Une méthode paramétrique et robuste de classification semi-supervisée avec rejet

Christophe Saint-Jean, Carl Frélicot

Une méthode paramétrique et robuste de classification semi-supervisée avec rejet

Christophe Saint-Jean* Carl Frélicot

L3I - Université de La Rochelle - 17042 La Rochelle Cedex 1
{christophe.saint-jean,carl.frelicot}@univ-lr.fr

Résumé

En classification et plus généralement en apprentissage, on distingue classiquement deux approches: le cas supervisé et le cas non supervisé. On peut combiner celles-ci dans un apprentissage semi-supervisé. Dans ce cadre, on enrichit un ensemble de données non étiquetées par un certain nombre d'exemples étiquetés. Ces derniers, en proportion généralement faible, servent à guider les algorithmes de classification via leur paramétrage et leurs conditions d'initialisation.
Dans cet article, nous rappelons quelques techniques actuelles de semi-supervision et références du domaine. Nous proposons une extension de l'algorithme de classification par partition que nous avons proposé dans (Saint-Jean *et al.*, 2000) pour prendre en compte la semi-supervision. Nous proposons également une heuristique pour le choix du paramètre du M-estimateur de Huber tenant compte de la supervision partielle. Enfin, nous présentons les résultats de notre approche sur des jeux de données artificiels et standards du domaine.

Mots-clés : Classification, Semi-supervision, Robustesse, Rejet, EM

1 INTRODUCTION

En classification non supervisée, on cherche à mettre en évidence des relations entre des objets en les regroupant en entités homogènes suivant une certaine mesure de similarité. Cette fonction est réalisable via la recherche d'extremum d'une fonctionnelle adéquate (Ex: Fuzzy C-Means (Bezdek, 1987) ou Expectation-Maximization (Dempster *et al.*, 1977)). Cependant, la complexité de l'espace à observer impose généralement l'utilisation de méthodes sous-optimales, sujettes aux extrema locaux. Dans ce cadre, le choix des paramètres et surtout des conditions initiales s'avère crucial pour le succès de la méthode. Nous allons illustrer ce problème par deux exemples pour montrer l'intérêt de la semi-supervision en classification.

⋆ Ce travail est financé par le Conseil Général de la Charente-Maritime

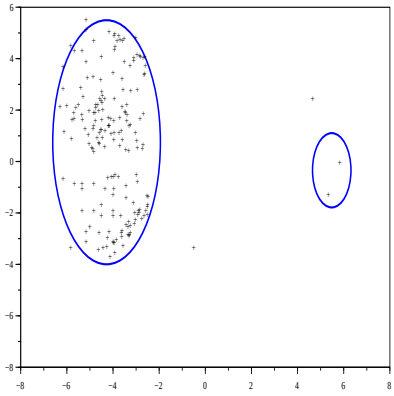

FIG. 1 – *Sensibilité aux outliers (2 classes demandées)*

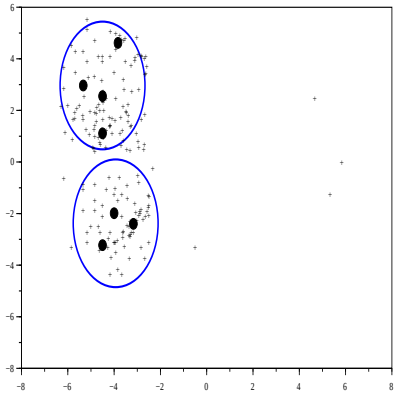

FIG. 2 – *Correction par la supervision*

Considérons le cas de deux classes séparées (Fig. 1) et quelques points isolés (*outliers*) qui peuvent être aberrants. La classification obtenue est le résultat de l'algorithme EM. L'estimation des paramètres des classes est pertubée par les *outliers* et l'algorithme privilégie des classes bien séparées plutôt des classes proches (même denses). Si l'on ajoute quelques points supervisés (● dans Fig. 2), on contraint suffisament le processus d'optimisation pour éviter un extremum local très éloigné de l'optimum global pour autant que les points supervisés soient pertinents.
La semi-supervision trouve également son intérêt dans des problèmes de type OU-Exclusif. Le jeu de données de la figure 3 comporte quatre classes où deux d'entre elles ont la même moyenne théorique. Une mauvaise initialisation peut être corrigée en supervisant quelques points adéquats (cf. Fig 4).

Bien évidemment, superviser même partiellement permet d'améliorer la précision de l'estimateur. Il en résulte une meilleure capacité en généralisation du modèle. De même, il a été montré (Hsieh, 1998) que la semi-supervision permet de limiter le phénomène de Hughes dû à la dimensionalité des données (Hughes, 1968).

Cet article est organisé comme suit. Nous évoquons tout d'abord le contexte de l'étude. Ensuite, nous rappelons l'algorithme de classification robuste avec rejet que nous avons proposé dans (Saint-Jean *et al.*, 2000). Puis, nous proposons une extension de notre approche à la semi-supervision et en présentons les résultats sur quelques jeux de données. Finalement, nous concluons sur les améliorations possibles et perspectives de notre travail.

