

**ClaSpec : Classification spectrale pour la segmentation
d'images couleur et de signaux audio**

Réunion : Présentation et analyse d'articles de base
sur la classification spectrale

**On Spectral Clustering: Analysis and an
algorithm**

Andrew Y. Ng, Michael I. Jordan and Yair Weiss

In Advances in Neural Information Processing Systems 14, 2001

Première partie : Illustration de la méthode et de quelques points théoriques

- Présentation d'un algorithme pour l'analyse spectrale de données
- Cette méthode est fondée sur l'utilisation des vecteurs propres principaux d'une matrice définie à partir de la distances entre les vecteurs à classer
- L'objectif de ce papier est de montrer les limites de convergence de cet algorithme

Les étapes de l'algorithme (1/2)

- Soient
 - $S = \{s_1, s_2, \dots, s_n\}$ un ensemble de vecteurs de \mathbb{R}^l
 - k le nombre de classes recherchées (supposé connu)
- Les étapes
 - 1) Création d'une matrice A de $\mathbb{R}^{n \times n}$ dont les valeurs A_{ij} décroissent en fonction de la distance entre les vecteurs s_i et s_j
 - 2) Définir la matrice diagonale D dont chaque valeur A_{ii} est la somme des valeurs de la $i^{\text{ème}}$ ligne.
 - 3) Construire la matrice $L = D^{-1/2} A D^{-1/2}$

Les étapes de l'algorithme (2/2)

- Les étapes (suite)
 - 1) Calculer les valeurs propres x_i de L et ne retenir que les k plus grandes
 - 2) Former la matrice $X=[x_1 \ x_2 \ \dots \ x_k]$ de $\mathbb{R}^{n \times k}$ et Y la matrice normalisée (par ligne) de X
 - 3) Appliquer une méthode de classification non supervisée (K-means) sur l'ensemble des lignes de Y (vecteur de \mathbb{R}^k)
 - 4) Si un ensemble de vecteurs ligne de Y appartiennent à une même classe alors les vecteurs si de même indice appartiennent à une même classe.

Illustration simple – Mise en place

- Nous partirons d'un exemple simple
- L'ensemble de vecteurs s est composé des vecteurs à deux dimensions suivants:
 - $s_1 = (-1, 2)$; $s_2 = (2, 3)$; $s_3 = (3, 2)$; $s_4 = (2, 1)$
- Nous supposons que chaque point représente le centre de 4 classes C_1 , C_2 , C_3 , C_4 .
 - $k = 4$

Illustration simple – étape n°1

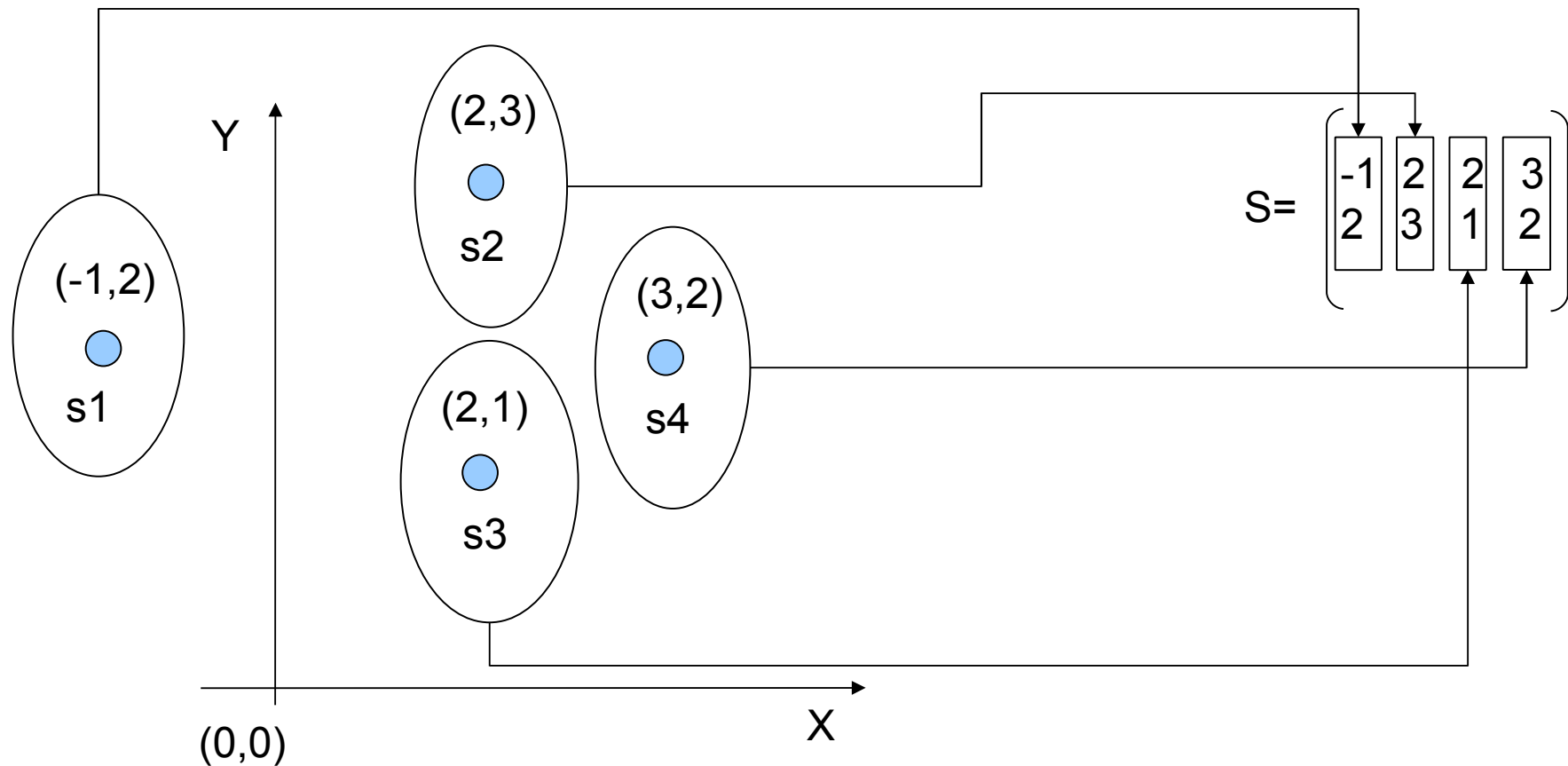
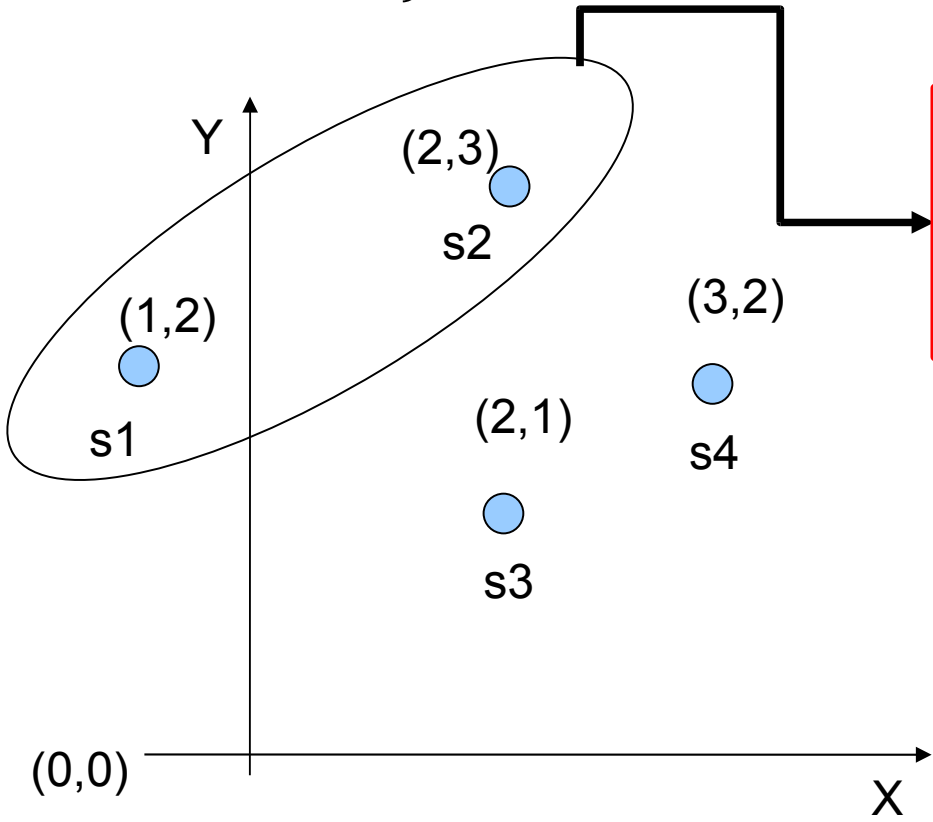


