

LES CARTES TOPOLOGIQUES

Quantification vectorielle

\mathcal{D} espace des données

A ensemble d'apprentissage $A \subset D \subset \mathbb{R}^n$

$$A = \{\mathbf{z}_i, i = 1, \dots, N\}$$

Réduire l'information de \mathcal{D} :

- En la résumant par un ensemble de **p référents**

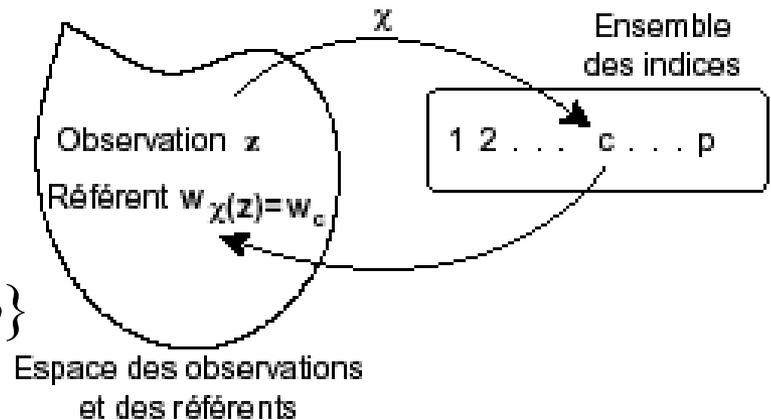
$$W = \{\mathbf{w}_c, c = 1, \dots, p\}$$

- En réalisant une partition de \mathcal{D} en **p** sous-ensembles par l'intermédiaire d'une **fonction d'affectation χ** :

$$\chi: D \rightarrow \{1, 2, \dots, p\}$$

$$P = \{P_1, P_2, \dots, P_p\}$$

$$P_c = \{\mathbf{z} \in D / \chi(\mathbf{z}) = c\}$$



Méthode des k-moyennes

- Minimiser la somme des inerties locales par rapport à χ et W

$$I(W, \chi) = \sum_{\mathbf{z}_i} \|\mathbf{z}_i - \mathbf{w}_{\chi(\mathbf{z}_i)}\|^2 = \sum_c \sum_{\mathbf{z}_i \in P_c \cap A} \|\mathbf{z}_i - \mathbf{w}_c\|^2$$



$$\sum_c I_c = \sum_c \sum_{\substack{\mathbf{z}_i \in A \\ \chi(\mathbf{z}_i) = c}} \|\mathbf{z}_i - \mathbf{w}_c\|^2$$

- L'inertie I_c représente l'erreur de quantification obtenue si l'on remplace chaque observations de P_c par son référent \mathbf{w}_c
- *Minimisation itérative qui fixe alternativement la partition (χ) puis minimise l'inertie (W)*

Phase d'affectation:

Pour un ensemble \mathbf{W} de référents fixe, la minimisation de I par rapport à χ s'obtient en affectant chaque observation \mathbf{z} au référent \mathbf{w}_c selon la nouvelle fonction d'affectation χ

$$\chi(z) = \operatorname{argmin}_r \|z - w_r\|^2 \quad (3)$$

Phase de minimisation:

La partition χ est fixée. La fonction $I(\mathbf{W}, \chi)$ est quadratique et convexe par rapport à \mathbf{W} . Le minimum global est atteint pour

$$\frac{\partial I}{\partial \mathbf{W}} = \left[\frac{\partial I}{\partial \mathbf{w}_1}, \frac{\partial I}{\partial \mathbf{w}_2}, \dots, \frac{\partial I}{\partial \mathbf{w}_p} \right]^p = 0$$


$$\forall c \sum_{\mathbf{z}_i \in P_c} (\mathbf{z}_i - \mathbf{w}_c) = 0 \quad \mathbf{w}_c = \frac{\sum_{\mathbf{z}_i \in P_c \cap A} \mathbf{z}_i}{n_c} \quad (4)$$

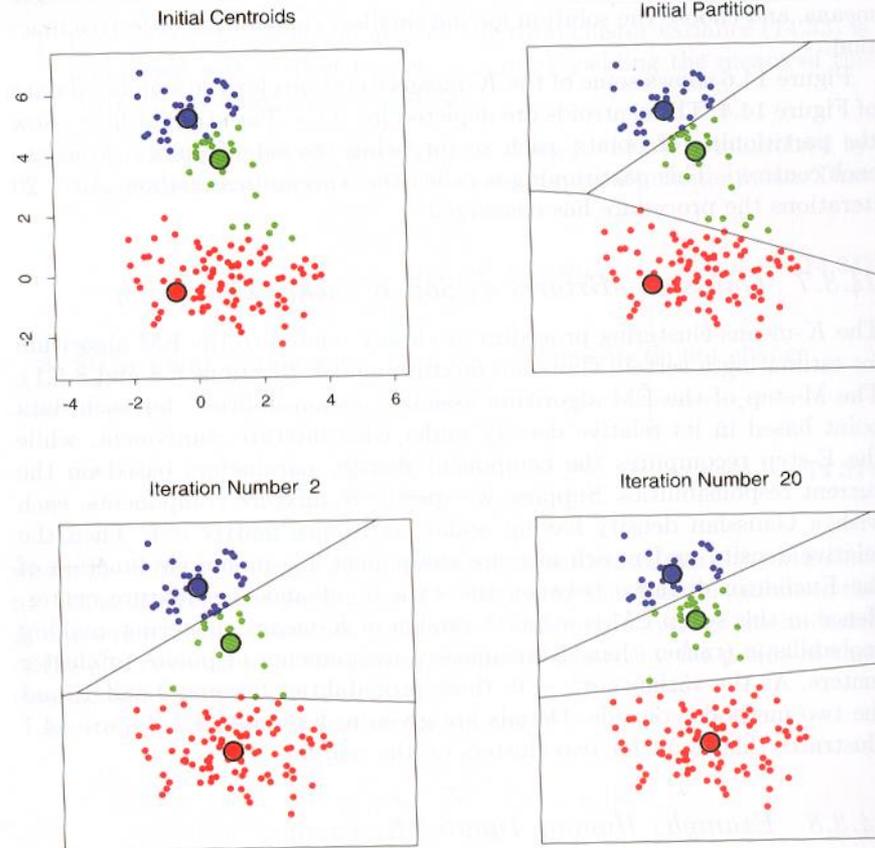
ALGORITHME

Algorithme des k-moyennes

- 1. Phase d'initialisation :** $t = 0$, choisir les p référents initiaux (en général d'une manière aléatoire), fixer le nombre maximal d'itérations N_{iter}
- 2. Étape itérative :** à l'itération t , l'ensemble des référents \mathcal{W}^{t-1} de l'étape précédente sont connus :
Phase d'affectation : mise à jour de la fonction d'affectation χ^t associée à \mathcal{W}^{t-1} : on affecte chaque observation \mathbf{z} au référent défini à partir de l'expression (3).
Phase de minimisation : calcul des nouveaux référents \mathcal{W}^t en appliquant l'équation (4).
- 3. Répéter** l'étape itérative jusqu'à ce que l'on atteigne $k > N_{iter}$ itérations ou une stabilisation de I .

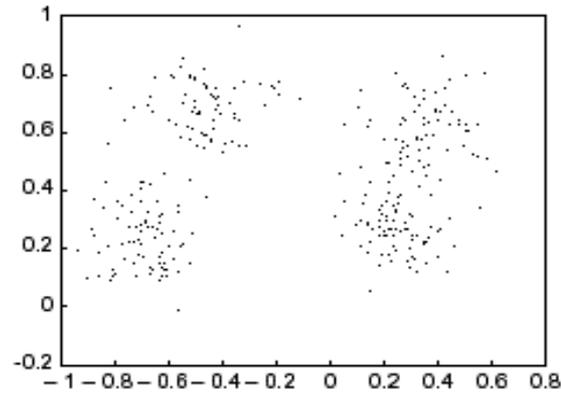
Apprentissage: itérations successives

464 14. Unsupervised Learning

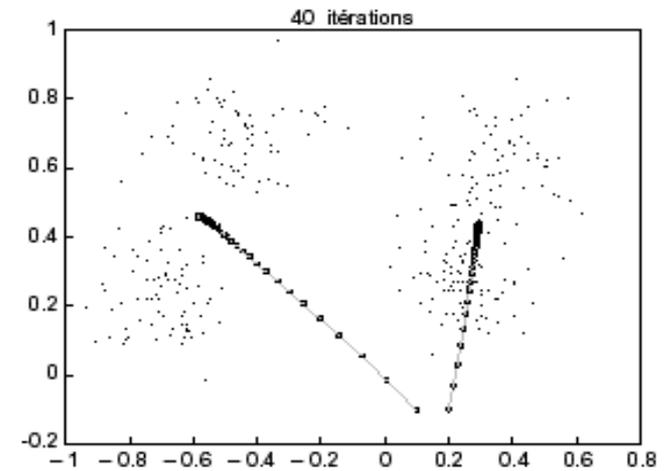


Apprentissage avec trois centres mobiles

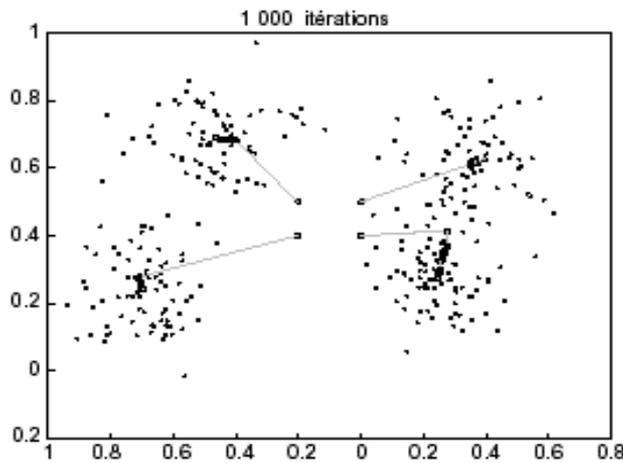
Sensibilité aux conditions initiales



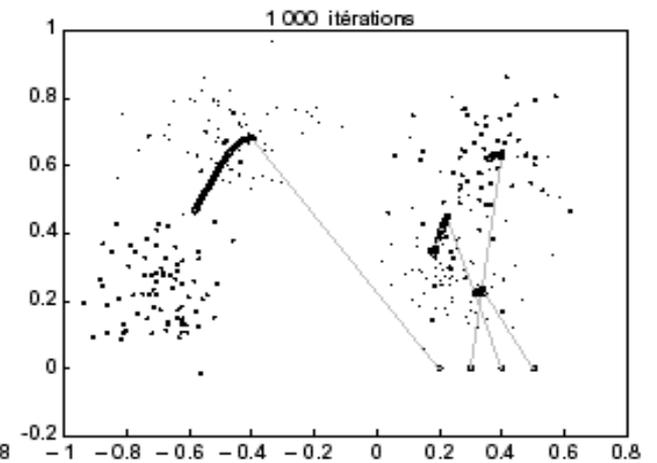
(a)



