

# COURS DE DATA MINING

## 8 : MODELISATIONS

### RESEAUX DE NEURONES ET DE KOHONEN

EPF – 4/ 5<sup>ème</sup> année - Option Ingénierie d’Affaires et de Projets - Finance

Bertrand LIAUDET

<b>8 : Modélisations - Réseaux de neurones et de Kohonen</b>	<b>2</b>
<b>Les réseaux de neurones.....</b>	<b>2</b>
1 : Généralités	2
2 : Architecture et principes de fonctionnement	3
3 : Type des données en entrée	4
4 : Exploitation des résultats	4
5 : Paramétrage de la couche cachée	5
6 : Valeurs des nœuds et des liaisons	5
7 : La fonction sigmoïde	5
8 : La SEC	6
9 : La rétropropagation	7
10 : Critères d’arrêt	9
11 : Modèle avec plusieurs variables cibles	10
12 : Interprétation des résultats : l’analyse de la sensibilité	10
<b>Applications .....</b>	<b>11</b>
Le fichier de churn	11
<b>Les réseaux de Kohonen.....</b>	<b>19</b>
1 : Généralités	19
2 : Les cartes auto-organisatrices (SOM)	19
3 : Les réseaux de Kohonen	20
4 : Avantages et inconvénients	20
5 : Interprétation des résultats : validité et interprétation des classes	21
<b>Applications .....</b>	<b>22</b>
Le fichier de churn	22

1<sup>ère</sup> édition : octobre 2008.

# 8 : MODELISATIONS

## - RESEAUX DE NEURONES ET DE KOHONEN

### Les réseaux de neurones

#### 1 : Généralités

##### Imiter le cerveau humain

L'inspiration à l'origine de la technique des « réseaux de neurones formels », communément appelé « réseau de neurones » vient du fait que le cerveau humain est un système apprenant qui n'est pas basé sur les principes de la logique formelle mais sur une structure, le cerveau humain, contenant environ 100 milliards de neurones reliés entre eux par 10 000 contacts synaptiques soit environ un million de milliards de synapses.

Les réseaux de neurones formels sont une tentative pour imiter le mécanisme d'apprentissage qui se produit dans le cerveau.

##### Neurone réel

Les neurones réels présentent trois régions principales : le corps cellulaire, les dendrites - prolongements relativement courts et arborescents du corps cellulaire - et l'axone, prolongement long et fibreux.

Un neurone utilise des dendrites pour rassembler des données d'entrée issues d'autres neurones. Ces données d'entrée sont combinées pour produire une réponse envoyée à d'autres neurones ou d'autres cellules.

Les axones transportent les influx en provenance du corps cellulaire vers d'autres cellules (la longueur d'un axone est très variable ; elle peut atteindre 1 m chez l'homme et près de 10 m chez la girafe).

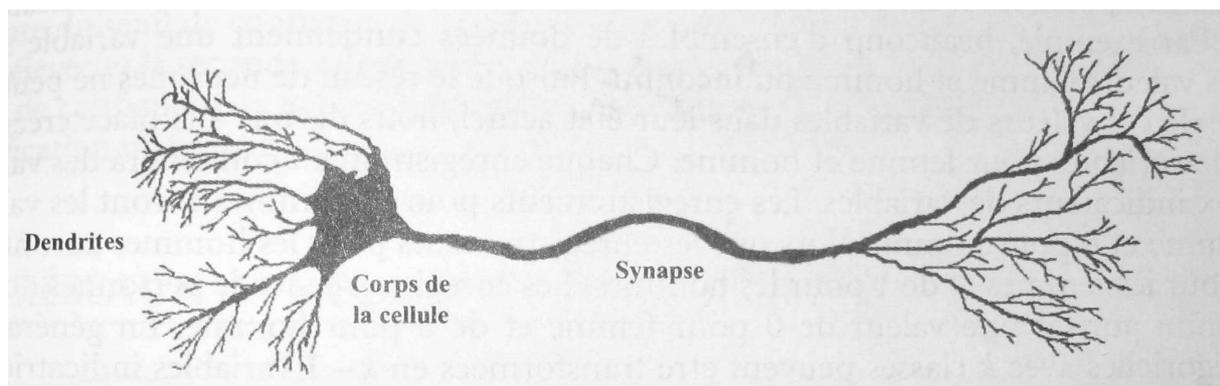


Schéma d'un neurone réel<sup>1</sup>

<sup>1</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

## Neurone formel

Les **données d'entrée** ( $x_i$ ) sont recueillies à partir des neurones du flux supérieur dans l'ensemble des données, et sont combinées dans une **fonction combinatoire** telle la somme. Cette fonction combinatoire est en entrée d'une **fonction d'activation** qui produit une **réponse envoyée en entrée d'autres neurones**.

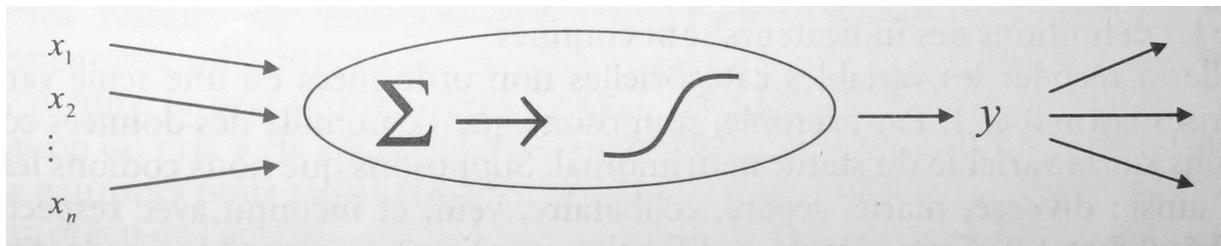


Schéma d'un neurone formel<sup>2</sup>

## Avantage des réseaux de neurones

Robuste aux données bruitées.

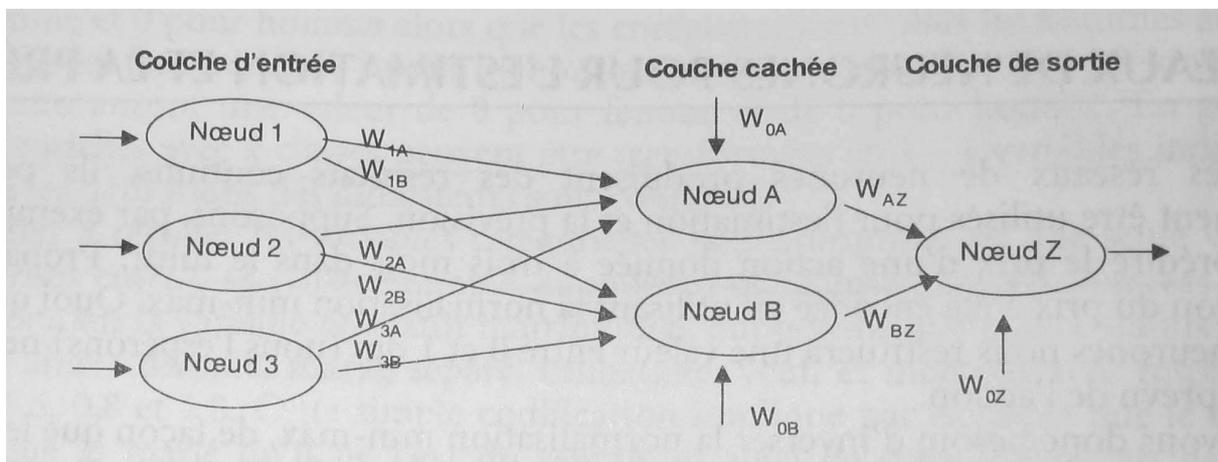
Permettent de modéliser de grandes variétés de comportements.

## Inconvénients

Les résultats sont assez opaques, à la différence de la méthode des arbres de décision.

La mise en œuvre, qui passe par un apprentissage, peut être longue.

## **2 : Architecture et principes de fonctionnement**



Exemple d'un petit réseau de neurones<sup>3</sup>

- Un réseau de neurones formels est disposé en couches de neurones formels.

<sup>2</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

<sup>3</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

- Les neurones sont appelés « nœuds ».
- La plupart des réseaux sont constitués de 3 couches successives : une **couche d'entrée**, une **couche cachée** et une **couche de sortie**. Toutefois, il peut y avoir 0 ou N couches cachées.
- D'une couche à l'autre, tous les nœuds de la première couche (nœuds « in ») sont reliés à tous les nœuds de la seconde (nœuds « out »).
- Chaque liaison a un poids : une valeur entre 0 et 1.
- Chaque nœuds des couches cachées et de sortie possède aussi un poids : une valeur entre 0 et 1.
- Le nombre de nœuds de la couche d'entrée dépend du nombre de variables prises en compte et de leur type. En simplifiant, on peut dire qu'on a un nœud par variable en entrée.
- Le nombre de couches cachées et le nombre de nœuds pour chaque couche cachée est paramétrable par l'utilisateur.
- En général, la couche de sortie ne contient qu'un nœud. Toutefois, elle peut en contenir plus. En simplifiant, on peut dire que ce nœud correspond à la variable de sortie.

### **3 : Type des données en entrée**

La valeur des données en entrée et en sortie doit être comprise entre 0 et 1.

#### **Traitement des variables numériques**

On applique une standardisation « min-max » aux données numérique :

$$x' = (x - \text{moy}(X)) / (\text{max}(X) - \text{min}(X))$$

Si on applique les résultats à une population dans laquelle le min et le max ont changé, on peut obtenir des résultats erronés.

#### **Traitement des variables catégorielles :**

Si elles sont ordonnées, on peut affecter à chaque catégorie une valeur comprise entre 0 et 1.

