.g. a K-components Gaussian isotropic mixture), let us denote the parameter vector by Θ
- The **hidden (latent) variables** Z are the assignment ones which map each data point x to a component of the mixture (a cluster)
- An arbitrary distribution q is chosen on variables Z
- **The log likelihood** $\log p(X|\Theta)$ **is equal to the sum of** :
  - The *complete log likelihood*, $\mathbb{E}_q \log p(X, Z|\Theta)$
  - The *entropy* of q, $H(q)$
  - The *Kullback-Leibler divergence*, $KL(q|p(Z|X, \Theta))$, between q and the posterior distribution of the hidden variables knowing the data points

SOM and variational EM

- To use the EM algorithm, the posterior distribution of the hidden variables knowing the data points has to be known. The **variational approach** consists in replacing this distribution by a simpler one
- Verbeek et al. constrain $p(Z|X,\Theta)$ to a subset of probability distributions that **fulfill topological constraints** corresponding to the prior structure of the SOM
- In addition, VerBeek et al. study the effects of **shrinking the neighborhood function** during training and conclude that it improves the quality of the solutions

The Generative Topographic Mapping (GTM)

[Bishop et al., 1998]

- **Mixture model** inspired by the SOM rather than an adaptation
- Uniform prior distribution on a fixed grid which is mapped via an **explicit smooth nonlinear mapping** to the data space
- The **constraints** induced on the data space are quite similar to the SOM constraints
- Once the model has been specify (by choosing the nonlinear mapping), its parameters are estimated via an **EM algorithm**
- The obtained algorithm is quite different from the SOM, but GTM can be reformulated in a way that is **close to the batch SOM** with probabilistic assignments (as in e.g. the STM)

Soft Topographic Maps for non numerical data

[Graepel et al., 1998]

- [Graepel et al., 1998] define an **extension of STM** for kernels and dissimilarities data
- The updates for the prototype coefficients are then expressed as

$$\gamma_{ki}(t+1) = \frac{\sum_{j=1}^{K} h_{jk}(t)\mathbb{P}(x_i \in C_j)}{\sum_{l=1}^{N}\sum_{j=1}^{K} h_{jk}(t)\mathbb{P}(x_l \in C_j)}, \tag{18}$$

  where $m_k(t) = \sum_{i=1}^{N} \gamma_{ki}^{t}\psi(x_i)$ and ψ is the embedding map
- It is only in a **Batch mode**

Aronszajn, N. (1950).
Theory of reproducing kernels.
Transactions of the American Mathematical Society, 68(3) :337–404.

Baruque, B. and Corchado, E. (2011).
Fusion methods for unsupervised learning ensembles, volume 322 of *Studies in Computational Intelligence*.
Springer.

Bourgeois, N., Cottrell, M., Deruelle, B., Lamassé, S., and Letrémy, P. (2015).
How to improve robustness in Kohonen maps and display additional information in factorial analysis : application to text mining.
Neurocomputing, 147 :120–135.

Cottrell, M., Fort, J., and Pagès, G. (1998).
Theoretical aspects of the SOM algorithm.
Neurocomputing, 21 :119–138.

de Bodt, E., Cottrell, M., and Verleisen, M. (2002).
Statistical tools to assess the reliability of self-organizing maps.
Neural Networks, 15(8-9) :967–978.

Erwin, E., Obermayer, K., and Schulten, K. (1992a).
Self-organizing maps : ordering, convergence properties and energy functions.
Biological Cybernetics, 67(1) :47–55.

Erwin, E., Obermayer, K., and Schulten, K. (1992b).
Self-organizing maps : stationnary states, metastability and convergence rate.
Biological Cybernetics, 67(1) :35–45.

Fort, J. (2005).
SOM's mathematics.
In *Workshop on Self-Organizing Maps (WSOM 2005)*, Paris, France.

Goldfarb, L. (1984).
A unified approach to pattern recognition.

Pattern Recognition, 17(5) :575–582.

Graepel, T., Burger, M., and Obermayer, K. (1998).
Self-organizing maps : generalizations and new optimization techniques.
Neurocomputing, 21 :173–190.

Hammer, B. and Hasenfuss, A. (2010).
Topographic mapping of large dissimilarity data sets.
Neural Computation, 22(9) :2229–2284.

Heskes, T. (1999).
Energy functions for self-organizing maps.
In Oja, E. and Kaski, S., editors, *Kohonen Maps*, pages 303–315. Elsevier, Amsterdam.

Heskes, T. (2001).
Self-organizing maps, vector quantization, and mixture modeling.
IEEE Transactions on Neural Networks, 12(6) :1299–1305.

Kohohen, T. and Somervuo, P. (1998).
Self-organizing maps of symbol strings.
Neurocomputing, 21 :19–30.

Mac Donald, D. and Fyfe, C. (2000).
The kernel self organising map.
In *Proceedings of 4th International Conference on knowledge-based Intelligence Engineering Systems and Applied Technologies*, pages 317–320.

Mariette, J., Olteanu, M., Boelaert, J., and Villa-Vialaneix, N. (2014).
Bagged kernel SOM.
In Villmann, T., Schleif, F., Kaden, M., and Lange, M., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2014)*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 45–54, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.

Mariette, J. and Villa-Vialaneix, N. (2016).
Aggregating self-organizing maps with topology preservation.
In *Proceedings of WSOM 2016*, Houston, TX, USA.

Forthcoming.

Olteanu, M. and Villa-Vialaneix, N. (2015a).
On-line relational and multiple relational SOM.
Neurocomputing, 147 :15–30.

Olteanu, M. and Villa-Vialaneix, N. (2015b).
Using SOMbrero for clustering and visualizing graphs.
Journal de la Société Française de Statistique.
Forthcoming.

Petrakieva, L. and Fyfe, C. (2003).
Bagging and bumping self organising maps.
Computing and Information Systems Journal, 9 :69–77.

Ritter, H., Martinetz, T., and Schulten, K. (1992).
Neural Computation and Self-Organizing Maps : an Introduction.
Addison-Wesley.

Rossi, F. (2014).
How many dissimilarity/kernel self organizing map variants do we need?
In Villmann, T., Schleif, F., Kaden, M., and Lange, M., editors, *Advances in Self-Organizing Maps and Learning Vector Quantization (Proceedings of WSOM 2014)*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 3–23, Mittweida, Germany. Springer Verlag, Berlin, Heidelberg.

Saavedra, C., Salas, R., Moreno, S., and Allende, H. (2007).
Fusion of self organizing maps.
In *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007).*

Verbeek, J. J., Vlassis, N., and Kröse, B. J. A. (2005).
Self-organizing mixture models.
Neurocomputing, 63 :99–123.

Villa, N. and Rossi, F. (2007).
A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph.

In *6th International Workshop on Self-Organizing Maps (WSOM 2007)*, Bielefield, Germany. Neuroinformatics Group, Bielefield University.

Vrusias, B., Vomvoridis, L., and Gillam, L. (2007).
Distributing SOM ensemble training using grid middleware.
In *Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2007)*, pages 2712–2717.