Cartes auto-organisatrices de Kohonen et typologisation de territoires

Thomas Soubiran

Ceraps - UMR 8026 du CNRS

Semin-R, 10 juin 2016

Introduction :

Objet de la présentation :

- Établir une typologie de territoires à partir de cartes auto-organisatrices de Kohonen
  - présentation de la méthode
  - intégration de la dimension spatiale des données dans l'analyse
  - utilisation du package Kohonen
  - visualisation des résultats
- Utilisation de l'extension PostGIS de PostGRESQL depuis R pour la manipulation des données spatiales

Plan :

Introduction
- Problématique

Cartes auto-organisatrice de Kohonen
- Introduction
- *K-means*
- SOM : théorie et estimation
- Intégration de la dimension spatiale des données

PostGIS

Application
- Le package Kohonen
- Variables
- Résultats

Problématique :

Contexte :

- Accès croissant à des données géoréférencées issues de la statistique publique en France
- L'Insee diffuse ainsi de plus en plus de données à la commune mais aussi (surtout) à un niveau *infracommunal* (Iris) au travers du Réseau Quetelet
- Plus généralement, diffusion de plus en plus large des « données publiques » notamment via Etalab (loi CADA modifiée, directive européenne de 2003,...)

Problématique :

Établir une typologie de territoires :

- Axée sur l'opposition rural-urbain mais sans postuler un dégradé entre les deux
- Caractérisation de types de territoires ruraux et urbains ainsi que sur les espaces intermédiaires (zones périurbaines, communes « multipolarisées »)
- Combinaison de différentes informations (caractéristiques des individus et des ménages mais aussi des territoires proprement dits comme l'occupation des sols ou les équipements)
- Intégration de la dimension proprement spatiale des données (« loi » de Tobler)

Analyse à l'Iris qui permet une plus grande précision que le niveau communal, particulièrement pour les territoires plus densément peuplés comme les zones urbaines

Classification des données :

La méthode de classification doit donc répondre aux impératifs suivants :

- Données continues :
  - proportions (en majorité)
  - variables de rapport mesurées sur d'autres échelles comme la densité
  - variables de dénombrement
  - relations le plus souvent non-linéaires entre les données
- Dimensions des données :
  - 50 885 unités (Iris) à classifier
  - 25×2 variables (pour l'instant)
- Prise en compte des relations entre les unités observées et leur voisinage plus que leur proximité proprement dite

  En effet, les données sont agrégées dans des unités de tailles variables (entre 0.02 et 97.93 km^2 pour la région NPdCP, la médiane se situant à 5.3 km^2). La distance sur la graphe a donc été préférée à la distance géographique.

Premiers essais par *K-medians* (package flexClust) puis cartes auto-organisatrices de Kohonen

Plan

Introduction
- Problématique

Cartes auto-organisatrice de Kohonen
- Introduction
- *K-means*
- SOM : théorie et estimation
- Intégration de la dimension spatiale des données

PostGIS

Application
- Le package Kohonen
- Variables
- Résultats

Cartes auto-organisatrices de Kohonen :

Le terme de cartes auto-organisatrices de Kohonen (*Self Organizing Maps* ou `Som`) désigne un ensemble de méthodes d'apprentissage non supervisé (Kohonen (2001)) :

- Cas particulier des réseaux de neurones artificiels où l'apprentissage alterne des phases de compétition et de coopération entre les « neurones » (-ie : classes)

Cartes auto-organisatrices de Kohonen :

Le terme de cartes auto-organisatrices de Kohonen (*Self Organizing Maps* ou `Som`) désigne un ensemble de méthodes d'apprentissage non supervisé (Kohonen (2001)) :

- Cas particulier des réseaux de neurones artificiels où l'apprentissage alterne des phases de compétition et de coopération entre les « neurones » (-ie : classes)
- Méthode de *réduction des dimensions* des données : les données de départ sont projetées sur un treillis (ou grille) de neurones le plus souvent à deux dimensions

  De ce point de vue, les `Som` s'apparentent aux méthodes factorielles (`ACP`, `AFC`,...) où les points sont projetés sur des axes minimisant la variance du nuage de points

  Néanmoins, les dimensions du treillis ne sont pas liées à celle des données. Plus généralement, il s'agit de méthodes dont l'inspiration et le raisonnement sont très différents

Cartes auto-organisatrices de Kohonen :

Le terme de cartes auto-organisatrices de Kohonen (*Self Organizing Maps* ou `Som`) désigne un ensemble de méthodes d'apprentissage non supervisé (Kohonen (2001)) :

- Cas particulier des réseaux de neurones artificiels où l'apprentissage alterne des phases de compétition et de coopération entre les « neurones » (-ie : classes)
- Méthode de *réduction des dimensions* des données : les données de départ sont projetées sur un treillis (ou grille) de neurones le plus souvent à deux dimensions
- La projection sur une grille permet de prendre en compte le *voisinage* des points afin de préserver la *topologie* des données
  Autrement dit, si deux objets sont similaires dans l'espace original, alors leur position *sur la grille* devrait être elle aussi similaire

  Note : les `Som` ne cherchent cependant pas à reproduire des *distance* à la différence du `MDS`).

