SLEMC : Apprentissage semi-supervisé enrichi par de multiples clusterings.

Cédric Wemmert*, Germain Forestier*

*Université de Strasbourg
LSIIT UMR 7005 CNRS/UDS
{wemmert,forestier}@unistra.fr

Résumé. La tâche de classification supervisée consiste à induire un modèle de prédiction en utilisant un ensemble d'échantillons étiquetés. La précision du modèle augmente généralement avec le nombre d'échantillons disponibles. Au contraire, lorsque seuls quelques échantillons sont disponibles pour l'apprentissage, le modèle qui en résulte donne généralement des résultats médiocres. Malheureusement, comme la tâche d'étiquetage est souvent très coûteuse en temps, les utilisateurs fournissent principalement un très petit nombre d'objets étiquetés. Dans ce cas, de nombreux échantillons non étiquetés sont généralement disponibles. Lorsque les deux types de données sont disponibles, les méthodes d'apprentissage semi-supervisé peuvent être utilisées pour tirer parti à la fois des données étiquetées et non étiquetés. Dans cet article nous nous concentrons sur le cas où le nombre d'échantillons étiquetés est très limité. Ainsi, définir des approches capables de gérer ce type de problème représente un verrou scientifique important à lever. Dans cet article, nous passons en revue et comparons les différents algorithmes d'apprentissage semi-supervisé existant, et nous présentons également une nouvelle manière de combiner apprentissage supervisé et non supervisé afin d'utiliser conjointement les données étiquetées et non étiquetées. La méthode proposée utilise des résultats de clustering pour produire des descripteurs supplémentaires des données, qui sont utilisés alors lors d'une étape d'apprentissage supervisé. L'efficacité de la méthode est évaluée sur différents jeux de données de l'UCI [1] et sur une application réelle de classification d'une image de télédétection à très haute résolution avec un nombre très faible d'échantillons étiquetés. Les expériences donnent des indications sur l'intérêt de combiner des données étiquetées et non dans le cadre de l'apprentissage automatique.

1 Introduction

Le nombre d'exemples étiquetés est un paramètre essentiel lors de la phase d'apprentissage en classification supervisée. Si trop peu d'exemples sont disponibles, le modèle prédictif induit par l'apprentissage aura des performances relativement faibles. Malheureusement, dans

1. Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

de nombreuses applications réelles, les objets étiquetés sont difficiles à obtenir. En effet, cela s'explique souvent par le coût important de l'intervention humaine dans le processus d'identification et de sélection des exemples. Nous pouvons citer par exemple la recherche d'objets dans une base de données à partir de quelques échantillons saisis par l'utilisateur (recherche d'images basée sur le contenu, recommandation de sites web, etc.). Dans ces cas, très peu d'exemples étiquetés sont disponibles alors qu'un nombre important de données non-étiquetées sont disponibles (toutes les autres instances de la base de données). Pour l'exemple de la recommandation de sites web, il est difficile d'imaginer de demander à l'utilisateur d'étiqueter plus de sites qu'il n'en connait, puisque son objectif est précisément de trouver des sites similaires, mais qu'il ne connait pas encore. Le même problème apparait en vente à distance, pour les systèmes de recommandation d'articles, qui se basent uniquement sur les achats déjà réalisés par l'acheteur. Enfin, dans les applications nécessitant une identification visuelle par l'utilisateur pour la création des exemples, leur nombre est généralement assez faible par rapport au volume des données dans lesquelles la recherche va s'effectuer.

Un autre problème important est de parvenir à réaliser une classification très précise, lorsque le rapport entre le nombre d'exemples étiquetés et le nombre d'attributs les décrivant est très faible. Si le nombre d'attributs est élevé, les classifieurs standards nécessitent beaucoup d'exemples pour obtenir de bon taux de classification. Cette observation est connue comme étant le phénomène de Hughes (Hughes, 1968). Dans le cadre de la télédétection, les capteurs hyperspectraux peuvent désormais produire des données comportant un nombre important de bandes spectrales (jusqu'à 200 valeurs réelles par pixel de l'image). Avec de telles données, plus de détails peuvent être observés sur la couverture du sol, c'est-à-dire que le nombre de classes d'intérêt augmente. Plus d'attributs et plus de classes, ce qui implique un besoin plus important en terme d'exemples, qui sont généralement coûteux en temps et en argent à acquérir. La même observation peut être réalisée avec les méthodes d'analyse basées objets des images à très haute résolution spatiale (taille d'un pixel inférieur à 5m). Avec ce type d'images, une première étape de segmentation est d'abord réalisée afin de construire des régions. Celles-ci sont alors décrites par un ensemble d'attributs spectraux, spatiaux ou contextuels trop important en regard du nombre d'exemples disponibles.

Alors que les données étiquetées sont rares et souvent insuffisantes par rapport à la dimension de l'espace de recherche, les données non-étiquetées sont disponibles en très grande quantité. Des travaux récents montrent que ces données peuvent être utilisées afin d'améliorer la qualité d'une classification supervisée (Blum et Mitchell, 1998; Nigam et al., 2000a; Seeger, 2002). On parle alors de *classification semi-supervisée*.

La méthode proposée dans cet article est sensiblement différente des approches existantes, car nous proposons d'utiliser plusieurs classifieurs non supervisés afin de créer une nouvelle description des exemples basée sur les clusters proposés. Ensuite, un classifieur supervisé est appliqué dans ce nouvel espace de données. Comme l'objectif des classifieurs non-supervisés est de créer des clusters qui maximisent la similarité intra-cluster conjointement à la dissimilarité inter-cluster, aucun exemple étiqueté n'est nécessaire, mais aucune étiquette n'est attribuée aux clusters trouvés. La classification non-supervisée (ou *clustering*) peu être vue comme un moyen de résumer la distribution des objets étiquetés dans leur espace.

Dans cet article, nous présentons tout d'abord des travaux récents sur le sujet et justifions l'intérêt de notre méthode dans la section 2. Ensuite, plusieurs algorithmes semi-supervisés ainsi que notre approche sont décrits plus précisément dans la section 3. Section 4, nous pré-

sentons plusieurs expériences réalisées sur les jeux de données classiques de l'UCI (Newman et al., 1998). Les résultats obtenus sont comparés avec les autres méthodes semi-supervisées ainsi que des méthodes supervisées, afin de quantifier l'apport de l'étape de clustering dans le processus d'apprentissage. Nous donnons aussi des résultats obtenus sur des données réelles dans le domaine de la classification d'images de télédétection, et plus précisément, d'une image à très haute résolution spatiale d'une zone urbaine de la région de Strasbourg (France). Enfin, nous concluons et exposons quelques pistes de travaux futurs.

2 État de l'art

Plusieurs travaux ont montré que les données non-étiquetées pouvaient permettre d'améliorer la qualité de la classification lorsque peu d'exemples étaient disponibles (Goldman et Zhou, 2000; Blum et Mitchell, 1998; Joachims, 1999; Nigam et al., 2000b; Bennett et al., 2002; Chawla et Karakoulas, 2005; Gabrys et Petrakieva, 2004; Zhou et al., 2007; Raskutti et al., 2002; Deodhar et Ghosh, 2007; Bouchachia, 2007; Cai et al., 2009).

Gabrys et Petrakieva (2004) ou Bouchachia (2007) proposent de classer ces méthodes en trois catégories principales : les *méthodes de pré-étiquetage*, pour lesquelles les données non-étiquetées sont étiquetées à l'aide d'un classifieur initial entraîné sur l'ensemble des exemples disponibles, les *méthodes de post-étiquetage*, qui consistent à étiqueter des clusters construits sur l'ensemble des données disponibles en fonction de leur composition en terme d'exemples étiquetés, et les *approches semi-supervisées*, qui utilisent à la fois les données étiquetées ou non durant le processus de clustering.

2.1 Méthode de pré-étiquetage

Une première manière d'utiliser les objets non-étiquetées est mise en œuvre dans la méthode dite de *co-training* développée par Blum et Mitchell (1998). L'idée principale est d'utiliser deux classifications complémentaires afin d'étiqueter itérativement les données non-étiquetées. Cela implique qu'il existe deux ensembles d'attributs indépendants et complémentaires pour décrire les données.

Afin d'étendre cette méthode et de permettre d'éviter l'indépendance et la redondance entre les ensembles d'attributs, ce qui n'est pas très réalistes dans le cas d'applications réelles, Goldman et Zhou (2000) ont présenté une stratégie de co-training qui utilise les données non-étiquetées afin d'améliorer les performances d'un classifieur supervisé. Leur méthode utilise aussi deux classifieurs supervisés différents qui sélectionnent les futurs objets à étiqueter. Les expériences montrent que la méthode augmente significativement la précision de l'algorithme ID3. Dans (Nigam et al., 2000b), une version basée sur l'algorithme *Expectation Maximization* (EM) est proposée et appliquée à de la classification de textes.

Plus récemment, Raskutti et al. (2002) ont présenté une méthode de co-training qui ne nécessite pas obligatoirement d'utiliser deux classifieurs complémentaires. L'idée est de produire une "vue" différente des données en calculant une classification non-supervisée sur tout le jeu de données (étiqueté et non-étiqueté). Alors, les données originales et la nouvelle représentation sont utilisées pour créer deux prédicteurs indépendants et appliquer le co-training.

Dans (Zhou et al., 2007), les auteurs présentent une approche de co-training qui suppose d'avoir deux représentations des données. La méthode utilise la corrélation entre les deux vues

afin de produire de manière itérative de nouveaux exemples positifs et négatifs. Les expériences montrent que cette technique surpasse les autres approches de co-training, sur des cas pour lesquels un seul exemple étiqueté existe.

Contrairement au co-training, ASSEMBLE (Bennett et al., 2002) ne nécessite pas d'avoir des vues multiples sur les données. Il incorpore itérativement des exemples auto-étiquetés par un processus de boosting. À chaque itération, des objets non-étiquetés le sont grâce à l'ensemble d'exemples existant et ajoutés à l'ensemble d'apprentissage.