(b)

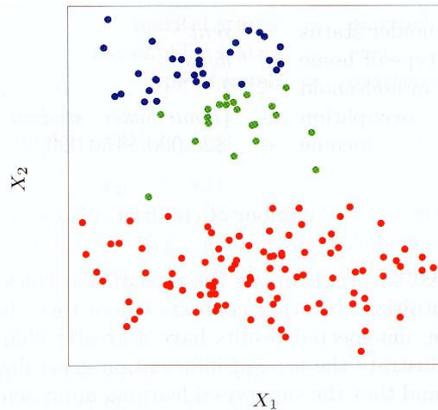


(c)

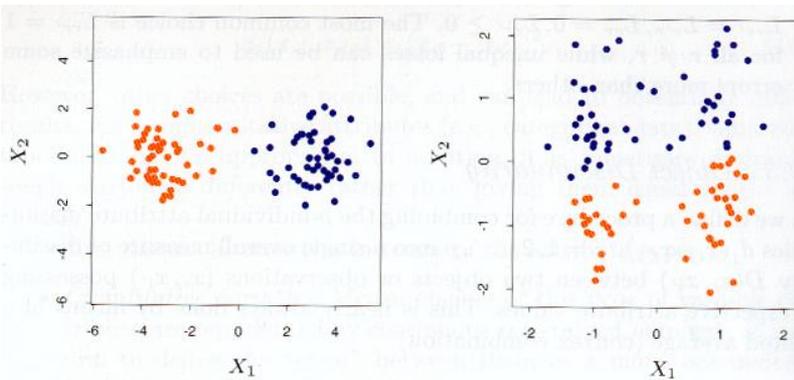


(d)

Problèmes d'apprentissages liés à l'algorithme



Les données présentent 2 groupes qui ne sont pas retrouvées par l'utilisation de 3 centres mobiles



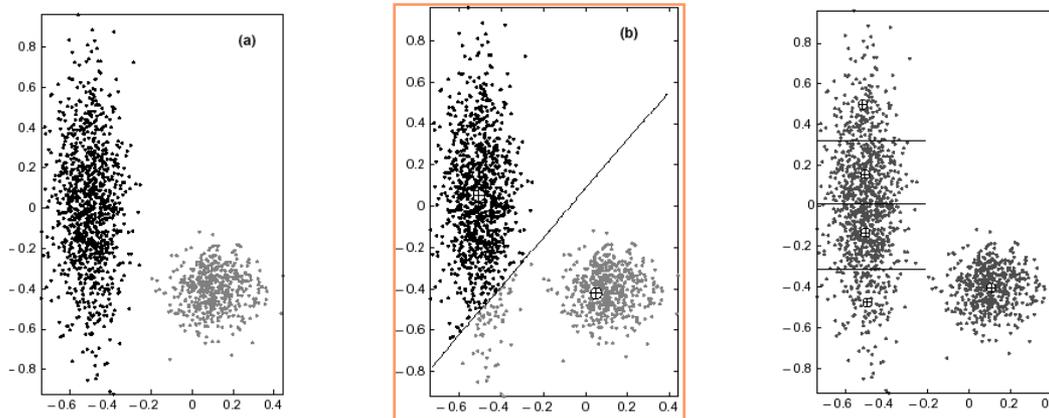
(Effet de la normalisation sur la classification)

Gauche: partition pour $k=2$ sur les données

Droite: partition pour $k=2$ obtenue après normalisation des données (centrées, réduites).

La bonne partition n'est plus retrouvée.

Comportement de l'algorithme des k-moyennes en fonction des densités sous-jacente



(a) Données simulées selon deux distributions gaussiennes de matrice de variance-covariance différentes

(b) référents et partition obtenue à la convergence avec deux référents

(c) avec cinq référents;

Cartes topologique

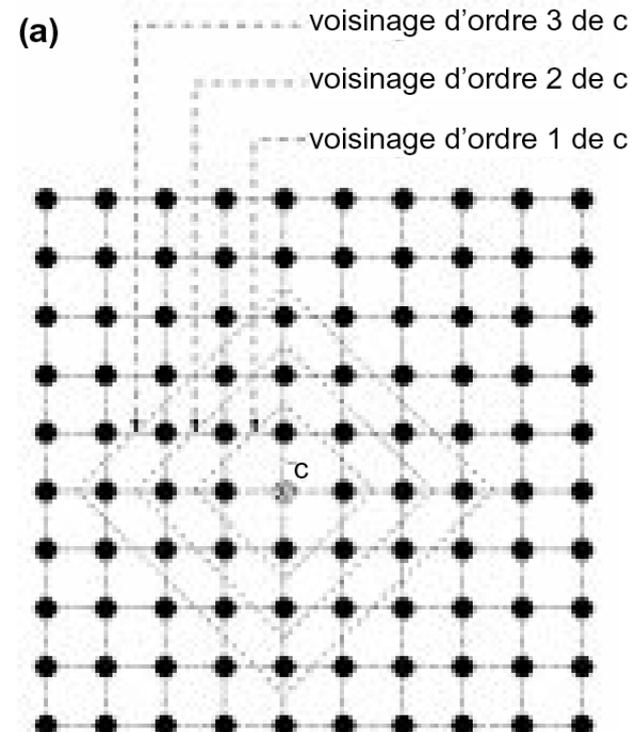
- L'ensemble des indices est maintenant ordonné.
- Il s'agit d'un espace discret (\mathcal{C}) de faible dimension à des fins de visualisation (1-D, 2-D).
- \mathcal{C} ensemble de **neurones** connectés par une structure de graphe non-orienté muni d'une distance discrète δ sur \mathcal{C} et d'une **structure de voisinage**:

$$V_c(d) = \{r \in \mathcal{C}, \delta(c, r) \leq d\}$$



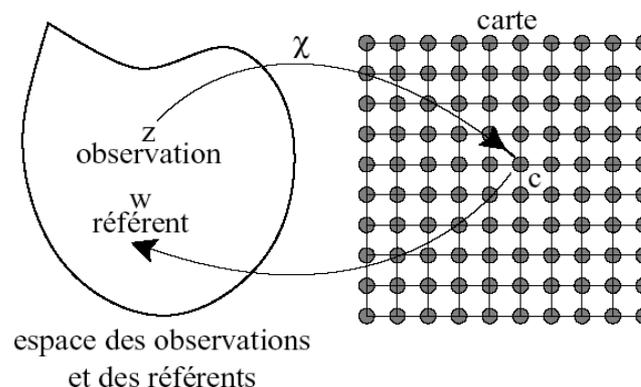
Voisinage d'ordre d

$\delta(c,r) =$ + court chemin sur le graphe



Quantification par carte

- Chaque neurone c de C est associé à un vecteur référent \mathbf{w}_c de l'espace des données D
- L'apprentissage approxime la densité sous-jacente des données tout en cherchant à respecter une contrainte de conservation de la topologie de la carte C
- *Deux neurones c et r « voisins » par rapport à la topologie discrète de la carte C sont associés à deux vecteurs référents \mathbf{w}_c et \mathbf{w}_r proches dans l'espace des données D .*



Les indices des référents sont ordonné par la topologie

NOTION DE CONSERVATION DE LA TOPOLOGIE



Apprentissage

Apprentissage de la carte = Minimisation d'une fonction de coût

$$J_{som}^T(\chi, \mathcal{W}) = \sum_{\mathbf{z}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} K^T(\delta(c, \chi(\mathbf{z}_i)) \| \mathbf{z}_i - \mathbf{w}_c \|^2)$$

La valeur de T détermine la taille du voisinage

$$V_c^T = \{r \in \mathcal{C} / K^T(\delta(c, r)) > \alpha\}.$$

(T très proche de 0 → Algorithme des K-moyennes)

Le seuil α gère l'ordre des valeurs significatives prises en compte par le calcul.

- Chaque observation \mathbf{z} calcule une distance généralisée à l'ensemble des neurones de la carte par l'intermédiaire de la fonction K^T :

$$d^T(\mathbf{z}_i, \mathbf{w}_{\chi(\mathbf{z}_i)}) = \sum_{c \in \mathcal{C}} K^T(\delta(c, \chi(\mathbf{z}_i)) \| \mathbf{z}_i - \mathbf{w}_c \|^2)$$

Optimisation non adaptative des cartes topologiques

Minimisation itérative de $J^{\text{som}}(\chi, \mathcal{W})$ pour un paramètre T fixé:

Phase d'affectation:

L'ensemble \mathcal{W} des référents est fixé, la minimisation s'obtient en affectant chaque observation \mathbf{z} au référent \mathbf{w}_c selon la nouvelle fonction d'affectation χ^T

$$(18) \quad \chi^T(\mathbf{z}) = \arg \max_{r \in \mathcal{C}} \sum_{c \in \mathcal{C}} K_T(\delta(c, r)) \|\mathbf{z} - \mathbf{w}_c\|^2 = \arg \max_{r \in \mathcal{C}} d^T(\mathbf{z}, \mathbf{w}_r)$$

Phase de minimisation:

La partition χ^T est fixée. La fonction $J^{\text{som}}(\chi, \mathcal{W})$ est minimisée par rapport à l'ensemble des référents \mathcal{W} . La fonction étant convexe par rapport aux paramètres le minimum global est atteint pour

$$\mathbf{w}_c^T = \frac{\sum_{r \in \mathcal{C}} K(\delta(c, r)) \mathbf{Z}_r}{\sum_{r \in \mathcal{C}} K(\delta(c, r)) n_r}$$