Illustration simple – étape n°1 (suite)

$$S = \begin{pmatrix} -1 & 2 & 2 & 3 \\ 2 & 3 & 1 & 2 \end{pmatrix} \Rightarrow A = \begin{pmatrix} A_{ij} \end{pmatrix} = \begin{pmatrix} 1.0000 & 0.0067 & 0.0067 & 0.0003 \\ 0.0067 & 1.0000 & 0.1353 & 0.3679 \\ 0.0067 & 0.1353 & 1.0000 & 0.3679 \\ 0.0003 & 0.3679 & 0.3679 & 1.0000 \end{pmatrix}$$



$$(s_1, s_2) \Rightarrow A_{12}$$
$$A_{12} = \exp(-(\|s_1 - s_2\|^2 / 2\sigma^2))$$

$$(s_2, s_3) \Rightarrow A_{23}$$

Pour $\sigma=1$

$$(s_4, s_3) \Rightarrow A_{43}$$

Illustration simple – étape n°2 et 3

$$A = \begin{pmatrix} 1.0000 & 0.0067 & 0.0067 & 0.0003 \\ 0.0067 & 1.0000 & 0.1353 & 0.3679 \\ 0.0067 & 0.1353 & 1.0000 & 0.3679 \\ 0.0003 & 0.3679 & 0.3679 & 1.0000 \end{pmatrix}$$

D =

$$\begin{pmatrix} 1.0138 & 0 & 0 & 0 \\ 0 & 1.5100 & 0 & 0 \\ 0 & 0 & 1.5100 & 0 \\ 0 & 0 & 0 & 1.7361 \end{pmatrix}$$

L =

$$\begin{pmatrix} 0.9864 & 0.0054 & 0.0054 & 0.0003 \\ 0.0054 & 0.6623 & 0.0896 & 0.2272 \\ 0.0054 & 0.0896 & 0.6623 & 0.2272 \\ 0.0003 & 0.2272 & 0.2272 & 0.5760 \end{pmatrix}$$

Illustration simple – étape n°4 et 5

$$L = \begin{pmatrix} 0.9864 & 0.0054 & 0.0054 & 0.0003 \\ 0.0054 & 0.6623 & 0.0896 & 0.2272 \\ 0.0054 & 0.0896 & 0.6623 & 0.2272 \\ 0.0003 & 0.2272 & 0.2272 & 0.5760 \end{pmatrix}$$

↓
Vecteurs propres

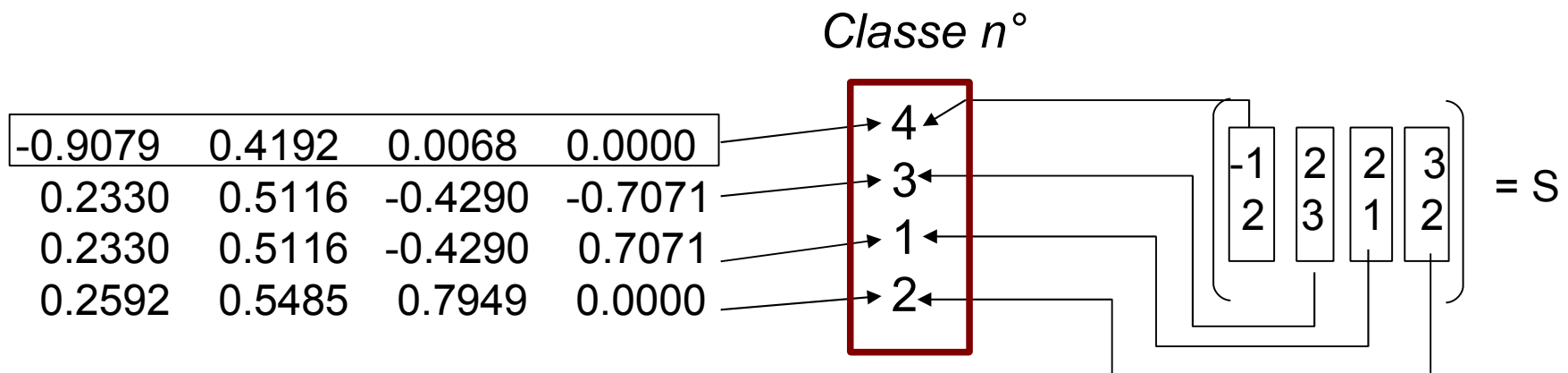
v1	v2	v3	v4
-0.9079	0.4192	0.0068	0.0000
0.2330	0.5116	-0.4290	-0.7071
0.2330	0.5116	-0.4290	0.7071
0.2592	0.5485	0.7949	0.0000