Carte auto-organisatrice de Kohonen (suite) :

Les Som constitue un exemple d'algorithme inspiré par (la représentation) d'un processus naturel (Turing)

- La méthode s'inspire du traitement de l'information sensorielle par le cortex cérébral des vertébrés
- Basé sur la théorie des neurones de Donald Hebb
- Simplification par T. Kohonen de l'approche d'abord proposée par Willshaw et von Malsburg dans la première moitié des années soixante-dix
- Les Som peuvent aussi être vues comme un version topologiquement contraintes des nuées dynamiques (*K-means*)

La présentation de la méthode par analogie au fonctionnement du cerveau nécessitant de solides connaissances en neurologie, la représentation des Som comme *K-means* modifiée sera privilégiée dans la suite de l'exposé.

Classification *K-means* : préambule

Les *K-means* peut être vu comme appartenant à une classe plus large d'algorithmes : les algorithmes combinatoires (Hastie et al. (2009))

- L'objet des algorithmes combinatoires est d'attribuer directement chacune des observations i à un groupe sans postuler un modèle probabiliste sous-jacent. L'assignation est une association de type « plusieurs-à-un » $k = C(i)$ avec $i \in \{1, \ldots, N\}$ et $k \in \{1, \ldots, K\}$.
- On cherche une fonction particulière $k = C^*(i)$ satisfaisant à différents critères d'optimalité. Dans le cas présent, il s'agit d'assigner des valeurs proches à un même *cluster* en utilisant une mesure de distance $d(x_i, x_{i'})$. Ce qui revient, entre autres choix possibles, à minimiser la fonction suivante :

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}) \tag{1}$$

  qui mesure la dispersion intra de chaque groupe.
- L'objectif est de trouver un sous-ensemble contenant une partition sous-optimale correcte, ce qui n'a rien de trivial (difficulté NP). Ainsi, le nombre d'assignations distinctes a pour expression :

$$A(N, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \begin{bmatrix} K \\ k \end{bmatrix} k^N \simeq K^N$$

  Oui, ça fait vite beaucoup

Classification *K-centroïdes* (suite)

En utilisant la norme euclidienne carrée, on obtient la méthode ***K-means***.

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \|x_i - x_{i'}\|^2 \tag{2}$$

L'équation (1) devient alors :

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2 \tag{3}$$

où $\bar{x}_k$ désigne la moyenne arithmétique de la classe k

Note : L'utilisation de la norme euclidienne carrée a notamment pour conséquence de donner plus de poids aux grandes distances ce qui la méthode sensible aux valeurs extrêmes.
Néanmoins $d(x_i, x_{i'})$ peut être remplacée par une mesure plus robuste comme la médiane. On obtient alors la méthode ***K-medians***.

Classification *K-means* : algorithme

C^* peut être trouvé en résolvant :

$$C^* = \arg \min_{C_i \{m\}_1^k} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2 \tag{4}$$

Soit, pour tout ensemble d'observation S, $\bar{x}_S = \arg\min_m \sum_{i \in S} \|x_i - m\|^2$du fait que la moyenne arithmétique minimise la norme euclidienne

Les paramètres peuvent être obtenus de la façon suivante :

1: Initialisation de l'algorithme en attribuant les observations aux classes $i, \ldots, K$
2: **repeat**
3: Actualiser les moyennes (centroïdes) de chaque classe
4: Assigner chaque observation à la classe la plus proche (-ie : distribuer les observations selon une partition de Voronoï (9))
5: **until** convergence

Propriétés :

Les ***K-means*** s'apparentent à un algorithme du gradient (*gradient descent*) ou l'agorithme EM (Espérance maximisation) (Bottou and Bengio (1995)) même s'il ne s'agit pas d'un algorithme probabiliste :

$$Q(w', w) = \sum_i (x_i - w'_{s_i(w)})^2 \tag{5}$$

La solution de (5) $\partial Q(w', w)/\partial w'_k = 0$ est identique à la solution (4) de $W(C)$.

On peut ainsi noter que :

- L'estimation des ***K-means*** est proche de l'estimation d'un modèle de mélange fini de lois normales
- De par l'inégalité de Jenks, l'alogrithme converge de façon monotone vers un *minimum local*
- La solution n'est pas pas nécessairement optimale et dépend fortement des valeurs de départ. Plusieurs exécutions consécutives de l'algorithm peuvent donner des résultats (très) différents
- C'est pourquoi le package flexClust propose la fonction `stepFlexclust()` qui estime un nombre définit par l'utilisateur de modèles en parallèle (grace au package multicore) et ne conserve que la meilleure solution
- Cette approche est néanmoins uniquement heurisitque et ne garantit rien
- Alternative : améliorer l'algorithme pour qu'il « colle » au mieux aux données

K-means et SOM :

Les SOM se distinguent notamment des *K-means* par la mise en relation des centroïdes et l'ajout d'une phase de coopération

- Lors des itérations *K-means*, les classes qui correspondent aux centroïdes sont indépendantes. Le classe sélectionnée est la seule à pouvoir s'adapter (*the winner takes it all*)
- Lors des itérations SOM, ces centroïdes sont placés sur une grille de neurones. De part cette structure en treillis, chaque classe a donc plusieurs classes qui lui sont adjacentes
- Après s'être adapté, le neurone sélectionné permet aussi à son voisinage de s'adapter en fonction de leur *distance* sur la grille au neurone retenu (coopération)

