Pôle Validation Département Pilotage Consolidé et Modèles
Direction Risques Groupe

Prévision du défaut sur le périmètre de TPE

XU Kai

6 novembre 2017

Table des matières

Introduction
Contexte

TPE : un sous-ensemble dont la chiffre d'affaires compris entre 1,5 et 5M€ ou montant total des engagements accordés supérieur à 1 M€.

Le groupe BPCE a accumulé des données volumineuses de TPE à la fois comportementales et financières en effectuant une fusion des données des réseaux Banques populaires, Caisse d'épargne et Natixis.

Ainsi, les données de TPE du groupe possèdent les caractéristiques de quantité volumineuse et de grande dimension, ce qui nous permet d'en faire une étude du point de vue de l'apprentissage statistique.

Introduction
Sujet

- *Objectif 1 : prévoir des clients défaillants de TPE*
  Des clients défaillants n'occupent que une petite proportion et leurs caractéristiques communs sont difficile à capter au sens général. Pour améliorer l'allocation des crédits, une meilleure identification des risques est nécessaire.
- *Objectif 2 : construire un cadre efficace pour la prévision*
  Basé sur une classe de réseau neuronal :carte Kohonen, nous essayons et modifions des modèles existants à notre besoins. Les modifications seront proposées à partir de nos travaux en vue de la prévision performante et robuste.

Sujet : Construire un cadre prédictif des défauts de TPE à partir de la carte Kohonen et proposer un nouveau modèle raffiné profitant des avantages des méthodes différentes.

Introduction
Variables disponibles

La base totale est un échantillon occupant une partie de TPEs dans 4 cohortes de Juin 2012 à Décembre 2013 à pas semestriel. Elle compte 18902 contrats :

- Variable d'intérêt : indicatrice de défaut. Au total, environ 3% de clients tombent en défaut chaque année.
- Variables explicatives : 202 variables parmi lesquelles 190 variables quantitatives et 12 variables qualitatives de 4 catégories :
  - donnée de compte
  - données externes
  - ratios financiers
  - signalétique : effectif de l'entreprise, forme juridique etc
- Les variables qualitatives seront transformées en nouvelles variables binaires, ce qui produisent environ 56 nouvelle variables.

Au final, 246 variables sont utilisée dans la prévision.

Données quantitatives :

Pour éviter les problèmes d'échelle et l'influence d'une hétérogénéité des variances, les variables quantitatives sont centrées et réduites.

La transformation du z-score :

$$\frac{V - \mu}{\sigma}$$

Soient μ l'espérance et σ l'écart-type des valeurs d'une variable explicative V.

Données qualitatives :

Une variable qualitative à *K* modalités est remplacée par *K* variables binaires, et chacune correspondant à une des modalités.

Exemple : un tableau regroupant les informations des chiens

Tableau original :

Individu	Taille	Poids
bass	Petite	Léger
beau	Grande	Moyen
boxe	Moyenne	Moyen

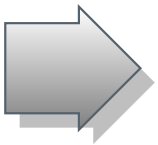


Tableau disjonctif complet :

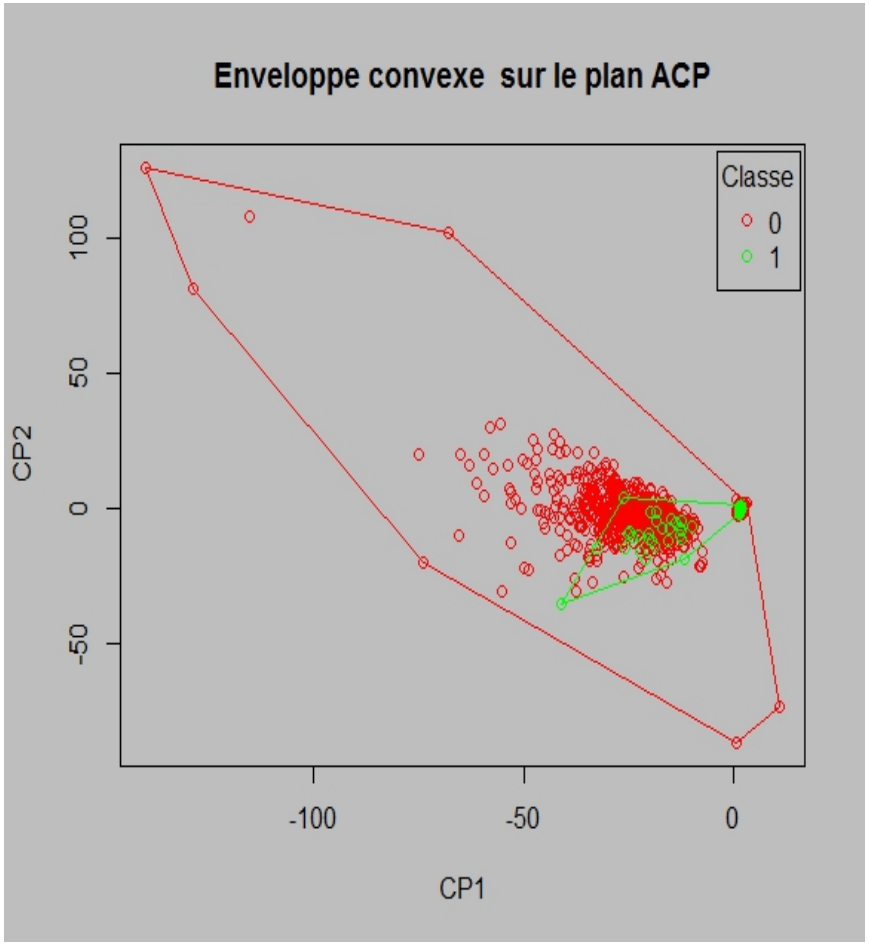
Individu	Taille P	Taille M	Taille G	Poids L	Poids M
bass	1	0	0	1	0
beau	0	0	1	0	1
boxe	0	1	0	0	1