Si elles ne sont pas ordonnées, la méthode précédent risque de conduire à des résultats erronés du fait de la création de voisinages irréels.

Chaque catégorie peut être alors être traitée comme une variable booléenne.

### **4 : Exploitation des résultats**

Les résultats sont compris entre 0 et 1.

#### **Classification**

On peut créer des classes, le nombre de classes choisi définissant l'amplitude.

Exemple : 4 classes d'amplitude  $1/4 = 0,25$ .

#### **Prévision d'une variable X**

Pour la prévision, le résultat sera « dénormalisé » :

$$x' = x * ( \max(X) - \min(X) ) + \min(X)$$

### Exemple :

Prévision du prix d'une action dont le min vaut 20, le max 30 et la sortie du réseau 0.69 :

$$\text{Prévision} = 0,69 * (30-20) + 20 = 26,9$$

## 5 : Paramétrage de la couche cachée

On peut choisir le nombre de nœuds de la couche cachée et le nombre de couche cachée.

Plus le nombre de nœuds augmente, plus le réseau est apte à identifier des phénomènes complexe.

Toutefois, un trop grand nombre de nœuds conduit à un sur-apprentissage dans l'échantillon d'apprentissage finalement nuisible aux échantillons de test.

## 6 : Valeurs des nœuds et des liaisons

- Lors de l'initialisation, un poids est donné aléatoirement à chaque liaison et à chaque nœud des couches cachées et de sortie. **L'ajustement de ces poids représente la clé du mécanisme d'apprentissage par le réseau de neurones.**
- Pour un individu, les nœuds de la couche d'entrée prennent la valeur normalisée des variables d'entrée du modèle.
- Pour un individu, les nœuds des couches cachées et de sortie prennent une valeur qui est une combinaison (une somme le plus souvent) des combinaisons linéaires des nœuds « in » et des poids correspondants.

Pour un nœud j donné, on a donc :

$$\text{NET } j = \text{Somme pour } i \text{ de } 0 \text{ à } N ( W_{ij} * X_i )$$

Avec :

NET : valeur du nœud dans le réseau.

i : allant de 0 à N, N étant le nombre de nœuds « in ».

$W_{ij}$  : poids de la liaison entre le nœud « i » qui est « in » et le nœud « j » qui est « out ».

$X_i$  : valeur du nœud « i », avec  $X_0 = 1$ .

## 7 : La fonction sigmoïde

Dans un neurone réel, les signaux sont envoyés entre les neurones quand la combinaison des données d'entrée dépasse un certain seuil : le **seuil d'activation**. Le comportement n'est pas linéaire car la réponse ne dépend pas linéairement de l'incrément de la stimulation.

La fonction qui modélise ce comportement est appelée : **fonction d'activation**. C'est une fonction non linéaire.

La fonction d'activation la plus commune est la **fonction sigmoïde** :

$$y = 1 / ( 1 + \exp(- x) )$$

soit :

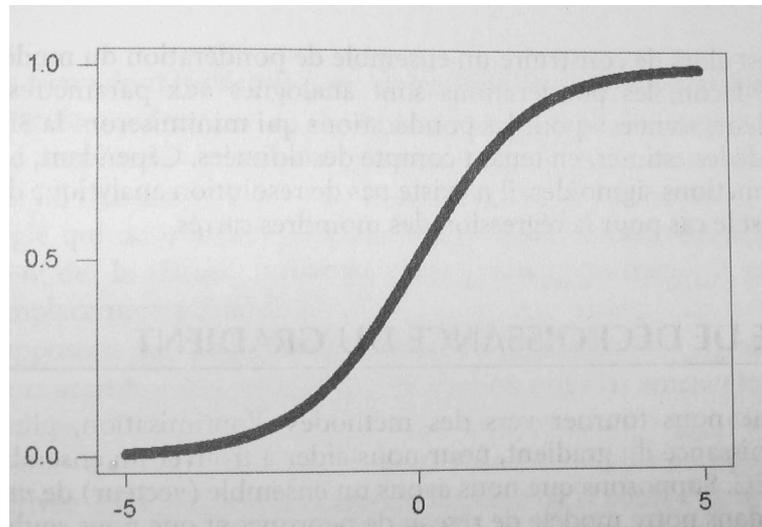
$$\text{SIG} ( \text{NET}(n) ) = 1 / ( 1 + \exp(-\text{NET}(n)) )$$

Avec :

exp : fonction exponentielle :  $\exp(1) = 2,7$ .

NET(n) : NET du nœud « n ».

SIG ( NET(n) ) : sigmoïde du NET du nœud « n ».



Graphique de la fonction sigmoïde<sup>4</sup>

La fonction sigmoïde est telle que lorsque les données d'entrée sont proches du centre de l'intervalle,  $f(x)$  est linéaire. Lorsque les données d'entrée s'éloignent du centre,  $f(x)$  est curviligne. Lorsque les données sont très éloignées du centre,  $f(x)$  devient quasiment constante.

Donc les incréments du NET d'un nœud produisent des incréments variables de SIG(NET) : près du centre, un incrément du NET produit un incrément linéaire du SIG. Plus on s'écarte du centre, moins l'incrément du NET a d'effet sur le SIG. Loin du centre un incrément du NET ne produit pas d'incrément du SIG.

La fonction sigmoïde est aussi appelée : **fonction d'écrasement** : elle écrase les extrêmes.

On va appliquer la fonction sigmoïde à la valeur NET de chaque nœud.

## 8 : La SEC

Les réseaux de neurones sont une **méthode supervisée** : on choisit une **variable cible**.

Chaque individu avec ses variables en entrée passe à travers le réseau et fournit un résultat dans le nœud de sortie. Cette valeur de sortie est comparée à celle de la variable cible.

**Erreur de prévision = valeur de la donnée réelle - valeur de sortie**

Cette erreur est analogue à celle des modèles de régression.

En général, les modèles à réseau de neurone calculent une somme des erreurs au carré (SEC) :

**SEC = Somme pour tous les enregistrements (donnée réelle – donnée en sortie)<sup>2</sup>**

<sup>4</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

Le problème consiste donc à **minimiser la valeur de SEC en fonction de l'ensemble des valeurs de pondération** des nœuds et des liaisons.

## 9 : La rétropropagation

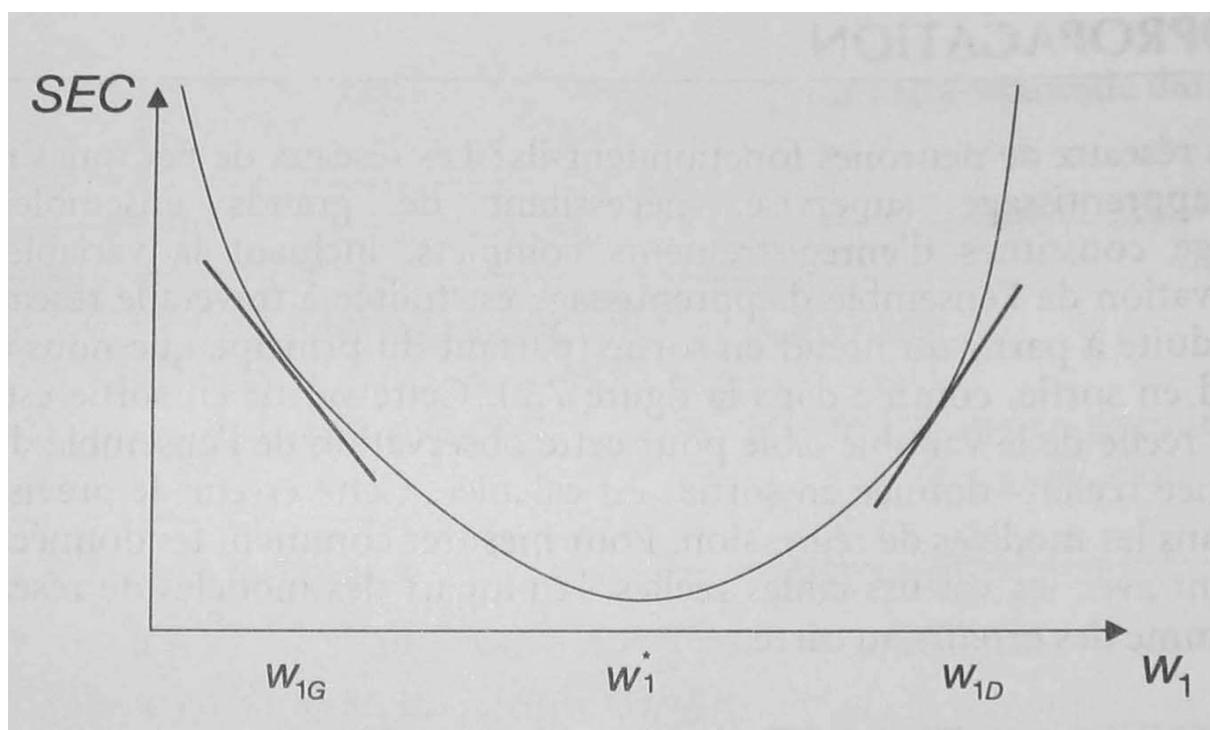
En raison de la nature non linéaire de la fonction sigmoïde, il n'existe **pas de résolution analytique de la minimisation de la SEC**.

La rétropropagation met en œuvre des calculs mathématiques et algorithmiques complexes que nous ne présentons pas ici.

Nous présentons seulement les principaux concepts et paramètres qui entrent en jeu.

### Méthode de décroissance du gradient de la SEC pour ajuster les pondérations

Pour minimiser la SEC, on utilise la « **méthode de décroissance du gradient** » qui donne la direction dans laquelle il faut ajuster la pondération pour faire décroître la SEC.