2.2 Méthodes de post-étiquetage

L'idée de ces approches est de commencer par effectuer un clustering sur les données et d'évaluer par la suite la *pureté* de chaque cluster, en calculant leur composition en terme d'exemples étiquetés. N'importe quel algorithme de clustering peut être utilisé lors de cette première étape. Dans (Ghahramani et Jordan, 1994), une approche générique basée sur l'algorithme *Expectation Maximization* (EM) est proposée.

D'autres méthodes (Kothari et Jain, 2003; Eick et al., 2004) optimisent la pureté des clusters extraits du jeu de données. Elles proposent d'utiliser un algorithme génétique afin de raffiner itérativement l'appartenance des objets aux classes clusters, afin que de maximiser la pureté des clusters en terme d'étiquette.

2.3 Clustering semi-supervisé

Dans (Chawla et Karakoulas, 2005), une étude empirique de plusieurs techniques d'apprentissage semi-supervisé appliquées à différents type de jeux de données est présentée. Plusieurs expériences permettent d'évaluer l'influence de la taille des ensembles d'objets étiquetés et non-étiquetés, et l'effet du bruit dans les données exemples. L'article conclut que la performance des différentes méthodes dépend fortement du domaine d'application et de la nature des données. Cependant, l'utilisation conjointe de données étiquetées et non-étiquetées augmente significativement la qualité des résultats dans l'ensemble des cas.

Basu et al. (2002) ont défini deux variantes de l'algorithme K-means, qui permettent d'utiliser des données étiquetées pour améliorer la classification. L'idée principale est d'utiliser les objets étiquetés pour guider la phase d'initialisation des noyaux. Dans la première méthode, les noyaux initiaux sont déterminés uniquement grâce aux données étiquetées puis l'algorithme se déroule normalement. Tous les objets (étiquetés ou non) se déplacent à chaque itération et sont affectés au cluster le plus proche. Dans la seconde variante, les objets étiquetés restent dans leur cluster initial quelque soit la manière dont évolue leurs centres.

Une approche plus récente est exposée dans (Cai et al., 2009). L'idée est de calculer simultanément le clustering et la classification supervisée, plutôt que de procéder en deux étapes. Afin de le réaliser, les auteurs définissent une fonction d'objectif qui évalue à la fois la qualité de classification et de clustering en mélangeant deux termes : le taux de mauvaise classification (pour la partie supervisée) et l'impureté des clusters (pour l'aspect non-supervisé). Une méthode à peu près similaire est présentée dans (Deodhar et Ghosh, 2007) et appliquée à des données réelles en marketing.

2.4 Applications en télédétection

Comme mentionné précédemment, les méthodes semi-supervisées ont été appliquées dans plusieurs domaines différents (classification de textes, de sites web ou d'images). Dans cet article, nous nous intéressons au cas de la classification d'images des télédétection. Dans ce domaine, plusieurs travaux ont montré qu'il était possible de tirer parti des données non-étiquetées dans le processus de classification (Shahshahani et Landgrebe, 1994). Ils y proposent trois méthodes qui incorporent simultanément des données étiquetées et non-étiquetées dans des classifieurs paramétriques, non-paramétriques et semi-paramétriques. Des résultats sont montrés sur un petit extrait d'une images AVIRIS, qui mettent en avant les améliorations obtenues grâce à leur méthode.

Dans (Hoffbeck et Landgrebe, 1996), les auteurs présentent une approche basée sur une nouvelle estimation de la matrice de covariance. Elle produit des classifications plus précises que les méthodes standards d'estimation de la matrice de covariance, dans les cas où la taille du jeu d'apprentissage est faible. Plusieurs expériences sur la classification d'une image AVIRIS d'une zone d'agriculture du Nevada montrent l'efficacité de la méthode sur des données réelles.

Plus récemment, Jia et J.A. (2002) ont présentés une nouvelle méthode capable de traiter des données hyperspectrales. L'idée est d'effectuer un clustering de l'ensemble d'exemples étiquetés, puis d'associer plusieurs clusters aux différentes classes pour construire une représentation des classes sous forme de clusters. Puis, chaque pixel est classé en fonction de son cluster d'appartenance et de l'appartenance de ce cluster aux classes.

Morgan et al. (2004) ont proposé une autre approche résolvant partiellement le problème du trop petit nombre d'exemples étiquetés. Elle consiste à utiliser une technique de réduction de la dimensionalité qui s'adapte automatiquement à la taille du jeu d'exemples. Des expériences sont présentées sur la classification de données hyperspectrales HyMap (Hyperspectral Mapper). Les résultats montrent que même si le nombre d'attributs a été réduit et le nombre d'exemples disponibles est faible, les classifications produites conservent une grande précision.

La méthode que nous proposons dans cet article est quelque peu différente de celles décrites ci-dessus. En effet, toutes les méthodes de co-training utilisent les données étiquetées et non-étiquetées ensemble lors de la phase d'apprentissage. Si de nouveaux exemples sont disponibles, toute cette étape doit être recalculée afin de les prendre en compte, ce qui est souvent coûteux. Dans notre approche, la classification non supervisée peut être vue comme un pré-traitement qui n'est réalisé qu'une seule fois. Ensuite, en fonction des exemples disponibles, la classification supervisée est construite, ce qui est généralement très rapide, le nombre d'exemples étant très faible.

3 Méthodes semi-supervisées

Dans cette section, nous formalisons plus précisément les différentes approches que nous avons comparées à la nôtre lors de nos expériences.

Soit X un ensemble de n objets $x_j \in X$. Nous nous plaçons dans un cas de classification à q classes avec m exemples étiquetés et l objets non-étiquetés. Nous faisons l'hypothèse que m est très faible et $l >> m$.

Soit L l'ensemble des objets étiquetés de X :

$$L = \{(x_1, y_1), \ldots, (x_m, y_m)\} \tag{1}$$

avec $y_i \in \{1, \ldots, q\}$ les étiquettes des classes des exemples.
Soit U l'ensemble des objets non-étiquetés de X :

$$U = \{(x_{m+1}, 0), \ldots, (x_{m+l}, 0)\} \tag{2}$$

avec 0 signifiant qu'aucune étiquette n'est associée à cet objet.

L'objectif de la classification semi-supervisée est de construire un classifieur basé sur tout l'ensemble d'apprentissage X. Ce classifieur peut être vu comme une fonction associant une des q classes à chaque objet de x. Il peut être défini formellement de la manière suivante :

$$y = C_X(x) \ : \ y \in \{1, \ldots, q\} \tag{3}$$

3.1 Méthodes de pré-étiquetage

La première famille de méthodes étudiée se composent des *méthodes de pré-étiquetage*. L'idée est de réaliser une première classification C_L uniquement sur les données étiquetées (L). Ensuite, les données non-étiquetées sont étiquetées en fonction de cette classification.

Dans la méthode de *static labeling* (**SL**) Gabrys et Petrakieva (2004), les données non-étiquetées sont toutes étiquetées en une étape en appliquant simplement C_L à U. Un nouvel ensemble de données W est alors construit :

$$W = \{(x_j, y_j) \ : \ y_j = C_L(x_j), x_j \in U\} \tag{4}$$

Ensuite, lors d'une seconde étape, la classification finale est calculée comme :

$$y = C_{L \cup W}(x) \tag{5}$$

L'algorithme décrivant cette méthode est présenté par l'Algorithme 1.

Algorithm 1 Static labeling (**SL**)

1: construire une classification C_L à partir d'un ensemble d'exemples étiquetés L
2: soit $W = \{(x_j, y_j) \ : \ y_j = C_L(x_j), x_j \in U\}$
3: construire le classifieur final $C_{L \cup W}$

Une autre méthode de pré-étiquetage, appelée *dynamic labeling* (**DL**), est proposée dans (Gabrys et Petrakieva, 2004). Comme pour la méthode précédente, un classifieur C_L est construit à partir de l'ensemble d'apprentissage. Ensuite, plutôt que d'étiqueter l'ensemble des objets de U en une fois, ils le sont de manière itérative, un à la fois. Un objet x_j de U est choisi et étiqueté en fonction de C_L. Il est alors ajouté dans L et un nouveau classifieur est entraîné à partir de $L \cup \{x_j\}$. Le processus est itéré jusqu'à ce que tous les objets non-étiquetés soient étiquetés. Cet algorithme est présenté par l'Algorithme 2.

L'ordre dans lequel sont choisis les objets est défini en fonction de leur degré de confiance dans la clasification de chacun.

Algorithm 2 Dynamic labeling (**DL**)

soit $U' = U$ et $W' = \varnothing$
construire un classifieur $C_{L \cup W'}$
for all $x_j \in U'$ choisi en fonction de leur degré de confiance dans la classe **do**
  $W' := W' \cup \{(x_j, t_j) \ : \ t_j = C_{L \cup W'}(x_j)\}$
  $U' := U' \setminus \{(x_j, 0)\}$
end for
construire la classification finale $C_{L \cup W'}$

3.2 Méthodes de post-étiquetage

À l'inverse des méthodes de pré-étiquetage qui utilisent les données non-étiquetées pour améliorer un classifieur initial, les *méthodes de post-étiquetage* commencent par construire un clustering sur tous les objets de l'ensemble, mais sans tenir compte des étiquettes et exemples disponibles. Ensuite, les données non-étiquetées de chaque cluster sont étiquetées avec l'étiquette majoritaire de leur cluster.

