Multiple Kernel Self-Organizing Maps

Madalina Olteanu[(2)], Nathalie Villa-Vialaneix[(1,2)], Christine Cierco-Ayrolles[(1)]

http://www.nathalievilla.org

{madalina.olteanu,nathalie.villa}@univ-paris1.fr,

christine.cierco@toulouse.inra.fr

ESANN 2013, April 24th

(1) INRA (2) SAMM UNIVERSITÉ PARIS 1 – PANTHÉON - SORBONNE –

Outline

Data, notations and objectives

Data: D datasets $(x_i^d)_{i=1,\ldots,n,d=1,\ldots,D}$ all measured on the same individuals or on the same objects, $\{i,\ldots,n\}$, each taking values in an arbitrary space $\mathcal{G}_d$.

Data, notations and objectives

Data: D datasets $(x_i^d)_{i=1,\ldots,n,d=1,\ldots,D}$ all measured on the same individuals or on the same objects, $\{i, \ldots, n\}$, each taking values in an arbitrary space $\mathcal{G}_d$.

Examples:

- $(x_i^d)_i$ are n observations of p numeric variables;
- $(x_i^d)_i$ are n nodes of a graph;
- $(x_i^d)_i$ are n observations of p factors;
- $(x_i^d)_i$ are n texts/labels...

Data, notations and objectives

Data: D datasets $(x_i^d)_{i=1,\ldots,n,d=1,\ldots,D}$ all measured on the same individuals or on the same objects, $\{i,\ldots,n\}$, each taking values in an arbitrary space $\mathcal{G}_d$.

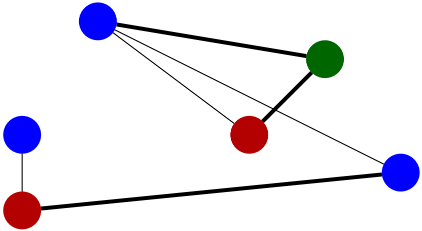
Examples:

- $(x_i^d)_i$ are n observations of p numeric variables;
- $(x_i^d)_i$ are n nodes of a graph;
- $(x_i^d)_i$ are n observations of p factors;
- $(x_i^d)_i$ are n texts/labels...

Purpose: Combine all datasets to obtain a map of individuals/objects: self-organizing maps for clustering $\{i,\ldots,n\}$ using all datasets.

Example [Villa-Vialaneix et al., 2013]

Data: A **network** with labeled nodes.



Examples: Gender in a social network, Functional group of a gene in a gene interaction network...

Purpose: find communities, i.e., groups of "close" nodes in the graph; "close" meaning:

- **densely connected** and sharing (comparatively) a few links with the other groups ("communities");
- but also **having similar labels**.

Outline

Kernels/Multiple kernel

What is a kernel?

$(x_i) \in \mathcal{G}$, $(K(x_i, x_j))_{ij}$ st: $K(x_i, x_j) = K(x_j, x_i)$ and $\forall\,(\alpha_i)_i,\ \sum_{ij} \alpha_i \alpha_j K(x_i, x_j) \geq 0$. In this case **[Aronszajn, 1950]**,

$$\exists\ (\mathcal{H}, \langle .,. \rangle),\ \Phi : \mathcal{G} \rightarrow \mathcal{H} \text{ st : } \qquad K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$$

Examples:

- **nodes of a graph**: Heat kernel $K = e^{-\beta L}$ or $K = L^+$ where L is the Laplacian **[Kondor and Lafferty, 2002, Smola and Kondor, 2003, Fouss et al., 2007]**
- **numerical variables**: Gaussian kernel $K(x_i, x_j) = e^{-\beta \|x_i - x_j\|^2}$;
- **text**: **[Watkins, 2000]**...

Kernel SOM

[Mac Donald and Fyfe, 2000, Andras, 2002, Villa and Rossi, 2007]

$(x_i)_i \subset \mathcal{G}$ described by a kernel $(K(x_i, x_{i'}))_{ii'}$.

