

---

# Avantages et Inconvénients de la version batch de l'algorithme de Kohonen

Jean-Claude Fort<sup>1</sup>, Patrick Letremy<sup>2</sup>, Marie Cottrell<sup>2</sup>

<sup>1</sup>IEL et MATISSE-SAMOS  
Université Nancy I  
54506 Vandoeuvre-lès-Nancy  
fortjc@iecn.u-nancy.fr

<sup>2</sup>MATISSE-SAMOS, UMR CNRS 8595  
Université Paris I  
90, rue de Tolbiac  
75634 Paris Cedex 13  
pley,cottrell@univ-paris1.fr

---

*RÉSUMÉ. L'algorithme d'auto organisation de Kohonen a été défini à l'origine comme un algorithme adaptatif, stochastique, destiné à modéliser l'apprentissage de certaines caractéristiques du système nerveux. Mais il est maintenant largement utilisé pour faire des classifications, de l'analyse exploratoire et de la visualisation de données. Il apparaît comme un outil très performant de fouille de données. Dans ce cadre, on est tenté d'utiliser une version déterministe de l'algorithme, plus rapide à mettre en oeuvre, apparentée à l'algorithme des centres mobiles de Forgy. Le but de ce papier est d'étudier les avantages et les inconvénients de cette méthode. Pour cela, on donne des arguments théoriques et on présente plusieurs exemples.*

*MOTS-CLÉS : auto organisation, SOM, classification, centres mobiles, algorithme de Newton*

---

## 1. Introduction

Les cartes auto organisatrices (SOM) de Teuvo Kohonen ([KOH 89], [KOH 95]) sont utilisées dans de nombreux domaines traditionnels d'analyse de données. Voir par exemple [KAS 97], [DEB 98], [OJA 99], [COT 97] etc.

L'algorithme originel est une extension de l'algorithme compétitif simple (Simple Competitive Learning, SCL) qui introduit la notion de voisinage entre les classes. Il permet d'obtenir des classes organisées reproduisant les proximités dans l'espace des données.

Les deux algorithmes sont des algorithmes stochastiques adaptatifs, ce qui signifie qu'à chaque présentation d'une nouvelle observation, l'algorithme met à jour les valeurs des vecteurs-codes (ou représentants) des classes. Mais pour les deux algorithmes, il existe des algorithmes déterministes équivalents, qui utilisent toutes les données à chaque étape. Pour l'algorithme SCL, il s'agit de l'algorithme classique des centres mobiles [FOR 65]. Pour l'algorithme de Kohonen, on parle d'algorithme de Kohonen batch, abrégé en KBATCH, [KOH 99].

## 2. Les versions stochastiques (SCL et SOM)

Notations :

1. On note  $d$  la dimension des données, et  $X_0(i)$  la position initiale du vecteur code (de dimension  $d$ ), où  $i$  est une unité élément de  $I$ , ensemble des unités.
2. La suite des données observées est notée  $\omega_t$   $t \geq 1$ , ce sont aussi des vecteurs de dimension  $d$ .
3. On note  $\sigma(i,j)$  la fonction de voisinage qui est symétrique, fonction de  $|i - j|$ , elle mesure le lien entre l'unité  $i$  et l'unité  $j$ ,  $\sigma(i,i) = 1$  et elle décroît avec la distance entre  $i$  et  $j$ .
4. On note  $\varepsilon(t)$  le gain (ou paramètre d'adaptation). La suite  $\varepsilon(t)$  est constante ou décroissante.,
5. Si  $\sigma(i,j) = 0$  dès que  $i \neq j$ , et vaut 1 pour  $i = j$ , cela correspond au cas SCL.

Alors l'algorithme fonctionne en deux étapes. A chaque itération, les vecteurs codes sont notés  $X_t(i)$ , et on présente une donnée  $\omega_{t+1}$ .

- 1) On détermine l'unité gagnante  $i_0(t+1)$  définie par

$$i_0(t+1) = \text{Arg min}_i \|\omega_{t+1} - X_t(i)\|$$

- 2) On modifie les vecteurs codes selon la règle suivante :

$$X_{t+1}(i) = X_t(i) + \varepsilon(t)\sigma(i_0(t+1),i)(\omega_{t+1} - X_t(i))$$

Si les vecteurs codes sont  $x = (x(i))$ ,  $i \in I$ , on définit les classes associées en posant

$$C_i(x) = \left\{ \omega \text{ t.q. } \|x(i) - \omega\| = \min_j \|x(j) - \omega\| \right\}.$$

La classe  $C_i$  est l'ensemble des données pour lesquelles l'unité  $i$  est la gagnante. L'ensemble des  $(C_i(x))$  est la mosaïque de Voronoï définie par  $x$ .

L'étude des propriétés mathématiques de cet algorithme reste partielle (voir [COT 98]) excepté dans le cas unidimensionnel (données réelles et structure unidimensionnelle des voisinages). Le bon cadre pour ces études est la théorie de l'approximation stochastique. On sait que l'algorithme SCL dans le cas en général et l'algorithme SOM [RIT 92] dans le cas d'une fonction voisinage fixe et d'un nombre fini de données peuvent être considérés comme

des algorithmes de gradient stochastique associés respectivement à la distorsion classique (ou somme des carrés intra) :

$$D(x) = \sum_{i \in I} \int_{C_i(x)} \|x(i) - \omega\|^2 \mu(d\omega)$$

et à la distorsion étendue aux classes voisines :

$$D_{ext}(x) = \sum_{i \in I} \sum_j \sigma(i, j) \int_{C_j(x)} \|x(i) - \omega\|^2 \mu(d\omega)$$

où  $\mu$  est la densité des données. Elle peut être continue ou discrète dans le cas de SCL, mais doit être discrète avec support fini pour SOM.