Phase de coopération :

La phase de coopération où les voisins du BMU (*Best Matching Unit*) sont actualisés a pour expression (Yin (2008)) :

$$\mathbf{w}_{k,t+1} = \mathbf{w}_{k,t} + \alpha_t \eta(\nu_t, k, \sigma_t)(\mathbf{x}_t - \mathbf{w}_{\nu,t}) \tag{6}$$

où

- $\mathbf{x}_t$ désigne le vecteur de données courant utilisé pour déterminer le BMU
- $\mathbf{w}$ désigne les coordonnées du neurone dans l'espace des données (-ie : centroïdes, aussi appelés, poids ou *code vector*)
- k désigne l'indice du neurone mis-à-jour
- ν_t désigne l'indice du neurone sélectionné à l'itération t (-cf. *supra* (4)) :

$$\nu_t = \arg\min_{k \in \mathcal{K}} \|\mathbf{x}_t - \mathbf{w}_k\|^2 \tag{7}$$

- α désigne le rythme d'apprentissage. Ce paramètre doit être fourni en argument à la fonction. Il détermine le poids attribué aux données de départ. Ce poids réduit progressivement et de façon monotone à chaque itération
- $\eta(\nu_t, k, \sigma_t)$ désigne la fonction de voisinage
  Cette fonction peut prendre plusieurs forme comme un cercle, un rectangle ou le noyau d'une fonction de densité normale :

$$\exp\left(-\frac{\|\mathbf{r}_\nu - \mathbf{r}_k\|^2}{2\sigma_t^2}\right) \tag{8}$$

où $\mathbf{r}_k$ désigne les coordonnées du neurone k sur la grille et σ_t et le diamètre du voisinage (-cf. estimateur de Nadaraya–Watson)

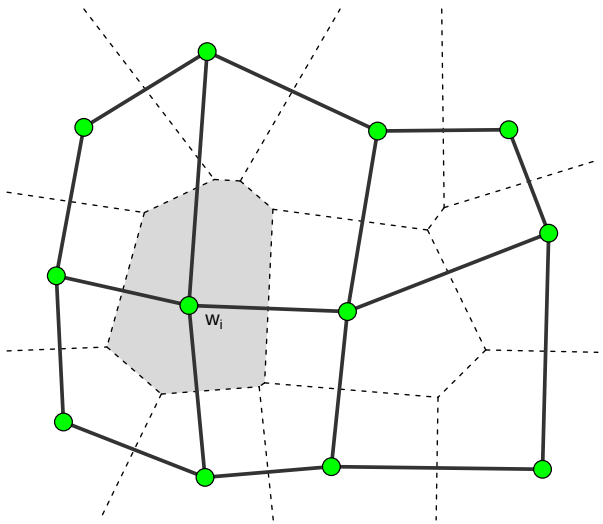
SOM et tesselation de Voronoï :

De par l'utilisation de la distance euclidienne, les SOM peuvent être vues comme une *tesselation de Voronoï* de l'espace des données (van Hulle (2011)) :

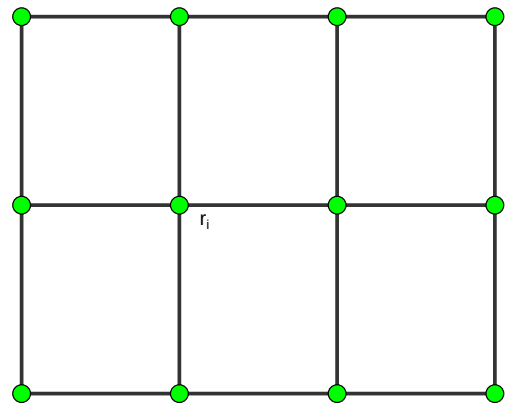
$$\hat{V}_i = \{\mathbf{x} \in \Omega \quad | \quad \|\mathbf{x} - \mathbf{w}_\nu\| \leq \|\mathbf{x} - \mathbf{w}_k\| \text{ pour } , \nu \neq k\} \tag{9}$$

où V_i désigne une partition de l'ensemble Ω tel que $V_\nu \cap V_k = \emptyset$ pour $\nu \neq k$ et $\cup_{k=1}^{K} \overline{V}_k = \overline{\Omega}$

À chaque neurone correspond une région de l'espace dont les frontières sont les bissectrices des lignes joignant chaque paires $\mathbf{w}$ de neurones :



(a) Poids $\mathbf{w}$



(b) Grille $\mathbf{r}$

Algorithme :

Les paramètres d'une SOM peuvent être obtenus de la façon suivante :

1: Initialisation de t, α, $\mathbf{w}_k$ et σ
2: **repeat**
3: Sélection d'un vecteur $\mathbf{x}_t$
4: Sélection du BMU ν

$$\nu_t \leftarrow \arg\min_{k \in \Omega} \|\mathbf{x}_t - \mathbf{w}_k\|^2$$

5: Actualisation des poids de ν et de ses voisins

$$\mathbf{w}_{k,t+1} \leftarrow \mathbf{w}_{k,t} + \alpha_t \eta(\nu_t, k, \sigma_t)(\mathbf{x}_t - \mathbf{w}_{\nu,t})$$

6: Mise-à-jour de t, α et σ
7: **until** $t <$ maxiter

Note : ceci n'est qu'une possibilité parmi différents algorithmes

À la différence des *K-means*, les propriétés des SOM comme la convergence n'ont été établies que dans un nombre limité de cas et certains auteurs suggèrent même qu'une démonstration générale est impossible

Et pourtant, ça marche plutôt bien en pratique...