Prototypes are defined in $\mathcal{H}$: $p_j = \sum_i \gamma_{ji}\phi(x_i)$; energy is calculated in $\mathcal{H}$:

1: **Prototypes initialization**: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ij}^0 \Phi(x_i)$ st $\gamma_{ij}^0 \geq 0$ and $\sum_i \gamma_{ij}^0 = 1$
2: **for** $t = 1 \rightarrow T$ **do**
3: Randomly choose $i \in \{1, \ldots, n\}$
4: **Assignment**

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \|x_i - p_j^{t-1}\|_{\mathcal{H}}$$

5: **for all** $j = 1 \rightarrow M$ **do Representation**
6: $\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j)) \left(\mathbf{1}_i - \gamma_j^{t-1}\right)$
7: **end for**
8: **end for**

δ: distance between neurons, h: decreasing function ($(h^t)_t$ diminishes with t), $\mu_t \sim 1/t$.

Kernel SOM

[Mac Donald and Fyfe, 2000, Andras, 2002, Villa and Rossi, 2007]

$(x_i)_i \subset \mathcal{G}$ described by a kernel $(K(x_i, x_{i'}))_{ii'}$.

Prototypes are defined in $\mathcal{H}$: $p_j = \sum_i \gamma_{ji}\phi(x_i)$; energy is calculated in $\mathcal{H}$:

1: **Prototypes initialization**: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ij}^0 \Phi(x_i)$ st $\gamma_{ij}^0 \geq 0$ and $\sum_i \gamma_{ij}^0 = 1$
2: **for** $t = 1 \to T$ **do**
3: Randomly choose $i \in \{1, \ldots, n\}$
4: **Assignment**

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \sum_{ll'} \gamma_{jl}^{t-1}\gamma_{jl'}^{t-1} K^t(x_l, x_{l'}) - 2\sum_l \gamma_{jl}^{t-1} K^t(x_l, x_i)$$

5: **for all** $j = 1 \to M$ **do Representation**
6: $\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j))\left(\mathbf{1}_i - \gamma_j^{t-1}\right)$
7: **end for**
8: **end for**

δ: distance between neurons, h: decreasing function ($(h^t)_t$ diminishes with t), $\mu_t \sim 1/t$.

Combining kernels

Suppose: each $(x_i^d)_i$ is described by a kernel K_d, kernels can be combined (e.g., **[Rakotomamonjy et al., 2008]** for SVM):

$$K = \sum_{d=1}^{D} \alpha_d K_d,$$

with $\alpha_d \geq 0$ and $\sum_d \alpha_d = 1$.

Combining kernels

Suppose: each $(x_i^d)_i$ is described by a kernel K_d, kernels can be combined (e.g., **[Rakotomamonjy et al., 2008]** for SVM):

$$K = \sum_{d=1}^{D} \alpha_d K_d,$$

with $\alpha_d \geq 0$ and $\sum_d \alpha_d = 1$.
Remark: Also useful to integrate different types of information coming from different kernels on the same dataset.

multiple kernel SOM

Prototypes initialization: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \Phi(x_i)$ st $\gamma_{ji}^0 \geq 0$ and $\sum_i \gamma_{ji}^0 = 1$

Kernel initialization: set (α_d^0) st $\alpha_d^0 \geq 0$ and $\sum_d \alpha_d = 1$ (e.g., $\alpha_d^0 = 1/D$); $K^0 \leftarrow \sum_d \alpha_d^0 K_d$

for $t = 1 \rightarrow T$ **do**

  Randomly choose $i \in \{1, \ldots, n\}$

  Assignment

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \|x_i - p_j^{t-1}\|^2_{\mathcal{H}(K^t)}$$

  for all $j = 1 \rightarrow M$ **do** **Representation**

    $\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j)) \left(\mathbf{1}_i - \gamma_j^{t-1}\right)$

  end for

end for

multiple kernel SOM

Prototypes initialization: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \Phi(x_i)$ st $\gamma_{ji}^0 \geq 0$ and $\sum_i \gamma_{ji}^0 = 1$