Modélisation suivie

Pour réaliser ce regroupement prédictif, il nous faut examiner tout d'abord la structure topologique des observation :

- Existence d'un séparateur linéaire dans l'espace d'entrées : enveloppe convexe
- Examen de la proximité topologique des clients de même classe : t-SNE

Enveloppe convexe :



- Enveloppe convexe d'une classe est l'ensemble convexe le plus petit parmi ceux qui la contiennent
- Examiner la séparation linéaire
- Nous la visualiser sur le plan ACP
- Classe 0 : Clients sains
  Classe 1 : Clients défaillants

t-SNE (t-distributed stochastic neighbor embedding)

- Trouver une projection optimale des points $\mathbf{x}_1, \ldots, \mathbf{x}_N$ à grande dimension dans un espace de 2 dimensions
- La fonction à optimiser :

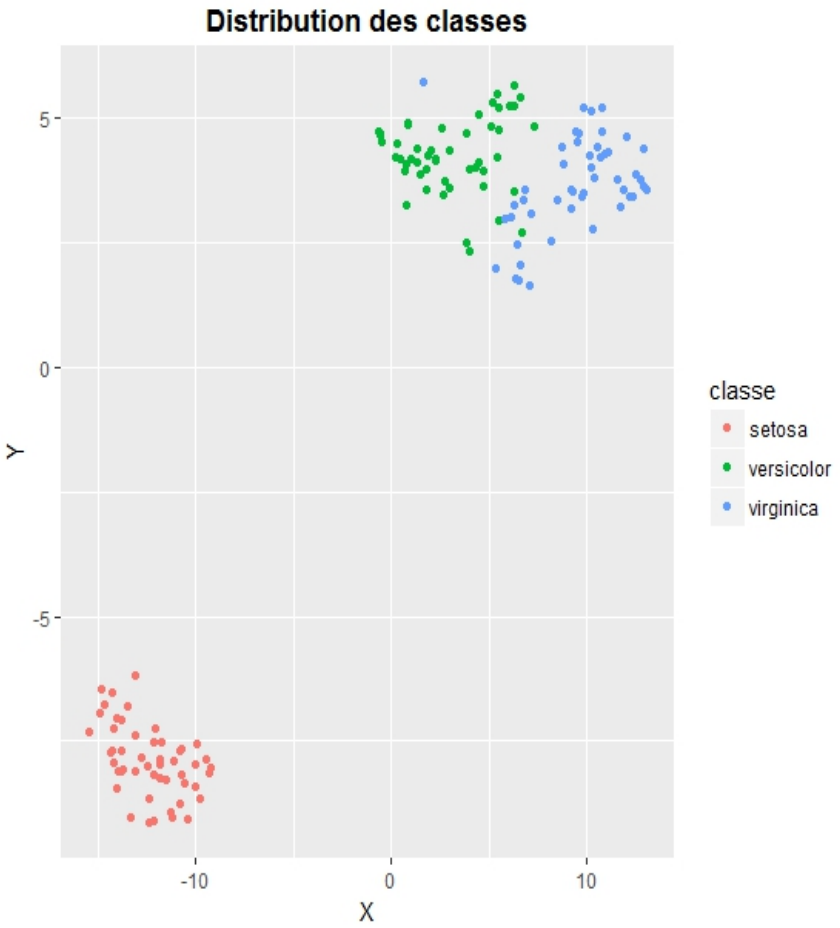
$$\inf_{y_1,\ldots,y_N} KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Les points originaux $\mathbf{x}_1, \ldots, \mathbf{x}_N$ et leurs projections $\mathbf{y}_1, \ldots, \mathbf{y}_N$:

- $p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$
- $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$
- $q_{j|i} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$
- $q_{ij} = \frac{q_{j|i} + q_{i|j}}{2N}$

Modélisation suivie
Structure topologique

Base de Iris :



Base de TPE :

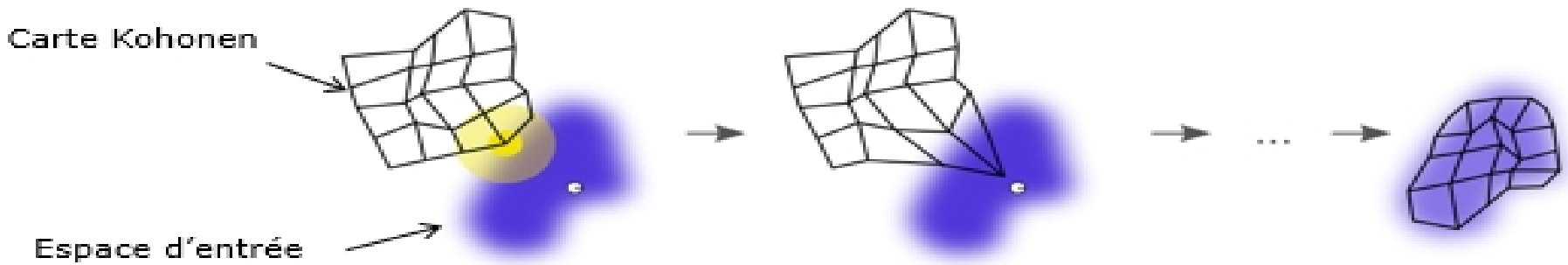
Apprentissage semi-supervisé
Carte Kohonen

L'analyse précédente se concentre au niveau individuel des observations.

Carte Kohonen : une classe de réseau de neurones artificiels qui étudie la répartition de données dans un espace à grande dimension. Elle met l'accent sur la préservation des relations topologiques des sous-groupes.

Nous utilisons la carte Kohonen pour rechercher le regroupement le plus agréable à faire la classification.

Apprentissage de la carte Kohonen :



L'adaptation de la carte est faite par une compétition entre les neurones en utilisant des vecteurs poids $\vec{w}$ appartenant aux neurones.