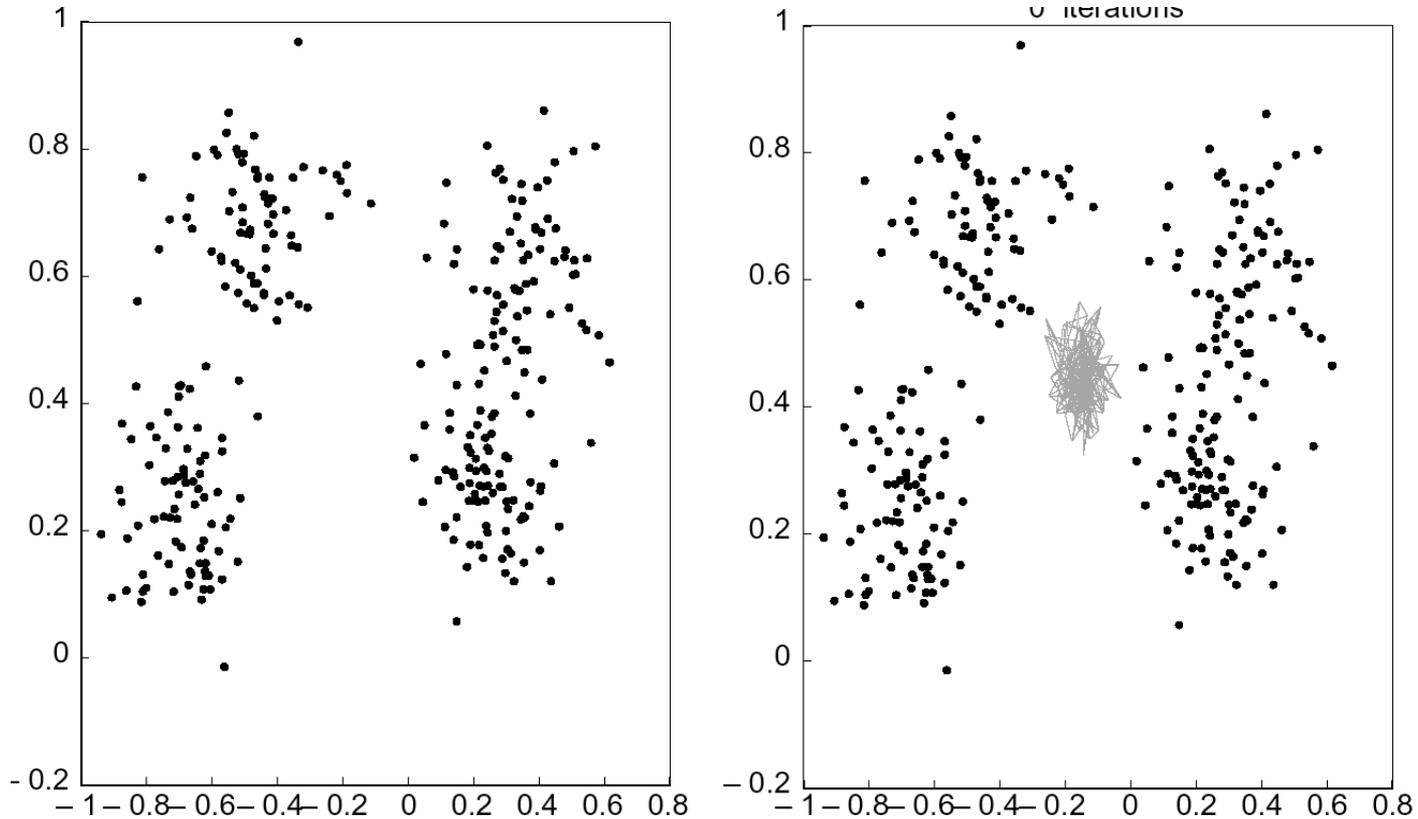
\mathbf{Z}_r représente la *somme* de toutes les observations affectées au neurone r
 n_r le *nombre* de ces observations

Minimisation à voisinages constants

Algorithme « nuées dynamiques » des cartes topologiques : T fixé

- 1. Phase d'initialisation : $t = 0$** Choisir les p référents initiaux (en général, d'une manière aléatoire), la structure et la taille de la carte, le nombre d'itérations N_{iter}
- 2. Étape itérative t .** L'ensemble des référents W^{t-1} de l'étape précédente est connu,
 - phase d'affectation : mise à jour de la fonction d'affectation χ^t associé à W^{t-1} . On affecte chaque observation \mathbf{z}_i au référent défini à partir de l'expression (18) ;
 - étape de minimisation : appliquer l'équation (19) afin de déterminer l'ensemble des nouveaux référents W^t .
- 3. Répéter** l'étape itérative jusqu'à ce que l'on atteigne N_{iter} itérations ou une stabilisation de J_{som}^T .

Exemple: Données simulées



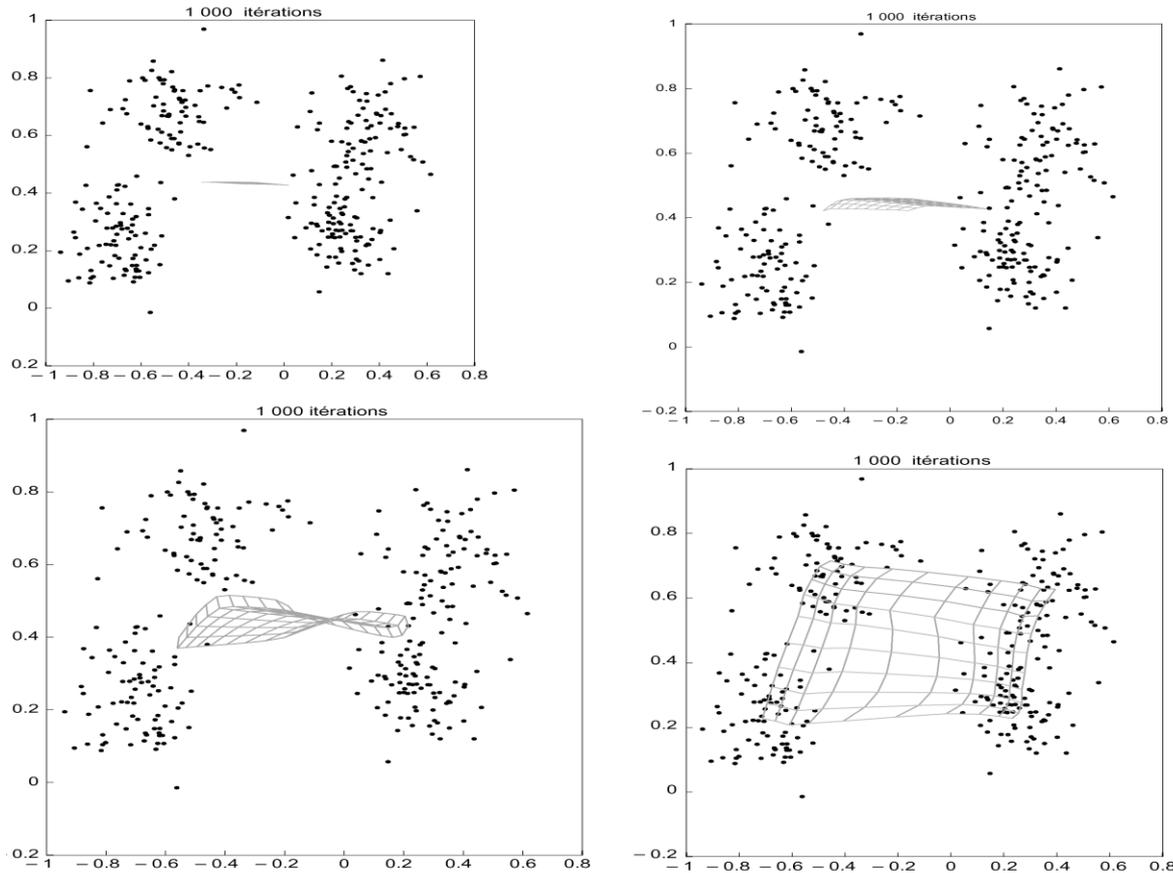
Données simulées selon quatre gaussiennes qui se recouvrent partiellement
Initialisation des référents selon une loi normale $N(0, 0.01)$

A gauche les données

A droite: *Représentation simultanée des données et des référents au moment de la phase d'initialisation de l'algorithme.*

Carte topologique obtenue à la convergence pour T fixé

de haut en bas et de gauche à droite $T=10$, $T=5$; $T=3$; $T=1$



*L'ordre induit par la carte sur les référents permet de visualiser
L'ordre topologique*

Algorithme des cartes topologiques

- La minimisation à T fixée est répétée un certain nombre de fois en faisant décroître la valeur de T .
- L'ordre apparaît pour les grandes valeurs de T , la carte se déploie alors pour recouvrir les données et faire diminuer la variance intra.

Les paramètres déterminants de la minimisation sont

- *L'intervalle de variation de T , la valeur initiale T^{\max} et la valeur finale T^{\min}*
- *Le nombre de fois où l'étape itérative est effectuée*
- *La manière dont le paramètre décroît dans l'intervalle $[T^{\max}, T^{\min}]$*

Optimisation globale des cartes topologiques

- 1. Phase d'initialisation.** Effectuer l'algorithme nuées dynamiques des cartes auto-organisatrices pour la valeur $T = T_{max}$, $t = 0$
- 2. Étape itérative t .** L'ensemble des référents ${}^{\circ}W^{t-1}$ de l'étape précédente sont connus. Calculer la nouvelle valeur de T en appliquant la formule :

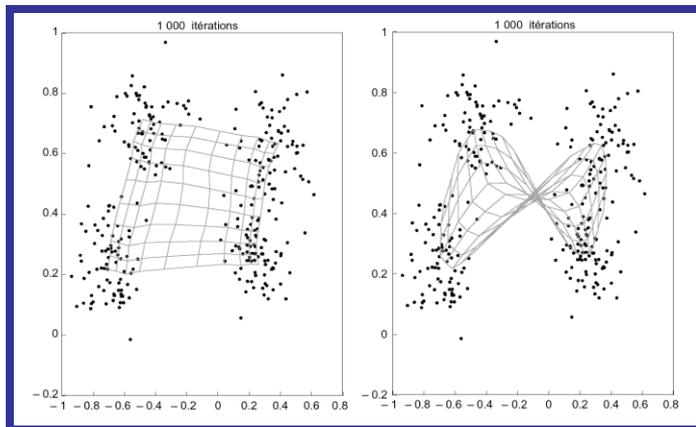
$$T = T_{max} \star \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter} - 1}} \quad (20)$$

Pour cette valeur du paramètre T , effectuer les deux phases suivantes :

phase d'affectation : mise à jour de la fonction d'affectation χ^t associée à ${}^{\circ}W^{t-1}$. On affecte chaque observation z_i au référent défini à partir de l'expression (18) ;