Courbe de l'évolution du SEC en fonction du poids<sup>5</sup>

La courbe ci-dessus montre une évolution parabolique de la SEC en fonction d'une seule pondération. C'est une simplification qui permet de montrer que la dérivée de la courbe donne la pente et nous dit dans quel sens il faut ajuster le poids.

### La rétropropagation

La rétropropagation consiste à **ajuster les poids des nœuds et des liaisons en remontant du nœud de la couche de sortie aux nœuds de la couche d'entrée**.

---

<sup>5</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

En général, les réseaux font la mise à jour après chaque calcul de la valeur de sortie d'un enregistrement.

Cet ajustement sera fonction de :

- L'erreur de prévision
- Le taux d'apprentissage : valeur comprise entre 0 et 1.

### Le taux d'apprentissage ( $\eta$ )

Le taux d'apprentissage est un paramètre qui favorise l'évolution de la SEC vers le minimum.

Quand le taux d'apprentissage est faible, les ajustements sont faibles.

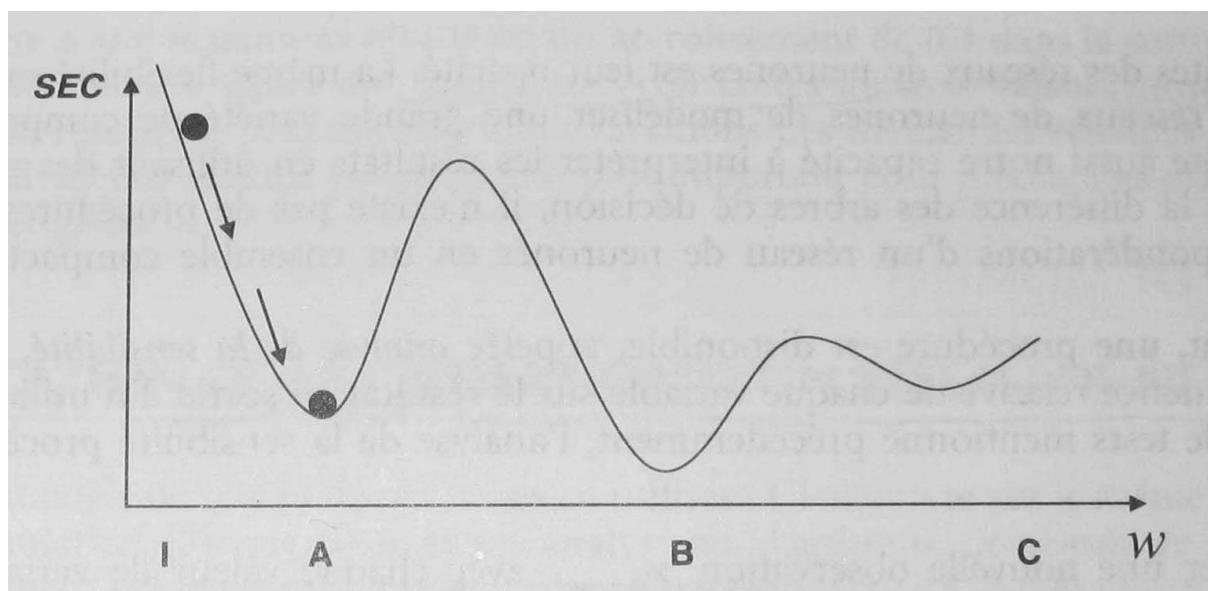
Quand le taux d'apprentissage est fort, les ajustements sont forts. Mais un taux d'apprentissage trop fort fait dépasser la SEC optimum.

Le taux d'apprentissage peut évoluer au cours de l'apprentissage. Au début, il est élevé pour s'approcher rapidement de la solution. Quand le réseau commence à converger, le taux est graduellement réduit pour ne pas dépasser la SEC optimum.

### Le terme de moment ( $\alpha$ )

Le terme de moment est un paramètre supplémentaire qui favorise l'évolution de la SEC vers le minimum.

Intuitivement, on peut comprendre son fonctionnement ainsi : la courbe d'évolution de la SEC en fonction des poids n'est pas une simple parabole. Elle contient plusieurs minimums ou « paliers ». Le terme de moment permet d'éviter que la recherche du meilleur minimum s'arrête à un palier intermédiaire ou qu'il se trouve avant ou après le meilleur palier.



Courbe de l'évolution du SEC en fonction du poids<sup>6</sup>

On peut interpréter cette courbe en disant que le terme de moment favorise le fait de ne pas s'arrêter au palier A, le fait de ne pas aller au palier C, le fait de s'arrêter au palier B.

<sup>6</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

## 10 : Critères d'arrêt

L'algorithme peut traiter tous les enregistrements de l'ensemble des données d'apprentissage un nombre de fois indéterminé. Il faut donc déterminer un critère d'arrêt.

### Le temps : à éviter

La modélisation par réseau de neurones peut prendre plusieurs heures ! Le temps peut donc être un critère d'arrêt si on est pressé, mais il risque de conduire à un modèle peu efficace.

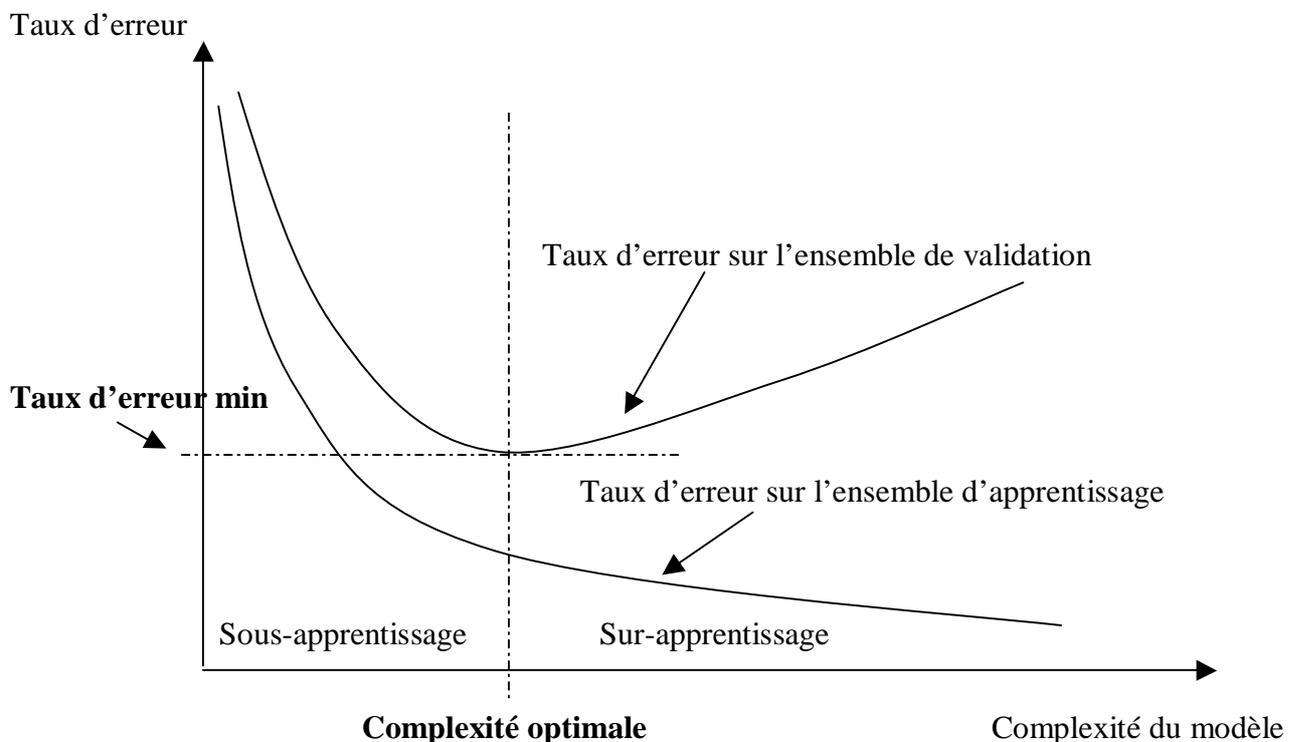
### Minimiser la SEC dans l'ensemble d'apprentissage : à éviter !

La SEC peut être un critère d'arrêt mais elle risque de conduire à un sur-apprentissage mémorisant des caractéristiques idiosyncrasiques (propres aux individus indépendamment de leur groupe).

### Minimiser la SEC dans l'ensemble de validation

En même temps qu'on minimise la SEC dans l'ensemble d'apprentissage, on vérifie qu'on la minimise aussi dans l'ensemble de validation. Quand elle croît dans l'ensemble de validation, c'est qu'on commence à entrer en phase de sur-apprentissage. C'est un bon critère d'arrêt.

#### ➤ Rappel du problème du sur-apprentissage :



Le modèle doit atteindre le taux d'erreurs minimum pour l'ensemble de validation.

## **11 : Modèle avec plusieurs variables cibles**

On peut avoir plusieurs variables cibles dans un réseau de neurone. La couche de sortie aura alors autant de nœud qu'il y a de variable cible. La prédiction se fera sur le n-uplet de valeur des variables cibles.

## **12 : Interprétation des résultats : l'analyse de la sensibilité**

Un inconvénient des réseaux de neurones est leur opacité.

Il fournit une fonction de prédiction mais cette fonction n'est pas traduisible, comme dans les arbres de décision, en un ensemble de règles intuitivement compréhensibles.

Cependant, on peut mesurer l'influence relative de chaque variable sur le résultat en sortie.

L'analyse de la sensibilité effectue cette mesure.