On recherche 4 classes donc on retient les 4 meilleurs vecteurs propres

$$X = \begin{pmatrix} -0.9079 & 0.4192 & 0.0068 & 0.0000 \\ 0.2330 & 0.5116 & -0.4290 & -0.7071 \\ 0.2330 & 0.5116 & -0.4290 & 0.7071 \\ 0.2592 & 0.5485 & 0.7949 & 0.0000 \end{pmatrix} = Y = \text{Vecteur } X \text{ normalisé}$$

Illustration simple – étape n°6 et 7

- Application d'une méthode de classification non supervisée (k-means) sur l'ensemble des vecteurs ligne de Y



Généralisation – cas idéal

■ Hypothèse

- Nombre de classes connus (k)
- Nombre de vecteurs par classe > 1
- La distance deux à deux entre les vecteurs de deux classes distinctes est infinie

La matrice A est block diagonale

$$\begin{pmatrix} A^{(11)} & 0 & 0 \\ 0 & A^{(22)} & 0 \\ 0 & 0 & A^{(23)} \end{pmatrix}$$

$k=3$

La matrice L est block diagonale

$$\begin{pmatrix} L^{(11)} & 0 & 0 \\ 0 & L^{(22)} & 0 \\ 0 & 0 & L^{(23)} \end{pmatrix}$$

Généralisation – cas idéal

- L'ensemble des vecteurs propres de L est l'union des ensembles de vecteurs propres de chaque block $L^{(ii)}$
- Comme dans l'exemple précédent, on ne garde que les k meilleurs vecteurs propres
- Et on suit les même étapes

Généralisation – cas idéal vers cas général

- Cadre de la démonstration
 - A est « presque block diagonal »
 - Il peut exister une perturbation mais suffisamment faible pour espérer rester proche du cas idéal.
 - On peut quantifier cette stabilité en analysant la différence entre la $k^{\text{ième}}$ et la $k+1^{\text{ième}}$ valeur propre:
 - $\delta = |\lambda_k - \lambda_{k+1}|$
 - λ_{k+1} doit être inférieur à λ_k
- Deux autres hypothèses
 - Classes denses (vecteurs proches)
 - Classes éloignées

Seconde partie : Étude pratique et limites de l'algorithme

- Objectifs
 - se familiariser avec le fonctionnement de l'algorithme de *Ng & all*
 - souligner les limites flagrantes de la méthode
- Outils
 - Quelques routines Scilab

Quand tout se passe bien...

- Rappel du cas idéal

- Cas idéal

- proximités non-nulles => objets de même classe
- matrice de similarité idéale
 - parties grise
=> valeur = 0
 - parties noires
=> valeur >= 0



- Rôle du filtrage gaussien

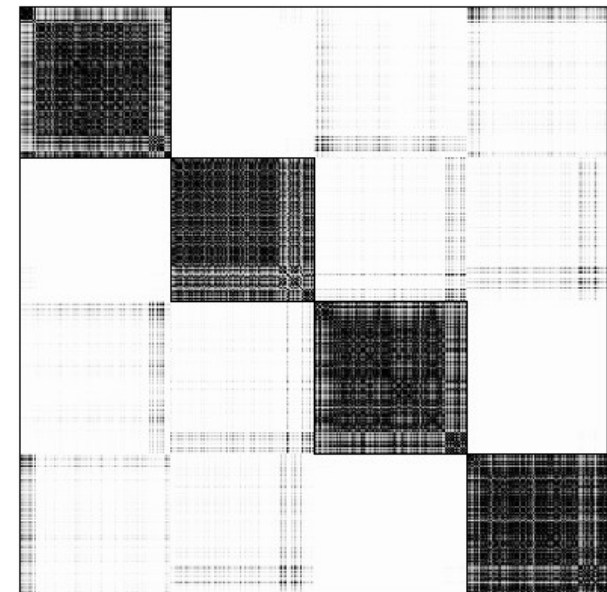
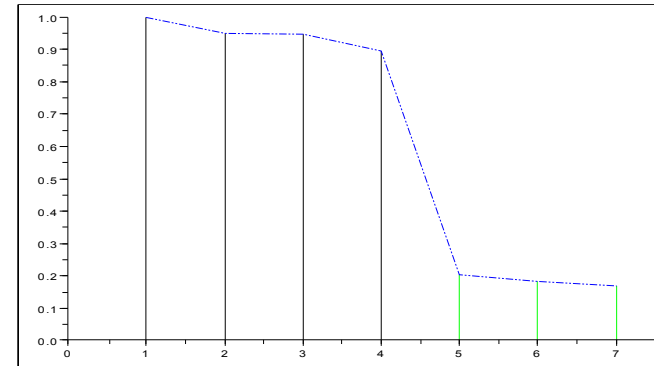
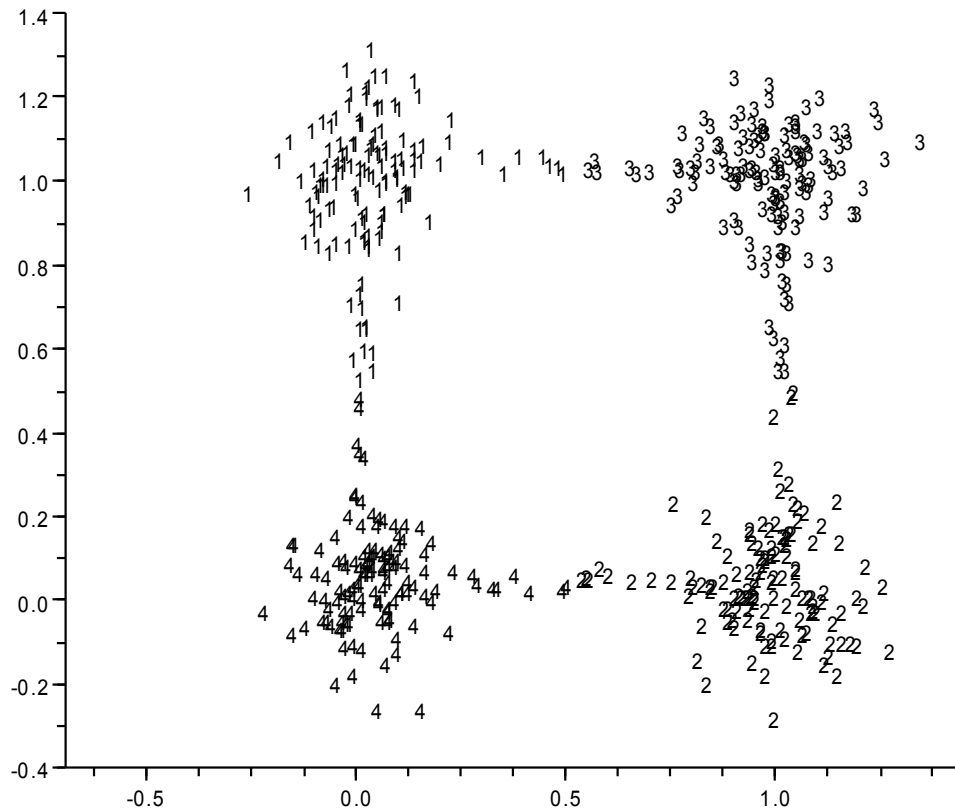
$$A(i, j) = \exp\left(\frac{-d^2(i, j)}{\sigma^2}\right)$$