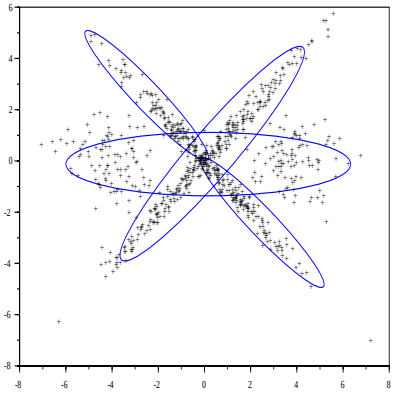

FIG. 3 – *Une mauvaise initialitisation sur un jeu de données de type XOR (4 classes demandées)*

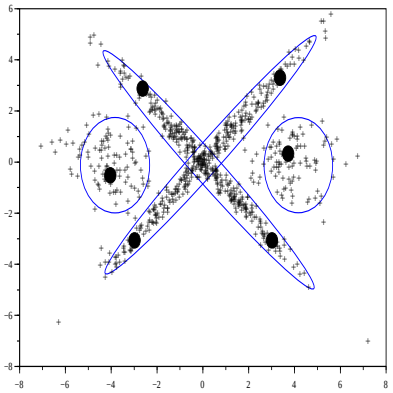

FIG. 4 – *Correction par la supervision*

2 CONTEXTE DE L'ÉTUDE

Dans ce papier, nous nous plaçons dans le contexte de la classification et du modèle de mélange sur les données pour la reconnaissance des formes. En classification, on dispose généralement d'un grand ensemble de données non étiquetées alors que les données étiquetées sont plus rares. Il existe de nombreuses façons de faire coopérer ces deux types de données suivant le cadre dans lequel on se place. Selon Labzour et al. (Labzour *et al.*, 1998), W. Pedrycz (Pedrycz, 1985) a introduit la notion de supervision partielle en modifiant l'algorithme des Fuzzy C-Means(FCM) (Bezdek, 1987). Ces travaux ont été repris une dizaine d'années plus tard par A. Bensaid et al. (Bensaid *et al.*, 1996) (Amar *et al.*, 1997) qui ont étendu le paradigme à la classification hiérarchique.
Comme nous l'avons dit plus haut, la semi-supervision peut s'effectuer en contraignant les estimateurs utilisés dans les algorithmes de classification. Dans le cas des FCM, on remplacera la minimisation de

$$Q = \sum_{j=1}^{C} \sum_{i=1}^{N} u_{ij}^{p} d_{ij}^{2} \tag{1}$$

par la minimisation de

$$J = Q + \sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ij} - f_{ij} b_i)^{p} d_{ij}^{2} \tag{2}$$

où les u_{ij} et les d_{ij} représentent respectivement le degré d'appartenance et la distance de la donnée x_i à la classe C_j. Le facteur de flou p varie dans $]1, +\infty]$.

La matrice f correspond aux degrés d'appartenance des points supervisés désignés par le vecteur booléen b (b_i=1 si le i-ème point est supervisé, 0 sinon). Bien évidemment, on dérive de nouvelles équations de mises à jour des u_{ik} et des paramètres des classes (Pedrycz & Waletzky, 1997).

Dans le cadre probabiliste et pour un algorithme du type EM, la semi-supervision s'effectue naturellement en fixant les probabilités a posteriori des points supervisés de la manière suivante :

$$P(C_j|x_i) = \begin{cases} 1 & \text{si } x_i \text{ appartient à la classe } C_j \\ 0 & \text{sinon} \end{cases} \quad (3)$$

Cette approche s'apparente aux travaux de Tadjudin S. et Landgrebe D. (Tadjudin & Landgrebe, 1998) et ceux décrits dans (Nigam *et al.*, 2000). Nous nous plaçons également dans ce contexte (cf. section 4).
Récemment, plusieurs techniques de classification ont été étendues à la semi-supervision. Parmi elles, citons les machines à vecteurs de support (Bennett & Demiriz, 1998) (Fung & Mangasarian, 1999), les algorithmes génétiques (Demiriz *et al.*, 1999) et les algorithmes avec des prototypes-points (Labzour *et al.*, 1998). Avant de présenter notre approche de la semi-supervision, nous allons rappeler notre méthode de classification robuste avec rejet.

3 MÉTHODE DE CLASSIFICATION ROBUSTE AVEC REJET

Dans un article précédent (Saint-Jean *et al.*, 2000), nous avons proposé une méthode de classification robuste avec rejet. Celle-ci est une variante de l'algorithme Expectation-Maximization (EM) (Dempster *et al.*, 1977) dont nous rappelons ici le principe.

3.1 Rappels sur l'algorithme EM

L'algorithme EM est utilisé en classification pour estimer les paramètres Θ_j des classes formant le mélange $f(x;\Theta) = \sum_{j=1}^{C} \pi_j f(x;\Theta_j)$ à partir des données observées $\chi = \{x_1, x_2, \ldots, x_N\}$. Pour cela, on cherche à maximiser la vraisemblance $\mathcal{L}(\Theta) = P(\chi|\Theta)$. Sous l'hypothèse d'indépendance des x_i, la log-vraisemblance $\log \mathcal{L}(\Theta)$ s'écrit :

$$\log \mathcal{L}(\Theta) = \sum_{i=1}^{N} \log \sum_{j=1}^{C} \pi_j f(x_i;\Theta_j) \quad (4)$$

Cette maximisation étant difficile à effectuer, on introduit une variable aléatoire supplémentaire Z (appelée également donnée incomplète) correspondant à

la classe d'appartenance de chacun des éléments à classifier. On écrit pour l'élément x_i :

$$z_{ij} = \begin{cases} 1 & \text{si } x_i \text{ appartient à la classe } C_j \\ 0 & \text{sinon} \end{cases}$$

Généralement, les $z_i = (z_{i1}, \ldots, z_{iC})$ sont pris comme des réalisations indépendantes suivant une distribution multinomiale $Mult_k(1,\pi)$ où $\pi = (\pi_1, \ldots, \pi_C)^T$.