## Applications

### Le fichier de churn

Réseau de neurones

Pour avoir des précisions sur le mode d'emploi, on utilise le bouton « ? » dans la fenêtre en cours.

On va utiliser 2 méthodes :

- Rapide.
- Multiple. Cette méthode crée plusieurs réseaux de topologie différente (le nombre exact de réseaux créés dépend des données d'apprentissage).

Ces deux méthodes sont assez rapides.

- Elagage exhaustif. Cette méthode donne souvent les meilleurs résultats, c'est aussi la plus lente. Cette méthode peut se révéler très lente, tout particulièrement avec les ensembles de données volumineux.

## Résultats avec la méthode rapide

The screenshot displays the results of a neural network analysis. It is organized into several sections:

- Analyse**:
  - Précision estimée : 92,421
  - Couche d'entrée : 13 neurones
  - Couche cachée 1 : 3 neurones
  - Couche de sortie : 1 neurones
- Importance relative des entrées**: A table showing the relative importance of input features.

Appels au service client	0.748625
Conso de jour en minutes	0.703762
International	0.291153
Conso internationale en minutes	0.178311
Conso de nuit en minutes	0.153778
Nb msg	0.08636
Mail	0.0757285
Nb dappels de jour	0.0627624
Nb dappels de nuit	0.0520928
Nb dappels internationaux	0.0509746
- Champs**:
  - Cible**: Churn? (radio button selected)
  - Entrées**: A list of input features with checkboxes. The checked features are: Appels au service client, Conso de jour en minutes, Conso de nuit en minutes, Conso internationale en minutes, International, Mail, Nb dappels de jour, Nb dappels de nuit, Nb dappels internationaux, and Nb msg.
- Créer des paramètres**:
  - Utiliser les données partitionnées: faux
  - Méthode: Rapide

La synthèse affiche :

- La précision : 92,42
- Les variables en entrée et la variable cible.
- La méthode : rapide et la topologie du réseau : 1 couche cachée et 3 nœuds cachés.
- Les attributs les plus influents dans la prédiction du churn :
  - Appel au service client et consommation jour
  - International et les appels au service client.
- La durée du calcul (pas sur notre graphique) : 2 s

## Résultats avec la méthode multiple :

**Analyse**

- Précision estimée : 95,178
- Couche d'entrée : 13 neurones
- Couche cachée 1 : 7 neurones
- Couche cachée 2 : 5 neurones
- Couche de sortie : 1 neurones

**Importance relative des entrées**

Conso de jour en minutes	0.779745
Appels au service client	0.551973
International	0.299785
Conso de nuit en minutes	0.220306
Conso internationale en minutes	0.11119
Mail	0.098513
Nb msg	0.0977003
Nb dappels internationaux	0.0898257
Nb dappels de nuit	0.0683568
Nb dappels de jour	0.0257146

**Champs**

- Cible**
  - Churn?
- Entrées**
  - Appels au service client
  - Conso de jour en minutes
  - Conso de nuit en minutes
  - Conso internationale en minutes
  - International
  - Mail
  - Nb dappels de jour
  - Nb dappels de nuit
  - Nb dappels internationaux
  - Nb msg

**Créer des paramètres**

- Utiliser les données partitionnées: faux
- Méthode: Multiple

La synthèse affiche :

- La précision : 95,17
- Les variables en entrée et la variable cible.
- La méthode : rapide et la topologie du réseau : 2 couches cachées et 12 nœuds cachés.
- Les attributs les plus influents dans la prédiction du churn :
  - Appel au service client et consommation jour
  - International et consommation nuit
- La durée du calcul (pas sur notre graphique) : 1mn 23 s

Du point de vue des paramètres observés, les deux méthodes se valent à peu près.

## Comparaison entre les deux méthodes a partir de l'analyse du « churn »

Répartition du churn dans la population initiale :

Valeur ▲	Proportion	%	Comptage
False.		85,51	2850
True.		14,49	483

Churn?  
 False.       True.

Répartition du churn d'origine dans le churn de « modélisation rapide »

Valeur ▲	Proportion	%	Comptage
False.		92,29	3076
True.		7,71	257

Churn?  
 False.       True.

Répartition du churn d'origine dans le churn de « modélisation multiple »

Valeur ▲	Proportion	%	Comptage
False.		84,73	2824
True.		15,27	509

Churn?  
 False.       True.

On constate que la modélisation rapide arrive à 509 churn à true, contre 257 dans la modélisation rapide. Autrement dit, la modélisation rapide donne un résultat plus proche de la situation d'origine dans laquelle il y a 483 churn à true.

On constate toutefois que les deux modélisations donnent un résultat assez mauvais : moins de la moitié des churn modélisé à true correspondent à un churn true dans la population d'origine.

## Comparaison avec la modélisation par règle de décision



Valeur	Proportion	%	Comptage
False.	89,56	89,56	2985
True.	10,44	10,44	348

Churn?  
 False.  True.

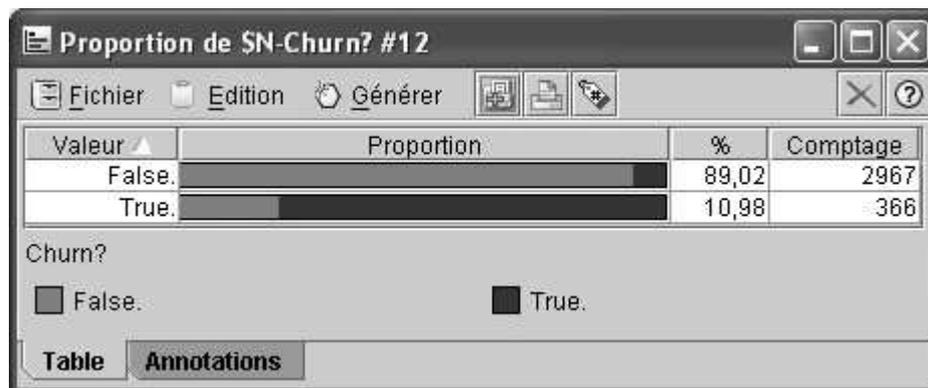
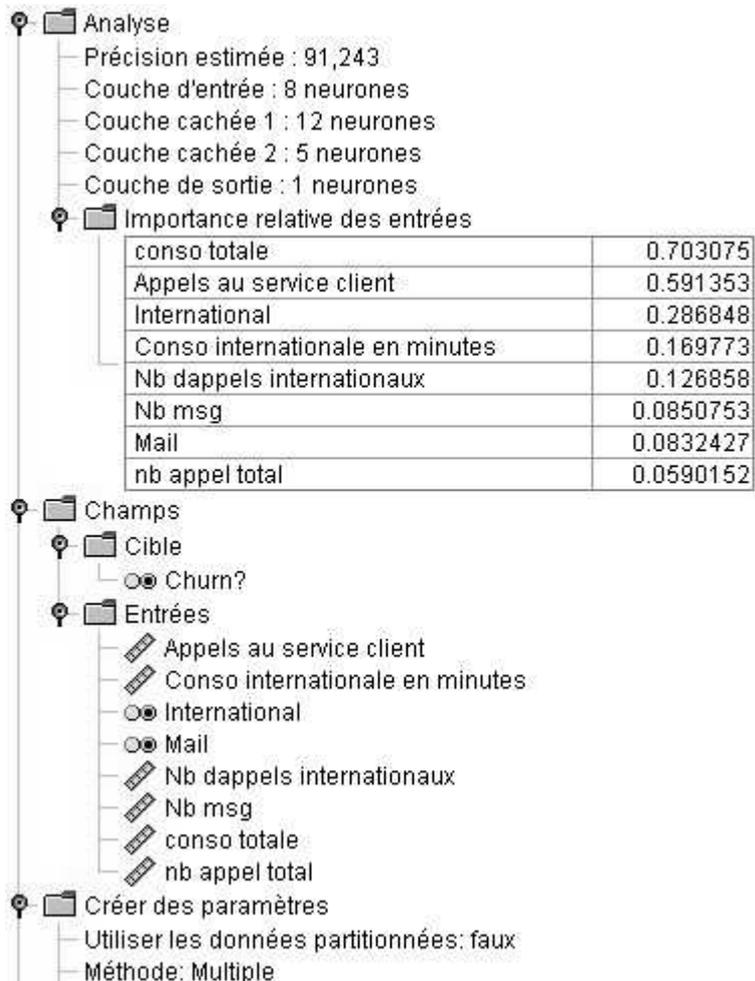
Table Annotations

On voit que la modélisation par règles de décision, on arrive à 348 churn à true, ce qui est moins bon qu'avec la modélisation par neurones multiples. Par contre les 348 churn à true correspondre massivement aux churn à true d'origine.

## Création d'un attribut « somme des consommation »

On crée un attribut qui somme les consommation de jour, soirée et nuit, ainsi que le nombre d'appels.

### ➤ Résultat de la modélisation neurones-multiple



On voit que le résultat est plus intéressant qu'avec les consommation séparées : la concordance entre le churn à true calculé et le churn à true d'origine est meilleure.

A noter qu'on pourrait accéder au chiffrage avec le nœud Sortie / Matrice.