Soit K_l, $l = 1, \ldots, k$, les clusters produits par la première étape de classification non-supervisée, et c_{lj}, $j = 1, \ldots, q$ le nombre d'objets étiquetés avec la classe j dans le cluster l :

$$c_{lj} = \|\{(x_i, y_i) \in L \ : \ (x_i, y_i) \in K_l, y_i = j\}\| \tag{6}$$

Étiquetage des clusters à la majorité La méthode de post-étiquetage présentée ici est appelée *étiquetage des clusters à la majorité* (**CLM**) Gabrys et Petrakieva (2004) et est décrite par l'Algorithme 3. Elle se compose de trois étapes :

- La première consiste à étiqueter tous les clusters contenant au moins un exemple étiqueté. L'étiquette affectée à chaque objet du cluster est l'étiquette majoritairement présente parmi les objets étiquetés dans le cluster.
- La seconde étape affecte l'étiquette du cluster le plus similaire aux clusters ne contenant aucun exemple. La mesure de similarité $\Delta(K_j, K_k)$ dépends de la méthode de clustering utilisée et estime la distance entre deux K_l et K_k.
- Enfin, lors de la troisième étape, le classifieur final est construit en fonction du nouvel ensemble d'objets étiquetés.

Optimisation de la pureté Une autre famille de méthodes Eick et al. (2004) est basée sur l'optimisation d'un critère de *pureté* d'un clustering K construit initialement à partir des données. L'évaluation de la pureté Π d'un cluster est basée sur deux critères :

- l'impureté de classe qui mesure le pourcentage d'*exemples minoritaires* dans les différents clusters de K ;
- le nombre de clusters k qui doit être maintenu à un niveau plutôt bas dans la majorité des cas.

Les *exemples minoritaires* sont des objets étiquetés appartenant à la classe qui n'est pas la classe majoritaire dans le cluster. Comme défini précédemment, la classe majoritaire d'un cluster K_l est $y_{K_l} = \arg\max_{j \in \{1 \ldots q\}} (c_{lj})$. Ainsi, les exemples minoritaires $m(K_l)$ d'un

Algorithm 3 Étiquetage des clusters à la majorité (**CLM**)

construire un clustering $K = \{K_l, l = 1 \dots k\}$ à partir de X
let $LK := \varnothing$
for all $K_l, l = 1 \dots k$ **do**
  if $\sum_{j=1}^{q} c_{lj} \neq 0$ **then**
    $y_{K_l} = \arg\max_{j \in \{1 \dots q\}} (c_{lj})$
    $W_l = \{(x_i, y_{K_l}), x_i \in K_l\}$
    $LK := LK \cup K_l$
  end if
end for
for all $K_u \ : \ \sum_{j=1}^{q} c_{uj} = 0$ **do**
  $K_m = \arg\max_{K_l \in LK} (\Delta(K_l, K_u))$
  étiqueter tous les objets du cluster K_u avec l'étiquette y_{K_m}
  $W_u = \{(x_i, y_{K_m}), x_i \in K_u\}$
end for
construire le classifieur final $C_{W_1 \cup \dots \cup W_k}$

cluster K_l peuvent être défini par :

$$m(K_l) = \{(x_i, y_i) \in K_l \ : \ y_i \neq y_{K_l}\} \tag{7}$$

et les exemples minoritaires $M(K)$ du clustering K par :

$$M(K) = \{m(K_l) \ : \ \forall l \in [1 \dots k]\} \tag{8}$$

La pureté se définit alors comme :

$$\Pi(K) = \mathsf{impurity}(K) + \eta \times \mathsf{penalty}(k) \tag{9}$$

avec

$$\mathsf{impurity}(K) = \frac{\|M(K)\|}{n}$$

et

$$\mathsf{penalty}(k) = \begin{cases} \sqrt{\frac{k-q}{n}} & k \geqslant q \\ 0 & k < q \end{cases}$$

Le paramètre η ($0 < \eta < 2$) détermine la pénalité associée au nombre de clusters k afin qu'il ne devienne pas trop important.

Le premier algorithme défini par Eick et al. Eick et al. (2004) est un algorithme glouton appelé *Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart* (**SRIDHCR**) Eick et al. (2004). Plusieurs objets sont sélectionnés aléatoirement (entre q et $2q$ objets) pour être les représentants initiaux des clusters, qui sont créés en leur affectant les objets leur étant les plus proches. Ensuite, un objet est ajouté ou supprimé de l'ensemble des représentants. La qualité $\Pi(K)$ est évaluée et l'algorithme itère jusqu'à ce qu'il n'y ait plus d'amélioration significative de la qualité du résultat. En général, l'algorithme est lancé r fois et la meilleure solution est conservée. Une description précise est donnée par l'Algorithme 4.

Algorithm 4 Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart) (**SRIDHCR**)

for $i = 1$ to r **do**
  $\mathsf{rep} := \{(x_i, y_i) \in L \ : \ \text{choisis aléatoirement}\}$ and $q \leqslant \|\mathsf{rep}\| \leqslant 2q$
  while non fin **do**
    soit $s^1 = \mathsf{rep} \cup (x_i, y_i) \ : \ x_i \notin \mathsf{rep}$
    soit $s^2 = \mathsf{rep} \setminus (x_i, y_i) \ : \ x_i \in \mathsf{rep}$
    soit $S = \arg\min_{s \in \{s1, s2\}} \Pi(s)$
    if $\Pi(S) < \Pi(\mathsf{rep})$ **then**
      $\mathsf{rep} := S$
    else if $\Pi(S) = \Pi(\mathsf{rep})$ et $\|S\| > \|\mathsf{rep}\|$ **then**
      $\mathsf{rep} := S$
    else
      fin
    end if
  end while
end for
la meilleure solution est conservée

Le second algorithme, *Supervised Clustering using Evolutionary Computing* (**SCEC**), essaie de trouver les meilleurs représentants des clusters par une approche évolutionnaire (Eick et al., 2004). Un premier ensemble de p clusterings est généré aléatoirement puis des opérateurs génétiques sont appliqués pour créer les générations futures. Trois opérateurs sont utilisés dans SCEC :

- mutation : un représentant est remplacé par un autre objet qui n'est pas déjà un représentant dans une des solutions ;
- croisement : cet opérateur crée une nouvelle solution à partir de deux solutions existantes ; l'intersection des représentants des deux solutions est insérée dans la nouvelle solution, et chaque représentant restant de chacune des solution est inséré ou non avec une probabilité de 50% dans la nouvelle solution ;
- copie : une solution de la génération actuelle est copiée dans la génération suivante.

Les nouveaux représentants sont choisis aléatoirement par un tournoi d'ordre k sur l'ensemble des représentants construits à partir des opérateurs. Le processus est itéré sur un nombre fixe de N générations. L'algorithme complet est décrit par l'Algorithme 5.

3.3 Clustering semi-supervisé

Le dernier type de méthodes, appelées approches par *clustering semi-supervisé*, utilisent les données étiquetées ou non en même temps. L'idée est que le clustering des données doit être guidé par les exemples étiquetés.

Refined clustering La méthode *refined clustering* (**RC**) (Gabrys et Petrakieva, 2004) se compose de deux étapes :

Algorithm 5 Supervised Clustering using Evolutionary Computing (**SCEC**)

1: $pop_0 = \{S_r = \{s_i^r = (x_i, y_i) \in L$: choisis aléatoirement$\}$
   $1 \leqslant r \leqslant ps\}$
2: **for** $i = 1$ à N **do**
3: $mu_i = \texttt{mutation}(pop_{i-1})$
4: $cr_i = \texttt{croisement}(pop_{i-1})$
5: $co_i = \texttt{copie}(pop_{i-1})$
6: $pop_i =$ select_by_k_tournament(mu_i, cr_i, co_i)
7: **for all** $S_r \in pop_i$ **do**
8: soit K^r le clustering correspondant aux représentants S_r
9: **for all** $l \in [1 \dots k]$ **do**
10: $K_l^r = \{(x_j, y_j)$: $\texttt{dist}(x_j, s_l^r)$ est minimal $\}$
11: **end for**
12: calculer $\Pi(S_r)$
13: **end for**
14: **end for**
15: la meilleure solution S_r est conservée

– création d'un nouvel ensemble de données totalement étiqueté à partir de toutes les données disponibles ($L \cup U$) ;
– application d'un algorithme des k-plus proches voisins pour construire le classifieur final.

La première étape est évidemment la plus cruciale et peut être décomposé en deux sous-parties :

– la création d'un nouvel ensemble de données, incorporant à la fois les objets étiquetés et non-étiquetés et représentant un maximum d'information ;
– l'étiquetage de ce nouvel ensemble par la méthode CLM décrite précédemment (Section 3.2).

La création du nouvel ensemble est réalisé grâce à une approche divisive. L'idée est de scinder les clusters existants contenant des objets étiquetés par différentes classes. Un seuil est défini comme le ratio représentatif minimal d'une classe dans un cluster.

De plus, afin d'éviter la disparition des classes minoritaires, celles n'étant présentes que dans un seul cluster sont conservées en scindant le cluster en deux. L'algorithme complet est présenté par l'Algorithme 6.

Méthodes de seeding Finalement, nous utilisons aussi dans cette étude les méthodes dites de *seeding* proposées par Basu et al. (2002). Ces méthodes sont des variantes de l'algorithme de partitionnement K-means. L'objectif de K-means est de générer un k-partitionnement $K = \bigcup_{i=1}^{k} K_i$ de l'ensemble des données X. Chaque cluster K_i est représenté par son centre de gravité μ_i. Comme indiqué précédemment, nous nous intéressons à un problème de classification à q classes à partir d'un ensemble de données $X = L \cup U$ où L est l'ensemble des données étiquetées. L'idée est de guider l'algorithme K-means en utilisant les objets étiquetés L comme noyaux initiaux. Un premier q-partitionnement $\{S_i\}_{i=1\dots q}$ de L est calculé en regroupant les objets de L ayant la même étiquette dans un même cluster. Une hypothèse forte est qu'il existe

Algorithm 6 Refined clustering (**RC**)

construire un clustering $K = \{K_l, l = 1 \dots k\}$ sur X avec k relativement petit
for all $K_l, l = 1 \dots k$ **do**
  if $\sum_{j=1}^{q} c_{lj} \neq 0$ **then**
    while $nbc > 1$ **do**
      $nbc := 1$
      for all $m \in [1 \dots q]$ **do**
        soit $r_m = \frac{c_{lm}}{\sum_{j=1}^{q} c_{lj}}$
        if $r_m < \Theta$ et $\forall i \in [1 \dots k], i \neq l, g_{im} = 0$ **then**
          $K' := scission(K, l)$
        else if $r_m \geqslant \Theta$ **then**
          $nbc := nbc + 1$
          $K' := scission(K, l)$
        end if
      end for
    end while
  end if
end for
appliquer l'algorithme **CLM** à K'

au moins un objet pour chacun des clusters S_i, c'est-à-dire que l'on dispose d'au moins un exemple par classe.