- « bloc-diagonalisation » de la matrice de similarité
 - en conservant les proximités élevées
 - » *sigma* : référence de la « distance de proximité »
 - en atténuant fortement les proximités faibles

Quand tout se passe bien...

- Courbes 4-blobs

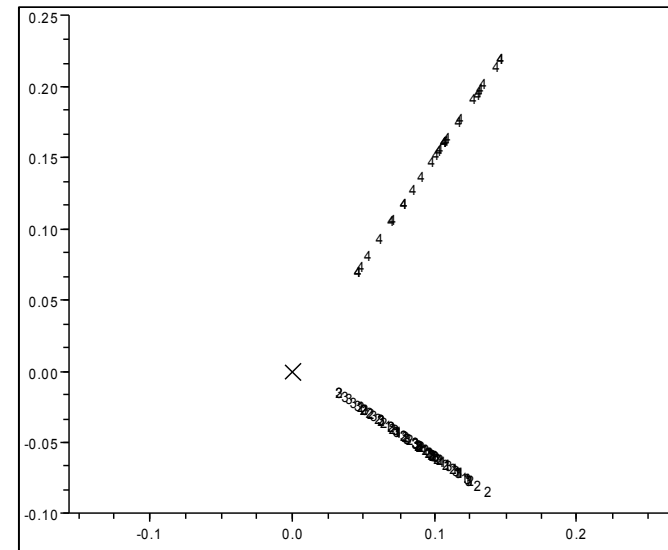
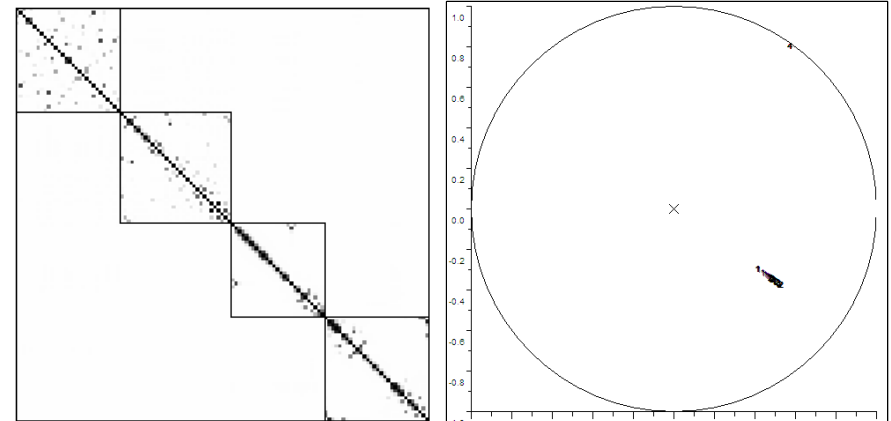
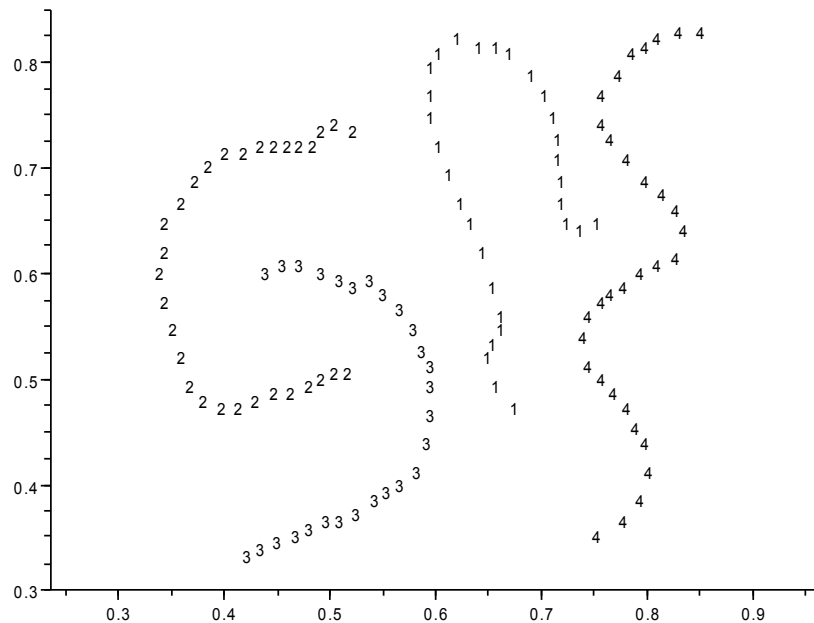
$\sigma = 0,3$