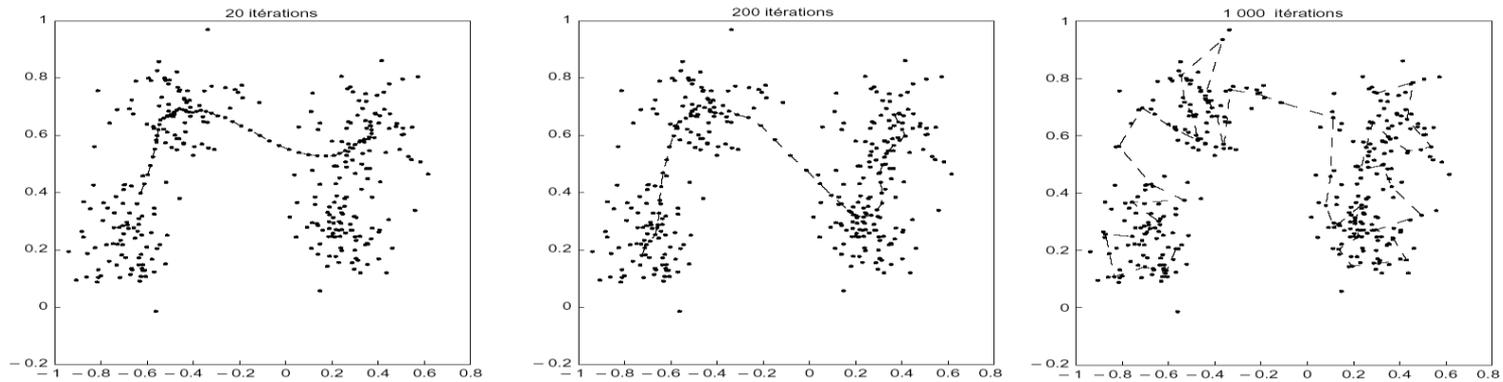
phase de minimisation : appliquer l'équation (19) afin de déterminer l'ensemble des nouveaux référents ${}^{\circ}W^t$.

- 3. Répéter** l'étape itérative jusqu'à que l'on atteigne $T = T_{min}$.

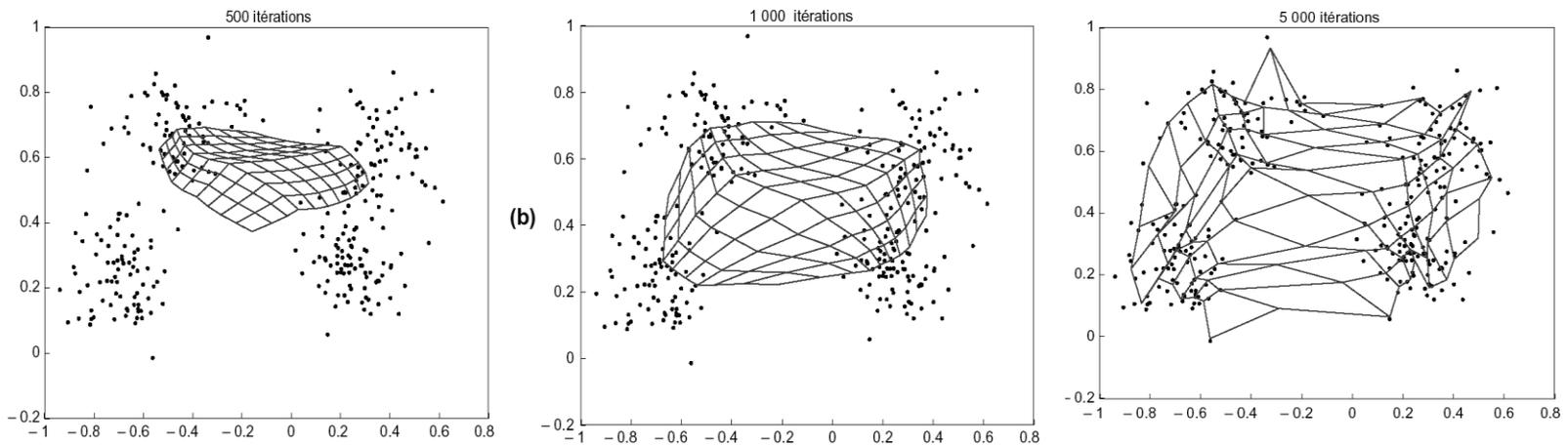


Une mauvaise gestion de la décroissance de T peut introduire des anomalies dans l'ordre

Evolution de l'apprentissage: données simulées 4 gaussiennes

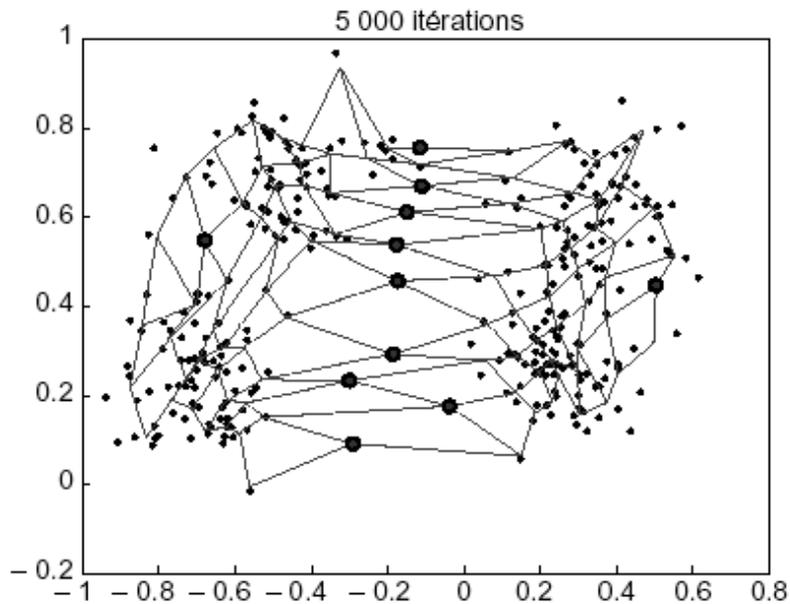


Carte 1-D 50 neurones après 20, 200, 1000 itérations



Carte 2-D 10*10 neurones après 500, 1000, 5000 itérations

Visualisation des différents sous-ensembles



Les frontières dessinées par les neurones n'ayant rien captés permettent de distinguer grossièrement les différents groupes.

Dans l'exemple les observations ont été découpées en deux sous-ensembles regroupant à chaque fois deux gaussiennes.

L'algorithme de Kohonen

Algorithme de Kohonen

1. Phase d'initialisation

- choisir la structure et la taille de la carte et les p référents initiaux (en général, d'une manière aléatoire) ;
- fixer les valeurs de T_{max} , T_{min} et le nombre d'itérations N_{iter} ; prendre $t = 0$.

2. Étape itérative t : l'ensemble des référents ${}^qW^{t-1}$ de l'étape précédente étant connus :

- choisir une observation \mathbf{z}_i (en général, d'une manière aléatoire) ;
- calculer la nouvelle valeur de T en appliquant la formule :

$$T = T_{max} \star \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}} \quad (23)$$

Pour cette valeur du paramètre T effectuer les deux phases suivantes :

- phase d'affectation : on suppose ${}^qW^{t-1}$ connu ; on affecte l'observation \mathbf{z}_i au neurone $\chi^t(\mathbf{z}_i)$ défini à partir de la fonction d'affectation (3) ;
- phase de minimisation : calcul de l'ensemble des nouveaux référents ${}^qW^t$; les vecteurs référents sont modifiés selon la formule (22) en fonction de leur distance au neurone sélectionné à l'étape d'affectation.

3. Répéter l'étape itérative en faisant décroître la valeur du paramètre T jusqu'à ce que l'on atteigne $t = N_{iter}$

Ordre Topologique

$$J_{som}^T = \sum_r \sum_{\mathbf{z}_i \in P_r \cap A} \sum_c K^T(\delta(c, r)) \|\mathbf{z}_i - \mathbf{w}_c\|^2 = \sum_c \sum_r \sum_{\mathbf{z}_i \in P_r} K^T(\delta(c, r)) \|\mathbf{z}_i - \mathbf{w}_c\|^2$$

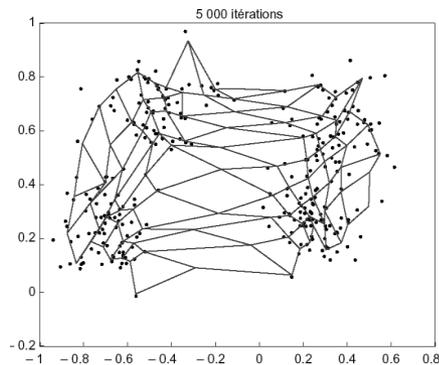
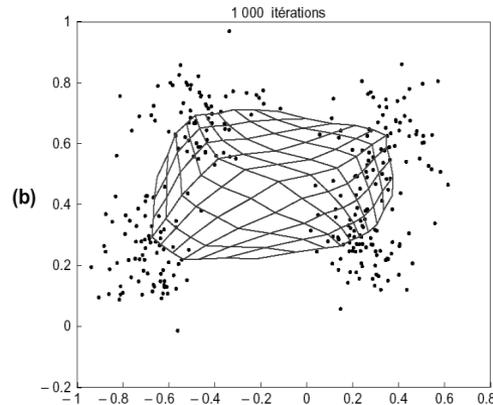
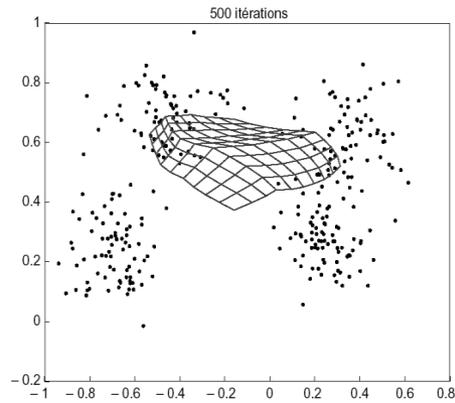
$$\begin{aligned} J_{som}^T &= \left[\sum_c \sum_{r \neq c} \sum_{\mathbf{z}_i \in P_r} K^T(\delta(c, r)) \|\mathbf{z}_i - \mathbf{w}_r\|^2 \right] + \left[K^T(\delta(c, c)) \sum_c \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_c\|^2 \right] \\ &= \frac{1}{2} \sum_c \sum_{r \neq c} K^T(\delta(c, r)) \left[\sum_{\mathbf{z}_i \in P_r} \|\mathbf{z}_i - \mathbf{w}_c\|^2 + \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_r\|^2 \right] \\ &\quad + K^T(\delta(c, c)) \sum_c \sum_{\mathbf{z}_i \in P_c} \|\mathbf{z}_i - \mathbf{w}_c\|^2. \end{aligned}$$