➤ Comparaison avec la modélisation par règles de décision



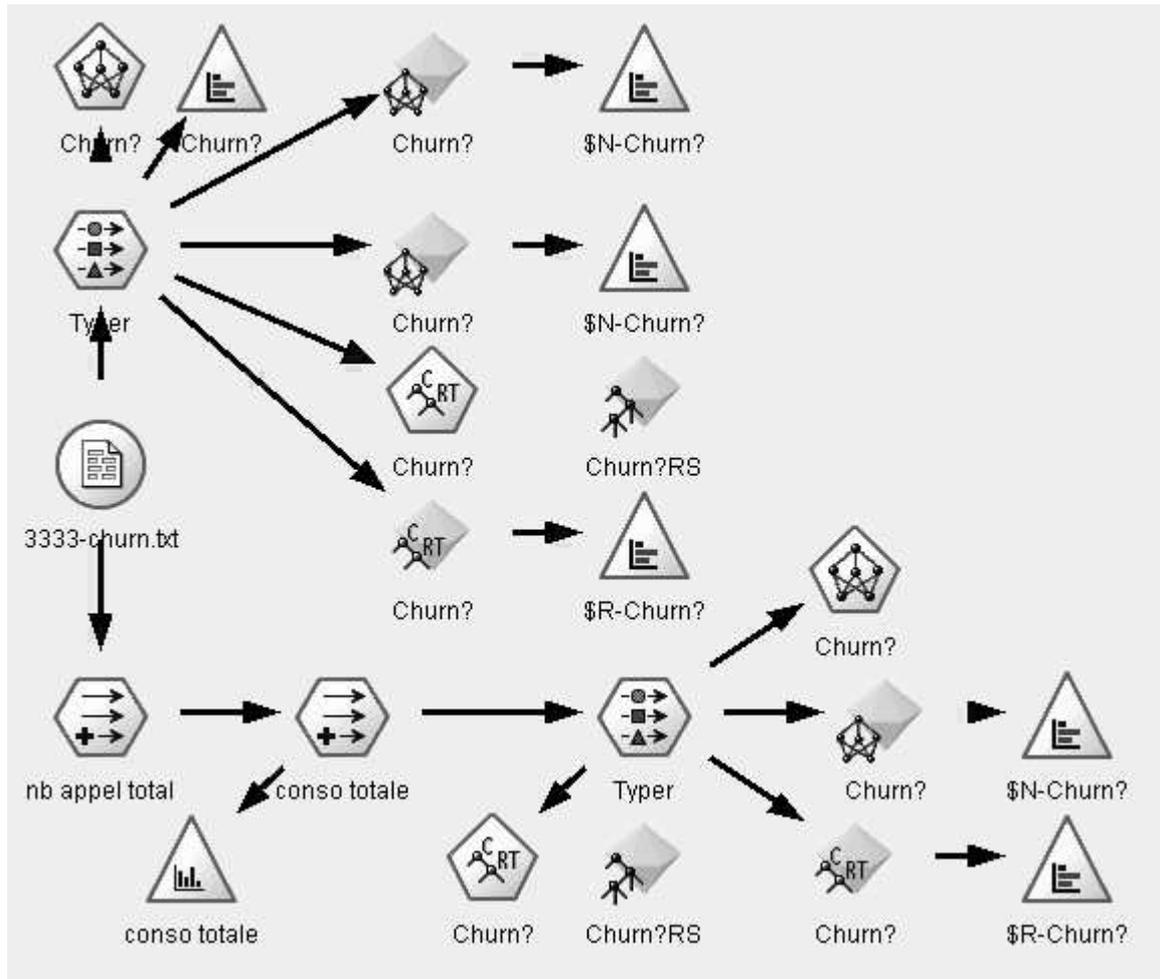
Valeur ▲	Proportion	%	Comptage
False.		89,98	2999
True.		10,02	334

Churn?  
 False.                       True.

Table    Annotations

On constate que les résultats se valent entre les deux modélisations.

## Flux clémentine



Flux Clémentine : Réseaux de neurones, Règles de décision.

## Les réseaux de Kohonen

### 1 : Généralités

Introduits en 1982.

Initialement appliqués à l'image et au son.

Mécanisme efficace de classification pour les données alpha-numérique.

C'est une espèce de réseau de neurones.

C'est aussi une espèce de carte auto-organisatrice (SOM, Self Organizing Map).

### 2 : Les cartes auto-organisatrices (SOM)

#### Généralités

Une carte auto-organisatrice est un procédé qui convertit un signal d'entrée complexe (plusieurs variables par exemple) en une nouvelle variables catégorielle : c'est donc un procédé de classification (modélisation non-supervisée).

Les SOM sont une généralisation de l'analyse en composantes principales.

Elle fonctionne comme un réseau de neurones sans variable cible et avec plusieurs nœuds dans la couche de sortie. La carte structure les nœuds en sortie en classes de nœuds.

#### Architecture

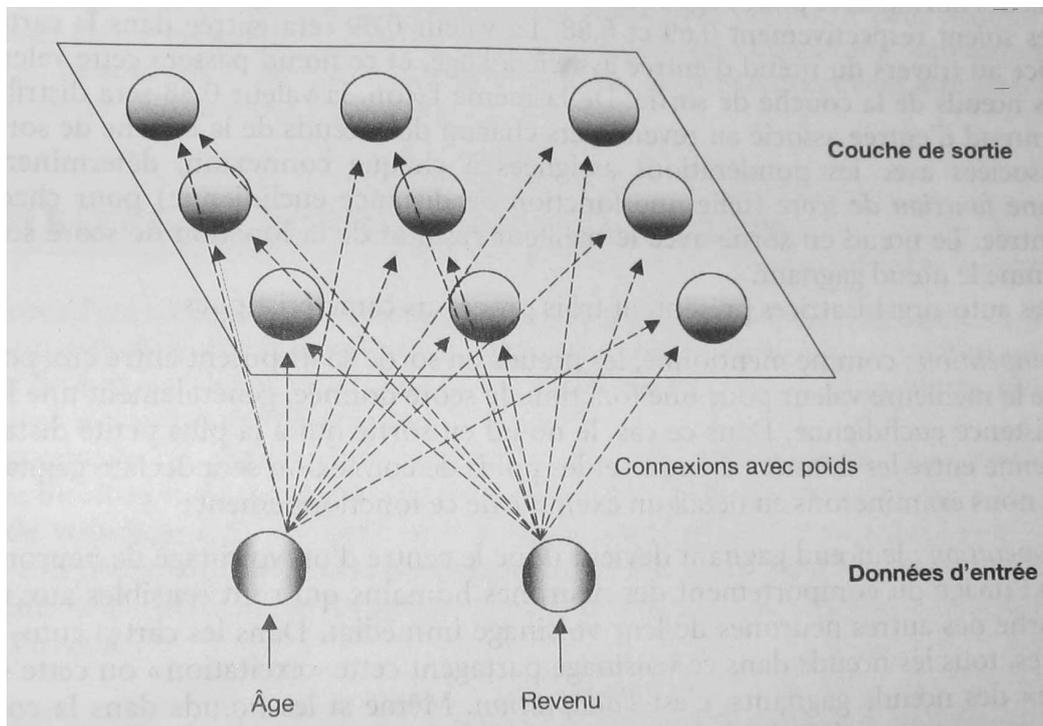


Schéma d'une carte auto-organisatrice (SOM)<sup>7</sup>

<sup>7</sup> D'après « Des données à la connaissance », de Daniel T. Larose. Vuibert 2005.

A la différence des réseaux de neurones, les SOM n'ont pas de couche cachée.

La couche de sortie contient plusieurs nœuds représenté sous la forme d'un treillis rectangulaire (un carré de dimension 3 dans l'exemple ci-dessus).

Le nombre de nœuds de la couche de sortie est fixé arbitrairement par l'utilisateur. Il définit le nombre maximum de classes.

### **Principes de fonctionnement : fonction de score**

Les valeurs des nœuds de la couche d'entrée (valeurs normalisées des variables prises en compte par le modèle) sont distribuées dans les nœuds de la couche de sortie après transformation en fonction des pondérations du réseau : on parle de « **fonction de score** ». Cette fonction est généralement une fonction de distance euclidienne.

Le nœud de sortie qui a le meilleur résultat (on dit le « **meilleur score** ») est le « nœud gagnant » : il reçoit l'individu en question. Le meilleur score c'est la plus petite distance entre les poids de connexion et les données d'entrée.

### **Principe de fonctionnement : liaison de voisinage des nœuds de la couche de sortie**

Comme dans tous les réseaux de neurones, les nœuds d'une même couche, et particulièrement de la couche de sortie, ne sont pas reliés entre eux.

Toutefois, les poids des nœuds de voisinage du nœud gagnant sont **adaptés** pour favoriser leur victoire en cas de données similaires.

C'est ce qu'on appelle **la coopération et l'adaptation** des nœuds de la couche de sortie. L'adaptation, c'est ce qui correspond à **l'apprentissage**.

Comme dans tous les réseaux de neurones, les nœuds d'une même couche, et particulièrement de la couche de sortie, ne sont pas reliés entre eux.

## **3 : Les réseaux de Kohonen**

Les réseaux de Kohonen sont des cartes auto-organisatrice (SOM) avec une variation dans la technique d'apprentissage.

Dans les réseaux de Kohonen, les nœuds dans le voisinage du nœud gagnant ajustent leur poids en utilisant une combinaison linéaire du vecteur d'entrée et du vecteur de poids en cours.

## **4 : Avantages et inconvénients**

### **Avantages**

Les réseaux de Kohonen permettent une classification raffinée.

### **Inconvénients**

La lecture des résultats n'est pas facile

Le nombre de classe maximum est fixée a priori. C'est aussi l'ordre de grandeur du nombre de classes final.

## **5 : Interprétation des résultats : validité et interprétation des classes**

Comme pour toute analyse de classification, plusieurs techniques peuvent être utilisées pour valider et interpréter les classes.

### **Analyse exploratoire des classes obtenues**

Il faut faire une analyse exploratoire de chaque classe pour comparer les répartitions des différentes variables dans chacune des classes et les répartitions de variables cibles dans les différentes classes.