Sous ces conditions, la log-vraisemblance complète s'écrit :

$$\log \mathcal{L}_c(\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{C} z_{ij} \log(\pi_j f(x_i; \Theta_j)) \tag{5}$$

L'algorithme EM maximise $\log \mathcal{L}_c$ en alternant deux phases :

E-Step (Expectation Step) :
On calcule l'espérance conditionnelle de $\log \mathcal{L}_c$:

$$Q(\Theta|\Theta^{(t)}) = E[Log\mathcal{L}_c(\Theta)|\chi,\Theta^{(t)}] \tag{6}$$

Comme $Log\mathcal{L}_c$ est une fonction linéaire des z_{ij}, ce calcul se limite à remplacer les z_{ij} par leur espérance conditionnelle :

$$E[z_{ij}|\chi,\Theta^{(t)}] = \frac{\pi_j f(x_i; \Theta_j^{(k)})}{\sum_{l=1}^{C} \pi_l f(x_i; \Theta_l^{(k)})} \tag{7}$$

Les z_{ij} sont les probabilités a posteriori d'appartenance de l'élément x_i à la classe C_j.

M-Step (Maximization Step) :
Dans cette phase, on recherche la valeur de Θ qui maximise (6). Cette étape dépend complètement du modèle des classes choisi que nous allons détailler par la suite.
L'utilisation de l'algorithme EM nécessite la connaissance du nombre de classes du modèle. Ce paramètre peut être estimé par des critères de validation tel que l'"Integrated Completed Likelihood" (ICL) (Biernacki *et al.*, 2000).

3.2 La modélisation des classes

Dans le cas de classes gaussiennes, les Θ_j représentent la probabilité a priori π_j, la moyenne μ_j, matrice de covariance Σ_j de la classe C_j. Nous avons enrichi ce modèle en considérant une classe comme un mélange de deux gaussiennes (cf. Fig. 5):

$$P(\hat{x}|C_j) = \underbrace{(1-\gamma_j)\mathcal{N}(x; \hat{\mu_j^r}, \hat{\Sigma_j^r})}_{(A)} + \underbrace{\gamma_j \mathcal{N}(x; \hat{\mu_j}, \alpha_j \hat{\Sigma_j})}_{(B)} \tag{8}$$

Le terme (A) est utilisé pour modéliser le noyau de la classe alors que le terme (B) permet de prendre en compte le bruit autour de la classe. Nous verrons par la suite que (B) est utilisé pour effectuer un rejet en distance.
Les γ_j et α_j permettent de contrôler respectivement la proportion entre les deux composantes et l'étalement de la seconde composante en modulant sa variance. Ainsi, γ_j est choisi entre 0 et 0.5 pour favoriser (A) et α_j est supérieur à 1.
Le calcul des $\hat{\mu_j^r}$ et $\hat{\Sigma_j^r}$ (estimateurs de μ_j et Σ_j) dans (A) est effectué de manière robuste en utilisant un M-estimateur. Dans (Saint-Jean *et al.*, 2000) nous avons testé plusieurs estimateurs. Ici, nous avons retenu le M-estimateur de Huber (Huber, 1981). Celui-ci est paramétré par une constante *h* permettant d'ajuster la taille du filtrage (cf. Fig. 7). Nous verrons plus tard que l'approche semi-supervisée permet d'en fixer automatiquement la valeur. L'algorithme 1 décrit cette procédure en même temps que le calcul de $\hat{\mu_j}$ et de $\hat{\Sigma_j}$.

Algorithme 1: Estimation robuste et itérative de la moyenne et de la matrice de covariance

Données : $\chi = \{x_1, \ldots, x_N\}$, z_{ji} l'estimation courante de $P(C_j|x_i)$, *h*

$$\hat{\mu}_j^r = \hat{\mu}_j = \frac{\sum_{i=1}^{N} z_{ji}\ x_i}{\sum_{k=1}^{N} z_{ji}}$$

$$\hat{\Sigma}_j^r = \hat{\Sigma}_j = \frac{\sum_{i=1}^{N} z_{ji}\ (x_k - \hat{\mu}_j^r)(x_i - \hat{\mu}_j^r)^T}{\sum_{i=1}^{N} z_{ji}}$$

répéter

  pour *i = 1 à N* **faire**

    $e_i = d_{\mathcal{M}}(x_i, C_j)$

    $w_i = \frac{\psi_{huber}(e_i, h)}{e_i}$

  $$\hat{\mu}_j^r = \frac{\sum_{i=1}^{N} w_i\ z_{ji}\ x_i}{\sum_{i=1}^{N} w_i\ z_{ji}}$$

  $$\hat{\Sigma}_j^r = \frac{\sum_{i=1}^{N} w_i\ z_{ji}\ (x_i - \hat{\mu}_j^r)(x_i - \hat{\mu}_j^r)^T}{\sum_{i=1}^{N} w_i\ z_{ji}}$$

jusqu'à *Critère d'arrêt*;

Le critère d'arrêt de cette procédure permet de borner le nombre d'itérations dans l'estimation. En effet, à chaque itération, le nombre de points possédant un poids significatif diminue et ce que l'on gagne en robustesse est perdu en précision. Le test de convergence peut être une combinaison de plusieurs types :

1. $t < t_{max}$ où t désigne le nombre d'itérations effectuées,
2. convergence des paramètres estimés,
3. taux maximal d'élimination $\alpha < \alpha_{max}$ où α est le pourcentage des données ayant un poids quasi-nul.