Induit l'ordre sur la carte, deux neurones proches doivent avoir des référents proches

Fonction de coût des k-moyennes pondéré par $K^T(0)$

T petit: carte topologique et k-moyennes coincident

Déroulement de l'algorithme



Première phase valeur de T grande, minimisation du premier terme de J

L'ordre se met en place

Deuxième phase valeur de T petite, minimisation du second terme de J

La variance intra est minimisée, les référents deviennent les centres de gravité de la partition

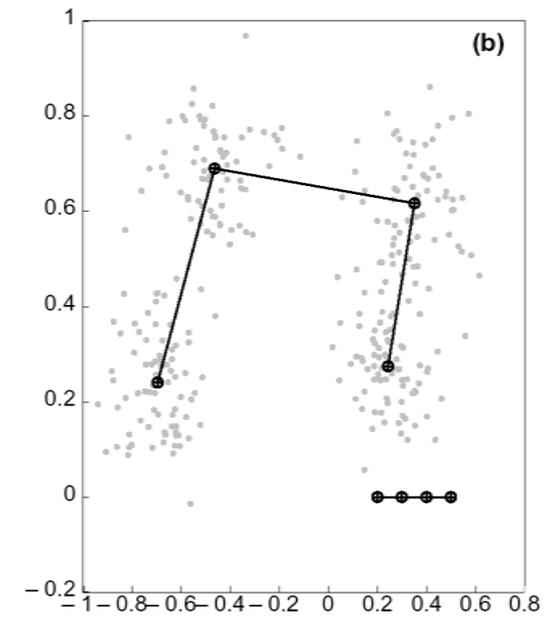
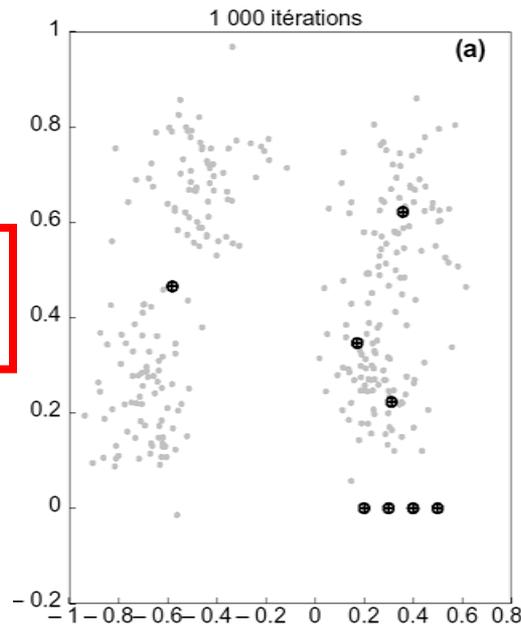
Interprétation des cartes topologique

CARTE TOPOLOGIQUE =

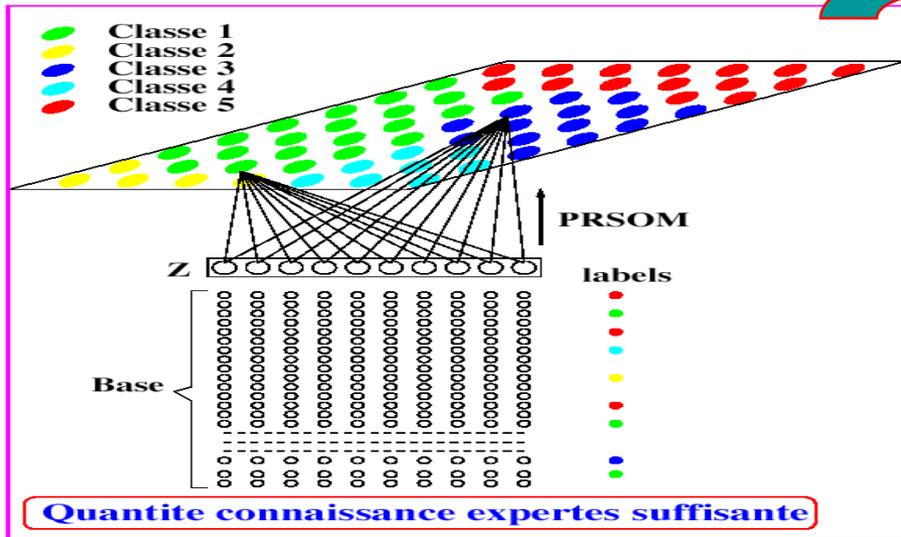
K-MOYENNES SOUS CONTRAINTE D'ORDRE

ou K-MOYENNES REGULARISE

*Comparaison: algorithme des
k-moyennes, carte topologique*



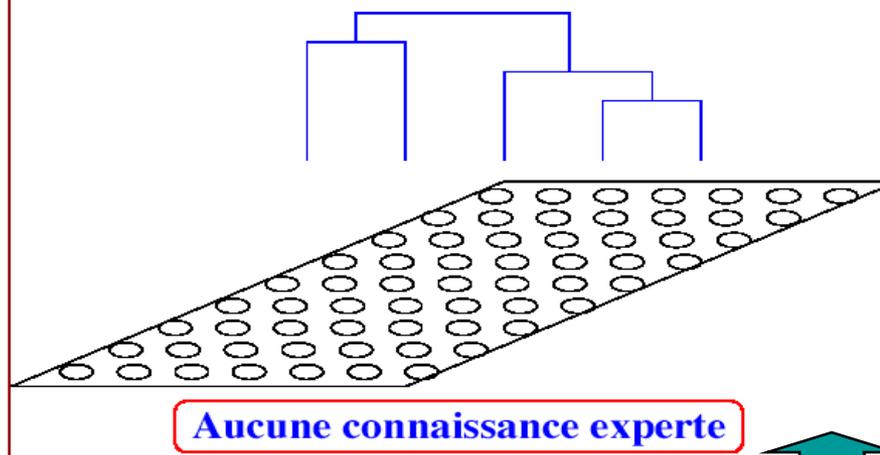
Classification et carte topologique



Base étiquetée statistiquement
représentative:

1. Affectation à l'aide de la fonction χ
2. Utilisation du vote majoritaire sur les différents sous-ensemble de la partition
3. Etiquetage du neurone

CAH sur les neurones + interpretation



Pas de connaissance experte

Sous l'hypothèse d'un bon ordre topologique

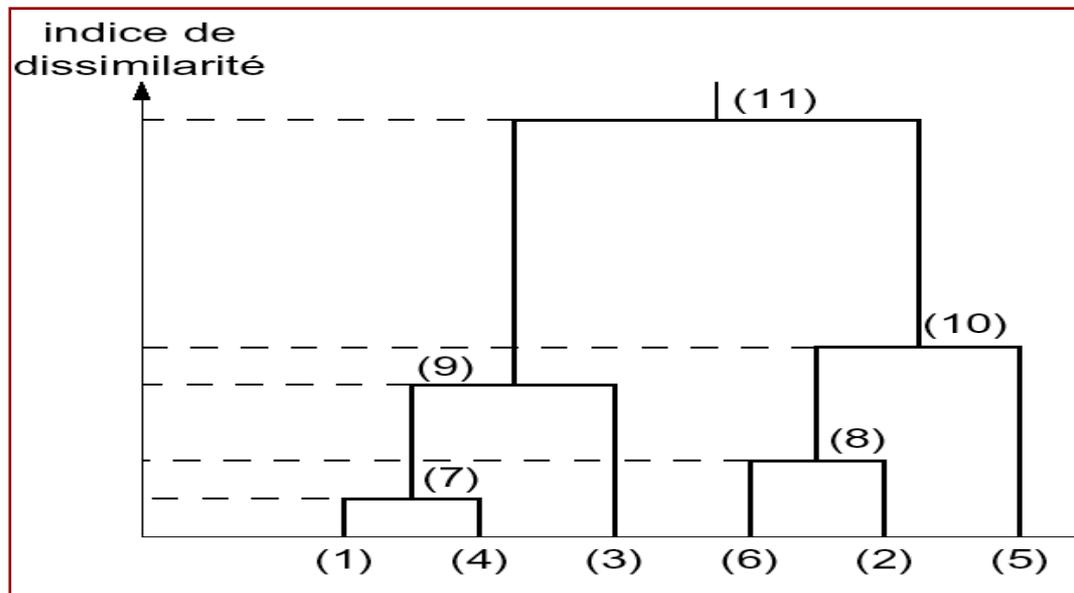
Il est très probable que deux neurones voisins
représentent des données de même classe.