### **Supprimer les variables cibles potentielles des variables d'entrée**

Si on prévoit d'analyser certaines variables cibles dans les classes obtenues, il faut supprimer ces variables des variables d'entrée de la modélisation de Kohonen.

### **Arbre de décision**

On fait un arbre de décision avec comme variable cible la variable de classification créée par le réseau de Kohonen.

### **Travailler sur deux sous-populations**

On peut diviser la population d'origine en deux aléatoirement et vérifier qu'on obtient des résultats concordants dans les deux sous-ensembles par observation des répartitions dans chaque classe et par analyse exploratoire de chaque classe.

## Applications

### Le fichier de churn

Réseau de Kohonen

Pour avoir des précisions sur le mode d'emploi, on utilise le bouton « ? » dans la fenêtre en cours.

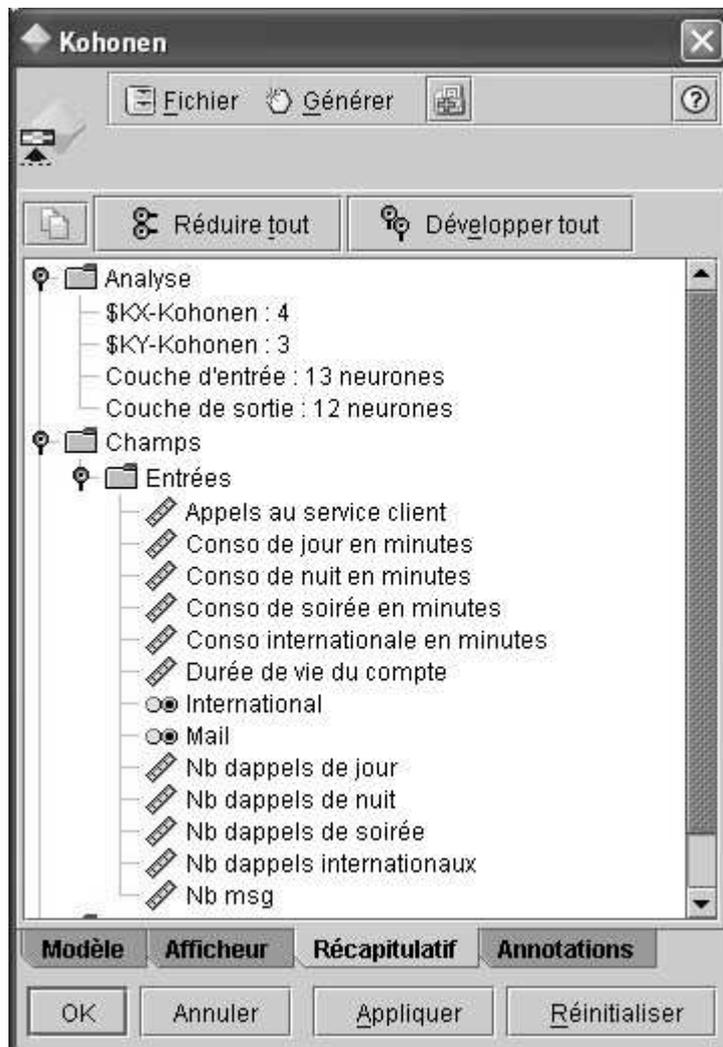
On va utiliser 2 méthodes :

- Le paramétrage par défaut
- Le paramétrage expert

### Résultats avec le paramétrage par défaut

Le paramétrage par défaut définit 4 fois 3 = 12 classes au maximum.

➤ Entrées du modèle :



## ➤ Classes trouvées



## ➤ Détails des classes

Pour analyser le détail des classes on peut :

- Utiliser l'onglet « afficheur » (cf. diagramme précédent).
- Créer un attribut correspondant à la nouvelle classe. Pour cela, faire un « générer nœud calculer » dans l'onglet modèle. Puis ajouter la formule suivante dans le nœud :

```
if ('$KX-Kohonen' = 0 and '$KY-Kohonen' = 0) then '00'  
elseif ('$KX-Kohonen' = 0 and '$KY-Kohonen' = 2) then '02'  
elseif ('$KX-Kohonen' = 1 and '$KY-Kohonen' = 0) then '10'  
elseif ('$KX-Kohonen' = 1 and '$KY-Kohonen' = 2) then '12'  
elseif ('$KX-Kohonen' = 2 and '$KY-Kohonen' = 0) then '20'  
elseif ('$KX-Kohonen' = 2 and '$KY-Kohonen' = 2) then '22'  
elseif ('$KX-Kohonen' = 3 and '$KY-Kohonen' = 0) then '30'  
elseif ('$KX-Kohonen' = 3 and '$KY-Kohonen' = 2) then '32'  
else '99' endif
```

Ensuite, il reste à faire une analyse exploratoire sur chaque classe.

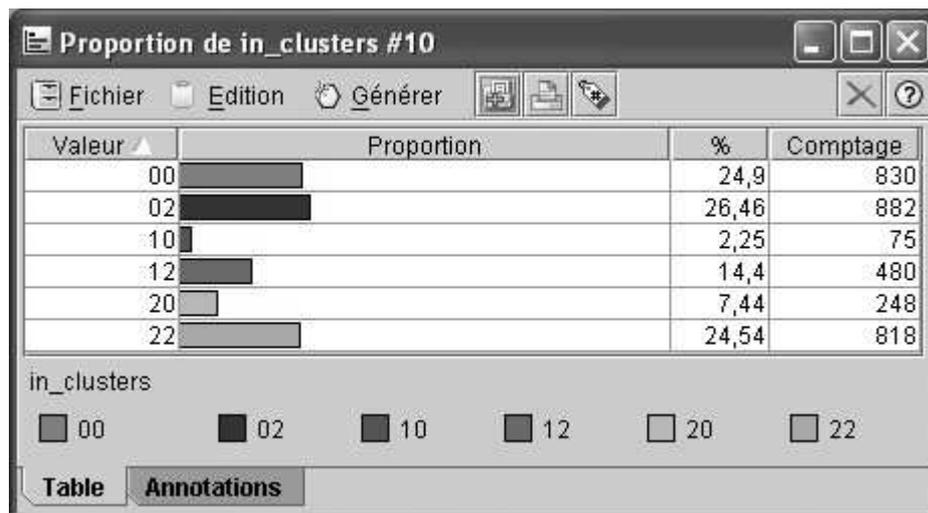
## Résultats avec le paramétrage expert

Dans le paramétrage expert, on choisit : largeur 3 et longueur 3.

Pour les 20 premiers cycle, le voisinage est à 2 et le taux d'apprentissage, « eta » décroît à partir de 0,3.

Pour les 150 cycles suivants, le voisinage est à 1 et le taux d'apprentissage, « eta » décroît à partir de 0,3.

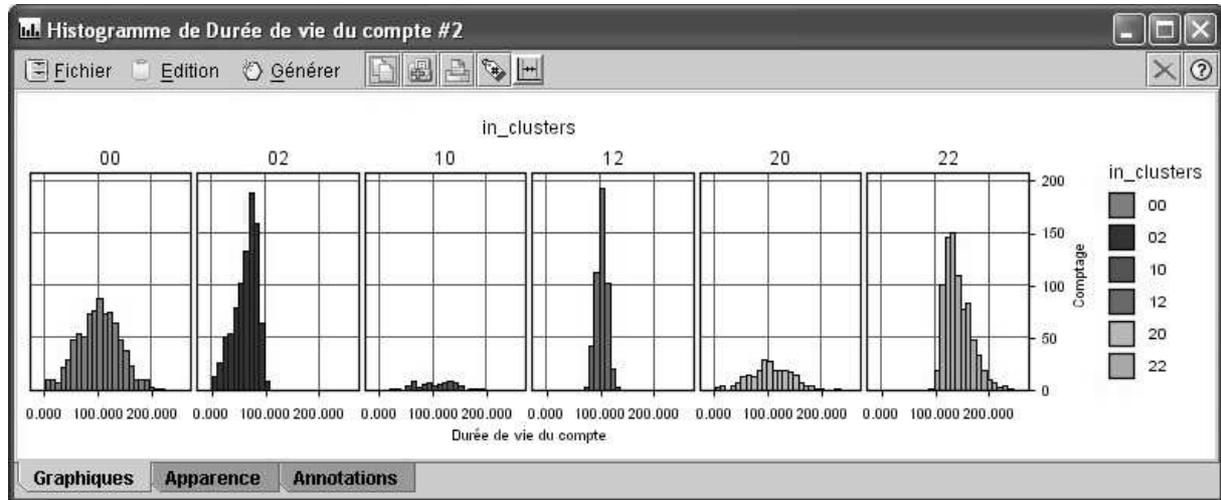
### ➤ Classes trouvées



## Interprétation des classes par analyse exploratoire

En utilisant l'afficheur et/ou en faisant une analyse exploratoire pour chaque classe, on arrive aux résultats suivants :

### ➤ Durée de vie des comptes



On a 3 classes classiques (00, 10 et 20).

La classe 2 est à durée de vie faible.

La classe 12 à durée de vie moyenne.

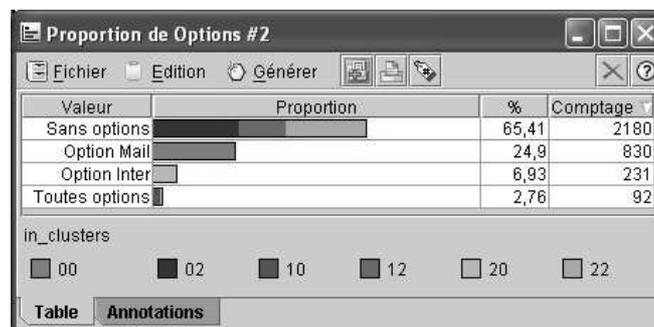
La classe 22 à durée de vie forte.