Nous avons utilisé une combinaison de (1) et (3) avec $\alpha_{max} = 50\%$ et $t_{max} = 4$. Pour (B), on estimera $\hat{\mu_j}$ et $\hat{\Sigma_j}$ à l'aide des formules usuelles de calcul de moyenne

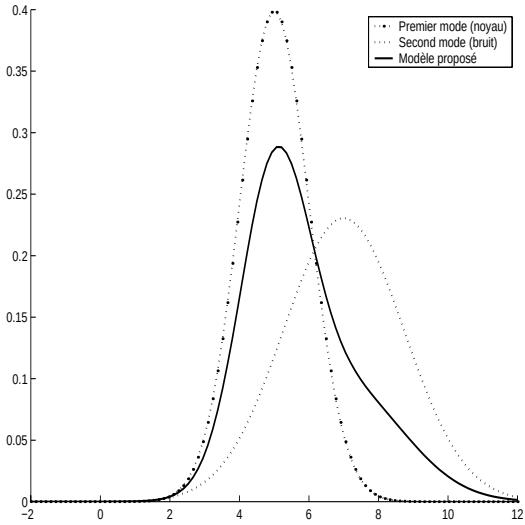

FIG. 5 – *Chaque classe est modélisée par un mélange de deux gaussiennes*

et de matrice de covariance. Au passage, on remarquera que l'on perd la propriété initiale de l'algorithme sur la croissance monotone de $Log\mathcal{L}$ du fait de l'utilisation de l'estimation robuste récursive. La phase de maximisation n'est plus strictement réalisée mais approximée. Ceci nous oblige à changer le critère d'arrêt de l'algorithme traditionnellement basé sur la stabilité $Log\mathcal{L}$. Dans notre cas, on s'arrêtera lorsqu'aucune amélioration de $Log\mathcal{L}$ n'a été constatée après un nombre minimal d'itérations (fixé à 10 itérations ici).

3.3 La règle d'affectation et le rejet

Le rejet est effectué lors de l'affectation finale des points aux classes. On crée une (C+1)-ème classe, appelée classe de rejet, dans laquelle on regroupe tous les seconds termes (B). Le mélange peut alors s'écrire sous la forme :

$$\hat{f}(x) \equiv f(x;\hat{\Theta}) = \sum_{j=1}^{C+1} \pi_j f(x;\hat{\Theta}_j) \qquad \text{avec} \tag{9}$$

$$f(x;\hat{\Theta}_j) = (1-\gamma_j)\,\mathcal{N}(x;\hat{\mu_j^r},\hat{\Sigma_j^r}) \qquad \forall j = 1,C \tag{10}$$

$$f(x;\hat{\Theta}_{C+1}) = \sum_{j=1}^{C} \gamma_j\,\mathcal{N}(x;\hat{\mu_j},\alpha_j\hat{\Sigma_j}) \tag{11}$$

L'affectation s'effectue ensuite suivant le critère du Maximum A Posteriori (MAP) parmi ces C+1 classes. La figure 6 montre deux exemples de cet algorithme.

Il est nécessaire de faire un compromis entre le taux de succès P_c, le taux d'erreur P_e et le taux de rejet P_r. En effet, on cherche à remplacer l'erreur par le rejet sans trop faire diminuer le taux de succès. Pour désigner le bon modèle Γ, on choisira celui qui maximise le critère:

$$C(\Gamma) = C_c * P_c(\Gamma) - C_e * P_e(\Gamma) - C_r * P_r(\Gamma) \tag{12}$$

où C_c,C_e,C_r représentent des coûts associés au succès, à l'erreur et au rejet.

4 EXTENSION À LA SEMI-SUPERVISION

Dans cette section, nous proposons une extension à la semi-supervision de la méthode de classification que nous venons de décrire. On utilise pour cela l'équation (3) qui fixe la probabilité a posteriori des points supervisés. Dans la suite, nous adopterons les notations suivantes:

$x_{i\bullet}$	i-ème point non supervisé de l'ensemble d'apprentissage
x_{ij}	i-ème point supervisé appartenant à la classe C_j
π_j	Probabilité a priori de la j-ième classe
z_{ij}	Estimation de la probabilité a posteriori $P(C_j\|x_i\bullet)$
N	Nombre total de points non supervisés
N_j	Nombre de points supervisés de la j-ième classe
$d_{\mathcal{M}}$	Distance de Mahalanobis

$$d^2_{\mathcal{M}}(x_i,C_j) = (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) \tag{13}$$

4.1 Mise à jour des paramètres des classes

Sous l'hypothèse d'un choix représentatif des points supervisés vis-à-vis de la population des classes, la mise à jour de la probabilité a priori de la classe C_j s'effectue suivant:

$$\pi_j = \frac{\sum_{i=1}^{N} z_{ij} + N_j}{N + N_j} \tag{14}$$

Le calcul des autres paramètres est décrit dans l'algorithme 2.