Kernel initialization: set (α_d^0) st $\alpha_d^0 \geq 0$ and $\sum_d \alpha_d = 1$ (e.g., $\alpha_d^0 = 1/D$); $K^0 \leftarrow \sum_d \alpha_d^0 K_d$

for $t = 1 \rightarrow T$ **do**

Randomly choose $i \in \{1, \ldots, n\}$

Assignment

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \sum_{ll'} \gamma_{jl}^{t-1} \gamma_{jl'}^{t-1} K^t(x_l, x_{l'}) - 2 \sum_l \gamma_{jl}^{t-1} K^t(x_l, x_i)$$

for all $j = 1 \rightarrow M$ **do** **Representation**

$\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j)) \left(\mathbf{1}_i - \gamma_j^{t-1}\right)$

end for

end for

Tuning $(\alpha_d)_d$ on-line

Purpose: minimize over $(\gamma_{ji})_{ji}$ and $(\alpha_d)_d$ the energy

$$\mathcal{E}((\gamma_{ji})_{ji}, (\alpha_d)_d) = \sum_{i=1}^{n} \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \left\| \phi^\alpha(x_i) - p_j^\alpha(\gamma_j) \right\|_\alpha^2 ,$$

Tuning $(\alpha_d)_d$ on-line

Purpose: minimize over $(\gamma_{ji})_{ji}$ and $(\alpha_d)_d$ the energy

$$\mathcal{E}((\gamma_{ji})_{ji}, (\alpha_d)_d) = \sum_{i=1}^{n} \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \left\|\phi^\alpha(x_i) - p_j^\alpha(\gamma_j)\right\|_\alpha^2 ,$$

KSOM picks up an observation x_i that has a contribution to the energy equal to:

$$\mathcal{E}|_{x_i} = \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \left\|\phi^\alpha(x_i) - p_j^\alpha(\gamma_j)\right\|_\alpha^2 \qquad (1)$$

Tuning $(\alpha_d)_d$ on-line

Purpose: minimize over $(\gamma_{ji})_{ji}$ and $(\alpha_d)_d$ the energy

$$\mathcal{E}((\gamma_{ji})_{ji}, (\alpha_d)_d) = \sum_{i=1}^{n} \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \left\| \phi^\alpha(x_i) - p_j^\alpha(\gamma_j) \right\|_\alpha^2 ,$$

KSOM picks up an observation x_i that has a contribution to the energy equal to:

$$\mathcal{E}|_{x_i} = \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \left\| \phi^\alpha(x_i) - p_j^\alpha(\gamma_j) \right\|_\alpha^2 \qquad (1)$$

Idea: Add a gradient descent step based on the derivative of (1):

$$\frac{\partial \mathcal{E}|_{x_i}}{\partial \alpha_d} = \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \Bigg(K_d(x_i^d, x_i^d) - 2 \sum_{l=1}^{n} \gamma_{jl} K_d(x_i^d, x_l^d) + \sum_{l,l'=1}^{n} \gamma_{jl}\gamma_{jl'} K_d(x_l^d, x_{l'}^d) \Bigg) = \sum_{j=1}^{M} h\left(\delta(f(x_i), j)\right) \|\phi(x_i) - p_j^d(\gamma_j)\|_{\mathcal{H}(K_d)}^2$$

adaptive multiple kernel SOM

Prototypes initialization: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \Phi(x_i)$ st $\gamma_{ji}^0 \geq 0$ and $\sum_i \gamma_{ji}^0 = 1$

Kernel initialization: set (α_d^0) st $\alpha_d^0 \geq 0$ and $\sum_d \alpha_d = 1$ (e.g., $\alpha_d^0 = 1/D$); $K^0 \leftarrow \sum_d \alpha_d^0 K_d$

for $t = 1 \rightarrow T$ **do**

  Randomly choose $i \in \{1, \ldots, n\}$

  Assignment

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \|x_i - p_j^{t-1}\|^2_{\mathcal{H}(K^t)}$$

  for all $j = 1 \rightarrow M$ **do Representation**

    $\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j)) \left(\mathbf{1}_i - \gamma_j^{t-1}\right)$

  end for

end for

adaptive multiple kernel SOM

Prototypes initialization: randomly set $p_j^0 = \sum_{i=1}^n \gamma_{ji}^0 \Phi(x_i)$ st $\gamma_{ji}^0 \geq 0$ and $\sum_i \gamma_{ji}^0 = 1$