La première méthode, *Seeded-Kmeans* (**SK**) (Basu et al., 2002), utilise simplement cette initialisation des noyaux plutôt qu'une initialisation aléatoire. L'algorithme K-means est ensuite déroulé normalement sans aucune modification. Une présentation complète de SK est donnée par l'Algorithme 7.

Algorithme 7 Seeded-Kmeans (**SK**)

$\mu_i^{(0)} = \frac{1}{|S_i|} \sum_{x \in S_i} x$ pour $i = 1 \dots q$
$t = 0$
repeat
  $K^{(t+1)} = \left\{ K_i^{(t+1)}, i = 1 \dots q \right\}$
  avec $K_i^{(t+1)} = \left\{ x \in X \ : \ i = \arg\min_h \|x - \mu_h^{(t)}\|^2 \right\}$
  $\mu_i^{(t+1)} = \frac{1}{|K_i^{(t+1)}|} \sum_{x \in K_i^{(t+1)}} x$ pour $i = 1 \dots q$
  $t = t + 1$
until convergence

Dans la seconde méthode, appelée *Constrained-Kmeans* (**CK**) (Basu et al., 2002), le partitionnement des objets étiquetés est utilisés comme dans la méthode *Seeded-Kmeans* pour initialiser le clustering. Cependant, les objets étiquetés ne sont pas réassignés à d'autres clusters durant l'exécution de l'algorithme. Ils sont contraints dans leur cluster d'origine. Cet algorithme est décrit pas l'Algorithme 8.

Apprentissage semi-supervisé enrichi par de multiples clusterings.

Algorithm 8 Constrained-Kmeans (**CK**)

1: $\mu_i^{(0)} = \frac{1}{|S_i|} \sum_{x \in S_i} x$ pour $i = 1 \dots q$
2: $t = 0$
3: **repeat**
4: $K^{(t+1)} = \left\{ K_i^{(t+1)}, i = 1 \dots q \right\}$
avec
$K_i^{(t+1)} = S_i \cup \left\{ x \in X \ : \ i = \arg\min_h \|x - \mu_h^{(t)}\|^2 \right\}$
5: $\mu_i^{(t+1)} = \frac{1}{|K_i^{(t+1)}|} \sum_{x \in K_i^{(t+1)}} x$ pour $i = 1 \dots q$
6: $t = t + 1$
7: **until** convergence

3.4 Apprentissage semi-supervisé enrichi par de multiples clusterings

La méthode que nous proposons, appelée *Semi-supervised learning enhanced by multiple clusterings* (**SLEMC**), pourrait être dans la catégorie des méthodes de post-étiquetage. En effet, elle essaie d'améliorer la classification en produisant tout d'abord un clustering sur l'ensemble de données. Celui-ci, calculé sur tout l'ensemble (étiqueté et non-étiqueté), regroupe les objets similaires ensemble. Ainsi, si les classes du problème sont bien séparées dans leur espace d'attributs, il est relativement simple d'affecter à chaque cluster la classe des objets exemples qui la composent.

Malheureusement, dans les problèmes réels, les classes ne sont pas bien séparées. Il arrive donc fréquemment d'avoir plusieurs objets étiquetés par différentes classes dans un même cluster, ou des clusters sans aucun exemple. Afin de résoudre ce problème, nous proposons d'utiliser un ensemble de clusterings.

Nous considérons b clusterings de l'ensemble de données X. Soit $\mathbf{K}$ cet ensemble de clusterings, $\mathbf{K} = \{K_1, \dots, K_b\}$. L'idée est d'affecter à chaque exemple étiqueté $(x_i, y_i),\ \forall i : 1 < i < m$ (avec m le nombre d'exemples étiquetés), un nouvel ensemble d'attributs :

$$v(x_i) = \left(K_1^i, \dots, K_b^i, y_i\right) \tag{10}$$

avec K_j^i le cluster affecté par la j^{me} méthode de clustering K_j à x_i. Ensuite, un modèle prédictif $C_V : X \to \{1, \dots, q\}$ est dérivé de ce nouvel ensemble de données $V = \{v(x_i)\}_{i=1}^m$, en utilisant un algorithme d'apprentissage supervisé classique. Finalement, l'étiquette $C_V(x_i)$ est affectée à chacun des objets non-étiquetés x_i de U.

L'algorithme complet est présenté par l'Algorithme 9.

4 Résultats

4.1 Évaluation sur des jeux de données artificielles

Dans cette section, nous comparons les méthodes semi-supervisées présentées précédemment sur différents jeux de données de l'UCI Newman et al. (1998). Le tableau 1 présente en détail les caractéristiques des des jeux de données utilisés.

Algorithm 9 Apprentissage semi-supervisé enrichi par de multiples clusterings (**SLEMC**)

soit L l'ensemble des exemples étiquetés disponibles
construire b clusterings $K = \{K_1, \ldots, K_b\}$ sur l'ensemble des données X
for all $(x_i, y_i) \in L$ **do**
    $v(x_i) = \left(K_1^i, \ldots, K_b^i, y_i\right)$
end for
construire un modèle prédictif C_V à partir de $V = \{v(x_i)\}_{i=1}^m$ en utilisant un algorithme classique d'apprentissage
utiliser C_V pour étiqueter U

Données	**#classes**	**#attributs**	**#objets**
iris	3	4	150
wine	3	13	178
ionosphere	2	34	351
diabetes	2	8	768
breast-w	2	9	699
anneal	5	38	898
remote	7	36	6435

TAB. 1 – *Informations sur les différents jeux de données utilisés.*

	1-NN	NB	C45	SL	DL
iris(2)	$78,02(\pm 13,48)$	$67,53(\pm 15,72)$	$56,52(\pm 11,13)$	$71,06(\pm 13,85)$	$79,27(\pm 14,18)$
iris(4)	$85,61(\pm 11,36)$	$76,08(\pm 13,58)$	$72,01(\pm 17,77)$	$81,80(\pm 14,02)$	$88,84(\pm 11,64)$
iris(8)	$91,50(\pm 4,94)$	$93,59(\pm 3,39)$	$90,59(\pm 5,51)$	$93,01(\pm 3,98)$	$93,33(\pm 2,93)$
iris(16)	$93,46(\pm 5,38)$	$94,20(\pm 3,54)$	$91,73(\pm 4,85)$	$93,70(\pm 3,33)$	$94,20(\pm 3,91)$
wine(2)	$76,67(\pm 13,889)$	$54,62(\pm 14,19)$	$51,32(\pm 15,07)$	$72,53(\pm 17,22)$	$80,92(\pm 15,29)$
wine(4)	$88,92(\pm 8,046)$	$77,88(\pm 11,89)$	$70,39(\pm 11,10)$	$87,27(\pm 9,57)$	$90,69(\pm 9,22)$
wine(8)	$91,95(\pm 5,250)$	$88,77(\pm 8,85)$	$79,90(\pm 7,44)$	$93,43(\pm 4,29)$	$94,76(\pm 3,52)$
wine(16)	$93,50(\pm 3,636)$	$95,53(\pm 3,73)$	$85,04(\pm 6,10)$	$95,52(\pm 3,67)$	$96,09(\pm 3,65)$
breast-w(2)	$81,88(\pm 16,93)$	$82,39(\pm 15,30)$	$70,68(\pm 17,74)$	$85,22(\pm 16,45)$	$84,72(\pm 17,98)$
breast-w(4)	$87,08(\pm 18,72)$	$86,65(\pm 19,26)$	$80,83(\pm 17,94)$	$87,94(\pm 19,24)$	$89,91(\pm 18,54)$
breast-w(8)	$89,14(\pm 17,13)$	$91,08(\pm 16,99)$	$84,02(\pm 16,09)$	$90,74(\pm 17,01)$	$91,01(\pm 17,06)$
breast-w(16)	$93,18(\pm 3,27)$	$94,69(\pm 0,80)$	$89,03(\pm 2,50)$	$94,48(\pm 1,02)$	$94,75(\pm 1,26)$
diabetes(2)	$60,62(\pm 9,40)$	$59,23(\pm 10,48)$	$55,18(\pm 11,99)$	$59,14(\pm 9,82)$	$60,45(\pm 10,95)$
diabetes(4)	$62,92(\pm 6,01)$	$61,06(\pm 7,74)$	$62,11(\pm 7,59)$	$62,06(\pm 6,07)$	$64,93(\pm 7,30)$
diabetes(8)	$65,00(\pm 5,05)$	$66,24(\pm 4,76)$	$64,35(\pm 6,53)$	$65,40(\pm 5,78)$	$67,67(\pm 5,35)$
diabetes(16)	$66,07(\pm 3,93)$	$69,56(\pm 4,29)$	$66,53(\pm 4,14)$	$69,13(\pm 3,35)$	$69,61(\pm 2,78)$
ionos.(2)	$57,36(\pm 10,62)$	$56,35(\pm 9,96)$	$50,35(\pm 6,58)$	$58,12(\pm 12,21)$	$59,70(\pm 14,00)$
ionos.(4)	$70,08(\pm 10,59)$	$68,18(\pm 9,84)$	$66,20(\pm 12,61)$	$72,09(\pm 10,01)$	$71,51(\pm 11,49)$
ionos.(8)	$74,08(\pm 10,19)$	$79,93(\pm 7,62)$	$77,21(\pm 7,04)$	$79,24(\pm 7,42)$	$76,77(\pm 6,32)$
ionos.(16)	$77,92(\pm 15,98)$	$82,70(\pm 16,16)$	$82,09(\pm 16,19)$	$80,35(\pm 15,79)$	$76,90(\pm 15,21)$
anneal(2)	$76,71(\pm 4,54)$	$75,88(\pm 4,06)$	$74,87(\pm 7,55)$	$75,26(\pm 7,19)$	$70,35(\pm 10,35)$
anneal(4)	$79,74(\pm 3,91)$	$78,13(\pm 3,34)$	$78,51(\pm 4,95)$	$78,01(\pm 5,64)$	$72,33(\pm 9,63)$
anneal(8)	$85,19(\pm 3,00)$	$81,53(\pm 3,78)$	$82,59(\pm 5,21)$	$80,49(\pm 4,79)$	$77,56(\pm 6,68)$
anneal(16)	$90,91(\pm 1,46)$	$87,27(\pm 3,42)$	$89,84(\pm 4,53)$	$87,16(\pm 2,89)$	$83,45(\pm 3,88)$
remote(2)	$85,51(\pm 13,29)$	$65,40(\pm 13,46)$	$51,41(\pm 12,94)$	$83,52(\pm 15,28)$	$86,01(\pm 13,21)$
remote(4)	$95,59(\pm 6,70)$	$79,63(\pm 11,16)$	$72,96(\pm 8,74)$	$94,07(\pm 8,14)$	$95,67(\pm 6,64)$
remote(8)	$98,98(\pm 1,55)$	$94,10(\pm 7,04)$	$84,39(\pm 5,93)$	$98,40(\pm 1,41)$	$99,27(\pm 0,72)$
remote(16)	$99,70(\pm 0,75)$	$98,81(\pm 2,20)$	$88,44(\pm 4,94)$	$99,18(\pm 1,57)$	$99,40(\pm 1,13)$