Quand tout se passe bien... avec un peu de chance

- Courbes spaghetti

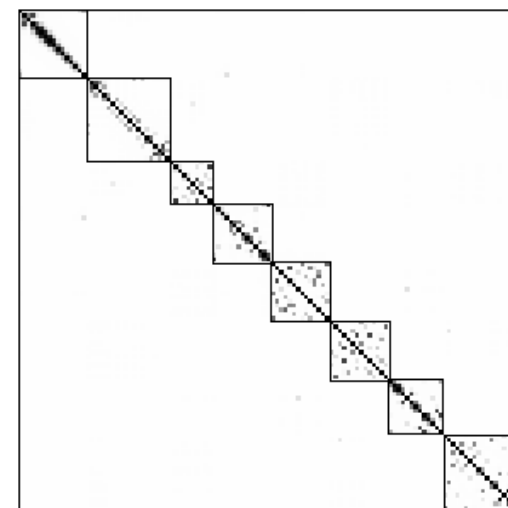
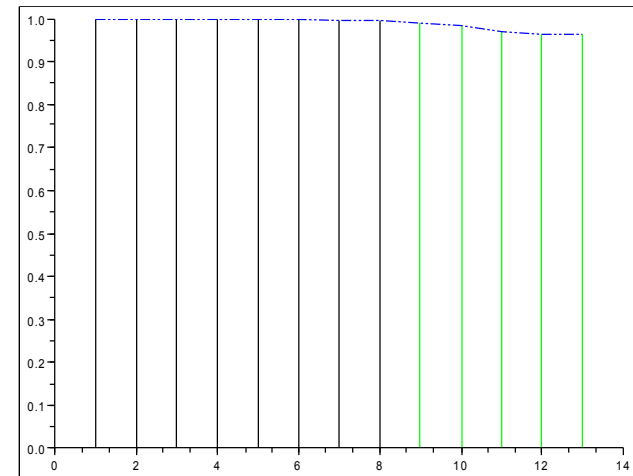
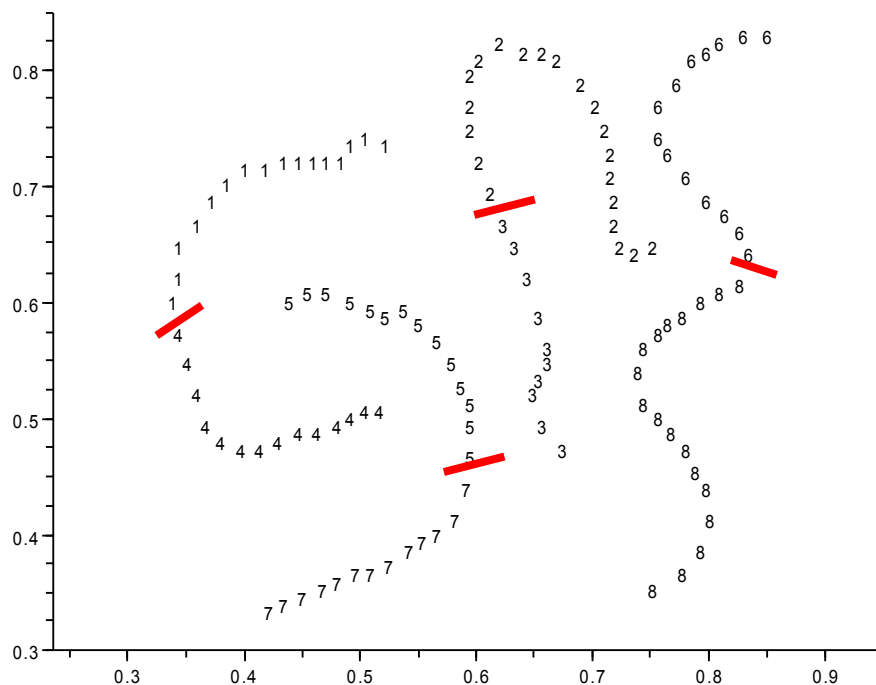
- $\sigma = 0,012$
- 4 classes



Quand tout se passe bien... avec un zest de chance

- Courbes spaghetti

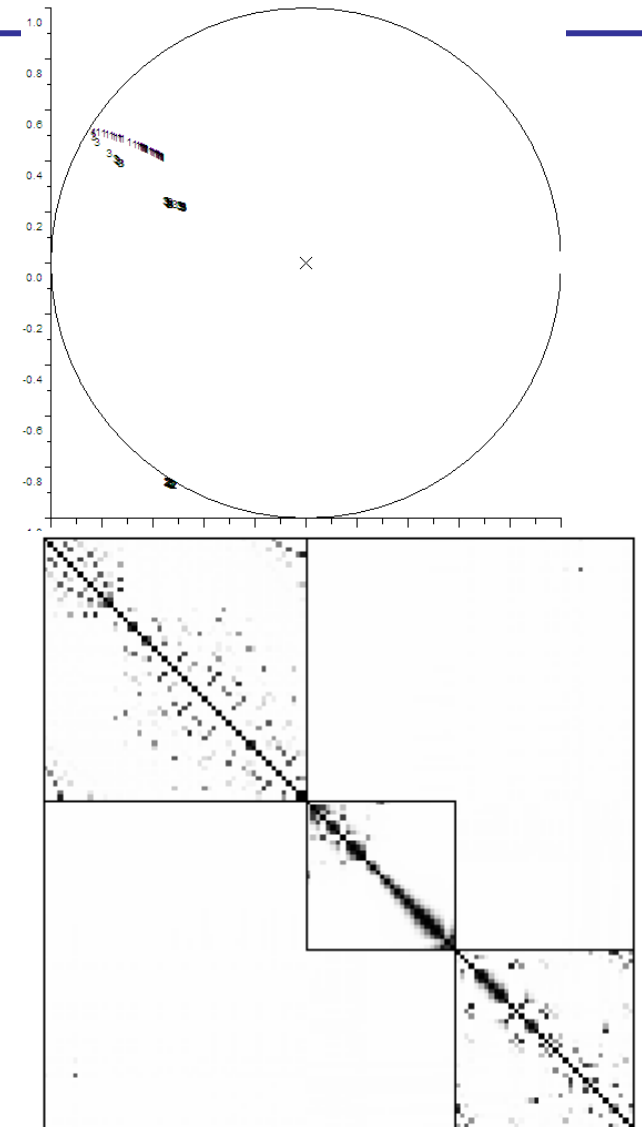
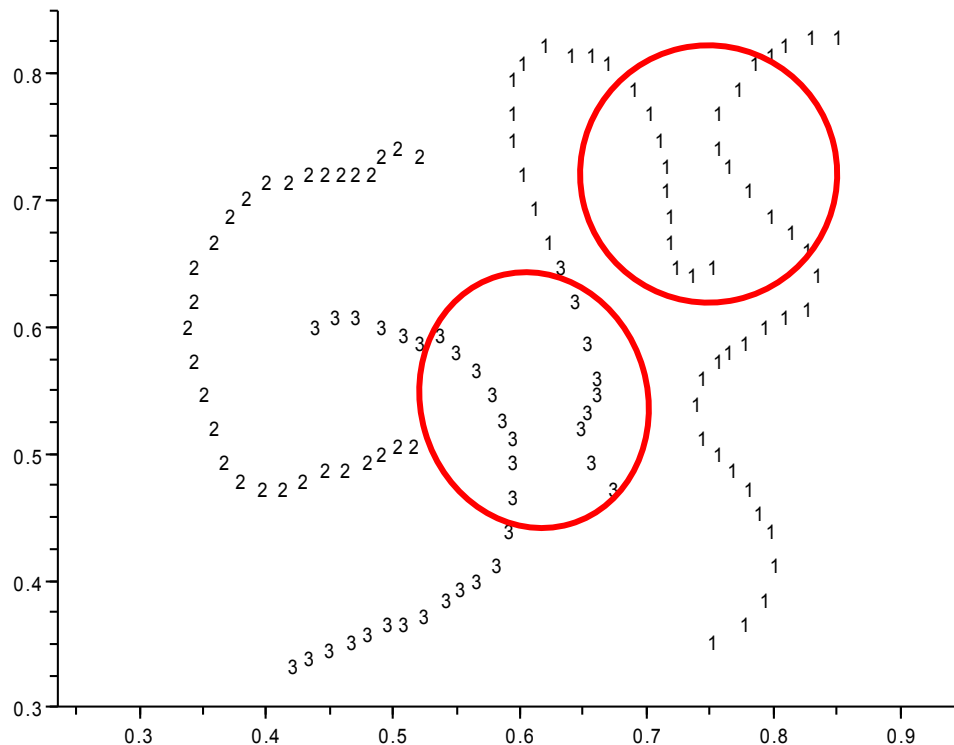
- $\sigma = 0,012$: un peu faible !
- 8 classes