Une Classification Ascendante hiérarchique
Permet de les retrouver

Classification hiérarchique

Algorithme de classification hiérarchique

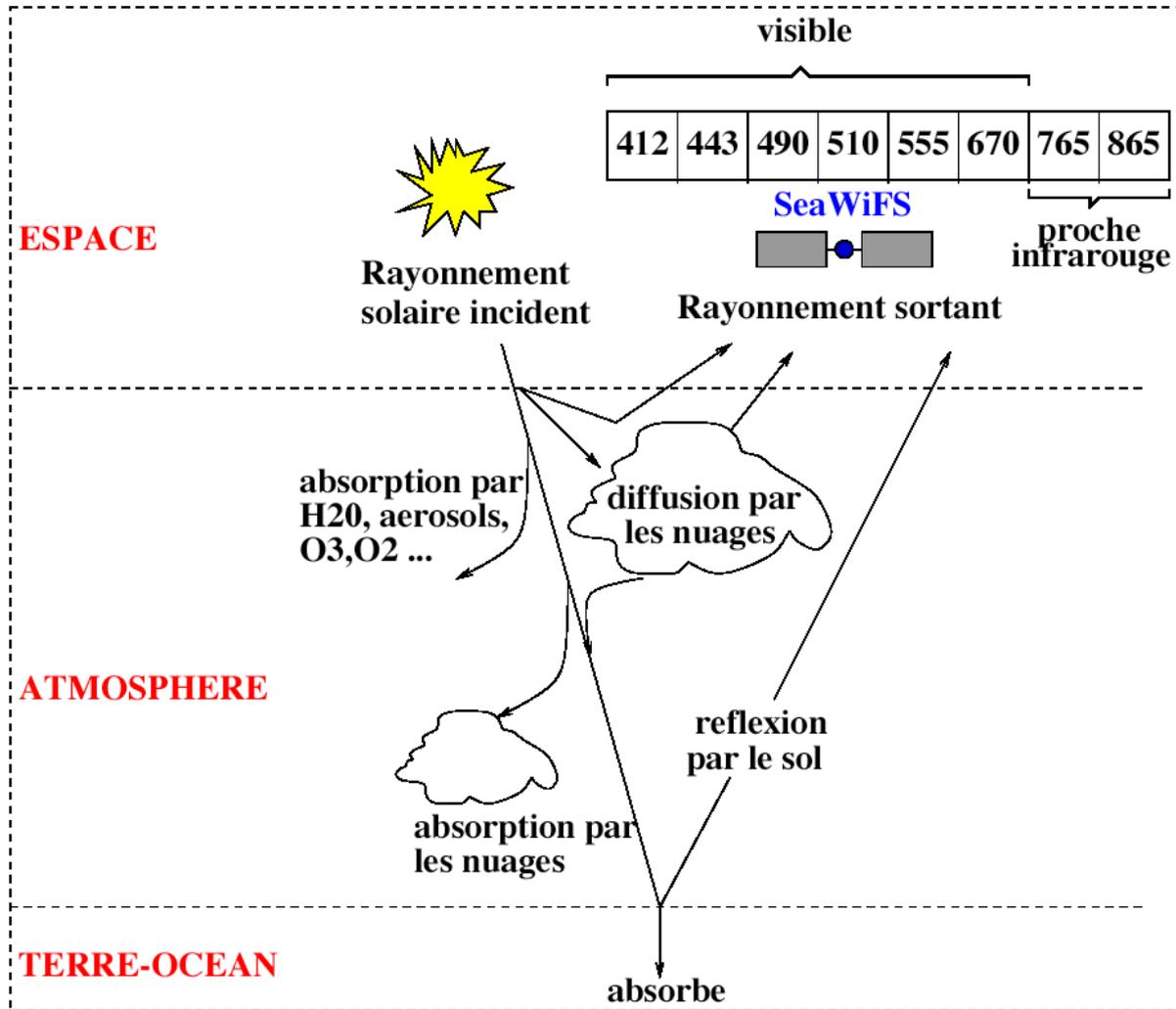
- 1. Initialisation.** Considérer la partition formée par les singletons ; chaque neurone est alors affecté à un sous-ensemble distinct. Choisir le nombre K de groupement que l'on souhaite former.
- 2.** Pour une partition donnée, trouver les deux sous-ensembles les plus proches au sens du critère de similitude choisi, et les fusionner de manière à former un seul sous-ensemble.
- 3.** Si le nombre de groupement de la partition courante est inférieur à K , revenir à l'étape (2), sinon l'algorithme se termine.



APPLICATION DES CARTES TOPOLOGIQUES

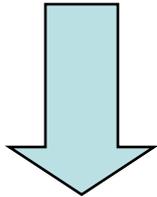
**Une application en télédétection satellitaire:
Etude de la couleur de l'océan**

Le problème physique



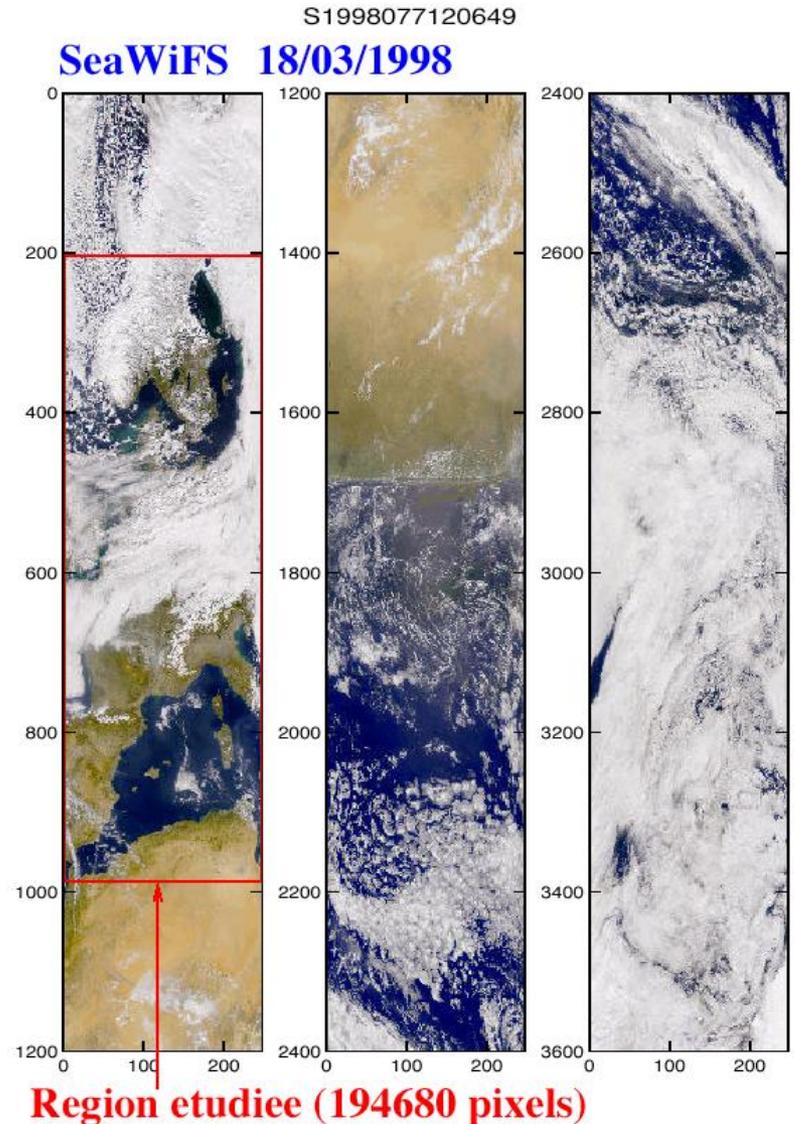
Les données

Image SeaWiFS (**Polder**): analyse de la même scène selon 8 (5) longueurs d'onde



On cherche à déterminer les différents types d'aérosols

Bandes k	Longueurs d'ondes (nanomètres) λ_k
1	412
2	443
3	490
4	510
5	555
6	670
7	765
8	865



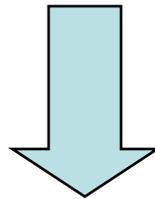
LES PREMIERES EXPERIENCES: ETUDE SUR UNE IMAGE

Les données: image SeaWifs sur la Afrique de l 'ouest et les Iles Canaries
5 janvier 1999 échantillonnée d'une manière homogène (1 ligne sur
10). Chaque observation est constitué par les 8 longueurs d'onde.

Apprentissage: carte topologique 10*10 (algorithme PRSOM),

Classification hiérarchique : 2 classes (Séparation mer nuages)

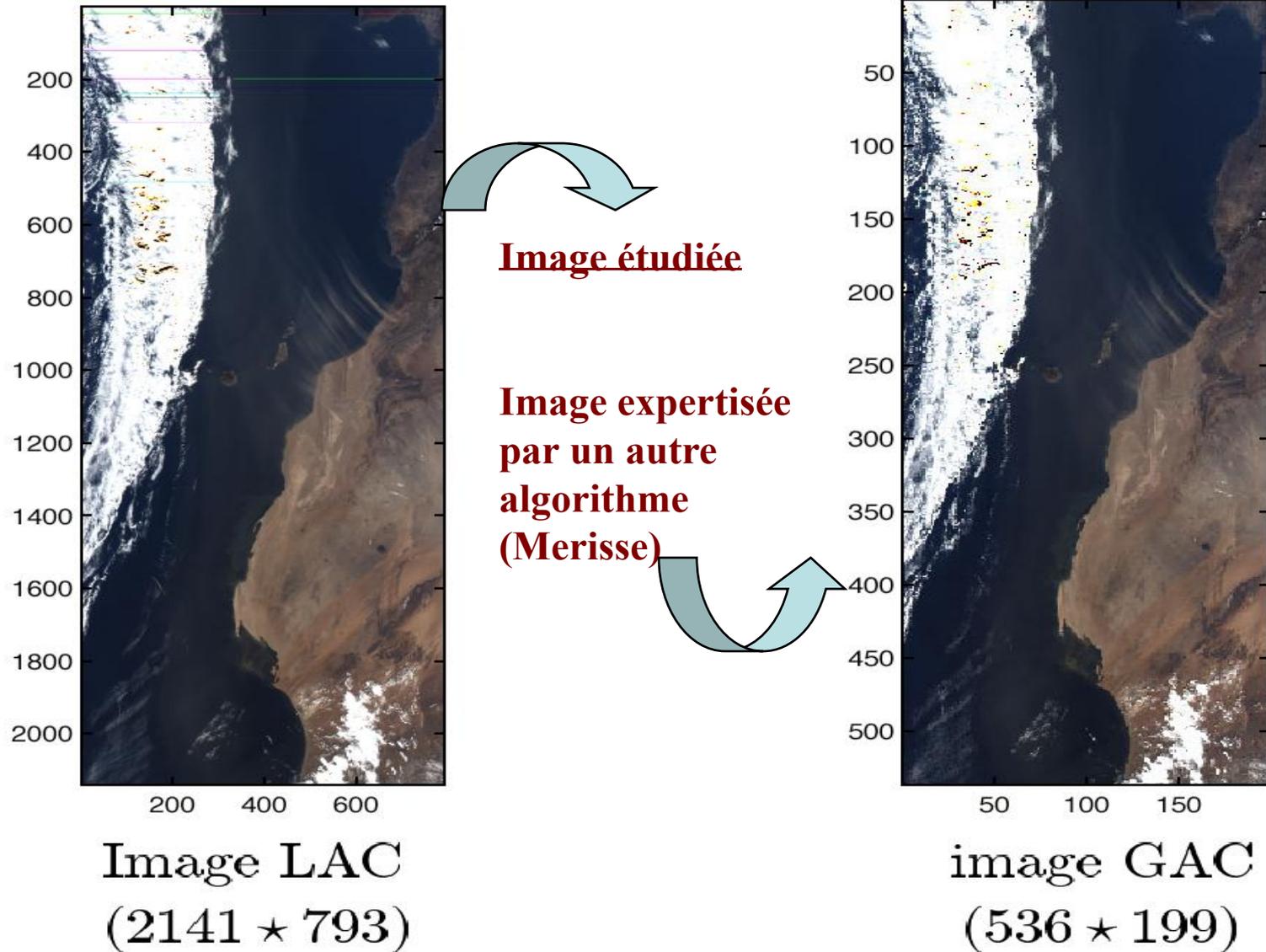
4 classes (Recherche des types d 'aérosols)