Variantes :

La littérature comporte de nombreuses variantes de ce qui vient d'être présenté (Kohonen (2001))

- La distance euclidienne peut être remplacée par d'autres mesures telles que la médiane ou d'autres normes
- SOM hiérarchiques (-cf *infra*)), apprentissage supervisé, logique floue
- Sujet connexe de la mesure de la qualité des partitions
- ...
- Extensions spatiales

Intégration de la dimension spatiale des données :

Les Som cherchent donc à préserver autant que possible la topologie des données. Néanmoins, elles n'intègrent pas explicitement la dimension géographique des données

Un des principe de l'analyse stipule en effet que, plus des observations sont proches géographiquement, plus elles tendent à présenter des valeurs similaires (première loi de la géographie de W. Tobler)

Pour cela, une solution consiste à augmenter les données d'une statistique d'autocrrelation spatiale Lisa qui est la version locale d'une statistique globale, le I de Moran (Arribas-Bel and Schmidt (2011))

Mesure globale de l'autocorrelation entre unités spatiales

Le I de Moran est mesure d'autocorrélation *globale* :

$$I = \frac{N}{S_0}\frac{\mathbf{e}^\intercal \mathbf{W}\mathbf{e}}{\mathbf{e}^\intercal \mathbf{e}} \tag{10}$$

$$= \frac{N}{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}} \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{11}$$

où N désigne le nombre d'unités, $\mathbf{e} = \mathbf{x} - \bar{\mathbf{x}}$. $\mathbf{W}$ désigne une matrice de poids liant les unités entre elles avec $w_{ii} = 0$ dont les cellules peuvent être binaire (par ex., si les unités sont contiguës) ou prendre d'autre valeurs.

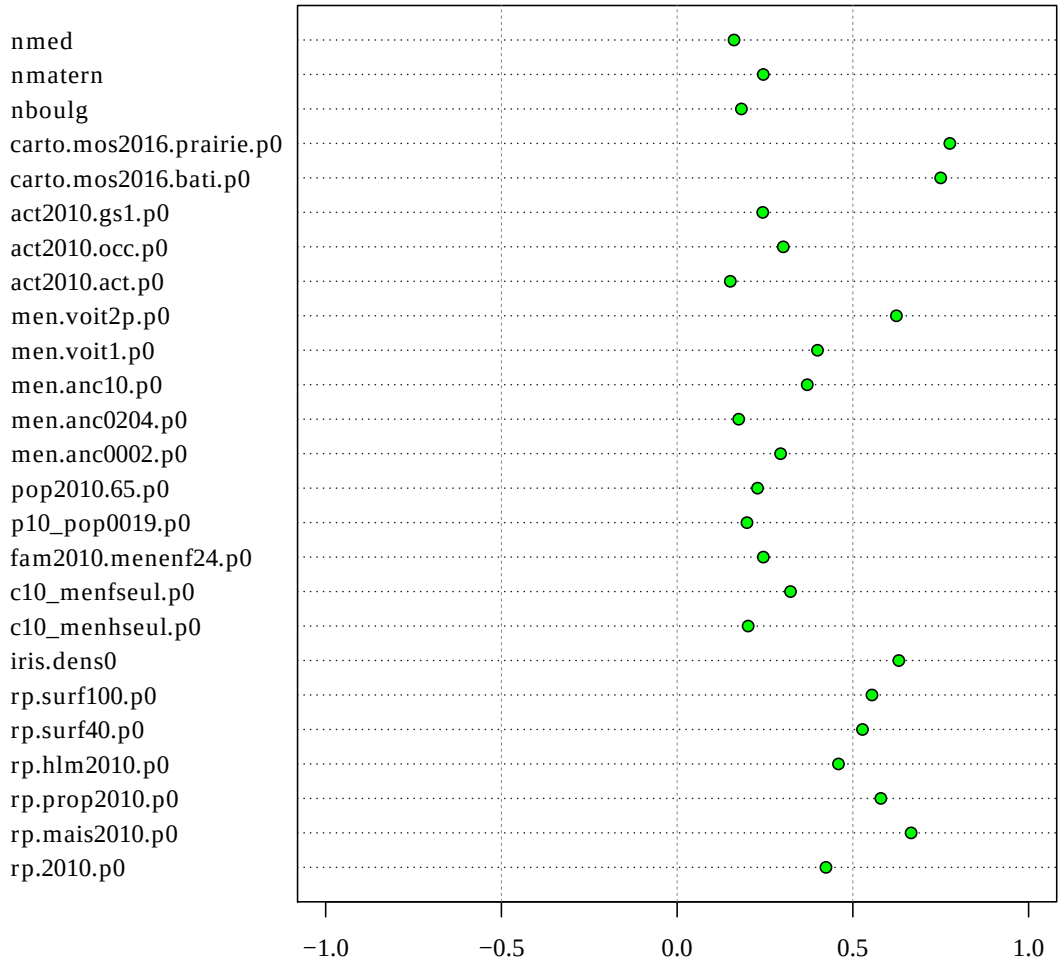
Le I de Moran peut être vu comme l'équivalent spatial du coefficient de corrélation de Pearson. Il permet de mesurer si des observations proches géographiquement tendent à présenter des valeurs similaires.