TAB. 2 – *Précision des résultats - partie I*

	CLM	RC	SCEC	SRIDHCR	SK
iris(2)	64, 15(±14, 09)	64, 54(±13, 67)	72, 46(±12, 89)	73, 38(±15, 32)	64, 97(±13, 85)
iris(4)	74, 92(±14, 48)	74, 86(±14, 12)	81, 16(±12, 52)	83, 43(±13, 04)	74, 60(±14, 39)
iris(8)	91, 63(±5, 18)	90, 26(±6, 74)	87, 58(±8, 47)	85, 62(±7, 58)	89, 47(±7, 65)
iris(16)	93, 21(±4, 97)	92, 84(±4, 96)	91, 97(±7, 41)	92, 09(±4, 94)	92, 59(±6, 12)
wine(2)	55, 82(±14, 58)	56, 18(±14, 87)	73, 65(±15, 86)	78, 47(±15, 64)	54, 73(±13, 55)
wine(4)	82, 98(±13, 65)	82, 98(±13, 33)	86, 45(±9, 73)	86, 53(±12, 49)	82, 29(±13, 11)
wine(8)	89, 84(±9, 12)	89, 53(±8, 72)	91, 28(±5, 72)	92, 46(±3, 60)	89, 43(±8, 58)
wine(16)	94, 87(±3, 79)	94, 55(±4, 11)	93, 25(±5, 65)	93, 41(±5, 31)	94, 22(±3, 90)
breast-w(2)	79, 44(±16, 79)	79, 48(±16, 76)	72, 59(±29, 74)	73, 56(±29, 98)	71, 91(±28, 48)
breast-w(4)	87, 44(±19, 62)	87, 60(±19, 59)	82, 76(±25, 13)	80, 89(±25, 21)	84, 06(±24, 94)
breast-w(8)	91, 27(±17, 03)	91, 12(±17, 00)	88, 37(±17, 01)	88, 11(±17, 09)	90, 90(±16, 98)
breast-w(16)	94, 62(±1, 28)	94, 67(±1, 35)	90, 36(±16, 86)	89, 73(±16, 88)	91, 58(±17, 05)
diabetes(2)	58, 61(±11, 21)	58, 38(±11, 31)	60, 43(±8, 83)	59, 35(±11, 51)	58, 14(±11, 44)
diabetes(4)	59, 09(±9, 88)	58, 02(±10, 14)	60, 03(±10, 02)	58, 60(±10, 58)	57, 53(±10, 36)
diabetes(8)	66, 15(±5, 70)	65, 73(±5, 10)	62, 41(±13, 10)	60, 41(±13, 02)	62, 88(±12, 63)
diabetes(16)	68, 76(±4, 81)	67, 79(±6, 31)	63, 90(±13, 45)	61, 63(±12, 31)	65, 03(±13, 64)
ionos.(2)	55, 96(±11, 46)	55, 51(±10, 72)	48, 09(±21, 33)	47, 91(±22, 05)	48, 07(±21, 88)
ionos.(4)	65, 74(±14, 88)	65, 40(±15, 53)	59, 70(±11, 03)	63, 95(±12, 40)	65, 88(±16, 11)
ionos.(8)	72, 74(±8, 33)	71, 34(±9, 76)	60, 10(±19, 42)	67, 08(±21, 45)	66, 75(±20, 40)
ionos.(16)	75, 78(±15, 01)	74, 10(±15, 33)	68, 11(±16, 77)	74, 98(±16, 51)	73, 14(±15, 28)
anneal(2)	71, 51(±9, 97)	69, 97(±10, 34)	71, 15(±10, 97)	68, 49(±11, 98)	69, 89(±10, 01)
anneal(4)	76, 40(±4, 25)	75, 10(±4, 33)	75, 92(±5, 59)	74, 32(±7, 81)	74, 06(±4, 68)
anneal(8)	78, 24(±6, 03)	76, 89(±6, 25)	79, 23(±5, 38)	76, 11(±7, 38)	75, 66(±6, 64)
anneal(16)	84, 06(±3, 88)	81, 80(±5, 60)	82, 89(±3, 75)	80, 99(±4, 72)	80, 21(±6, 66)
remote(2)	68, 65(±14, 41)	68, 73(±14, 46)	73, 21(±13, 37)	79, 73(±15, 25)	67, 89(±14, 04)
remote(4)	80, 94(±10, 64)	81, 07(±10, 15)	86, 95(±11, 07)	93, 90(±7, 69)	80, 04(±9, 25)
remote(8)	94, 97(±6, 54)	94, 87(±6, 45)	93, 47(±7, 30)	98, 30(±2, 42)	93, 96(±7, 23)
remote(16)	98, 22(±2, 83)	98, 29(±2, 48)	99, 48(±0, 94)	98, 74(±2, 41)	97, 85(±2, 78)

TAB. 3 – *Précision des résultats - partie II*

	CK	Simple+	Low+	Medium+	High+
iris(2)	65, 31(±14, 01)	68, 01(±15, 49)	68, 11(±15, 43)	69, 08(±15, 38)	68, 40(±15, 53)
iris(4)	74, 44(±14, 57)	76, 19(±13, 63)	75, 71(±13, 51)	75, 82(±13, 57)	75, 82(±13, 54)
iris(8)	88, 69(±8, 32)	93, 98(±3, 43)	93, 13(±3, 74)	92, 35(±3, 96)	92, 48(±3, 54)
iris(16)	92, 59(±6, 48)	94, 32(±4, 56)	92, 84(±4, 27)	91, 11(±4, 83)	92, 34(±4, 58)
wine(2)	54, 13(±13, 27)	54, 90(±14, 32)	54, 98(±14, 30)	55, 18(±14, 34)	54, 90(±14, 28)
wine(4)	82, 20(±13, 13)	78, 31(±12, 04)	78, 44(±12, 03)	79, 35(±11, 86)	78, 78(±11, 87)
wine(8)	89, 07(±8, 52)	89, 69(±8, 34)	90, 30(±7, 84)	90, 76(±7, 86)	90, 05(±7, 83)
wine(16)	94, 30(±4, 04)	95, 61(±3, 69)	95, 36(±3, 90)	95, 28(±4, 17)	95, 20(±4, 05)
breast-w(2)	71, 87(±28, 44)	75, 01(±28, 58)	75, 04(±28, 59)	75, 51(±28, 60)	75, 03(±28, 58)
breast-w(4)	84, 09(±24, 95)	83, 86(±24, 75)	83, 92(±24, 74)	84, 42(±24, 67)	83, 95(±24, 74)
breast-w(8)	90, 76(±16, 97)	91, 22(±17, 02)	91, 17(±17, 01)	91, 26(±17, 02)	91, 11(±17, 00)
breast-w(16)	91, 50(±17, 04)	91, 66(±17, 04)	91, 58(±17, 03)	91, 57(±17, 03)	91, 63(±17, 04)
diabetes(2)	58, 00(±11, 56)	59, 19(±10, 47)	59, 22(±10, 46)	59, 31(±10, 36)	59, 19(±10, 49)
diabetes(4)	57, 35(±10, 38)	61, 30(±7, 84)	61, 30(±7, 80)	61, 41(±8, 08)	61, 37(±7, 93)
diabetes(8)	62, 07(±12, 42)	64, 25(±12, 68)	64, 23(±12, 73)	63, 97(±12, 77)	64, 07(±12, 80)
diabetes(16)	64, 08(±13, 52)	67, 50(±13, 13)	67, 37(±13, 13)	66, 79(±13, 00)	67, 33(±13, 10)
ionos.(2)	47, 87(±21, 70)	48, 44(±21, 29)	48, 44(±21, 31)	48, 63(±21, 45)	48, 49(±21, 33)
ionos.(4)	65, 86(±16, 38)	68, 32(±9, 88)	68, 36(±9, 94)	69, 16(±9, 94)	68, 40(±9, 94)
ionos.(8)	66, 52(±20, 43)	75, 24(±21, 30)	75, 34(±21, 22)	75, 13(±21, 17)	75, 26(±21, 23)
ionos.(16)	72, 49(±15, 25)	82, 68(±16, 14)	82, 54(±16, 04)	80, 76(±15, 62)	82, 35(±15, 96)
anneal(2)	69, 60(±10, 21)	76, 03(±3, 96)	75, 91(±4, 00)	75, 80(±3, 86)	75, 64(±3, 78)
anneal(4)	73, 13(±5, 49)	78, 12(±3, 35)	78, 07(±3, 41)	77, 78(±3, 39)	77, 04(±3, 82)
anneal(8)	74, 84(±6, 61)	81, 54(±3, 92)	81, 30(±4, 16)	80, 84(±4, 13)	79, 44(±5, 03)
anneal(16)	79, 14(±7, 09)	87, 17(±3, 26)	86, 89(±3, 38)	85, 84(±3, 27)	83, 80(±4, 48)
remote(2)	67, 39(±13, 91)	65, 40(±13, 46)	65, 40(±13, 46)	65, 47(±13, 32)	65, 44(±13, 39)
remote(4)	79, 54(±8, 88)	79, 58(±11, 10)	79, 67(±11, 16)	79, 67(±11, 16)	79, 67(±11, 16)
remote(8)	93, 43(±7, 51)	94, 15(±7, 05)	94, 15(±7, 04)	94, 30(±7, 08)	94, 20(±7, 01)
remote(16)	97, 77(±3, 09)	98, 81(±2, 20)	98, 81(±2, 12)	98, 81(±2, 12)	98, 88(±2, 12)