Initialisation de la carte Kohonen

- Pour simplicité, effectuer l'initialisation aléatoire des vecteurs poids $\vec{w}$ selon la loi uniforme sur l'intervalle [0,1] de haute dimension

Compétiontion des neurones

- A instant t, une observation $x(t+1) \subset R^d$ est choisie aléatoirement et présentée au réseau
- Le neurone gagnant est donc déterminé par :

$$i_0 = \arg\inf_i \|x(t+1) - w_i(t)\|^2$$

Évolution des neurones

- Les vecteurs poids du neurone gagnant $w_{i_0}^t$ et ses voisins sont mis à jour par :

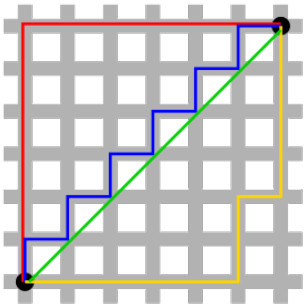
$$\begin{cases} w_i^{t+1} = w_i^t + \epsilon(t)(x(t+1) - w_i^t) & \forall i \in V_{r_t}(i_0) \\ w_i^{t+1} = w_i^t & \forall i \notin V_{r_t}(i_0) \end{cases}$$

- $V_{r_t}(i_0)$ est le voisinage de rayon $r(t)$ autour du neurone gagnant i_0 mesuré par la distance de Manhattan $d(\cdot, \cdot)$

On explique ici les notions concernant l'apprentissage de la carte Kohonen :

- Distance de Manhattan : Entre deux points A et B, de coordonnées respectives (X_A, Y_A) et (X_B, Y_B).

$$d(A, B) = |X_B - X_A| + |Y_B - Y_A|$$



Distance de Manhattan : chemins rouge, jaune et bleu ; ils sont équivalents au sens de la distance de Manhattan
Distance euclidienne : chemin vert

- Taux d'apprentissage $\epsilon(t)$: la vitesse à laquelle les vecteurs poids $\vec{w}$ sont ajustées
- Rayon $r(t)$: la distance de Manhattan de neurone gagnant à ses voisins les plus éloignés.

Les *hyperparamètres* qui conditionnent la carte Kohonen sont listées ci-dessous :

- Taille de la carte
- Taux d'apprentissage $\epsilon(t)$
- Rayon $r(t)$

Ils sont estimés par *validation croisée*, à la présence du manque de puissance de calcul, une validation croisée "hold one out" sera effectuée.

Évolution des hyperparamètres

Pour obtenir une convergence raisonnable de la carte, un décroissement exponentiel en fonction du temps est imposé sur le taux d'apprentissage $\epsilon(t)$ et le rayon $r(t)$:

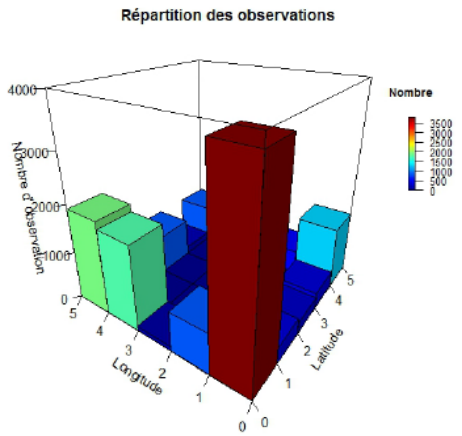
$$r(t) = r_0\left(\frac{r_T}{r_0}\right)^{\frac{t}{T}} \qquad \epsilon(t) = \epsilon_0\left(\frac{\epsilon_T}{\epsilon_0}\right)^{\frac{t}{T}}$$

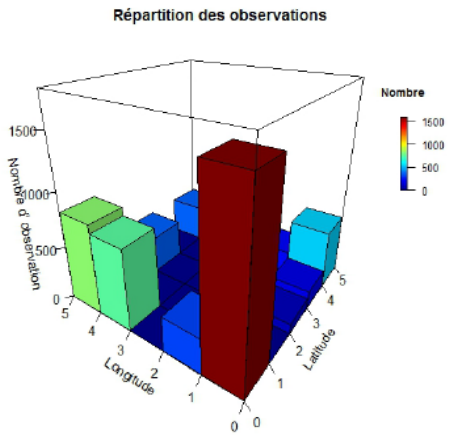
Estimation des hyperparamètres

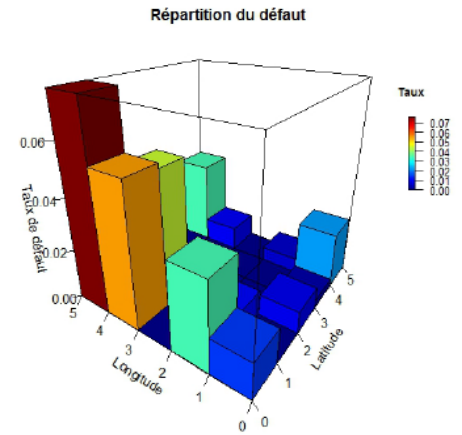
- Taille de la carte : 5×5 neurones
- Taux d'apprentissage $\epsilon(t)$: $\epsilon(0) = 0.04, \epsilon(T) = 0.01$; assez faible par rapport à sa limite supérieure 1 afin de éviter la convergence prématurée, ce qui peut conduire l'adaptation à stagner dans un optimum local.
- Rayon $r(t)$: $r(0) = 2.99, r(T) = 0.65$; de cette manière, l'interaction des neurones voisins autour de du neurone vainqueur s'affaiblit au fur et à mesure de l'apprentissage. En ce cas, la différence qui distingue des neurones voisins se renforce progressivement.