La significativité des paramètres estimés peut être testée avec un test de la forme :

$$z_I = \frac{I - E[I]}{\sqrt{V[I]}} = \frac{I - (-1/(N-1))}{\sqrt{E[I^2] - E[I]^2}}, \text{ avec } z_i \sim N(0,1) \tag{12}$$

Méthode alternative : test de permutation

I de Moran pour les IRIS de la région Nord Pas-de-Calais Picardie :



Source : INSEE enquête annuelle de recensement 2010

Note : tous les paramètres sont significatifs

Mesure locale de l'autocorrelation entre unités spatiales

Luc Anselin en a dérivé des mesures *locales* (ou LISA) de plusieurs mesures d'autocorrélation dont le I de Moran local (Anselin (1995)) :

$$I_i = \frac{N}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \sum_{j=1}^{N} w_{ij}(x_i - \bar{x})(x_j - \bar{x}) \tag{13}$$

Note : $I = \sum_i I_i / S_0$

I_i mesure l'association *d'une* unité avec ses voisins.

Comme pour le I de moran global, les moments de I_i peuvent être calculés. Par contre, la distribution du test équivalent à (12) est inconnue.

Le fichier a été augmenté des I de Moran locaux pour chacunes des variables retenues pour l'analyse

Code R pour le calcul des I de Moran locaux :

```r
## Liste des voisins
iris.nb0 <- poly2nb(npdcp.iris.geo0, row.names = npdcp.iris.geo0@data$iriscode)
## Calcul du I de Moran local pour toutes les colonnes du df original
localI.0 <- lapply(
    as.vector(colnames( df ),mode = "list")
  , function(vname, tpl, nb) localmoran(tpl[,vname], nb)
  , df
  , nb2listw( iris.nb0 , zero.policy=TRUE )
)
## Génération des noms de colonnes pour le df
names(localI.0) <- paste( colnames( df ), "localI", sep=".")
## Transformation en df
localI.1 <- as.data.frame(lapply(
    localI.0 , function(tpl)tpl[,1]
))
```

Intégration de la dimension spatiale des données :

Cette approche a l'avantage de la simplicité puisqu'elle ne nécessite pas de modifier le code source de la fonction estimant les paramètres des SOM.

Néanmoins :

- Les LISA sont-elles des statistiques exhaustives dans ce contexte ?
- Alternative : intégrer explictement la matrice $\mathbf{W}$ de (10) dans les calculs, par exemple dans la phase de compétition

Plan

Introduction
- Problématique

Cartes auto-organisatrice de Kohonen
- Introduction
- *K-means*
- SOM : théorie et estimation
- Intégration de la dimension spatiale des données

PostGIS

Application
- Le package Kohonen
- Variables
- Résultats

PostGIS :

PostGIS est une extension de PostGRESQL pour traiter les données spatiales :

- Basée sur les mêmes librairies que les packages R : GEOS (géométrie), PROJ.4 (projections), OGR (vecteur), GDAL (*raster*), ... Les interfaces sont d'ailleurs largement similaires (-ex : `ST_Intersects()` : `gIntersects()`, `ST_Intersection()` : `gIntersection()`, ...)
- Avantage de PostGIS : Possibilité d'utilisation d'index dans les requêtes.
- Les données peuvent donc être atteintes plus rapidement que dans une recherche séquentielle

  Utile, entre autre, pour agréger les données à l'Iris. Par exemple, pour réaliser l'intersection entre les Iris et la couche `ZONE_OCCUPATION_SOL` de la bd CARTO® de de l'IGN pour le seul département du Nord, il faut comparer 1 352 Iris avec 1 725 polygones, soit 2 332 200 comparaisons. Sachant que les polygones sont composés de 130 620 et 173 390 points respectivement.
- Facilite l'organisation et l'accès à de large collection de données spatiales
- Permet aussi de bénéficier de toute l'efficacité de PostGRESQL

Indexation des données spatiales :

La méthode d'indexation des données spatiales la plus répandue est sans doute le `R-Tree` proposée par Guttman

- Extension du `B-Tree`
- Structure hiérarchique
- Les objets proches sont regroupés dans des rectangles (*bounding boxes*)
- Ces rectangles sont eux-mêmes regroupés en rectangles qui eux-mêmes sont regroupés en rectangles. . .jusqu'à ce qu'il n'y ait plus qu'un seul rectangle englobant tous les points
- Les nœuds de l'arborescence stockent les coordonnés des différents rectangles ainsi que les références aux rectangles de taille inférieure contenus par ce rectangle
- La recherche se fait en partant de la racine de l'arborescence et, à chaque niveau, teste si les coordonnées recherchées sont contenus par l'un ou l'autre des rectangles d'un nœud jusqu'aux « feuilles » de l'arbre
- Est-ce que ça vous rappelle quelque chose ?
  L'utilisation de méthode de classification a ainsi été proposé pour la construction de l'arbre

Exemple de requête Sql avec PostGIS :