Sigma erroné : un premier dérapage

- Courbes spaghetti

- $\sigma = 0,02$: trop élevé !
- 4 classes



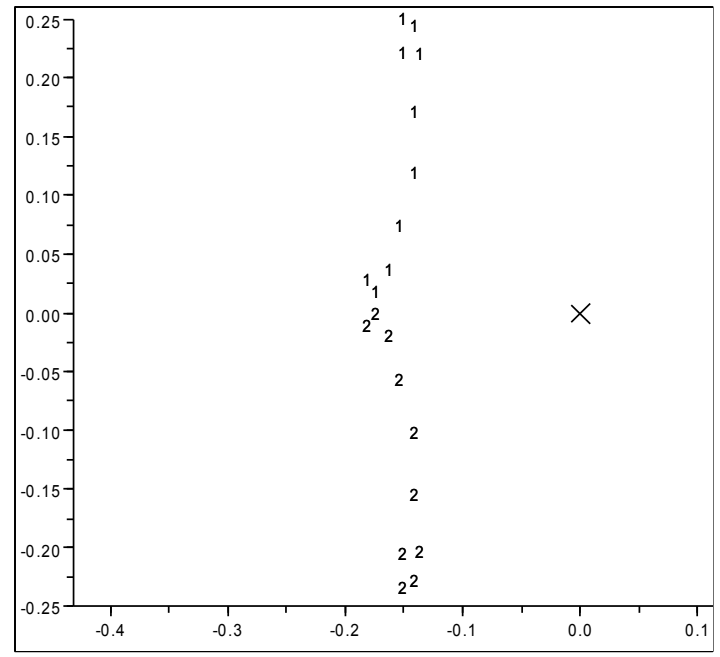
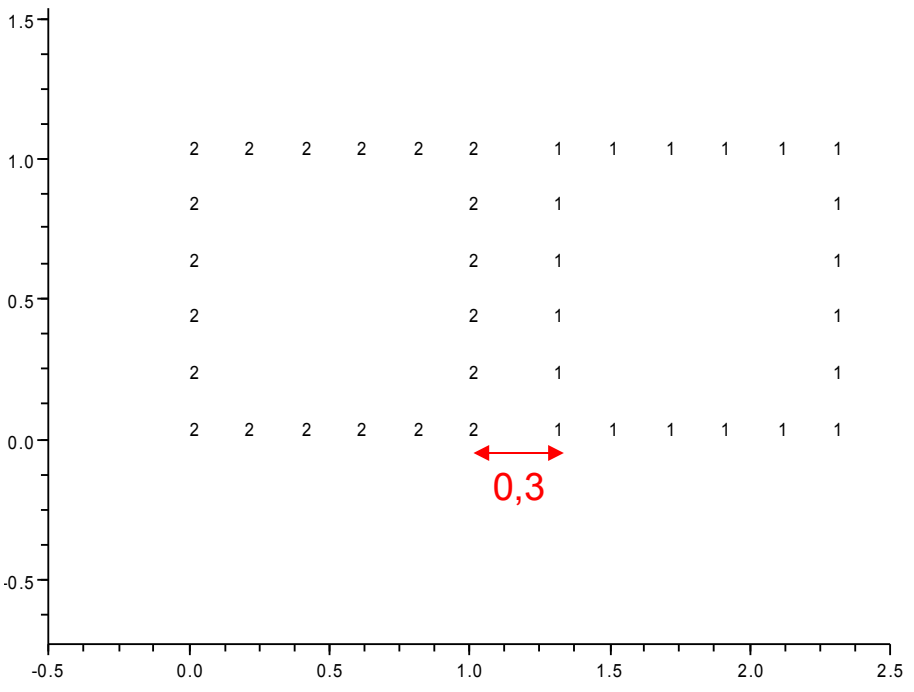
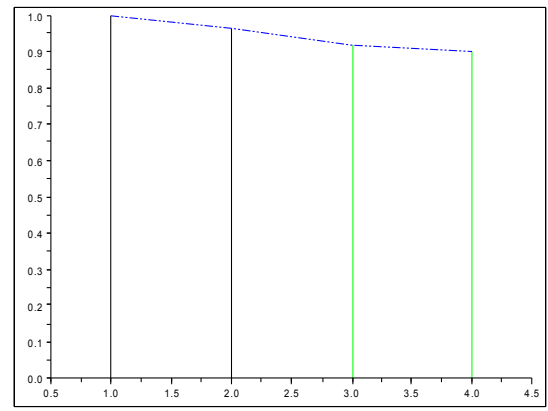
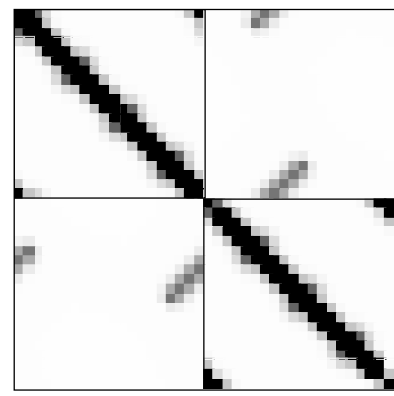
Dualité : *sigma* et nombre de classes obtenues