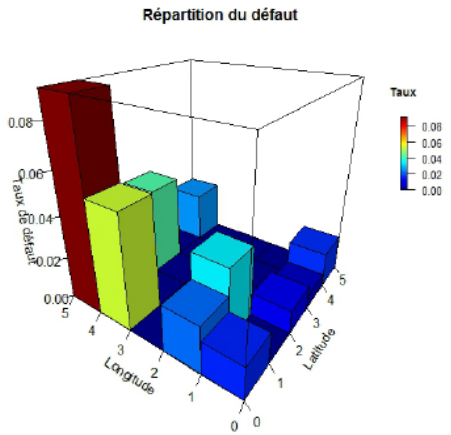
Modélisation suivie
Carte Kohonen

Carte Kohonen :





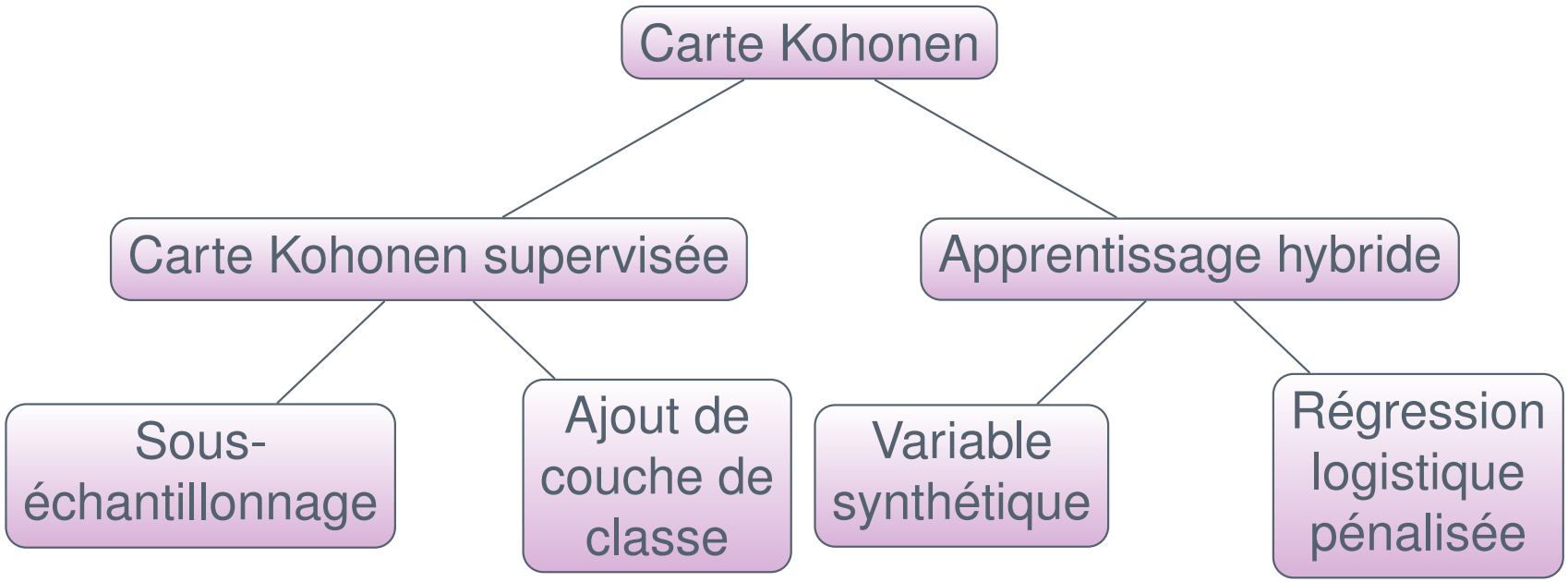




- Le neurone avec des observations le plus nombreuses et ses voisins possèdent les taux de défaut plus bas
- On ne réussit pas à identifier de neurone de défaut

Deux Modifications

Puisque un regroupement discriminant n'est pas produit, deux modifications sont prises en compte.

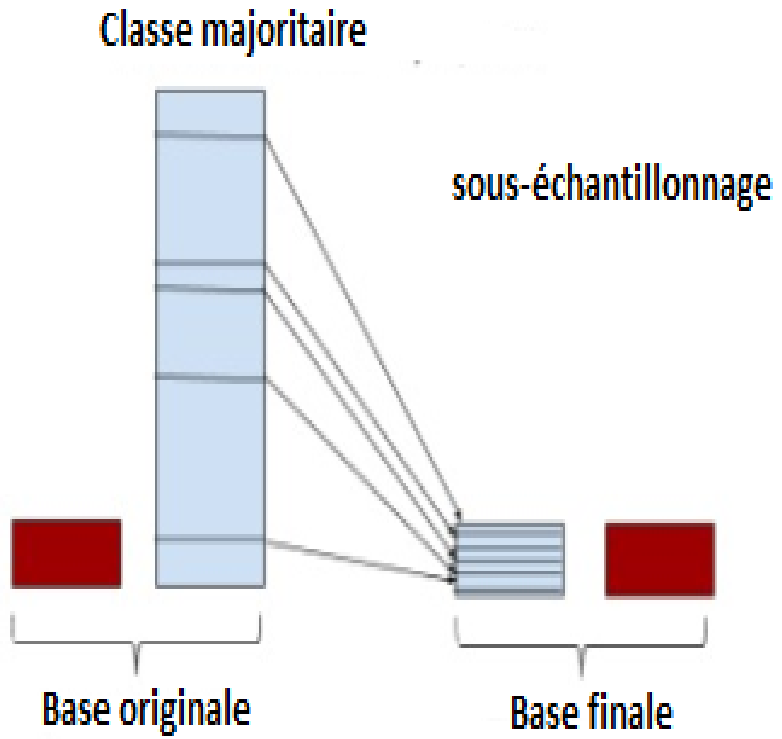


Modification I : Carte Kohonen supervisée

Idée : prendre en compte l'information dont nous disposons sur le défaut en la fusionnant dans la formation de la carte. De plus, une pondération entre les entrée et l'information de défaut est possible.