4.2 Paramétrage de l'estimateur

La supervision de quelques points permet d'automatiser le choix du seuil de Huber h_j. Plusieurs heuristiques sont possibles. Il est raisonnable de considérer comme pertinents les points plus proches du prototype de la classe que les points supervisés (que l'on ne remet pas en cause). Ce choix peut s'exprimer comme suit:

$$h_j = \max_{i \in [1,N_j]} (d_{\mathcal{M}}(x_{ij},C_j),h_{min}) \tag{15}$$

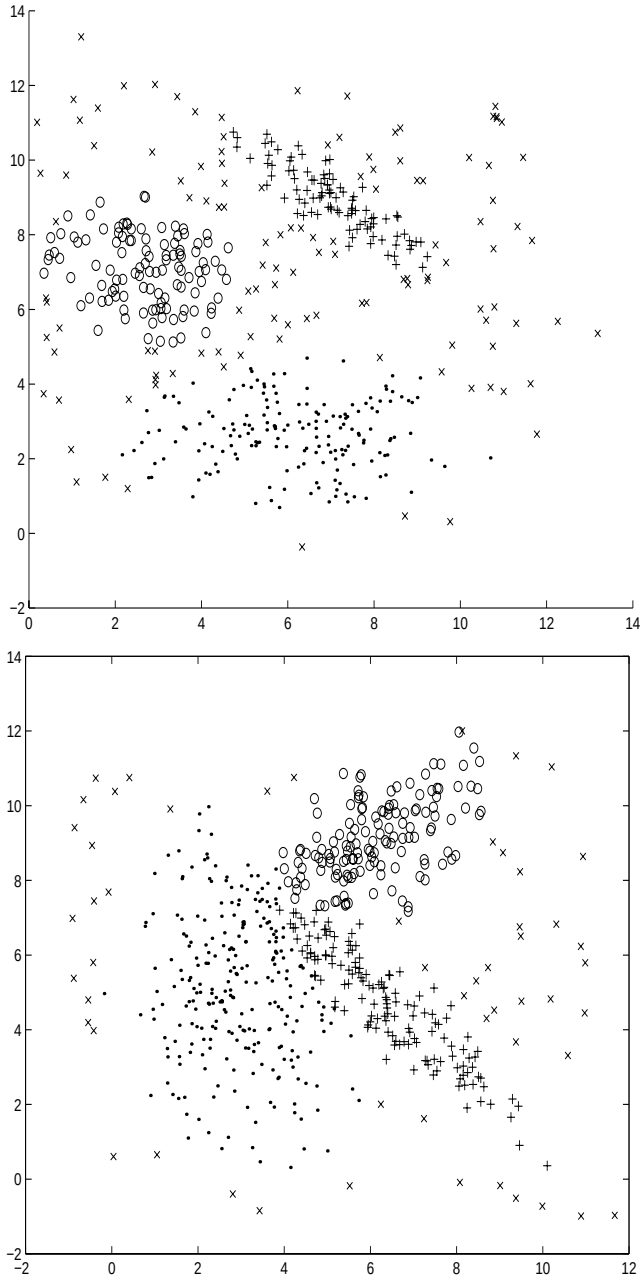

FIG. 6 – *Deux exemples de classification avec rejet - Les points rejetés sont les* ×

Algorithme 2: Mise à jour de la moyenne et matrice de covariance dans le cas semi-supervisé

Données : $\chi = \{x_1, \ldots, x_N\}$, z_{ji} l'estimation courante de $P(C_j|x_i)$, h_{min} le seuil minimal de l'estimateur

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} z_{ij} x_{i\bullet} + \sum_{i=1}^{N_j} x_{ij}}{\sum_{i=1}^{N} z_{ij} + N_j}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{N} z_{ij} (x_{i\bullet} - \mu_j)(x_{i\bullet} - \mu_j)^T + \sum_{i=1}^{N_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T}{\sum_{i=1}^{N} z_{ij} + N_j}$$

répéter

$h_j = \max_{i \in [1, N_j]}(d_{\mathcal{M}}(x_{ij}, C_j), h_{min})$

pour *i = 1 à N* **faire**

$e_{i\bullet} = d_{\mathcal{M}}(x_{i\bullet}, C_j)$

$w_{i\bullet} = \frac{\psi_{huber}(e_{i\bullet}, h_j)}{e_{i\bullet}}$

pour *i = 1 à* N_j **faire**

$e_{ij} = d_{\mathcal{M}}(x_{ij}, C_j)$

$w_{ij} = \frac{\psi_{huber}(e_{ij}, h_j)}{e_{ij}}$

$$\hat{\mu}_j^r = \frac{\sum_{i=1}^{N} w_{i\bullet}\ z_{ij}\ x_{i\bullet} + \sum_{i=1}^{N_j} w_{ij}\ x_{ij}}{\sum_{i=1}^{N} w_{i\bullet}\ z_{ij} + \sum_{i=1}^{N_j} w_{ij}}$$

$$\hat{\Sigma}_j^r = \frac{\sum_{i=1}^{N} w_{i\bullet}\ z_{ij}\ (x_{i\bullet} - \hat{\mu}_j^r)(x_{i\bullet} - \hat{\mu}_j^r)^T + \sum_{i=1}^{N_j} w_{ij}\ (x_{ij} - \hat{\mu}_j^r)(x_{ij} - \hat{\mu}_j^r)^T}{\sum_{i=1}^{N} w_{i\bullet}\ z_{ij} + \sum_{i=1}^{N_j} w_{ij}}$$

jusqu'à *Critère d'arrêt*;

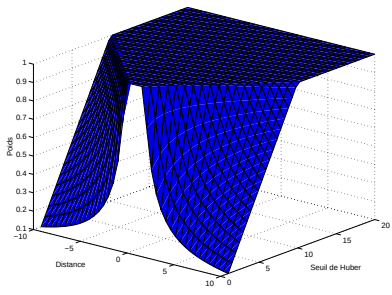

FIG. 7 – *M-Estimateur: Poids en fonction de la distance*

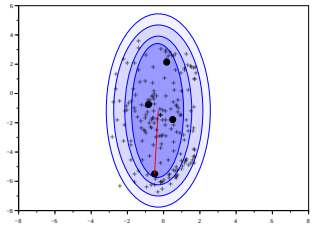

FIG. 8 – *Détermination du seuil de Huber à l'aide des points supervisés*

Il est indispensable de fixer un minimum h_{min} à h sans lequel l'estimation ne porterait que sur un trop petit nombre de points. La figure 8 représente les divers paliers de la fonction de seuil issus de cette heuristique, du plus foncé (poids le plus fort) au plus clair (poids le plus faible).