Agrégation des parcelles cultivées à l'Iris dans le département du Nord

```
select dcomiris
  , ST_asText(
    /* union des polygones résultant de l'intersection */
    ST_union(
      /* intersection des deux géométries */
      ST_Intersection(a.geom2154, b.geom2154 )
    )
  ) as geom2154
from iris.iris_059 as a
  , rgp.rpg2010_059 as b
/* filtre utilisant l'index */
where ST_intersects(a.geom2154, b.geom2154)
group by dcomiris;
```

L'exécution de la requête via la fonction `dbGetQuery()` crée un `data.frame` avec deux colonnes contenant le code de l'Iris et les polygones représentant les parcelles correspondant à cet Iris

```
iris59.rpg2010.wkt0 <- dbGetQuery(pgis.co, rq)
```

Récupération des données dans R :

Dans ce qui précéde, l'objet géométrique retourné est passé en argument à la fonction `ST_AsText()`

- La fonction `ST_AsText()` retourne une répresentation caractère de la géométrie de l'objet au format WKT (*Well-known text*) défini par l'*Open Geospatial Consortium*
- Exemple :

```
POLYGON((724242.7 7036482.3,724222.7 7036487.4,724130.04 7036449.81,...))
```

- Du côté de R, il faut utiliser la fonction **`readWKT`** du package **`rgeos`** en bouclant sur la colonne contenant les chaînes WKT ce qui, pour chacune d'entre elles, instancie l'objet *sp* approprié (**`SpatialLines`**, **`SpatialPoygons`**,...)
- Note : La fonction **`readWKT()`** n'accepte pas les chaînes WKT accompagnées de leur `SRID` qui identifie la projection utilisée (pourquoi ?). Il faut donc passer la projection en argument

Récupération des données dans R (code) :

```
##
## Version (TRÈS) simplifiée
## Présuppose notamment que de l'objet retourné par readWKT() est de type
     SpatialPolygons
##
as.sp.wkt <- cmpfun(function(
  wktgeo, prj, df = NULL, ID = NULL
){
  polygons <- list()
  for ( i in 1:length(wktgeo)){
    polygons <- c(
      polygons, readWKT(wktgeo[i], i, p4s = prj )
    )
  }
  polygons <- do.call("rbind", polygons)
  if( !is.null(df) )
    polygons <- SpatialPolygonsDataFrame(polygons,df)
  if( !is.null(ID) )
    return( spChFIDs(polygons,ID) )
  polygons
})
##
iris59.rpg2010.geo0 <- as.sp.wkt(
    iris59.rpg2010.wkt0$geom2154
  , EPSG$prj4[which(EPSG$code==2154)] ## SRID : RGF93
  , data.frame(iriscode = iris59.rpg2010.wkt0$dcomiris, stringsAsFactors = F)
  , iris59.rpg2010.wkt0$dcomiris ## ID
)
```

Plan

Introduction
- Problématique

Cartes auto-organisatrice de Kohonen
- Introduction
- *K-means*
- Som : théorie et estimation
- Intégration de la dimension spatiale des données

PostGIS

Application
- Le package Kohonen
- Variables
- Résultats

Le package Kohonen :

Les analyses ont été réalisées avec le package Kohonen (Wehrens and Buydens (2007)) qui comprend notament :

- des fonctions pour estimer différentes variantes de `SOM` : non-supervisées (`som()`), supervisées (`xyf()`,`bdk()`), hiérarchiques(`supersom()`)
- des méthodes `S3` pour interpréter et visualiser les résultats (`summary()`,`plot()`)
- une méthode pour attribuer les observations au neurone le plus proche (aussi accessible par l'élément `$unit.classif` de l'objet retourné)

Code R :

Code R minimal :

```
iris.df.som0 <- som(
    scale(iris.df)
  , grid  = somgrid(
      2, 5           ## taille de la grille
    , "rectangular" ## type de grille. Alternative : "hexagonal"
  ) #
  , n.hood = "square" ## forme du voisinage, valeur par défaut pour une grille
       rectangulaire
  , rlen   = 250 ## nombre d'itérations
)
```

Autres paramètres :

- `alpha` : rythme d'apprentissage, -cf. le paramètre α de (6)
  Vecteur de deux valeurs entre lesquelles α décroîtra linéairement à chaque itération
- `radius` : rayon du voisinage, -cf. le paramètre σ de (6)
  Valeur par défaut : `quantile(nhbrdist, 0.67) * c(1, -1)`. `nhbrdist` est calculée par un appel à la fonction `unit.distances()` qui calcule la matrice de distances entre toutes les unités.
- `init` : valeurs de départ
- ...