Modification I : Carte Kohonen supervisée
Sous-échantillonnage

Sous-échantillonnage :

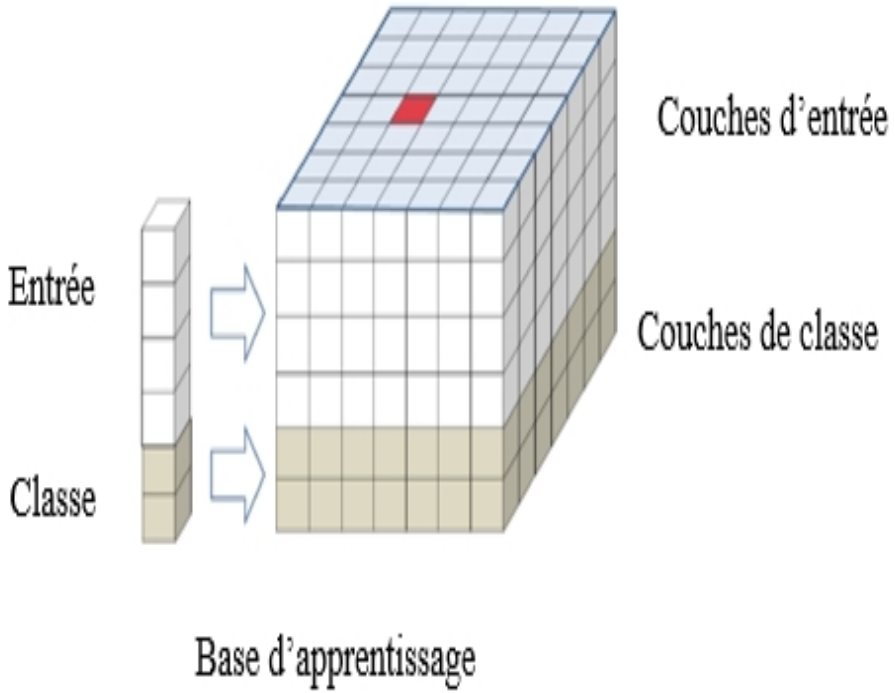


- Contre le déséquilibre des classes
- Diminuer des observations de classe majoritaire
- Réalisé par tirage aléatoire selon une loi uniforme

Modification I : Carte Kohonen supervisée

Carte Kohonen supervisée

Carte Kohonen supervisée :



- Profiter à la fois des entrées et des classes à l'étape d'apprentissage
- Possible de faire une pondération entre les couches de deux types
- La classification ne dépend que de l'entrée à l'étape de prévision

Modification I : Carte Kohonen supervisée
Choix de hyperparamètres

Sauf les hyperparamètres précédents, deux nouveaux hyperparamètres sont incorporés dans le modèle :

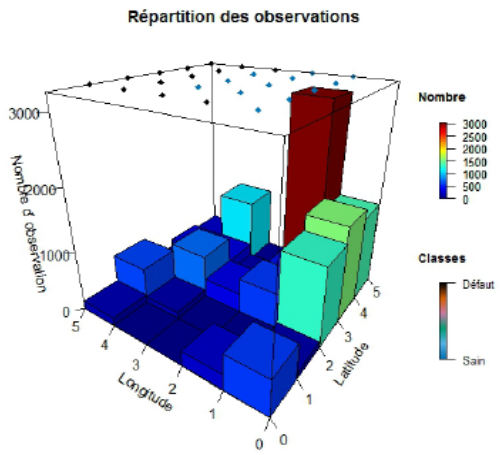
Nouveaux hyperparamètres

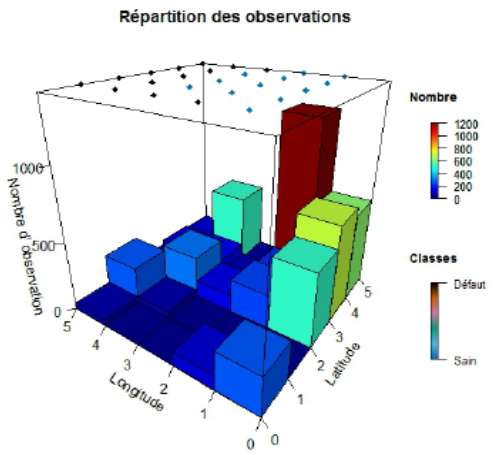
- la proportion de sous-échantillonnage p : la proportion entre nombre des clients sains sélectionnés et nombre des clients défaillants.
- le poids des couches wc : (wc_1, wc_2, wc_3) pour trois sous-couches ; wc_1 couche de d'entrée quantitative, wc_2 couche d'entrée qualitative et wc_3 couche de sortie.

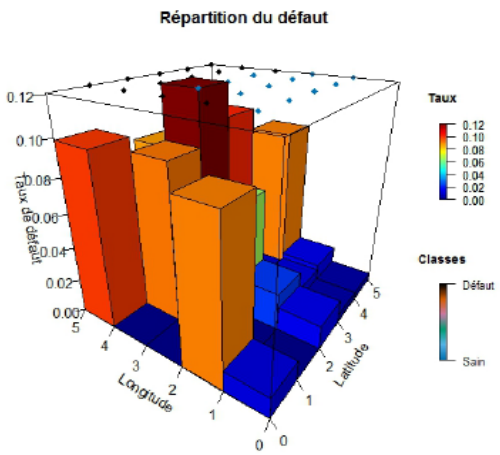
Estimation des hyperparamètres

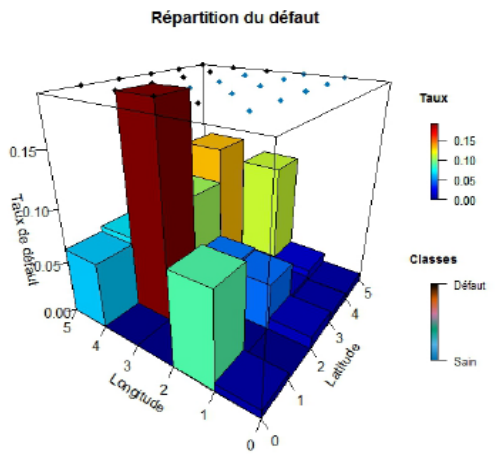
- Taille de la carte : 5×5 neurones
- Taux d'apprentissage $\epsilon(t)$: $\epsilon(0) = 0.04, \epsilon(T) = 0.01$
- Rayon $r(t)$: $r(0) = 2.99, r(T) = 0.65$
- la proportion de sous-échantillonnage p : 1.1
- le poids des couches wc : (3/8,1/8,1/2)