Kernel initialization: set (α_d^0) st $\alpha_d^0 \geq 0$ and $\sum_d \alpha_d = 1$ (e.g., $\alpha_d^0 = 1/D$); $K^0 \leftarrow \sum_d \alpha_d^0 K_d$

for $t = 1 \rightarrow T$ **do**

    Randomly choose $i \in \{1, \ldots, n\}$

    Assignment

$$f^t(x_i) \leftarrow \arg\min_{j=1,\ldots,M} \|x_i - p_j^{t-1}\|^2_{\mathcal{H}(K^t)}$$

    for all $j = 1 \rightarrow M$ **do** **Representation**

        $\gamma_j^t \leftarrow \gamma_j^{t-1} + \mu_t h^t(\delta(f^t(x_i), j)) \left(\mathbf{1}_i - \gamma_j^{t-1}\right)$

    end for

    Kernel update $\alpha_d^t \leftarrow \alpha_d^{t-1} + \nu_t \frac{\partial \mathcal{E}|_{x_i}}{\partial \alpha_d}$ and $K^t \leftarrow \sum_d \alpha_d^t K_d$

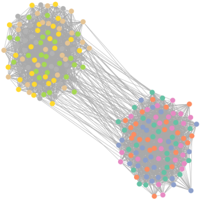
end for

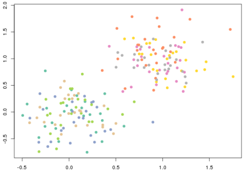
Outline

Example1: simulated data

Graph with 200 nodes classified in 8 groups:

- **graph**: Erdös Reyni models: groups 1 to 4 and groups 5 to 8 with intra-group edge probability 0.3 and inter-group edge probability 0.01;
- **numerical data**: nodes labelled with 2-dimensional Gaussian vectors: odd groups $\mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.3 & 0\\0 & 0.3\end{pmatrix}\right)$ and even groups $\mathcal{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0.3 & 0\\0 & 0.3\end{pmatrix}\right)$;
- **factor** with two levels: groups 1, 2, 5 and 7: first level; other groups: second level.





Only the knowledge on the three datasets can discriminate all 8 groups.

Experiment

Kernels: graph: L^+, numerical data: Gaussian kernel; factor: another Gaussian kernel on the disjunctive recoding

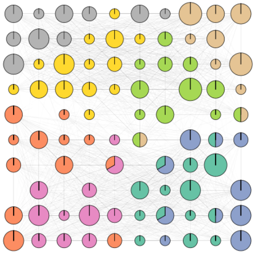
Experiment

Kernels: graph: L^+, numerical data: Gaussian kernel; factor: another Gaussian kernel on the disjunctive recoding
Comparison: on 100 randomly generated datasets as previously described:

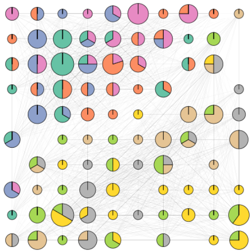
- multiple kernel SOM with all three data;
- kernel SOM with a single dataset;
- kernel SOM with two datasets or all three datasets in a single (Gaussian) kernel.