TAB. 4 – *Précision des résultats - partie III*

	Simple	Low	Medium	High
iris(2)	$78,11(\pm14,73)$	$80,29(\pm12,99)$	$85,70(\pm8,03)$	$82,31(\pm11,33)$
iris(4)	$83,17(\pm16,13)$	$84,97(\pm14,41)$	$86,24(\pm9,46)$	$81,69(\pm12,19)$
iris(8)	$88,95(\pm8,31)$	$89,34(\pm5,42)$	$90,06(\pm4,17)$	$85,88(\pm10,43)$
iris(16)	$90,74(\pm5,12)$	$88,76(\pm5,36)$	$87,90(\pm5,14)$	$86,91(\pm7,01)$
wine(2)	$84,41(\pm14,92)$	$84,90(\pm14,70)$	$91,88(\pm9,84)$	$84,05(\pm13,73)$
wine(4)	$93,85(\pm5,84)$	$94,97(\pm2,16)$	$95,02(\pm2,09)$	$88,13(\pm11,21)$
wine(8)	$95,12(\pm3,25)$	$95,07(\pm2,84)$	$94,66(\pm3,48)$	$91,69(\pm6,20)$
wine(16)	$95,77(\pm3,66)$	$95,36(\pm3,52)$	$95,61(\pm3,52)$	$91,38(\pm8,00)$
breast-w(2)	$73,84(\pm31,14)$	$82,87(\pm28,80)$	$84,47(\pm28,17)$	$79,89(\pm30,05)$
breast-w(4)	$86,79(\pm24,45)$	$87,58(\pm23,43)$	$88,10(\pm23,57)$	$86,56(\pm24,48)$
breast-w(8)	$91,50(\pm17,02)$	$91,24(\pm16,98)$	$91,14(\pm16,96)$	$89,69(\pm18,65)$
breast-w(16)	$91,08(\pm16,93)$	$90,85(\pm16,88)$	$90,84(\pm16,88)$	$87,84(\pm19,44)$
diabetes(2)	$60,16(\pm10,44)$	$60,43(\pm10,58)$	$59,31(\pm11,15)$	$59,49(\pm10,81)$
diabetes(4)	$60,74(\pm9,73)$	$59,43(\pm11,30)$	$59,32(\pm11,30)$	$59,05(\pm12,00)$
diabetes(8)	$61,22(\pm13,06)$	$61,69(\pm13,07)$	$60,38(\pm14,04)$	$61,42(\pm13,04)$
diabetes(16)	$62,41(\pm11,73)$	$63,12(\pm12,20)$	$62,79(\pm12,25)$	$62,73(\pm12,16)$
ionos.(2)	$51,54(\pm24,04)$	$56,55(\pm25,86)$	$61,69(\pm26,44)$	$57,56(\pm26,34)$
ionos.(4)	$72,57(\pm9,77)$	$72,69(\pm8,26)$	$73,75(\pm9,18)$	$70,39(\pm9,57)$
ionos.(8)	$67,73(\pm20,71)$	$68,70(\pm19,75)$	$71,53(\pm19,42)$	$67,61(\pm20,00)$
ionos.(16)	$74,07(\pm14,92)$	$71,84(\pm13,70)$	$72,37(\pm13,64)$	$69,86(\pm15,17)$
anneal(2)	$76,51(\pm4,17)$	$74,65(\pm6,08)$	$69,10(\pm8,30)$	$60,49(\pm8,47)$
anneal(4)	$76,80(\pm4,11)$	$75,59(\pm5,57)$	$71,55(\pm7,85)$	$60,82(\pm12,52)$
anneal(8)	$78,47(\pm4,10)$	$77,51(\pm5,29)$	$75,13(\pm5,56)$	$63,49(\pm8,20)$
anneal(16)	$80,20(\pm3,44)$	$79,66(\pm4,01)$	$75,55(\pm6,27)$	$62,84(\pm9,66)$
remote(2)	$83,60(\pm14,66)$	$86,89(\pm10,80)$	$93,06(\pm8,31)$	$88,39(\pm12,85)$
remote(4)	$88,47(\pm7,63)$	$93,04(\pm8,06)$	$95,10(\pm5,01)$	$94,81(\pm5,74)$
remote(8)	$93,52(\pm4,32)$	$96,03(\pm4,14)$	$95,89(\pm2,24)$	$93,91(\pm5,79)$
remote(16)	$94,07(\pm4,88)$	$96,88(\pm2,33)$	$96,88(\pm2,19)$	$97,11(\pm2,51)$

TAB. 5 – *Précision des résultats - partie IV*

données	1	2	3	Données enrichies	Données d'origine	↗
iris(2)	**85.70(Medium)**	**82.320(High)**	78,02(1-NN)	85.70(Medium)	78,02(1-NN)	+
iris(4)	88.84(DL)	**86.24(Medium)**	85,61(1-NN)	88.84(DL)	85,61(1-NN)	+
iris(8)	**93.99(Simple+)**	93,59(NB)	93,59(DL)	93.99(Simple+)	93,59(NB)	+
iris(16)	**94.32(Simple+)**	94.20(DL)	94.20(NB)	94.32(Simple+)	94.20(NB)	+
wine(2)	**91.89(Medium)**	**84.90(Low)**	**84.42(Simple)**	91.89(Medium)	76.67(1-NN)	+
wine(4)	**95.02(Medium)**	**94.98(Low)**	**93.85(Simple)**	95.022(Medium)	88.92(1-NN)	+
wine(8)	**95.13(Simple)**	**95.08(Low)**	94.77(DL)	95.13(Simple)	91.95(1-NN)	+
wine(16)	96.10(DL)	**95.77(Simple)**	**95.61(Medium)**	96.10(DL)	95.53(NB)	+
breast-w(2)	85.23(SL)	84.72(DL)	**84.47(Medium)**	85.23(SL)	82.40(NB)	+
breast-w(4)	89.91(DL)	**88.10(Medium)**	**87.59(Low)**	89.91(DL)	87.09(1-NN)	+
breast-w(8)	**91.50(Simple)**	**91.26(Medium+)**	**91.24(Low)**	91.50(Simple)	91.08 (NB)	+
breast-w(16)	94.75(DL)	94.68(RC)	**91.66(Simple+)**	94.75(DL)	94.69 (NB)	+
diabetes(2)	60.62(1-NN)	60.46(DL)	60.44(SCEC)	60.46(DL)	60.62(1-NN)	-
diabetes(4)	64.94(DL)	62.93(1-NN)	62.24(C45)	64.94(DL)	62.93(1-NN)	+
diabetes(8)	67.67(DL)	66.24(NB)	66.16(CLM)	67.67(DL)	66.24(NB)	+
diabetes(16)	69.61(DL)	69.56(NB)	69.14(SL)	69.61(DL)	69.56(NB)	+
ionos.(2)	**61.70(Medium)**	59.71(DL)	58.59(SL)	61.70(Medium)	57.37(1-NN)	+
ionos.(4)	**73.75(Medium)**	**72.69(Low)**	**72.67(Simple)**	73.75(Medium)	70.08(1-NN)	+
ionos.(8)	79.94(NB)	79.24(SL)	77.21(C45)	79.24(SL)	79.94(NB)	-
ionos.(16)	82.70(NB)	**82.68(Simple+)**	**82.54(Low+)**	82.68(Simple+)	82.70(NB)	-
anneal(2)	76.71(1-NN)	**76.51(Simple)**	**76.03(Simple+)**	76.51(Simple)	76.71(1-NN)	-
anneal(4)	79.75(1-NN)	78.51(C45)	78.13(NB)	78.12(Simple+)	78.13(NB)	-
anneal(8)	85.19(1-NN)	82.59(C45)	**81.55(Simple+)**	81.55(Simple+)	85.19(1-NN)	-
anneal(16)	90.92(1-NN)	89.85(C45)	87.27(NB)	87.18(Simple+)	90.92(1-NN)	-
remote(2)	**93.06(Medium)**	**88.39(High)**	**86.90(Low)**	93.06(Medium)	85.52(1-NN)	+
remote(4)	95.68(DL)	95.60(1-NN)	**95.10(Medium)**	95.68(DL)	95.60(1-NN)	+
remote(8)	99.27(DL)	98.99(1-NN)	98.41(SL)	99.27(DL)	98.99(1-NN)	-
remote(16)	99.70(1-NN)	99.48(SCEC)	99.41(DL)	99.41(DL)	99.70(1-NN)	-

TAB. 6 – *Les trois meilleurs résultats pour les expériences sur les différents jeux de données.*

Apprentissage semi-supervisé enrichi par de multiples clusterings.