Carte Kohonen supervisée :









- La carte supervisée peut identifier les neurones dominés par les classes différentes dans la base sous-échantillonnée
- En remettant les observations exclues de l'échantillon, la performance reste meilleure que celle de la carte non supervisée selon le lien entre les groupes et le défaut
- Cependant, l'apprentissage perd l'information d'entrée au cours du sous-échantillonnage

Modification I : Carte Kohonen supervisée

Matrice de confusion

Matrice de confusion :

	Classe réelle 0	Classe réelle 1
Classe estimée 1	FP (faux positifs)	VP (vrais positifs)
Classe estimée 0	VN (vrais négatifs)	FN (faux négatifs)

- Dans la base de validation, sur les 5480 clients sains, 3887 seront estimés comme sains, soit 71% des clients sains sont correctement prédits
- Sur les 189 clients défaillants, 133 seront estimés comme défaillants, soit 70.4% des clients défaillants sont correctement prédits.

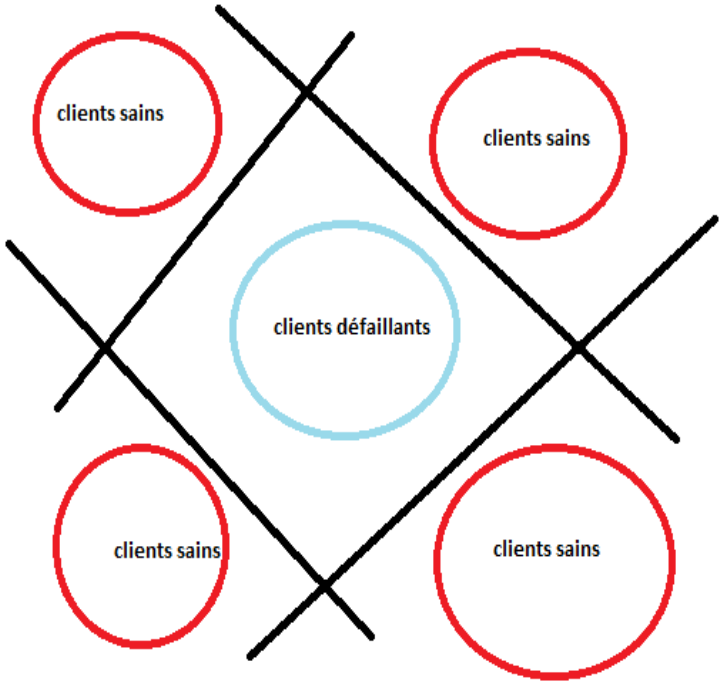
Modification II : Apprentissage hybride

Idée : prédire le défaut à la manière de l'apprentissage supervisé en estimant le lien entre des variables synthétiques x et des classes y au lieu de celui entre des entrées x et des classes y.

Grâce à la carte Kohonen, on est capable de raffiner les caractéristiques topologiques de l'espace d'entrée en utilisant les classifieurs locaux. De cette façon, l'apprentissage non-supervisé et l'apprentissage supervisé sont combinés pour atteindre meilleur performance.

Modification II : Apprentissage hybride
Hétérogénéité

Hétérogénéité :



- Les propriétés d'une partie des données sont différents que les autres.
- Fréquent dans la population dont les classes sont déséquilibrées

Regroupement des clients sains

- Entraîner une carte Kohonen classique sur tous les clients sains de la base d'apprentissage.
- Regrouper des clients sains selon les neurones pour former des sous-populations similaires à la structure topologique

Classifieur local

- Remettre des clients défaillant dans toutes les sous-populations générées dans la procédure précédente
- Construire des classifieurs locaux sur les sous-populations
- Le classifieur local choisi est un SVM (Machine à vecteur de support) muni de noyau de RBF : $h(x) = w^T \phi(x) + w_0$

  où $< \varphi(x), \varphi(x') >= K(x, x') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$

Modification II : Apprentissage hybride
Variable synthétique

Distance signée

Étant donné un hyperplan de séparation S, la fonction de la distance signée $f(x)$ est définie par les ensembles A et B qui sont caractérisés selon S

- $$f(x) = \begin{cases} d(x,S) & \text{si } x \in A \\ 0 & \text{si } x \in S \\ -d(x,S) & \text{si } x \in B \end{cases}$$
- Calculer les distances signées en utilisant des classifieurs locaux sur la base d' apprentissage
- Les distances signées servirent de variables synthétiques propres au modèle

Prévision de défaut

Effectuer une régression logistique pénalisée sur les variables synthétiques pour produire la probabilité d'être défaillant.

Régression logistique

Le but est de minimiser le risque moyen

$$\inf_{\beta_0,\beta} E[l(y_i, \beta_0 + \beta^T x_i)]$$

où $l(y, \beta_0 + \beta^T x) = -y\beta^T x + ln(1 + e^{y\beta^T x})$

Pénalisation de Lasso

- Selon le principe de rasoir d'Occam, un modèle parcimonieux est souvent mieux à interpréter.
- Par rapport à la norme l_0, la norme l_1 est plus facile à manipuler.
- Même si la norme l_1 n'est pas différentiable en 0, il existe beaucoup de algorithmes qui peuvent la traiter.