An example

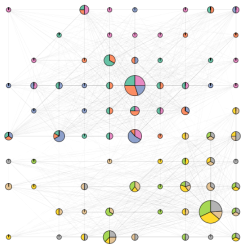
MK-SOM



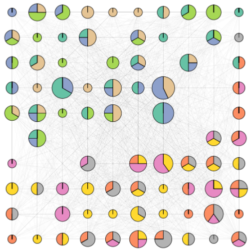
All in one kernel



Graph only



Numerical variables only

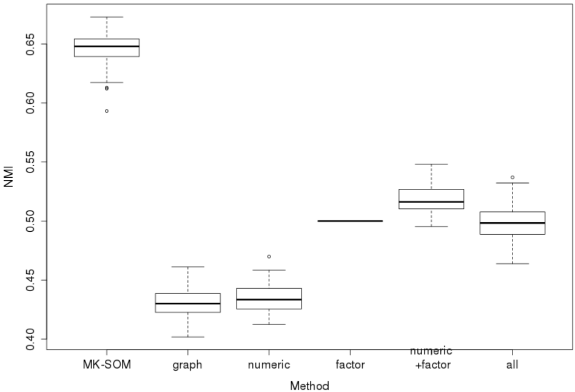


Numerical comparison I

(over 100 simulations) with **mutual information**

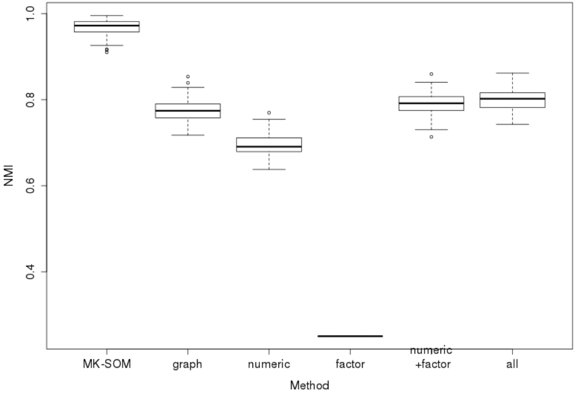
$$\sum_{ij} \frac{|C_i \cap \tilde{C}_j|}{200} \log \frac{|C_i \cap \tilde{C}_j|}{|C_i| \times |\tilde{C}_j|}$$

(adjusted version, equal to 1 if partitions are identical **[Danon et al., 2005]**).



Numerical comparison II

(over 100 simulations) with **node purity** (average percentage of nodes in the neuron that come from the majority (original) class of the neuron).



Conclusion

Summary

- integrating multiple sources information on SOM (e.g., finding communities in graph, while taking into account labels);
- uses multiple kernel and automatically tunes the combination;
- the method gives relevant communities according to all sources of information and a well-organized map;
- a similar approach also tested for relational SOM (**[Massoni et al., 2013]** for analyzing school-to-work transitions).

Conclusion

Summary

- integrating multiple sources information on SOM (e.g., finding communities in graph, while taking into account labels);
- uses multiple kernel and automatically tunes the combination;
- the method gives relevant communities according to all sources of information and a well-organized map;
- a similar approach also tested for relational SOM (**[Massoni et al., 2013]** for analyzing school-to-work transitions).

Possible developments

- **Main issue**: computational time; currently studying a sparse version.

Conclusion

Summary

- integrating multiple sources information on SOM (e.g., finding communities in graph, while taking into account labels);
- uses multiple kernel and automatically tunes the combination;
- the method gives relevant communities according to all sources of information and a well-organized map;
- a similar approach also tested for relational SOM (**[Massoni et al., 2013]** for analyzing school-to-work transitions).

Possible developments

- **Main issue**: computational time; currently studying a sparse version.

Thank you for your attention...

References

Andras, P. (2002).
Kernel-Kohonen networks.
International Journal of Neural Systems, 12:117–135.

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

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005).
Comparing community structure identification.
Journal of Statistical Mechanics, page P09008.

Fouss, F., Pirotte, A., Renders, J., and Saerens, M. (2007).
Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation.
IEEE Transactions on Knowledge and Data Engineering, 19(3):355–369.

Kondor, R. and Lafferty, J. (2002).
Diffusion kernels on graphs and other discrete structures.
In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322.

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.

Massoni, S., Olteanu, M., and Villa-Vialaneix, N. (2013).
Which distance use when extracting typologies in sequence analysis? An application to school to work transitions.
In *International Work Conference on Artificial Neural Networks (IWANN 2013)*, Puerto de la Cruz, Tenerife.
Forthcoming.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008).
SimpleMKL.

Journal of Machine Learning Research, 9:2491–2521.

Smola, A. and Kondor, R. (2003).
Kernels and regularization on graphs.
In Warmuth, M. and Schölkopf, B., editors, *Proceedings of the Conference on Learning Theory (COLT) and Kernel Workshop*, Lecture Notes in Computer Science, pages 144–158.

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)*, Bielefield, Germany. Neuroinformatics Group, Bielefield University.

Villa-Vialaneix, N., Olteanu, M., and Cierco-Ayrolles, C. (2013).
Carte auto-organisatrice pour graphes étiquetés.
In *Actes des Ateliers FGG (Fouille de Grands Graphes), colloque EGC (Extraction et Gestion de Connaissances)*, Toulouse, France.

Watkins, C. (2000).
Dynamic alignment kernels.
In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 39–50, Cambridge, MA, USA. MIT P.