### ➤ Options

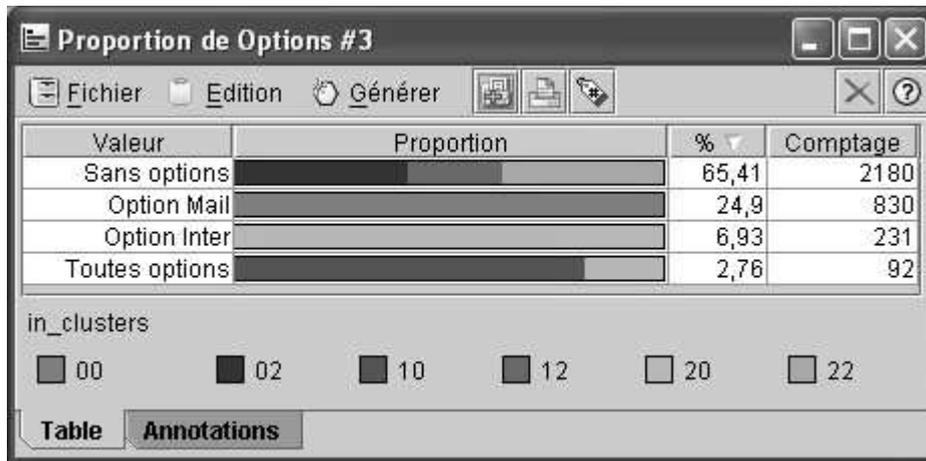
On peut aussi observer la répartition des classes dans les options. Pour cela, on crée un attribut calculer qui prend en compte les deux options.

#### **Attribut options :**

```
if ('Mail' and 'International') then 'Toutes options'  
elseif ('Mail' and not('International')) then 'Option Mail'  
elseif (not('Mail') and 'International') then 'Option Inter'  
else 'Sans options'  
endif
```



Répartition des classes dans les catégories d'options



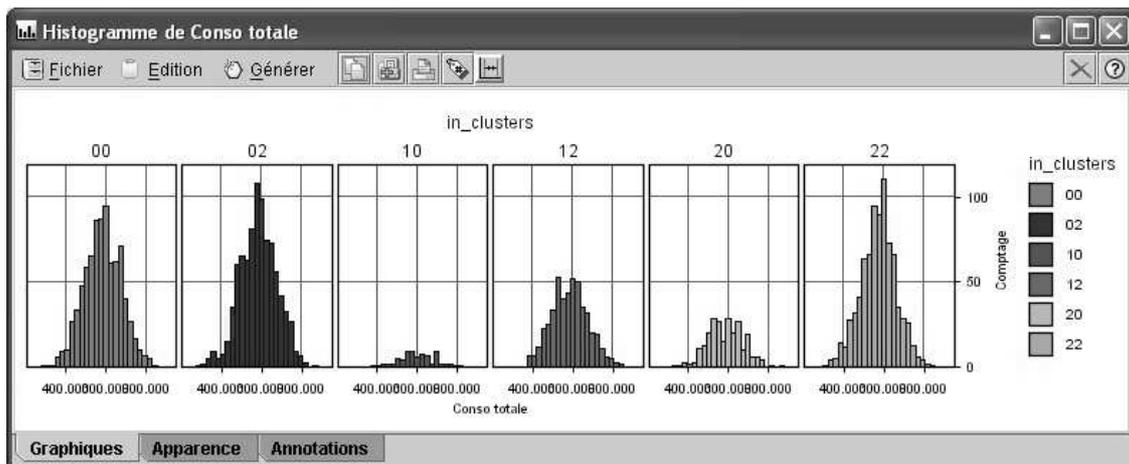
Répartition normalisée des classes dans les catégories d'options

Si on compare à la répartition des classes en fonction de la durée de vie, on constate que les 3 classes sans options se divisent en fonction de la durée de vie.

Les 3 autres classes (mail, inter et toutes options) ont une durée de vie de type gaussien.

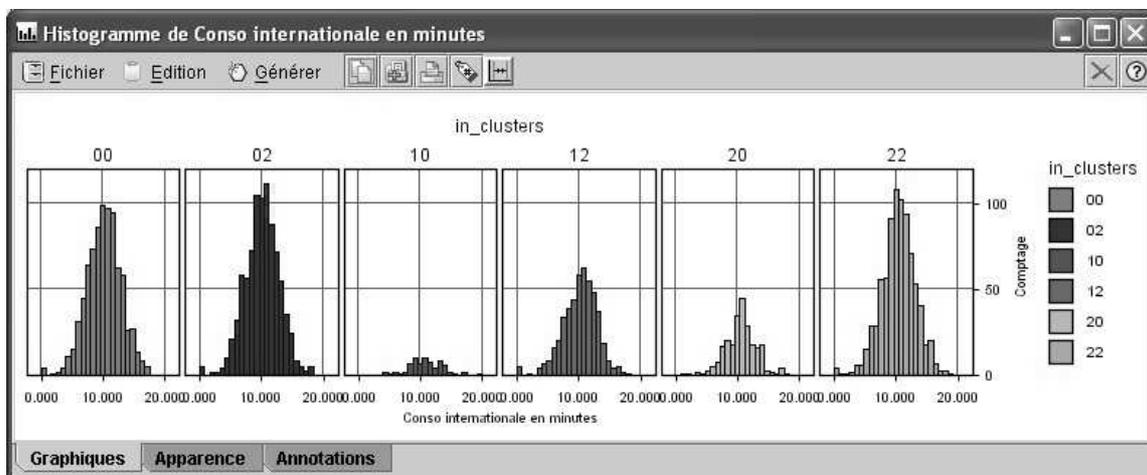
On constate aussi que la classe « inter » déborde un peu sur la classe toute option.

➤ Consommation totale



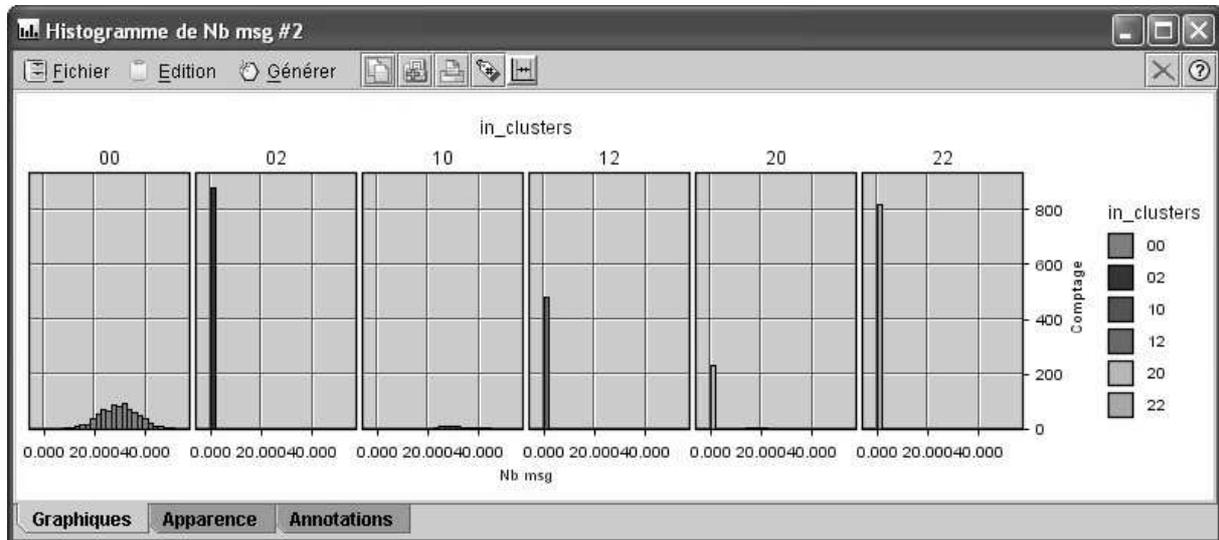
RAS.

➤ Consommation internationale



Ce graphique montre que le mode de consommation internationale est à peu près le même que les autres options. Ce qui veut dire que le produit option internationale est soit inintéressant, soit mal présenté.

➤ Nombre de messages



RAS.

➤ Répartition du churn dans les classes

Valeur	Proportion	%	Comptage
00		24,9	830
02		26,46	882
10		2,25	75
12		14,4	480
20		7,44	248
22		24,54	818

Churn?  
 False.  True.

- Le churn est sur-représenté en 10 et 20 (rose et vert pale) c'est-à-dire : dans l'option internationale et les toutes options.
- Le churn est sous représenté en 00 (bleu) c'est-à-dire : dans l'option mail.
- Les 3 autres classes correspondent aux sans options.

## Interprétation des classes par règles de décision

On peut faire une modélisation des règles de décision avec les classes de Kohonen comme attribut cible.