Régression logistique pénalisée

Une terme de pénalisation de type Lasso est ajoutée dans l'espérance de la perte logistique :

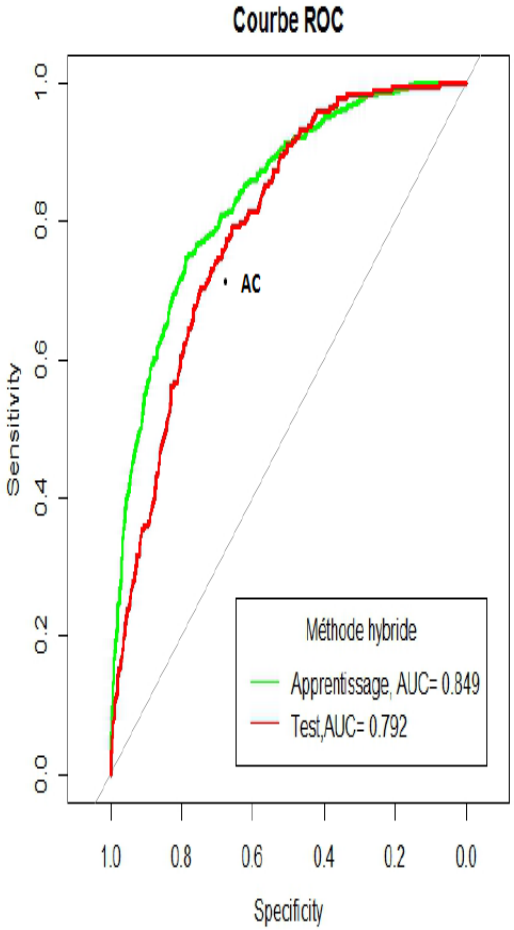
$$\inf_{\beta_0,\beta} \quad E[l(y_i, \beta_0 + \beta^T x_i)] + \lambda||\beta||_1$$

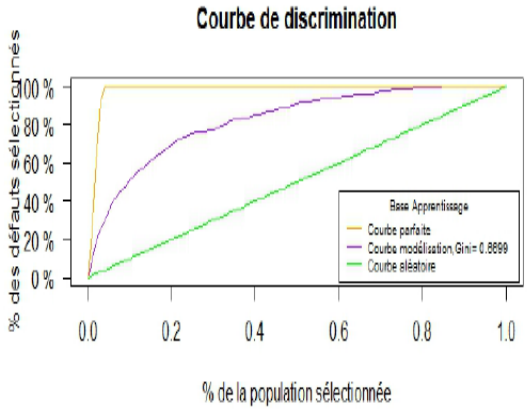
Pour décider le λ qui conditionne la régression, le critère d'information d'Akaike AIC sera utilisé. On choisira le modèle muni de le minimum de AIC_{min} comme le modèle de prévision.

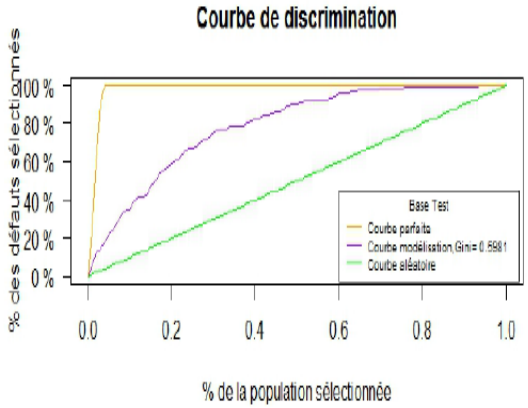
Estimation des hyperparamètres

- Taille de la carte : 9×9 neurones
- Taux d'apprentissage $\epsilon(t)$: $\epsilon(0) = 0.04, \epsilon(T) = 0.01$
- Rayon $r(t)$: $r(0) = 2.99, r(T) = 0.65$

Modification II : Apprentissage hybride
Résultat







- CA : ancien classifieur de la première modification, sa performance est strictement inférieur que celle de l'apprentissage hybride
- Erreur de généralisation est bien contrôlée selon la différence de la performance sur la base d'apprentissage et la base de validation.
- Les variables synthétiques en tant que prédicateurs sont propre au modèle au lieu de l'espace d'entrée

Modification II : Apprentissage hybride
Décomposition

Partie d'apprentissage non-supervisé

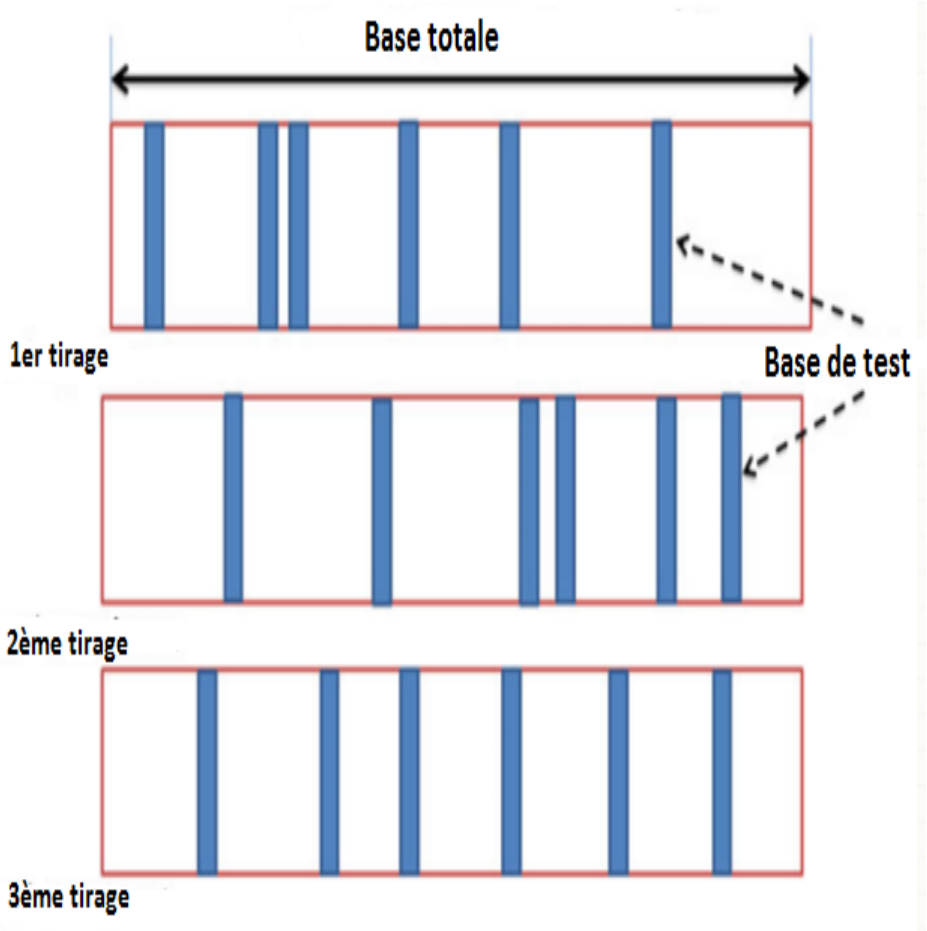
- Regroupement des clients défaillants en utilisant la carte Kohonen