- Rapport entre le paramètre du noyau gaussien et le nombre de classes extraites
 - décroître *sigma*, c'est :
 - accroître le voisinage de chaque point
 - donc, remplir la matrice de similarité
 - donc, compléter et accroître les blocs diagonaux
 - donc, réduire le nombre de classes extractibles
 - et inversement.
- *Sigma* est donc un paramètre **d'échelle**
 - il ne sera pas évident d'estimer conjointement *sigma* et *k* sans indications précises

Problèmes des classes trop proches

- Carrés proches

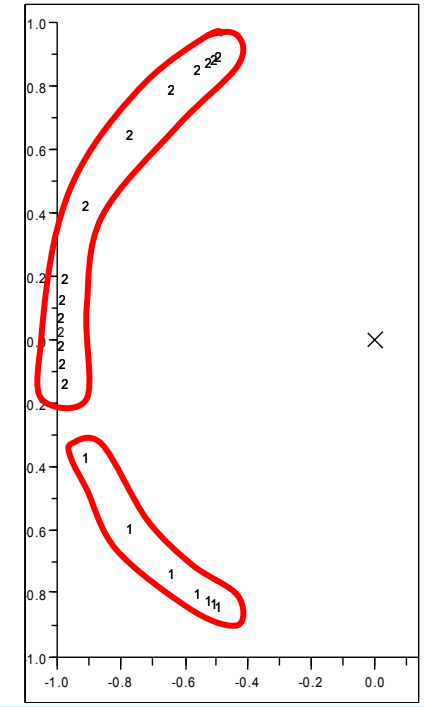
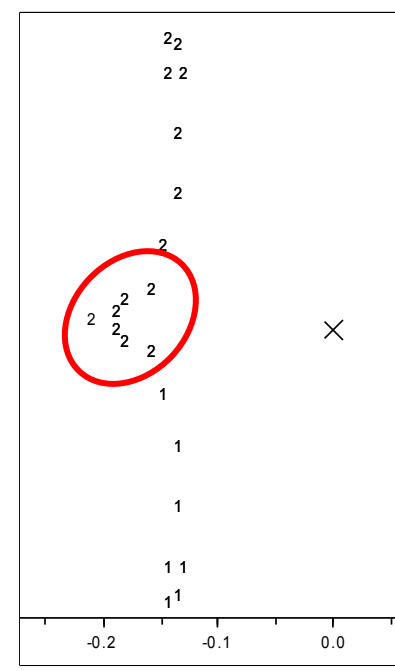
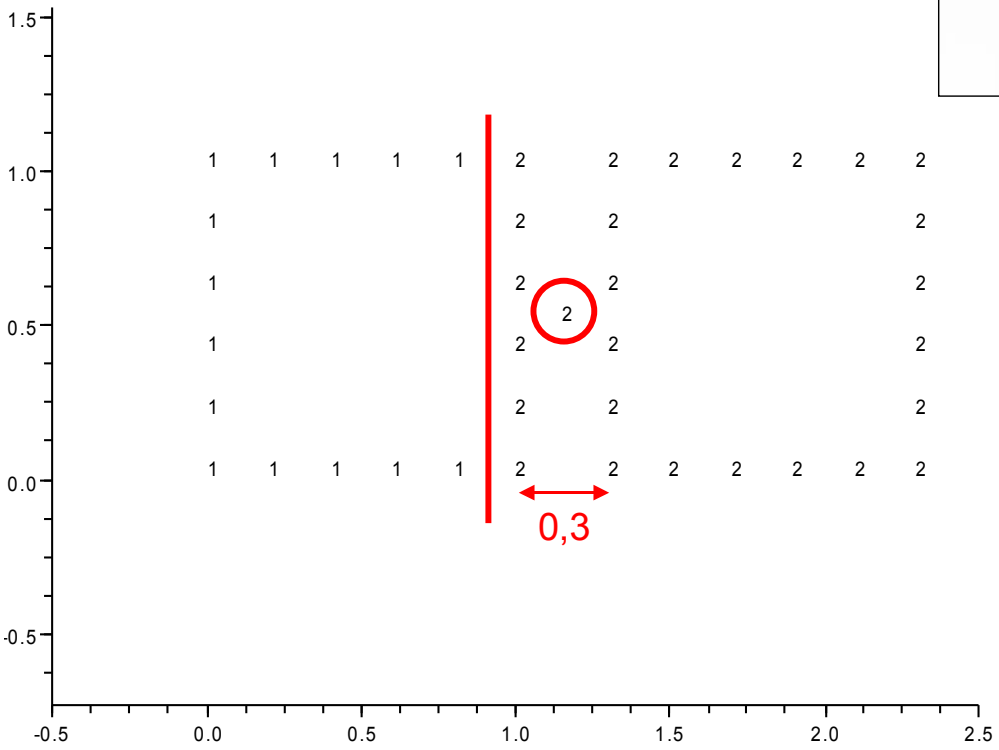
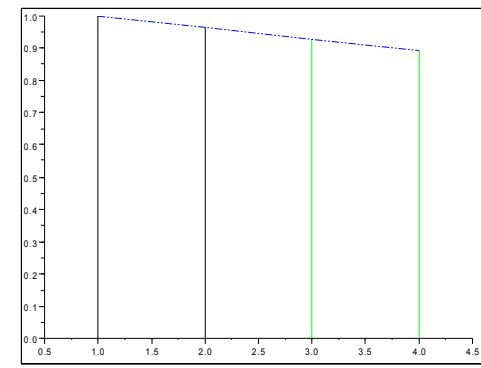
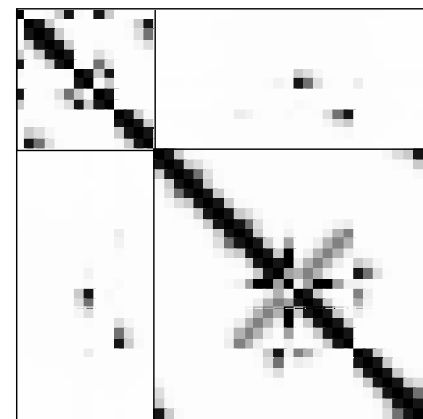
- $\sigma = 0,2$