TEST

Image entière

Image SeaWifs Afrique de l'ouest et Iles Canaries



Visualisation de la carte

696	863	638	551	685	305	418	408	1 041	495
1	2	3	4	5	6	7	8	9	10
1 553	546	264	409	718	953	1 035	940	745	533
11	12	13	14	15	16	17	18	19	20
1 053	1 036	1 578	839	167	512	1 035	940	1 041	495
21	22	23	24	25	26	27	28	29	30
1 098	792	142	738	168	514	686	434	671	1 191
31	32	33	34	35	36	37	38	39	40
1 004	1 206	657	550	397	401	452	506	402	361
41	42	43	44	45	46	47	48	49	50
1 755	576	1 441	933	292	459	471	521	1 301	475
51	52	53	54	55	56	57	58	59	60
1 706	889	1 406	391	569	480	512	4	556	584
61	62	63	64	65	66	67	68	69	70
3 372	1 506	458	510	567	512	314	651	495	608
71	72	73	74	75	76	77	78	79	80
2 016	619	365	472	616	644	339	655	442	419
81	82	83	84	85	86	87	88	89	90
2 873	877	697	487	643	620	592	750	718	932
91	92	93	94	95	96	97	98	99	100

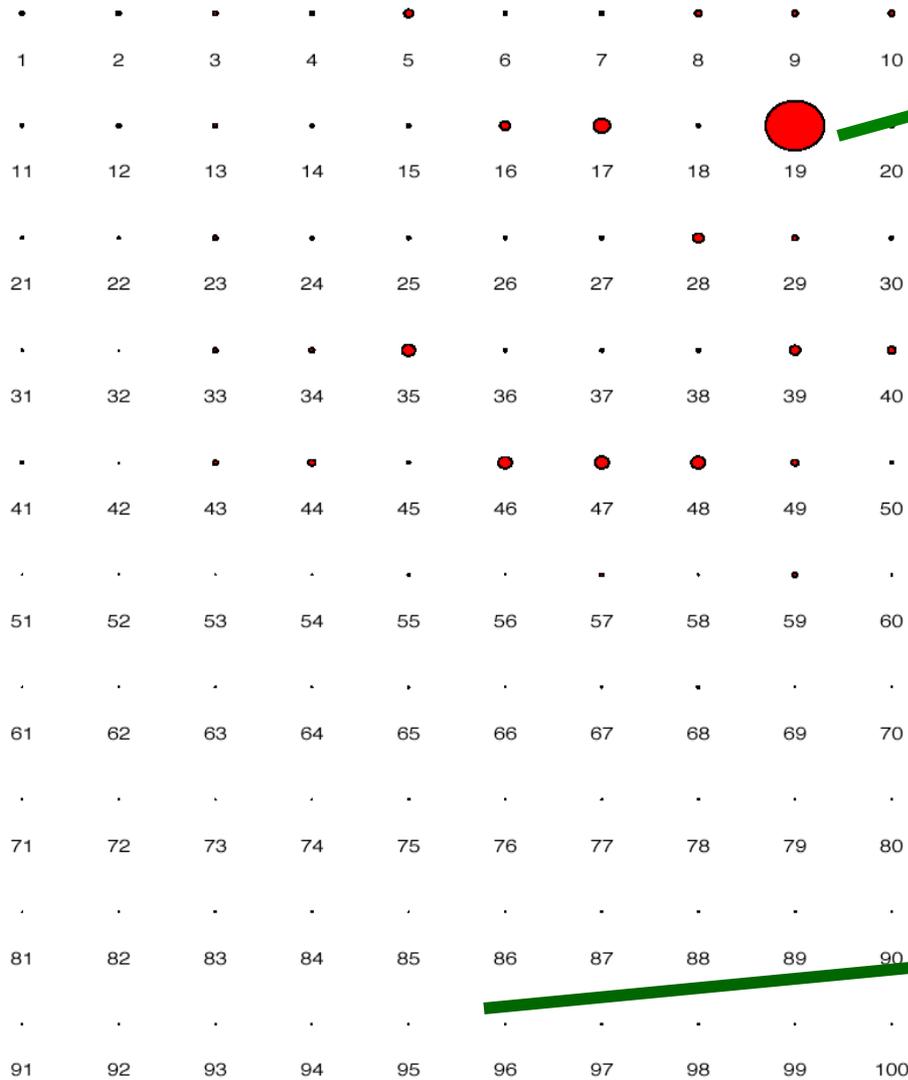
Numéro du neurone

Nombre de pixels captés

Le nombre de pixel capté par chaque neurone est une première indication sur la densité des données (spectre de R^8)

Carte SOM (10*10)

VISUALISATION DES VARIANCES



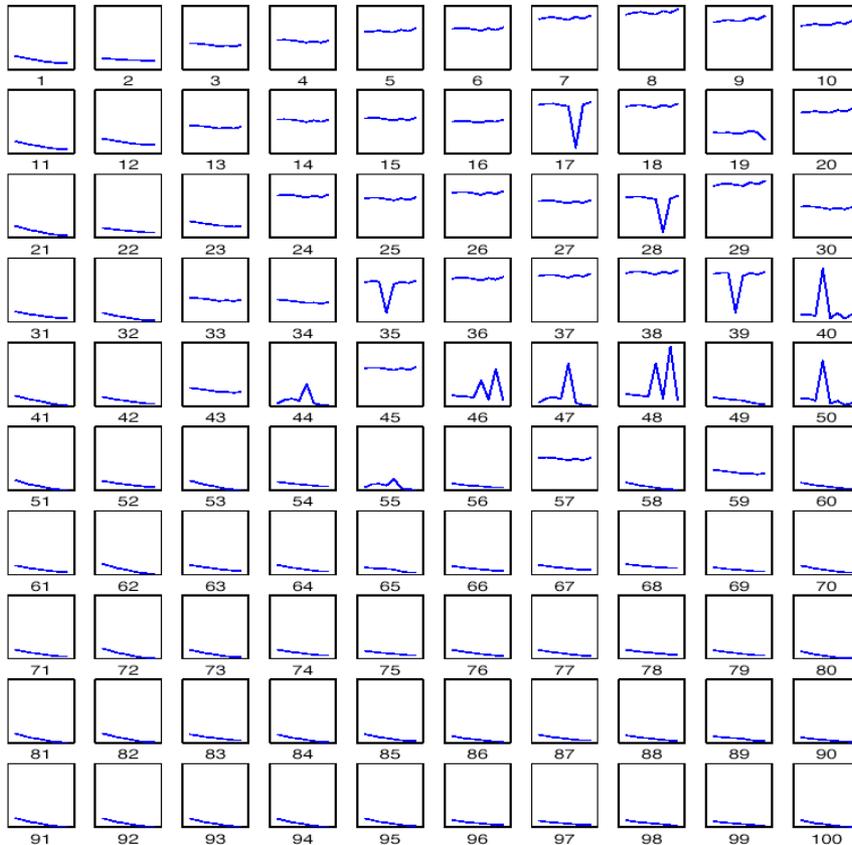
Zone de grande
variabilité =
nuage

Carte PRSOM (10*10)

La surface du disque est
proportionnelle
à la variance de la
gaussienne attachée au
neurone. Elle permet une
première analyse de la carte

Zone très dense = **mer**

VISUALISATION SIMULTANEE DES REFERENTS ET DE LA CARTE

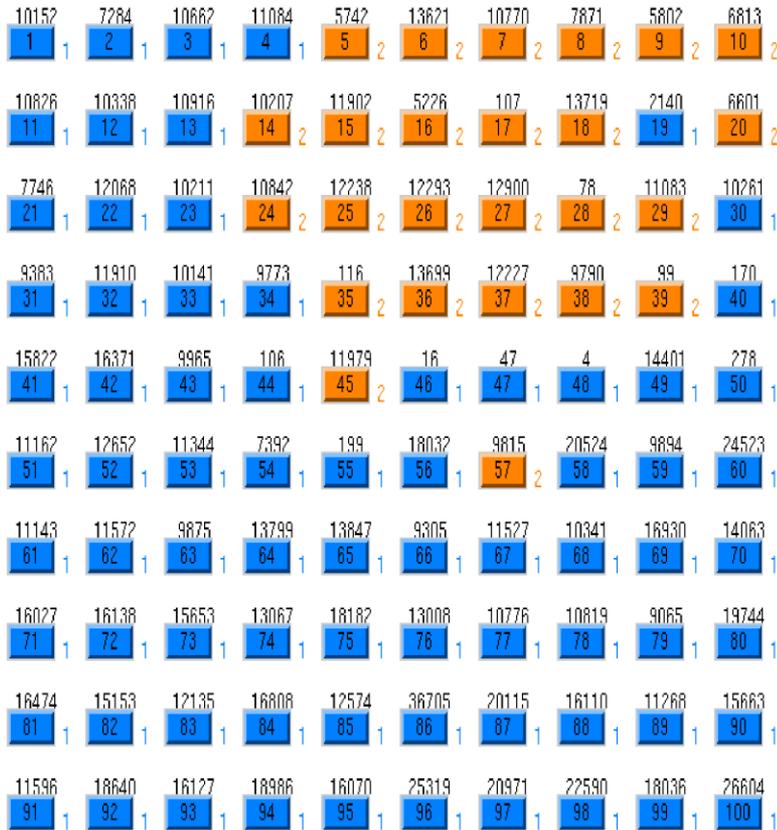


Permet d'interpréter l'ordre topologique obtenu en fonction des données. Chaque référent est un spectre de \mathbf{R}^8 .

La forme des spectres des référents varie de manière ordonnée selon les deux directions.