Partie d'apprentissage supervisé

- Les classifieurs locaux s'interviennent dans la génération des variables synthétiques x
- Des classes y prédites sont obtenues par le biais d'estimation du lien entre des variables synthétiques x et des classes y

Validation croisée
Évaluation de la robustesse

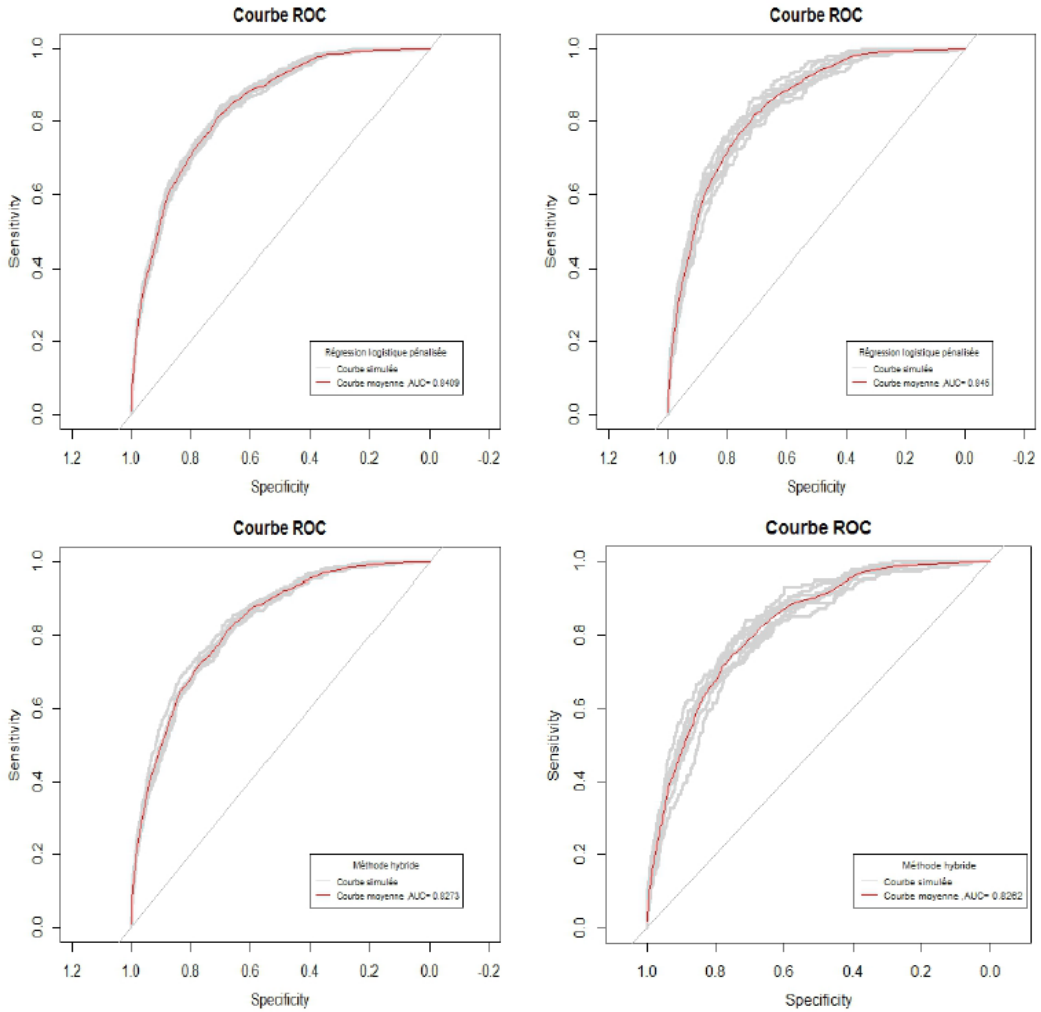
Validation croisée Monte Carlo stratifiée :



- Tirage aléatoire sans remise pour la construction de la base de test
- Sélectionner un échantillon au sein de chaque classe pour éviter le déséquilibre
- Pas de souci de choisir une partition raisonnable comme validation croisée k-fold

Validation croisée
Résultat

Performance des modèles :



- La comparaison est entre la régression pénalisée et la méthode hybride sur les variables sélectionnées
- La méthode hybride dont les variables explicatives propres au modèle est aussi performant que le modèle d'apprentissage supervisé dépendant des entrées originaux
- Le fléau de dimension est possible à éviter en utilisant les variables synthétiques

Perspective

- Selon Vapnik, la limite supérieure de l'erreur de test sera diminuée sous la taille de la base d'apprentissage N plus grande :

$$\Pr\left(\text{Erreur de test} \leqslant \text{Erreur d'apprentissage} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$

- La performance des classifieurs qui ont l'air de surapprentissage est possible d'être améliorée avec la base de donnée plus grande
- Une pondération des entrées peut être implémentée en ajustant la fonction de perte selon les cohortes. En ce cas, nous prendrons compte de l'effet du temps

Merci pour votre attention !