5 EXPÉRIMENTATIONS

Nous avons testé notre algorithme sur des données artificielles (cf Fig. 6) et d'autres jeux standards couramment utilisés en reconnaissance des formes (base UCI (Murphy & Aha, 1992)). La sélection des points supervisés a été effectuée par tirage aléatoire sans tenir compte des proportions des classes. Si le tirage n'est pas favorable, l'amélioration légitimement attendue peut être moindre. D'autre part, le nombre de paramètres de l'algorithme est trop important pour explorer complètement des intervalles sur ceux-ci sur tous les jeux de données. Ainsi, nous avons choisi de faire un tirage aléatoire sur les γ_j, α_j et taux de supervision pour diverses initialisations.
De plus, il n'existe pas, à notre connaissance, d'autres algorithmes de classification semi-supervisé avec rejet (même si certaines techniques sont adaptables). Nos résultats sont donc difficilement comparables directement avec l'existant.

5.1 Données artificielles

Nous avons repris les jeux de données utilisés précédemment en faisant varier les paramètres du modèle (identique pour chaque classe). Sur le jeu de données *Easy* (classes séparées), nous avons amélioré le taux d'erreur (0.28%) pour un taux de rejet de 10% (9% de supervision) contre 2% d'erreur sans supervision. Sur les données *Dif* (classes mélangées), nous avons également diminué le taux d'erreur (8.44%) par rapport au meilleur taux obtenu précedemment sans supervision (11.33%) dans (Saint-Jean *et al.*, 2000).

5.2 Données réelles

Les figures 9, 11, 13, 15 montrent respectivement les meilleurs résultats obtenus au sens du critère $C(\Gamma)$ parmi les diverses configurations paramétriques aléatoires ($\gamma_j \in [0.05,0.5]$, $\alpha_j \in [1,20]$ et le numéro d'initialisation de 1 à 50) pour différents taux de supervision avec des paramètres de coût fixés comme suit: Cc=1, Ce=2, Cr =2. L'abscisse répresente le taux de supervision (de 0 à 100%), l'axe de gauche, le taux de succès, l'axe de droite le taux d'erreur. Dans les figures 10, 12, 14, 16, le taux d'erreur est remplacé par le taux de rejet. L'aspect chaotique des courbes s'explique par le choix aléatoire des paramètres. Enfin, la table 1 reprend quelques valeurs nous paraissant pertinentes. Les cas 0% et 100% de supervision correspondent aux cas limites de non supervision et de supervision totale.