On obtient les résultats suivants :

Règles pour 00 - contient 1 règle(s)

**Règle 1 pour 00 (830; 1,0)**

si Mail in [ "yes" ]  
et International in [ "no" ]  
alors 00

Règles pour 02 - contient 2 règle(s)

**Règle 1 pour 02 (799; 0,974)**

si Mail in [ "no" ]  
et Durée de vie du compte <= 91.500  
et International in [ "no" ]  
et Durée de vie du compte <= 86.500  
alors 02

**Règle 2 pour 02 (66; 0,803)**

si Mail in [ "no" ]  
et Durée de vie du compte <= 91.500  
et International in [ "no" ]  
et Durée de vie du compte > 86.500  
et Nb dappels de jour <= 104  
alors 02

Règles pour 10 - contient 1 règle(s)

**Règle 1 pour 10 (92; 0,815)**

si Mail in [ "yes" ]  
et International in [ "yes" ]  
alors 10

Règles pour 12 - contient 2 règle(s)

**Règle 1 pour 12 (50; 0,76)**

si Mail in [ "no" ]  
et Durée de vie du compte <= 91.500  
et International in [ "no" ]  
et Durée de vie du compte > 86.500  
et Nb dappels de jour > 104  
alors 12

**Règle 2 pour 12 (472; 0,761)**

si Mail in [ "no" ]  
et Durée de vie du compte > 91.500  
et Durée de vie du compte <= 113.500  
et International in [ "no" ]  
alors 12

Règles pour 20 - contient 3 règle(s)

**Règle 1 pour 20 (83; 1,0)**

si Mail in [ "no" ]  
et Durée de vie du compte <= 91.500  
et International in [ "yes" ]  
alors 20

**Règle 2 pour 20 (55; 1,0)**

si Mail in [ "no" ]  
et Durée de vie du compte > 91.500  
et Durée de vie du compte <= 113.500  
et International in [ "yes" ]  
alors 20

**Règle 3 pour 20 (93; 1,0)**

si Mail in [ "no" ]  
 et Durée de vie du compte > 91.500  
 et Durée de vie du compte > 113.500  
 et International in [ "yes" ]  
 alors 20

Règles pour 22 - contient 1 règle(s)

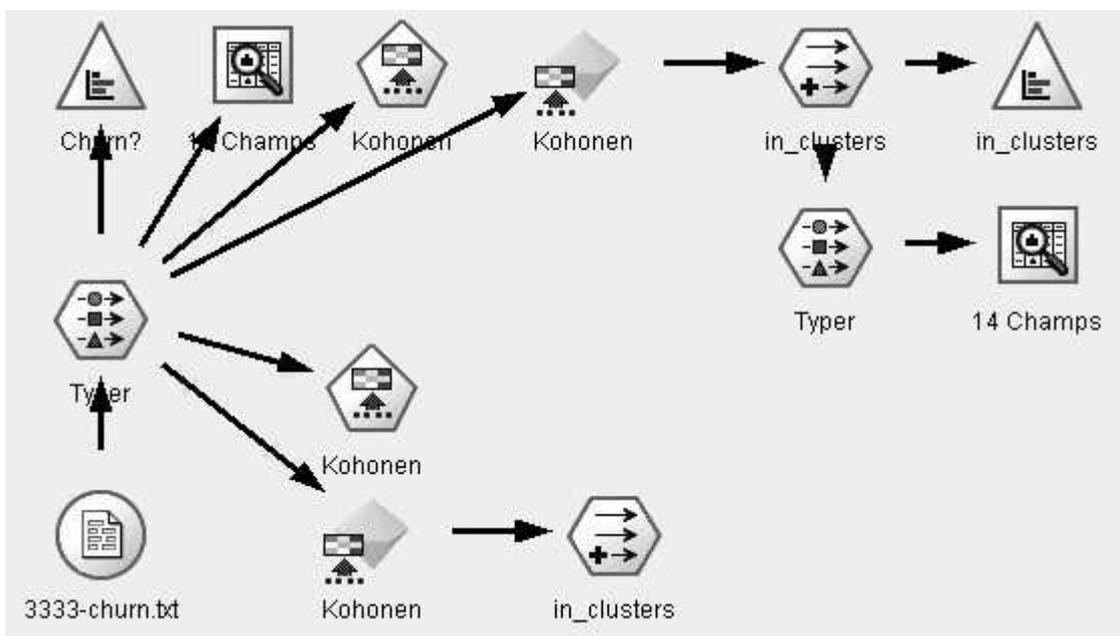
Règle 1 pour 22 (793; 0,938)

si Mail in [ "no" ]  
 et Durée de vie du compte > 91.500  
 et Durée de vie du compte > 113.500  
 et International in [ "no" ]  
 alors 22

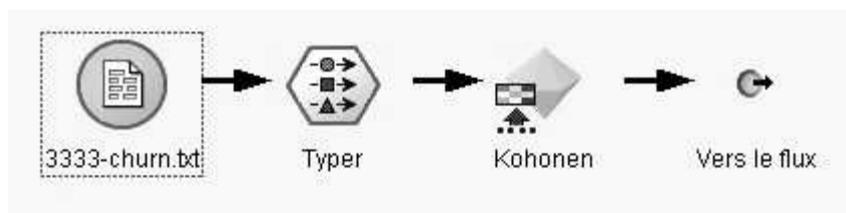
Par défaut : 02

On retrouve ce qu'on a mis au jour dans l'analyse exploratoire. A noter que si la production des règles est rapide, leur interprétation n'est pas facile à effectuer.

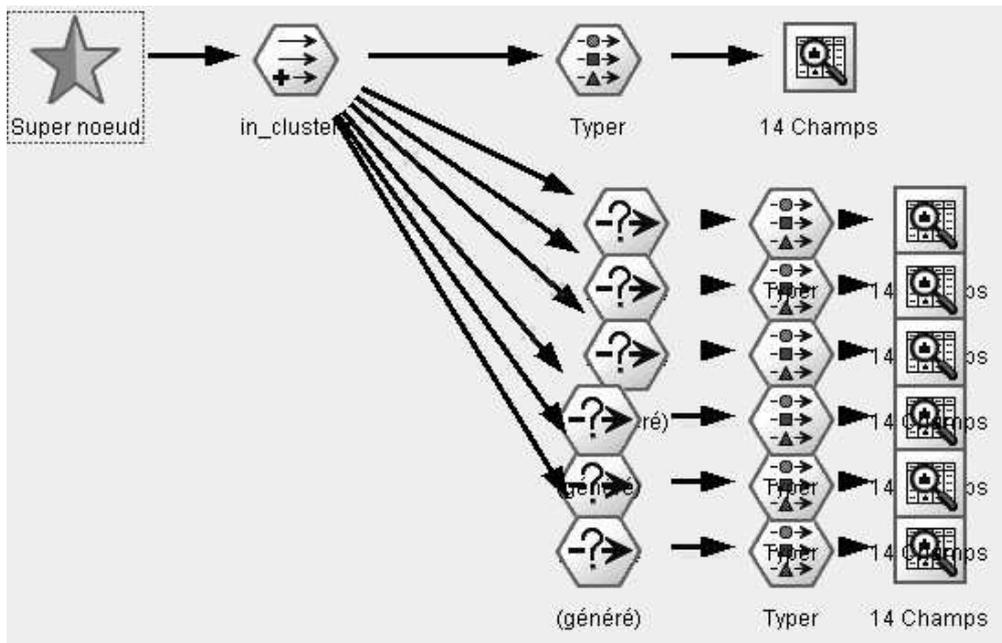
### Flux clémentine



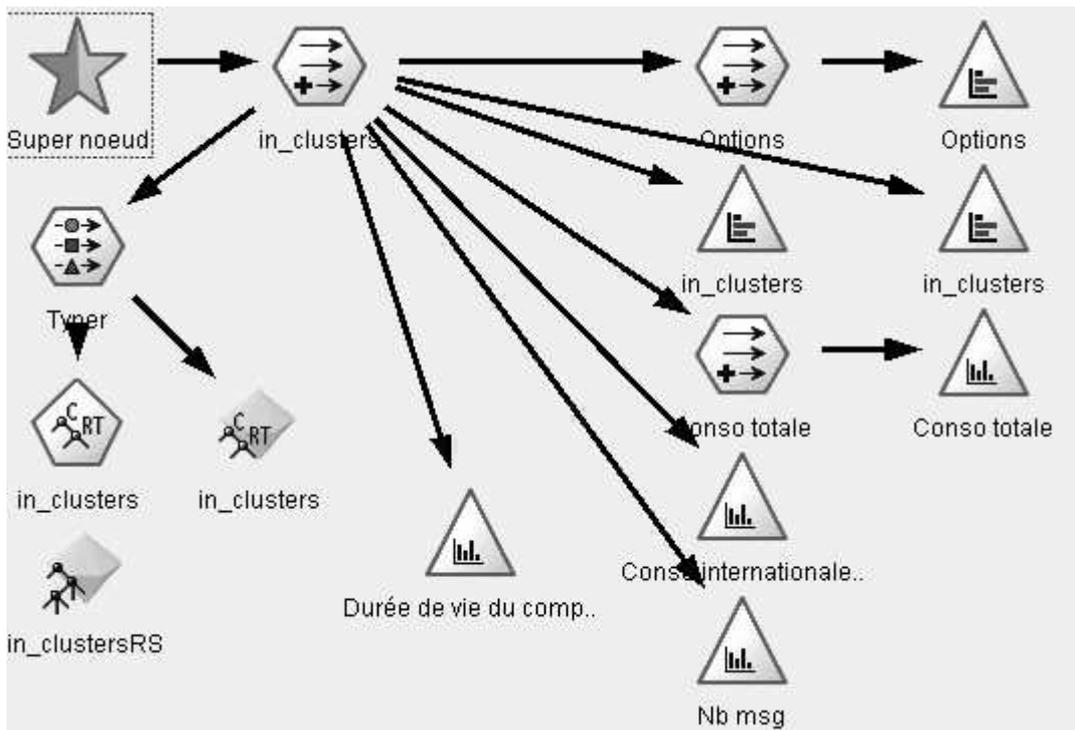
**Flux Clémentine : Réseaux de Kohonen.**



**Flux Clémentine : Super-nœud**



**Flux Clémentine : Analyse exploratoire**



**Flux Clémentine : Analyse exploratoire et arbre de décision**