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages :
  - Proportion de résidences principales
  - Proportion de maisons parmi les résidences principales
  - Proportion de propriétaires parmi les occupants des résidences principales
  - Proportion de HLM parmi les résidences principales
  - Proportion de surfaces des résidences principales en tranches ($s_i \leq$ 40 m^2, 40 m^2 $< s_i \leq$ 99 m^2, $s_i >$ 99 m^2)
  - Ancienneté dans le logement en tranches ($t_i \leq$ 2 ans, 2 ans $< t_i \leq$ 4 ans, 4 ans $< t_i \leq$ 9 ans, $t_i >$ 9 ans)
  - Nombre de voitures possédées par les membres du ménage (0, 1, > 1)

  Note : tous les champs renseignés dans les bulletins de recensement ne font pas diffusé (par ex. : le nombre d'étage du bâtiment)

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages
- Démographie :
  - Densité de la population
  - Proportion de ménages composés d'une seule personne
  - Proportion de ménages comportant au moins un enfant de ≤ 24 ans
  - Proportion d'individus ≤ 19 ans et ≥ 65 ans

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages
- Démographie
- Emploi :
  - Proportion d'actifs dans la population
  - Proportion d'actifs occupés
  - Proportion d'actifs occupés agriculteurs

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages
- Démographie
- Emploi
- Occupation des sols :
  - Proportion de terre agricoles
  - Proportion de bâti

Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages
- Démographie
- Emploi
- Occupation des sols
- Équipements :
  - Nombre de boulangeries
  - Nombre de maternelles
  - Nombre de médecins généralistes

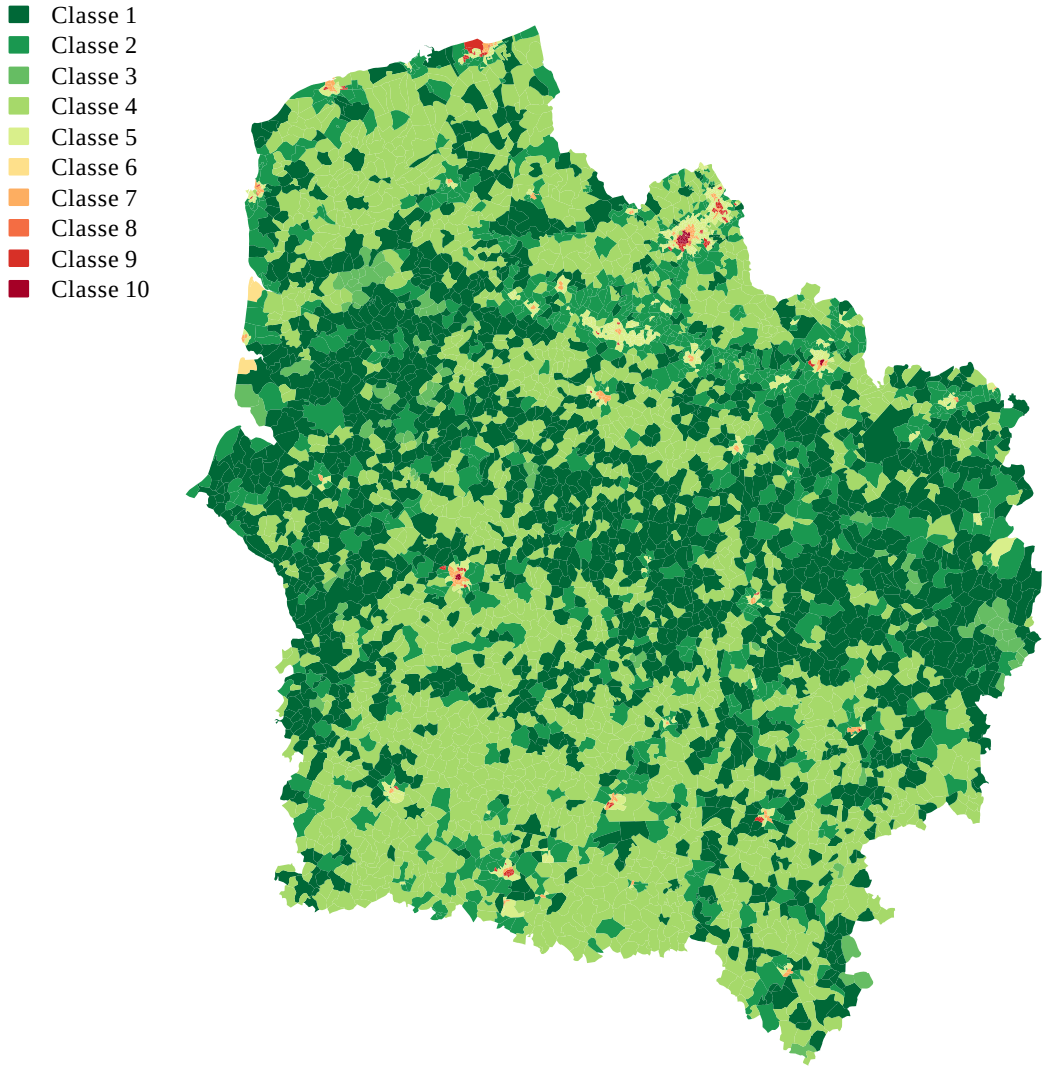
Variables retenues

Les variables retenues sont les suivantes (année 2010) :

- Caractéristiques des logements et des ménages
- Démographie
- Emploi
- Occupation des sols
- Équipements