Problèmes des classes trop proches

• Carrés proches

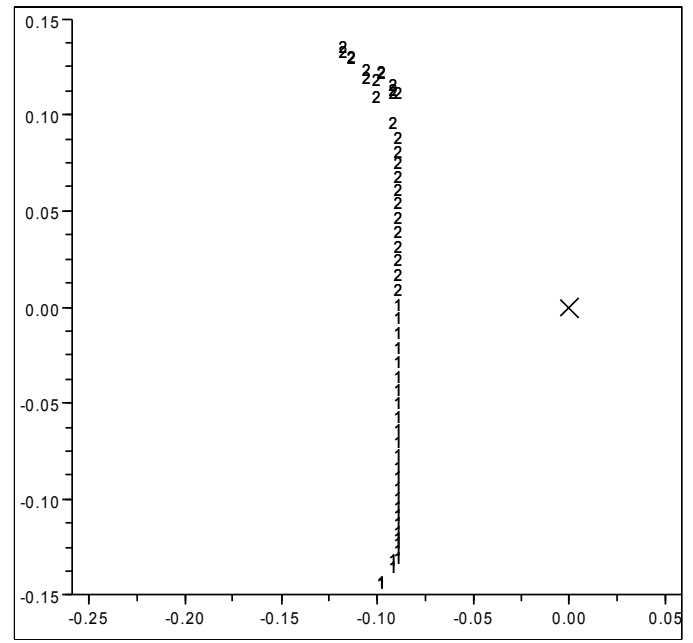
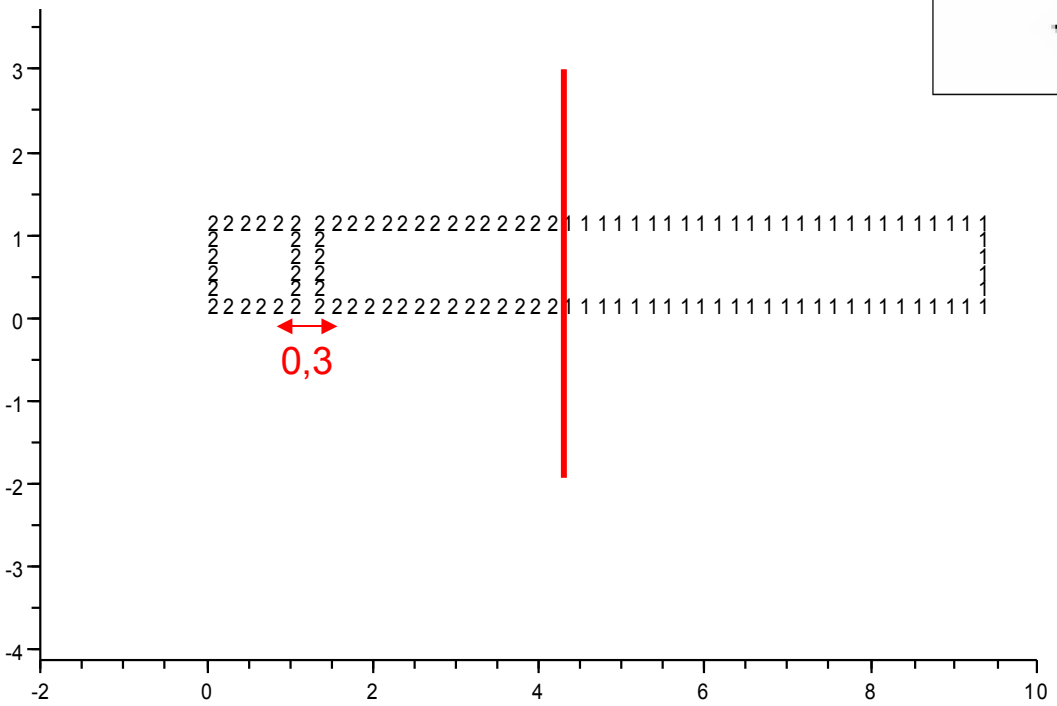
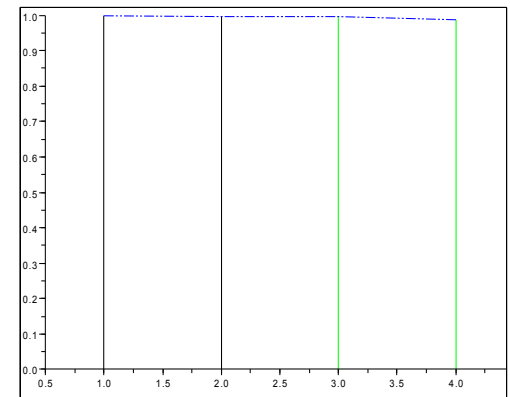
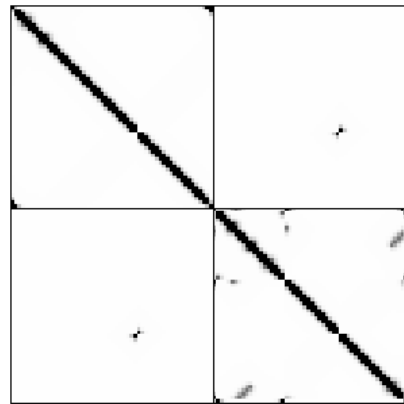
- $\sigma = 0,2$
- point perturbateur



Problèmes des classes trop proches

- Carrés proches

- $\sigma = 0,2$
- 2nd carré -> rectangle



Conclusion

- Classification spectrale : extraction de classes connexes
 - résultats spectaculaires avec les matrices de similarité quasiment bloc-diagonales
 - mais ça n'est qu'une « tendance »
 - lien avec les fonctions de coupes de classes définies comme des rapports de similarités inter-classes sur les similarités intra-classes ?
- Pertinence de la méthode fortement dépendante de :
 - la justesse des paramètres k et σ
 - la proximité des classes
 - comme beaucoup de méthodes
 - mais sensibilité particulière à la présence de points ambigus