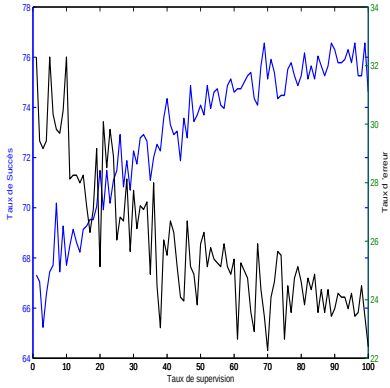

FIG. 9 – *Pima : Taux de Succès et d'erreur en fonction du taux de supervision*

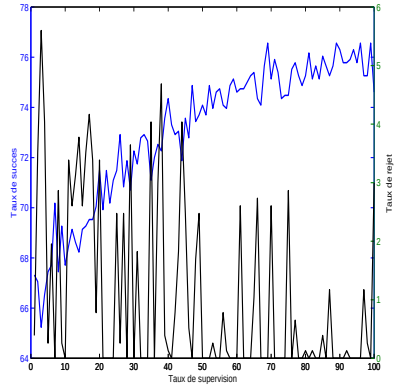

FIG. 10 – *Pima : Taux de Succès et de rejet en fonction du taux de supervision*

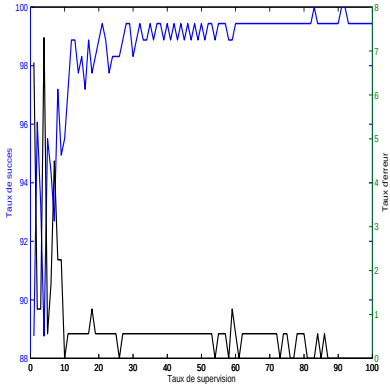

FIG. 11 – *Wine : Taux de Succès et d'erreur en fonction du taux de supervision*

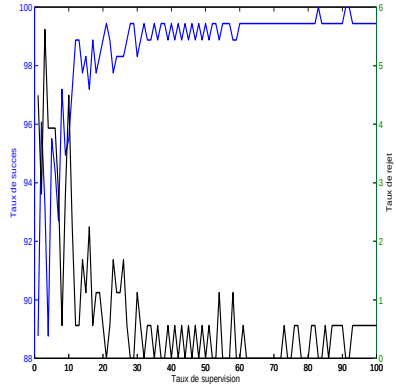

FIG. 12 – *Wine : Taux de Succès et de rejet en fonction du taux de supervision*

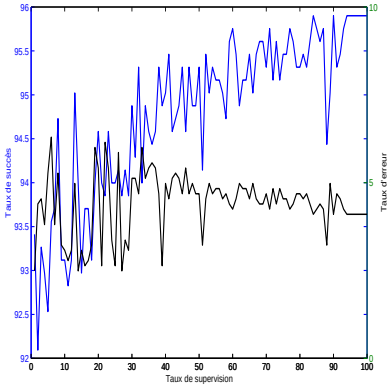

FIG. 13 – *BCW : Taux de Succès et d'erreur en fonction du taux de supervision*

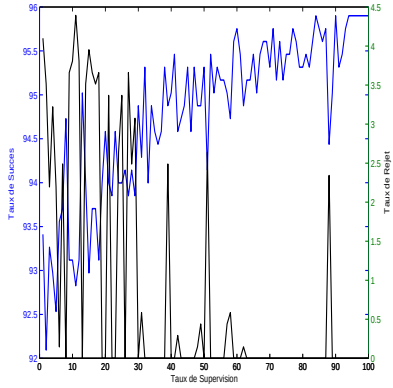

FIG. 14 – *BCW : Taux de Succès et de rejet en fonction du taux de supervision*

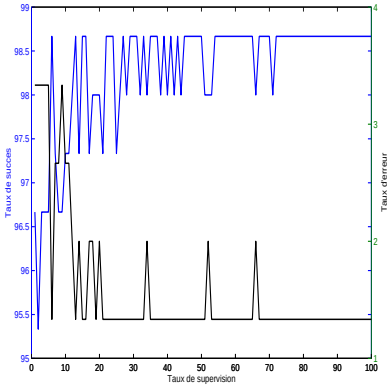

FIG. 15 – *Iris : Taux de Succès et d'erreur en fonction du taux de supervision*

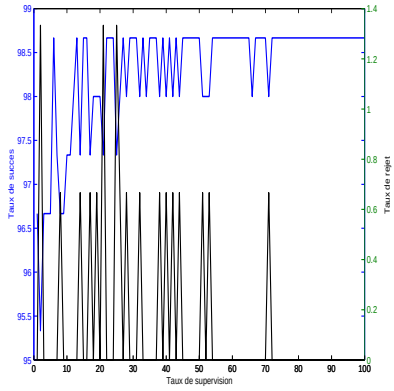

FIG. 16 – *Iris : Taux de Succès et de rejet en fonction du taux de supervision*

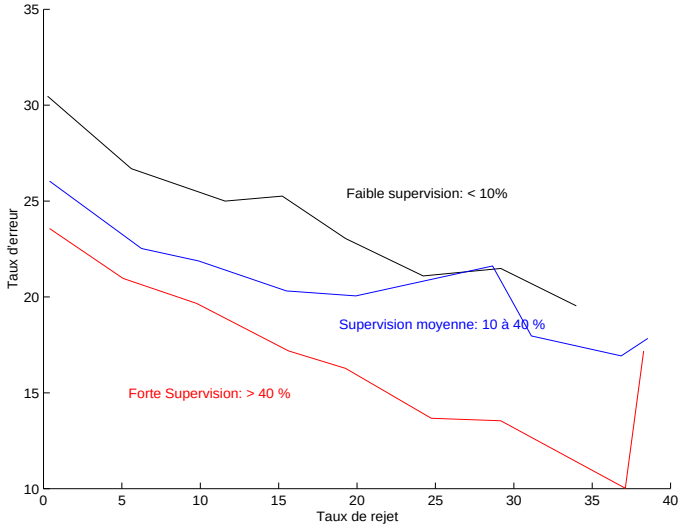

FIG. 17 – *Abaque du taux d'erreur avec le taux de rejet avec divers niveaux de supervision pour le jeu de données Pima*

Afin de pouvoir comparer avec des résultats publiés par ailleurs (Labzour *et al.*, 1998) (semi-supervision sans rejet), nous avons procédé à une resubstitution selon le critère MAP décrit plus haut. D'une manière générale, les résultats obtenus sont comparables avec ceux des méthodes supervisées[1].

Comme on pouvait s'y attendre, la supervision de quelques points permet de réduire le taux d'erreur (à coûts fixés) sur tous les jeux de données. En général, un taux de supervision faible permet de réduire de manière significative le taux d'erreur (sauf pour *Pima*) ou d'améliorer de manière significative le taux de succès.

Pour les données *Pima*, connues pour être très mélangées, seule une pénalisation forte du coût d'erreur ($C_e \gg 2$) permet de diminuer le taux d'erreur (16.66%) pour un taux de rejet important (15.74%), ceci au détriment du taux de succès (67.58%). Plus généralement, on constate que minimiser le taux d'erreur produit des classes plus concentrées sur les zones de fortes densités. Le choix des paramètres de coût dépend donc fortement de l'application visée.
L'abaque de la figure 17 montre, pour divers niveaux de supervision, la relation entre le taux de rejet et le taux d'erreur. Comme on pouvait s'y attendre, le rejet compense l'erreur potentielle de classification qui diminue lorsque la supervision augmente.