Même lorsque  $D$  est une fonction énergie, elle n'est pas partout différentiable et elle fournit seulement une information locale. En particulier, cela ne suffit pas pour fournir une preuve rigoureuse de convergence. Cependant on peut remarquer que si ces algorithmes sont convergents, ils convergent vers l'un des points d'équilibre de l'équation différentielle ordinaire (ODE) associée. Cette équation s'écrit avec les notations précédentes :

$$\frac{dx(i, u)}{du} = - \sum_{j \in I} \sigma(i, j) \int_{C_j(x(i, u))} (x(i, u) - \omega) \mu(d\omega).$$

Les points d'équilibre de cette équation vérifient alors

$$x^*(i) = \frac{\sum_j \sigma(i, j) \int_{C_j(x^*)} \omega \mu(d\omega)}{\sum_j \sigma(i, j) \mu(C_j(x^*))}.$$

Dans le cas de l'algorithme SCL, ceci exprime que chaque  $x^*(i)$  est le centre de gravité de sa classe. Plus généralement dans le cas de SOM,  $x^*(i)$  est le centre de gravité (pour la pondération définie par la fonction de voisinage) de l'union de  $C_i$  et des classes voisines.

On peut chercher à calculer ces solutions par un calcul itératif déterministe.

### 3. Les algorithmes batch

On en déduit immédiatement la définition des algorithmes batch de calcul des  $x^*(i)$ . Ces points sont obtenus comme limite de la suite définie par :

$$x^{k+1}(i) = \frac{\sum_j \sigma(i, j) \int_{C_j(x^k)} \omega \mu(d\omega)}{\sum_j \sigma(i, j) \mu(C_j(x^k))}.$$

Quand on ne prend pas en compte les voisinages, on retrouve exactement l'algorithme de Forgy (les centres mobiles). Dans le cas général, on définit l'algorithme de Kohonen version batch, que l'on note KBATCH.

On sait [FORT 01] que cet algorithme KBATCH est en fait un algorithme "quasi-Newtonien" (approximation de l'algorithme du gradient du second ordre qui minimise la

distorsion étendue  $D_{ext}$ ) C'est un algorithme "quasi-Newtonien" parce qu'on utilise seulement la diagonale de la matrice Hessienne. Malheureusement, il existe de nombreux ensembles disjoints où cette fonction  $D_{ext}$  est différentiable et il y a un minimum local à l'intérieur de chacun d'eux.

On sait que les algorithmes de Newton ne sont pas nécessairement des algorithmes de descente, ce qui entraîne que la fonction distorsion étendue peut croître pour certaines itérations. Cette propriété qui est un inconvénient peut s'avérer utile, car elle permet ensuite éventuellement la convergence vers un meilleur minimum.

Ces algorithmes déterministes sont, on le sait, extrêmement sensibles aux choix des conditions initiales. Nous montrerons dans la présentation de nombreux exemples [FORT 02] de ce fait. On verra alors que cet algorithme KBATCH qui en principe minimise la même fonction que l'algorithme SOM peut conduire à des états d'équilibre mal organisés, alors que l'algorithme stochastique SOM organise parfaitement.

Cela conduit à conclure que si l'algorithme KBATCH peut être séduisant par la rapidité de sa convergence et la simplicité de sa mise en oeuvre, le choix des conditions initiales s'avère être crucial du point de vue de la qualité de l'organisation, et qu'il est donc primordial de partir de conditions initiales choisies avec soin (maillage dans le premier plan principal par exemple).

#### 4. Bibliographie

- [COT 97] COTTRELL M., ROUSSET P.: The Kohonen algorithm: A Powerful Tool for Analysing and Representing Multidimensional Quantitative and Qualitative Data, *Proc. IWANN'97*, 1997.
- [COT 98] COTTRELL M., FORT J.C., PAGES G.: Theoretical aspects of the SOM Algorithm, WSOM'97, Helsinki 1997, *Neurocomputing* 21, 119-138, 1998.
- [DEB 98] DEBOECK G., KOHONEN T.: *Visual Explorations in Finance with Self-Organization Maps*, Springer, 1998.
- [FOR 65] FORGY B.W.: Cluster analysis of multivariate data: efficiency versus interpretability of classifications, *Biometrics*, 21, 3, 1965, p. 768.
- [FORT 01] FORT J.C., LETREMY P., COTTRELL M.: Stochastic on-line algorithm versus batch algorithm for quantization and Self Organizing Maps, *Second NNSP Conference*, Falmouth, September 2001.
- [FORT 02] FORT J.C., LETREMY P., COTTRELL M.: Batch Kohonen Algorithm, *ESANN 2002*, Bruges, 2002, p. 223-230.
- [KAS 97] KASKI S.: Data Exploration Using Self-Organizing Maps, *Acta Polytechnica Scandinavia*, 82, 1997.
- [KOH 89] KOHONEN T.: *Self-Organization and Associative Memory*, (3rd edition 1989), Springer, Berlin, 1984.
- [KOH 95] KOHONEN T.: *Self-Organizing Maps*, Springer, Berlin, 1995.
- [KOH 99] KOHONEN T.: Comparison of SOM Point Densities Based on Different Criteria, *Neural Computation*, 11, 1999, p. 2081-2095.
- [OJA 99] OJA E., KASKI S.: *Kohonen Maps*, Elsevier, 1999.
- [RIT 92] RITTER H., MARTINETZ T., SHULTEN K.: *Neural Computation and Self-Organizing Maps, an Introduction*, Addison-Wesley, Reading, 1992.