4.1.1 Protocole expérimental

Pour chaque expérience, les données ont été découpés en deux sous-ensembles, composés chacun de 50% des objets. Le premier ensemble est utilisé comme ensemble de données non-étiquetées pour effectuer le clustering dans les méthodes semi-supervisées. Le second ensemble permet de choisir quelques exemples étiquetés, considérés comme les connaissances disponibles. Le reste du second ensemble est utilisé pour évaluer la méthode (i.e. le calcul de la précision). Nous avons choisi d'évaluer les méthodes avec 2, 4, 8 et 16 exemples par classe.

Par exemple, pour le premier jeu de données (*iris*) qui contient trois classes, nous avons testé la méthode avec 6, 12, 24 et 48 objets étiquetés. Comme le nombre d'exemple est très faible, la qualité du résultat dépend fortement de leur sélection. C'est pourquoi, nous avons répété les expériences 30 fois et les résultats ont été moyennés. À chaque exécution, le jeu de données est coupé aléatoirement en deux.

Pour l'application de notre méthode SLEMC (section 3.4), nous avons tout d'abord choisi le nombre de clusterings qui seraient exécutés (i.e. combien d'attributs auraient les objets dans le nouvel espace des données), puis quelles méthodes allaient être appliquées. Nous avons défini quatre configurations différentes :

1. *Simple* : un algorithme EM (Expectation-Maximization (Dempster et al., 1977))
2. *Low* : un algorithme EM et un algorithme K-Means (MacQueen, 1967)
3. *Medium* : deux algorithmes EM et deux algorithmes K-Means
4. *High* : c algorithmes EM et c algorithmes K-Means (avec c le nombre de classes de l'ensemble de données).

De plus, nous avons testé une variante de la méthode SLEMC, qui conserve les descriptions initiales des objets et ajoutent les nouveaux attributs. Ces configurations sont appelées par la suite *Simple+*, *Low+*, *Medium+* et *High+*.

Chaque méthode a été lancée avec un nombre de clusters égal au nombre de classes réellement présentes dans le jeu de données, à l'exception de la configuration *High* pour laquelle les méthodes de clustering avait chacune k clusters, $k \in \{2, \ldots, c\}$, choisi aléatoirement.

Afin de faire un étude complète, nous avons aussi exécuté des méthodes de classification supervisées classiques sur les jeux de données testés. Pour chaque configuration, les exemples disponibles ont été utilisés et les données non-étiquetées ignorées. Nous avons choisi plusieurs algorithmes de différents types : arbre de décision (C4.5), bayésien naïf (NB) et un 1-plus proche voisin (1-NN).

4.1.2 Résultats obtenus

Les résultats pour chaque jeu de données et pour chaque expériences sont présentés dans les tableaux 2, 3, 4 et 5, où les valeurs présentées sont les moyennes et écarts-types des précisions sur les 30 exécutions pour chaque configuration et pour chaque jeu de données. Pour mémoire, le nombre d'exemples utilisé est indiqué au début de chaque ligne.

Le tableau 6 résume pour plus de lisibilité les quatre tableaux précédents en présentant pour chaque configuration et chaque jeu de données, les trois meilleures méthodes. Dans ce tableau, nous pouvons observer que la méthode proposée, SLEMC, donne de meilleurs résultats que les méthodes supervisées et les autres approches semi-supervisées lorsque le nombre d'exemples est très faible (2 ou 4 exemples par classe).

Sur les données *iris* et *wine* nos méthodes sont en première position pour 3 des 4 configurations. Sur les données *wine*, nos méthodes sont présentes 10 fois sur 12 parmi les trois meilleurs résultats en considérant toutes les configurations. Pour certaines configurations sur certains jeux de données, la différence de précision entre nos méthodes et les autres méthodes semi-supervisées semble importante. Par exemple, pour la configuration (2) sur le jeu de données *wine*, notre méthode avec la configuration *Medium* obtient une précision de 91.888% alors que DL n'obtient que 80.924% en précision. Cela apparaît aussi sur le jeu de données *remote* pour la configuration (2) car notre méthodes obtient une précision de 93.065% alors que la méthode DL obtient uniquement 86.015%.

Concernant les différentes configurations que nous avons proposés pour SLEMC, la configuration *Medium* (deux EM et deux KMeans) semble donner les meilleurs résultats puisqu'elle apparaît 10 fois sur l'ensemble du tableau présentant les trois meilleurs résultats. Cela renforce notre intuition qu'ajouter plus de clusterings améliore le résultat final, car les objets sont décrits avec plus de détails (i.e. avec plus d'attributs). Cependant, la méthode ne semble pas tirer profit de la variation du nombre de clusters, car la configuration *High* ne parvient pas à être plus performante que la configuration *Medium* qui elle utilise un nombre constant de clusters. Cela peut s'expliquer par le fait que les jeux de données utilisés montrent généralement une relative correspondance entre les classes et les clusters.

Nous observons aussi que dans la plupart des cas, les configurations qui ne conservent pas la description des objets (*Low*, *Simple*, *Medium* et *High*) donnent de bons résultats lorsque le nombre d'objets étiquetés est très faible (2 et 4), alors que les autres configurations conservant les anciens attributs (*Low+*, *Simple+*, *Medium+* et *High+*) donnent de meilleurs résultats quand le nombre d'exemples est plus grand (8 et 16). Cela peut s'expliquer en observant que quand le nombre d'exemples est bas, leur description est trop faible (manque d'information) pour construire un classifieur performant. Cependant, comme les attributs ajoutés par les clusterings sont calculés à partir de 50% des données, les descriptions enrichies comportent plus d'information. Quand le nombre d'exemples augmente, la description des objets est plus significative et plus précise que l'information brute apportée par les clusterings.

Comme indiqué dans l'introduction, si l'espace des données n'est pas corrélé avec l'information de classe, l'utilisation d'un ou plusieurs clusterings est inutile. Cette affirmation est confirmée par les résultats obtenus sur les jeux de données $anneal$ et $diabetes$, pour lesquels les résultats des approches semi-supervisées sont inférieurs à ceux du 1-plus proche voisin, bien que le nombre d'exemples soit très faible.

Parmi les autres méthodes semi-supervisées, DL a donné des résultats particulièrement bons, puisqu'il est présent 16 fois dans le tableau 6. En second vient la méthode SL qui est présente 4 fois. Ces résultats mettent en avant que les techniques de pré-étiquetage semblent être plus performantes que les autres, notamment sur ce type de données.

Nous avons aussi réalisé une expérience pour étudier l'importance de la quantité de données non-étiquetées dans nos méthodes. À nouveau, nous avons utilisé 50% des données comme données non-étiquetées (comme précédemment), mais aussi réalisé la même expérience avec 25% et 10% des données. Les tests ont été effectués sur le jeu de données *iris* et avec 2, 4 et 8 exemples par classe. Les résultats sont présentés sur la Figure 1 sur laquelle chaque courbe représente l'évolution de la précision en fonction de la quantité de données non-étiquetées disponibles. Comme attendu, la méthode tire profit de l'ensemble des données non-étiquetées, puisque l'on remarque aisément que la qualité des résultats augmentent en

fonction du nombre d'objets étiquetés disponibles.

4.2 Résultats sur une image de télédétection

Comme présenté en introduction, dans le domaine de la télédétection, le problème du faible rapport entre le nombre d'attributs et le nombre d'exemples est très important lorsque l'on traite des images hyperspectrales ou à très haute résolution spatiale. En fait, dans ce dernier cas, la classification est réalisée en trois étapes. Tout d'abord, une segmentation de l'image est réalisée, qui produit un ensemble de régions (i.e. groupes de pixels homogènes). Ensuite, ces régions sont caractérisées par plusieurs attributs (spectraux, géométriques, spatiaux, etc.). Finalement, c'est ce nouvel ensemble de données qu'il faut classer, qui se compose généralement de peu de régions (par rapport au nombre de pixels dans l'image) mais décrites par beaucoup d'attributs.

Nous présentons dans cette section des résultats obtenus sur la classification d'un extrait d'une image à très haute résolution d'une zone urbaine de la ville de Strasbourg (France) fournie par le laboratoire Image, Ville, Environnement de l'Université de Strasbourg (ERL 7230). L'image traitée est une image Quickbird (pansharpening) avec 4 bandes spectrales et une résolution spatiale de 0.7 m par pixel. L'extrait est visible sur la Figure 2.

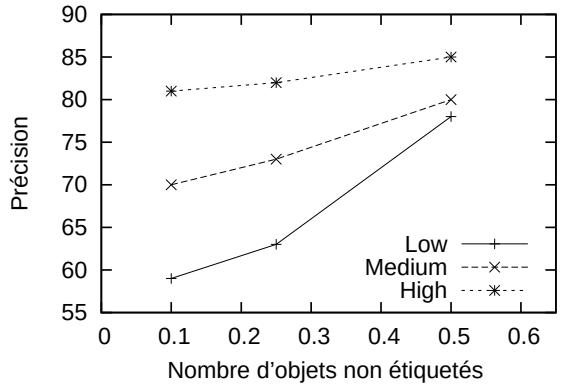
Après avoir réalisé une segmentation de cette image à l'aide de l'algorithme de ligne de partage des eaux Vincent et Soille (1991), chaque région a été caractérisée par les attributs ci-dessous :

- 8 attributs représentant la moyenne et l'écart-type des valeurs de chaque bande spectrale pour pixels composant la région ;
- 8 attributs représentant la moyenne et l'écart-type des valeurs normalisées de chaque bande spectrale ;
- 2 attributs représentant la moyenne et l'écart-type de la moyenne de toutes les bandes spectrales ;
- 2 attributs calculés comme la moyenne et l'écart-type de l'indice de végétation NDVI (*Normalized Difference Vegetation Index*) des pixels composant la région ;
- 1 attribut représentant l'aire couverte par la région ;
- 1 attribut calculé comme l'élongation de la forme géométrique représentant la région ;
- 1 attribut mesurant l'adéquation entre la forme géométrique représentant la région et sa boîte englobante orientée.