1. http://www.phys.uni.torun.pl/kmk/projects/datasets.html

Jeu de données	Taux de supervision	Taux de succès	Taux d'erreur	Taux de Rejet
Breast Cancer Wisconsin N=683,C=2,Dim=9	0.%	91.95%	5.12%	2.93%
	1.02%	93.41%	2.49%	4.1%
	100.%	95.9%	4.1%	0.%
Iris (Anderson) N=150,C=3,Dim=4	0.%	96.67%	2.66%	0.67%
	5.33%	98.66%	1.33%	0.%
	100.%	98.66%	1.33%	0.%
Pima Indians Diabetes N=768,C=2,Dim=8	0.%	66.93%	29.04%	4.04%
	6.77%	70.18%	29.82%	0.%
	100.%	76.56%	23.44%	0.%
Wine N=178,C=3,Dim=13	0.%	89.32%	5.62%	5.06%
	1.78%	96.07%	1.12%	2.81%
	100.%	100.%	0.%	0.%

TAB. 1 – *Résultats extraits sur quelques jeux standards*

Sur le jeu de données *Iris*, Labzour et al. (Labzour *et al.*, 1998) ont obtenu un taux d'erreur de 3.33% pour un taux de supervision de 10% tenant compte des proportions des classes. Sans cette hypothèse, notre algorithme divise ce taux d'erreur par 2 (1.33%) pour deux fois moins de supervision (5.87%). On remarque aussi que la supervision totale n'améliore pas les résultats.

Par ailleurs, nous avons pu constater que le système traite correctement les *outliers*. En effet, ils sont modélisés par la seconde composante des classes (Eq. 8-terme(A)) et par conséquent rejetés dans la phase d'affectation. Enfin, l'estimation robuste joue parfaitement son rôle qui tente de contrer l'attraction naturelle des centres vers les zones de fort mélange.

6 CONCLUSIONS ET PERSPECTIVES

Dans cet article, l'intêret de la semi-supervision a été rappelé à travers quelques exemples. Nous avons proposé un algorithme de classification semi-supervisée qui se distingue des approches classiques par l'introduction du rejet (classe suplémentaire modélisant les *outliers*) et l'utilisation de M-estimateurs robustes. Une méthode originale de choix du paramètre de l'estimateur de Huber tenant compte de la supervision partielle a été proposée. Les premières expérimentations effectuées sur divers jeux de données donnent des résultats prometteurs. Malheureusement, la comparaison directe de notre méthode avec d'autres est difficile. Bien évidemment, le choix des points à superviser influe largement sur la qualité de l'apprentissage comme il a été remarqué dans (Cohn *et al.*, 1996). Nos recherches actuelles visent à intégrer dans l'algorithme une véritable stratégie de sélection pour tenir compte au mieux des connaissances d'un éventuel expert. L'objectif à terme est d'intégrer notre algorithme dans une application réelle.

RÉFÉRENCES

AMAR A., LABZOUR N. & BENSAID A. (1997). Semi-supervised hierarchical clustering algorithms. In *Sixth Scandinavian Conference on Artificial Intelligence*, p. 232–239, Helsinki, Finland.

BENNETT K. & DEMIRIZ A. (1998). Semi-supervised support vector machines. *In Proceedings of Advances in Neural Information Processing Systems*, **11**, 368–374.

BENSAID A., HALL L., BEZDEK J. & CLARKE L. (1996). Partially supervised clustering for image segmentation. *Pattern Recognition*, **5**(29), 859–871.

BEZDEK J. (1987). *Pattern Recognition With Fuzzy Objective Function Algorithms (Second Edition)*. New-York: Plénum.

BIERNACKI C., CELEUX G. & GOVAERT G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(7), 719–725.

COHN D., GHAHRAMANI Z. & JORDAN M. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, **4**, 129–145.

DEMIRIZ A., BENNETT K. & EMBRECHTS M. (1999). Semi-supervised clustering using genetic algorithms. *Intelligent Engineering Systems Through Artificial Neural Networks*, p. 809–814.

DEMPSTER A. P., LAIRD N. M. & RUBIN D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, **39**, 1–38.

FUNG G. & MANGASARIAN O. (1999). *Semi-Supervised Support Vector Machines for Unlabeled Data Classification*. Rapport interne, Data Mining Institute Technical Report 99-05.

HSIEH P. (1998). *Classification of high dimensional data*. PhD thesis, School of electrical and computer engineering, Purdue University, West lafayette, Indiana 47907-1285.

HUBER P. (1981). *Robust Statistics*. New York: John Wiley.

HUGHES G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, **14**(1), 55–63.

LABZOUR N., BENSAID A. & BEZDEK J. (1998). Improved semi-supervised point-prototype clustering algorithms. *IEEE Int'l Conf. on Fuzzy Systems , IEEE World Congress on Computational Intelligence*.

MURPHY P. M. & AHA D. W. (1992). UCI Repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html.

NIGAM K., MCCALLUM A., THRUN S. & MITCHELL T. (2000). Text classification from labeled and unlabeled documents using em. *Machine learning*, **39**(2/32), 103–134.

PEDRYCZ W. (1985). Algorithms of fuzzy clustering with partial supervision. *Pattern Recogition Letters*, **3**(1), 13–20.

PEDRYCZ W. & WALETZKY J. (1997). Fuzzy clustering with partial supervision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **27**(5), 787–795.

SAINT-JEAN C., FRÉLICOT C. & VACHON B. (2000). Clustering with EM: complex models vs. robust estimation. *in Proceedings SPR 2000, Valence (Espagne)*.

TADJUDIN S. & LANDGREBE D. (1998). Robust parameter estimation for mixture model. In *In proceedings of International Geoscience and Remote Sensing Symposium*, Seattle, USA.