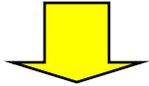
Carte SOM (10*10) codage par intensité (codage1)

Comparaison de la classification SOM+CAH en deux classes avec le produit distribué par SeaWifs



Carte SOM 10*10

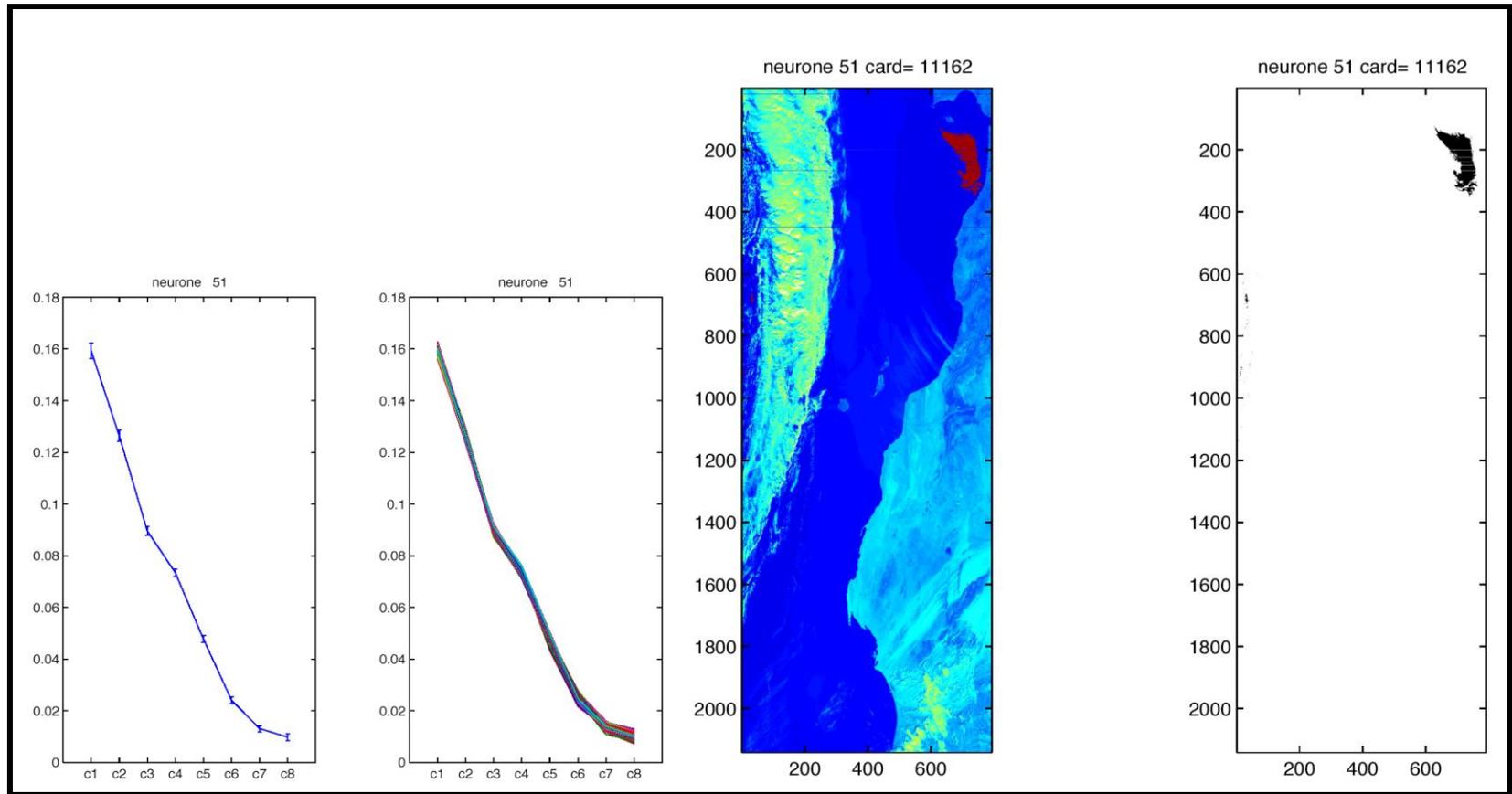
	PR SOM + CAH	
	Nuages	Mer visible
Nuages SeaWiFS	0.91	0.09



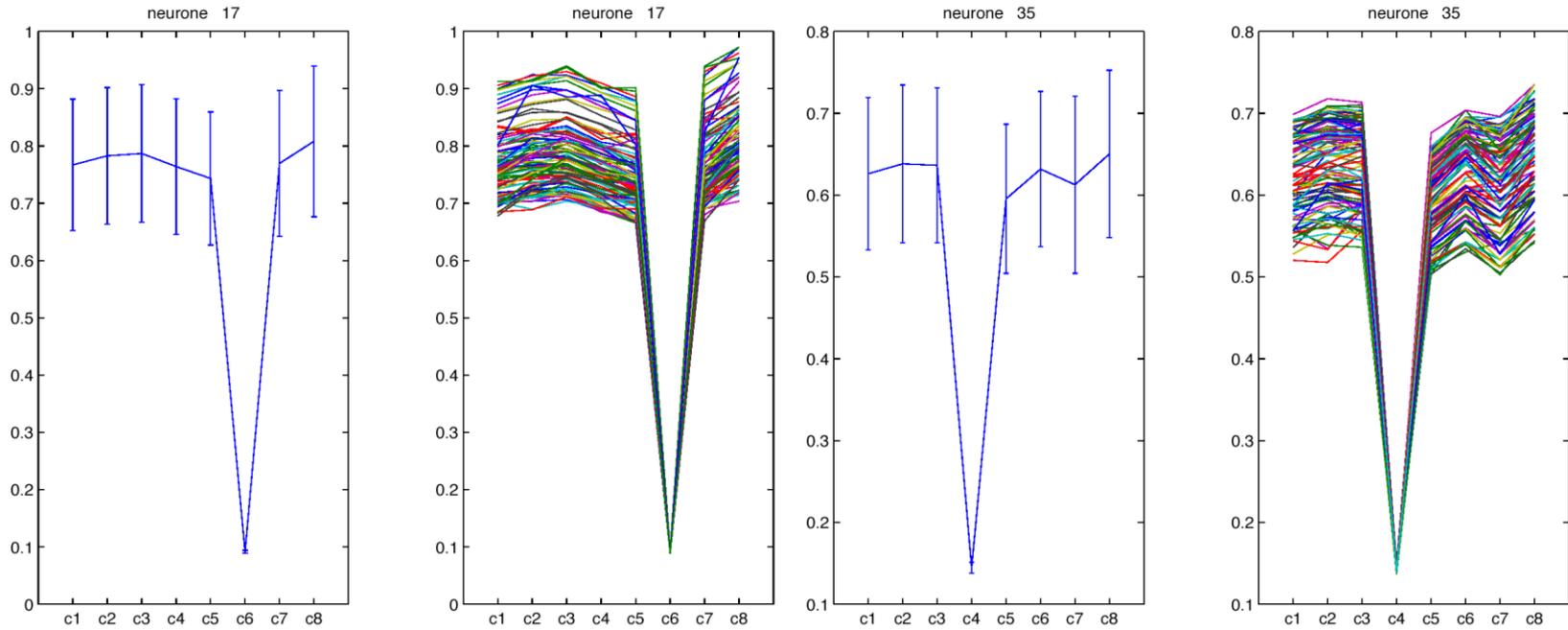
9% des pixels ne sont pas écartés par SOM et peuvent être analysés en terme d'aérosols

Visualisation d'un neurone, des spectres associés

Localisation du sous-ensemble de pixels sur l'image



Visualisation de la partition: détection des données aberrantes

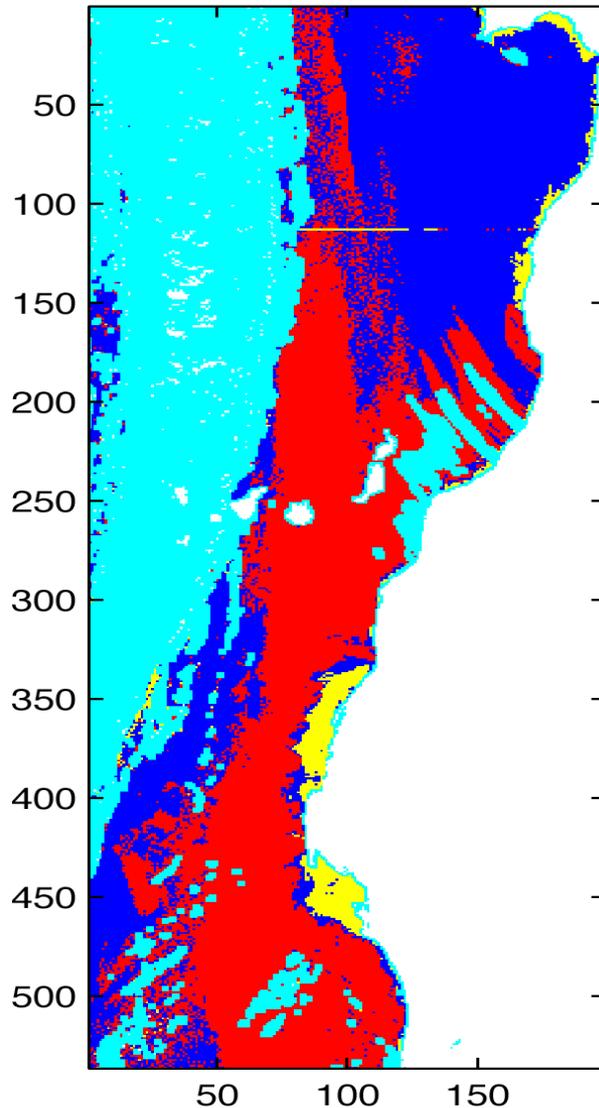


Spectre et variance des neurones 17 et 32

Spectres des pixels de l'image affectés au neurone 16, 32

Tous les pixels pour lesquels un canal n'a pas fonctionné sont affectés au même neurone

Image GAC étiqueté à l'aide d'un autre algorithme (Merisse)



Aérosol désertique: bleu foncé

Aérosol marin: rouge

Nuages: bleu clair

Eaux du cas2: jaune

La nature des aérosol ne pouvant pas être observée, la comparaison avec d'autres méthodes permet de juger de la cohérence de la classification obtenue.

Les types proposés par l'algorithme Merisse sont utilisés pour étiqueter les neurones

Matrice de confusion

comparaison entre la classification de la carte et celle de l'expert

		PRsOM + vote majoritaire			
		Classe1	Classe2	Classe3	Classe4
Expert	Aérosols marins	0.8	0.04	0	0.16
	nuages	0.03	0.91	0.01	0.05
	Cas 2	0.03	0.22	0.71	0.03
	Aérosols désertiques	0.1	0.04	0	0.86