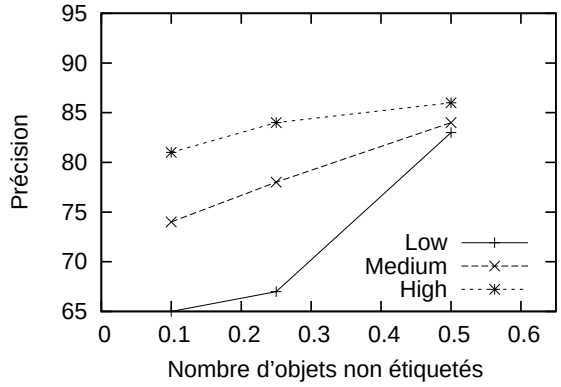
Finalement, les données générées étaient composées de 186 objets décrits par 23 attributs, et représentés par 3 classes. Comme dans les expériences précédentes, nous avons testé trois configurations pour étudier l'importance du nombre de clusterings. Les trois configurations sont *Low* (deux EM), *Medium* (deux EM et deux K-Means) et *High* (6 EM et 6 K-Means). Nous avons aussi comparé les résultats avec trois méthodes classiques de classification supervisée (C4.5, 1-plus proche voisin, bayésien naïf), et les autres méthodes semi-supervisées présentées dans la section 3. À nouveau, les tests ont été réalisés 30 fois et les résultats moyennés. Pour chaque expérience, les exemples sont tirés au hasard.

Les résultats obtenus avec différents nombre d'exemples sont donnés dans les tableaux 2, 3, 4 et 5, et les trois meilleurs résultats sont présentés dans la tableau 6 (jeu de données *remote*, dernière ligne).

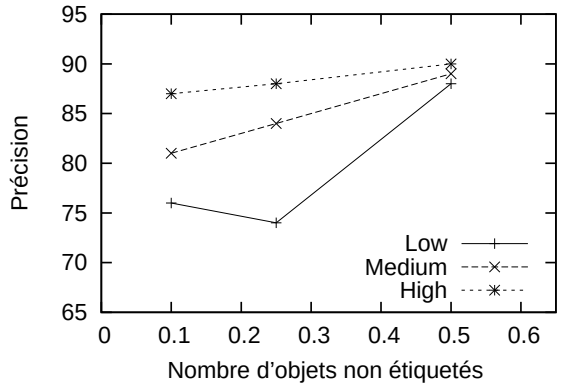
Ces résultats confirment que la configuration *Medium* est la plus performante car elle parvient à donner les meilleurs résultats lorsque très d'exemples sont disponibles. Cela confirme



(a) 2 exemples étiquetés par classe.



(b) 4 exemples étiquetés par classe.



(c) 8 exemples étiquetés par classe.

FIG. 1 – *Expériences sur* iris *avec un nombre variable de nombre d'objets non-étiquetés.*

FIG. 2 – *Extrait d'une image Quickbird© image [Strasbourg, France - 900×900 pixels - 0.7m per pixel - 2001]*

aussi qu'un nombre minimal d'exemples est nécessaire pour obtenir des classifications supervisées de bonne qualité, mais que ce manque peut être partiellement rattrapé par l'utilisation du clustering sur les données non-étiquetées.

5 Conclusion

Dans cet article, nous avons présenté comment plusieurs résultats de clustering pouvaient être combinés à un classifieur supervisé, afin d'améliorer sa précision lorsque peu d'exemples sont disponibles. La méthode présentée a montré de bons résultats quand le nombre d'exemples étiquetés est effectivement très faible et que des données non-étiquetées sont disponibles.

Les expériences conduites nous donnent un aperçu de l'amélioration que l'on peut attendre du fait d'incorporer via un clustering les données non-étiquetées lors de l'apprentissage supervisé. De plus,une expérience sur des données réelles montrent l'intérêt de développer de telles approches dans le domaine de l'analyse automatique d'images par exemple.

Cependant, certaines questions restent ouvertes. Comment choisir efficacement les clusterings à utiliser en fonction d'un jeu de données spécifique ? Est-il plus pertinent de générer de la diversité parmi eux comme c'est le cas en ensemble clustering ? Comment savoir qu'un jeu de données est adapté à l'usage de ce type de méthode ? Ces questions nous donnent quelques directions de recherche afin d'essayer d'améliorer le travail présenté ici.

Références

Basu, S., A. Banerjee, et R. J. Mooney (2002). Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, pp. 27–34.

Bennett, K. P., A. Demiriz, et R. Maclin (2002). Exploiting unlabeled data in ensemble methods. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 289–296.

Blum, A. et T. Mitchell (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, pp. 92–100.

Bouchachia, A. (2007). Learning with partly labeled data. *Neural Computing and Applications 16*(3), 267–293.

Cai, W., S. Chen, et D. Zhang (2009). A simultaneous learning framework for clustering and classification. *Pattern Recognition 42*(7), 1248–1286.

Chawla, N. V. et G. J. Karakoulas (2005). Learning from labeled and unlabeled data : An empirical study across techniques and domains. *Journal of Artificial Intelligence Research 23*, 331–366.

Dempster, A. P., N. M. Laird, et D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39*(1), 1–38.

Deodhar, M. et J. Ghosh (2007). A framework for simultaneous co-clustering and learning from complex data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 250–259.

Eick, C., N. Zeidat, et Z. Zhao (2004). Supervised clustering–algorithms and benefits. In *IEEE International Conference on Tools with Artificial Intelligence*, pp. 774–776.

Gabrys, B. et L. Petrakieva (2004). Combining labelled and unlabelled data in the design of pattern classification systems. *International journal of approximate reasoning 35*(3), 251–273.

Ghahramani, Z. et M. I. Jordan (1994). Supervised learning from incomplete data via an em approach. In *Advances in Neural Information Processing Systems 6*, pp. 120–127. Morgan Kaufmann.

Goldman, S. et Y. Zhou (2000). Enhancing supervised learning with unlabeled data. In *International Conference on Machine Learning*, pp. 327–334.

Hoffbeck, J. P. et D. A. Landgrebe (1996). Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(7), 763–767.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory 14*(1), 5 – 63.

Jia, X. et R. J.A. (2002). Cluster-space representation for hyperspectral data classification. *IEEE Transactions on Geoscience and Remote Sensing 40*(3), 593–598.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning*, pp. 200–209.

Kothari, R. et V. Jain (2003). Learning from labeled and unlabeled data using a minimal number of queries. *IEEE Transactions on Neural Networks 14*(6), 1496–1505.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Morgan, J. T., J. Ham, M. M. Crawford, A. Henneguelle, et J. Ghosh (2004). Adaptive feature spaces for land cover classification with limited ground truth data. *International Journal of Pattern Recognition and Artificial Intelligence 18*(5), 777–799.

Newman, D., S. Hettich, C. Blake, et C. Merz (1998). UCI repository of machine learning databases.

Nigam, K., A. K. McCallum, S. Thrun, et T. Mitchell (2000a). Text classification from labeled and unlabeled documents using EM. *Machine Learning 39*(2/3), 103–134.

Nigam, K., A. K. McCallum, S. Thrun, et T. M. Mitchell (2000b). Text classification from labeled and unlabeled documents using EM. *Machine Learning 39*(2/3), 103–134.

Raskutti, B., H. L. Ferrá, et A. Kowalczyk (2002). Combining clustering and co-training to enhance text classification using unlabelled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 620–625.

Seeger, M. (2002). Learning with labeled and unlabeled data. Technical report, University of Edinburgh.

Shahshahani, B. M. et D. Landgrebe (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing 32*(5), 1087–1095.

Vincent, L. et P. Soille (1991). Watersheds in digital spaces : An efficient algorithm based on immersion simulations. *IEEE Pattern Analysis and Machine Intelligence 13*(6), 583–598.

Zhou, Z.-H., D.-C. Zhan, et Q. Yang (2007). Semi-supervised learning with very few labeled training examples. In *AAAI International Conference on Artificial Intelligence*, pp. 675–680.

Summary

The supervised classification task involves inducing a predictive model using a set of labeled samples. The accuracy of the model usually increases as more labeled samples are available. When only few samples are available for the learning task, the resulting model generally provides poor results. When labeled samples are difficult to get, unlabeled samples are, on the contrary, generally easily available. When both types of data are available, semi-supervised learning approaches can be adopted to leverage from the labeled and the unlabeled data. In this paper we focus on the case where the number of labeled samples is very limited. As the labeling task is costly and time consuming, users generally provide a very few number of labeled objects. Thus, designing approaches able to leverage from very limited number of labeled samples is a challeging issue.

In this article, we review and compare different semi-supervised learning algorithms and we also introduce an innovative way to combine supervised and unsupervised learning in order to use both labeled and unlabeled samples. The proposed method uses clustering results to produce extra features used in a supervised learning step. The efficiency of the method is evaluated on various UCI datasets and on the classification of a very high resolution remote sensing image, when the number of labeled samples is very low. The experiments provide some insights on how combining labeled and unlabled data can be usefull in pattern recognition.