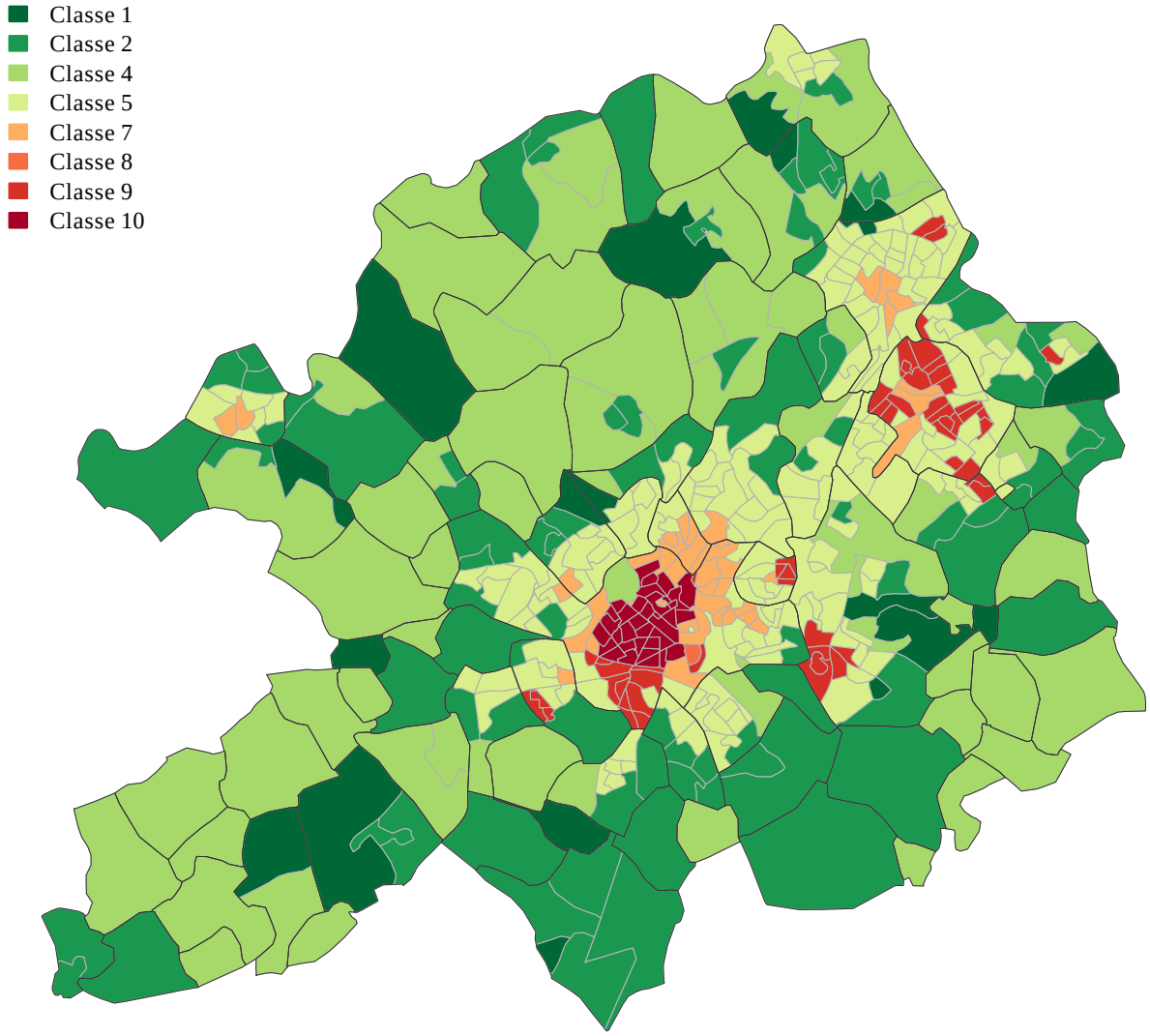
Note : les analyses ont été réalisées dans les contours de la nouvelle région Nord Pas-de-Calais Picardie pour l'instant (5 056 Iris au total)

Typologie des IRIS de la région Nord Pas-de-Calais Picardie



Source : IGN base Contours IRIS ® , INSEE enquête annuelle de recensement 2010

Typologie des IRIS de la métropole lilloise



Source : IGN base Contours IRIS ® , INSEE enquête annuelle de recensement 2010

Merci pour votre attention

Références bibliographiques I

Anselin, L. (1995). Local Indicators of spatial Association – LISA. *Geographical Analysis*, 27(2) :93–115.

Arribas-Bel, D. and Schmidt, C. R. (2011). Self-organizing Maps and the US Urban Spatial Structure. Working paper, GeoDa Center for geopspatial analysis and computation.

Bottou, L. and Bengio, Y. (1995). Convergence Properties of the K-means Algorithms. In Tesauro, G. and Touretzky, D., editors, *Advances in Neural Information Processing Systems 7*, chapter 10, page 585–592. MIT Press, Cambridge, MA.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2 edition.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, 3 edition.

van Hulle, M. M. (2011). Self-organizing maps. In Rozenberg, G., Bäck, T., and Kok, J. N., editors, *Handbook of Natural computing ; Theory, Experiments and Applications*, chapter 19, pages 585–622. Springer.

Wehrens, R. and Buydens, L. (2007). Self- and Super-organising Maps in R : the kohonen package. *J. Stat. Softw.*, 21(5).

Yin, H. (2008). The Self-Organizing Maps : Background, Theories, Extensions and Applications. In Fulcher, J. and Jain, L. C., editors, *Computational Intelligence : A Compendium*, number 115 in Studies in Computational Intelligence, page 715–762. Springer.