Classification faiblement supervisée : arbre de décision probabiliste et apprentissage itératif

Riwal Lefort*,** Ronan Fablet**
Jean-Marc Boucher**

*Ifremer/STH, Technopole Brest Iroise - 29280 Plouzane, France
http://www.ifremer.fr
**Telecom Bretagne/LabSTICC, Technopol Brest Iroise
CS83818, 29238 Brest Cedex, France
riwal.lefort@telecom-bretagne.eu
http://www.telecom-bretagne.eu

Résumé. Dans le domaine de la fouille de données, il existe plusieurs types de modèles de classification qui dépendent de la complexité de l'ensemble d'apprentissage. Ce papier traite de la classification faiblement supervisée pour laquelle l'ensemble d'apprentissage est constitué de données de labels inconnus mais dont les probabilités de classification *a priori* sont connues. Premièrement, nous proposons une méthode pour apprendre des arbres de décision à l'aide des probabilités de classification *a priori*. Deuxièmement, une procédure itérative est proposée pour modifier les labels des données d'apprentissage, le but étant que les *a priori* faibles convergent vers des valeurs binaires, et donc vers un *a priori* fort. Les méthodes proposées sont évaluées sur des jeux de données issus de la base de données UCI, puis nous proposons d'appliquer ces méthodes d'apprentissage dans le cadre de l'acoustique halieutique.

1 Introduction

Dans le domaine de la fouille de données, de nombreuses applications nécessitent le développement de modèles de classification stables et robustes. On peut citer par exemple, la reconnaissance d'objets (Crandall and Huttenlocher, 2006), la reconnaissance de texture d'images (Lazebnik et al., 2005), la reconnaissance d'évènement dans des vidéos (Hongeng et al., 2004), ou encore l'analyse de scènes dans des images (Torralba, 2003). Ce type de problème correspond à deux étapes principales : la définition d'un vecteur de descripteurs qui décrit l'objet considéré et le développement d'un modèle de classification pour les objets considérés. La seconde étape requière une phase d'apprentissage dont le procédé dépend des caractéristiques de l'ensemble d'apprentissage. Par exemple, en classification semi-supervisée (Chapelle et al., 2006), l'ensemble d'apprentissage est constitué de quelques données labélisées, complétées par un ensemble conséquent d'exemples sans label. Pour certaines applications, les performances de classification sont alors identiques au cas de l'apprentissage supervisé.

De manière plus générale, ce papier traite de l'apprentissage faiblement supervisé qui inclue à la fois l'apprentissage supervisé, l'apprentissage semi-supervisé, et l'apprentissage non

supervisé. L'idée est que chaque exemple de l'ensemble d'apprentissage est associé à un vecteur de probabilité qui donne l'*a priori* de chaque classe et qui constitue la seule connaissance relative aux exemples. Soit $\{x_n, \pi_n\}_n$ l'ensemble d'apprentissage, où x_n contient les descripteurs du n^{eme} exemple d'apprentissage et $\pi_n = \{\pi_{ni}\}_i$ est le vecteur qui donne les probabilités *a priori* pour x_n, i indexant la classe. Ainsi, le cas de l'apprentissage supervisé correspond à $\pi_{ni} = 0$ si l'exemple n n'est pas de la classe i et $\pi_{ni} = 1$ sinon. Pour l'apprentissage semi-supervisé, une partie des vecteurs π_n seront binaires comme pour le cas de l'apprentissage supervisé, l'autre partie sera constituée de vecteurs de distributions uniformes sur les classes. Enfin, en apprentissage non supervisé, les classes sont équiprobables.

D'un point de vue applicatif, l'apprentissage semi-supervisé couvre plusieurs spécialités. Tout d'abord, dans le domaine de la reconnaissance d'objets dans des images : chaque image est annotée de manière binaire telle que la présence ou l'absence des classes dans les images soit connue (Crandall and Huttenlocher, 2006) (Ulusoy and Bishop, 2005) (Ponce et al., 2006) (Weber et al., 2000) (Fergus et al., 2006) (Schmid, 2004). Dans ce cas, pour chaque objet d'une image, l'annotation peut être vue comme une information *a priori* sur les classes. Deuxièmement, les objets peuvent être annotés par un expert, ou l'application elle-même produit des incertitudes sur les classes, et donc des *a priori* (Rossiter and Mukai, 2007). Ce type d'application est typique de la télédétection et notamment de l'interprétation d'images (Van de Vlag and Stein, 2007). Le cas de l'acoustique halieutique est un cas typique d'apprentissage faiblement supervisé (Fablet et al., 2008) (Lefort et al., 2009). Des images de la colonne d'eau sont acquises par un sondeur acoustique, l'objectif étant de classer les bancs de poissons dans les images en fonction de leur classe (le banc de poissons étant finalement associé à une seule classe). L'unique connaissance *a priori* est celle de la proportion des classes dans les images qui est donnée par chalutage, le résultat de la pêche étant représentatif de la proportion des classes présentes dans l'image au moment de la pêche. Enfin, l'apprentissage faiblement supervisé est employé dans le cas des cascades de classificateurs probabilistes (Maccormick and Blake, 2000) (Neville and Jensen, 2000). C'est le cas de l'apprentissage semi-supervisé itératif (Chapelle et al., 2006). Au lieu de transmettre une information " dure " sur les labels, les probabilités issues des sorties des classificateurs sont conservées et utilisées pour l'apprentissage d'un autre classificateur. Dans ce cas, l'emploi des classificateurs probabilistes diminue le risque de propagation d'erreurs.

En premier lieu, dans ce papier, une méthode d'apprentissage faiblement supervisé qui s'appuie sur les arbres de décision est proposée. La plupart des classificateurs probabilistes utilisent des modèles génératifs qui utilisent l'algorithme EM (Neal and Hinton, 1998) (Mc Lachlan and Krishnan, 1997). Ces classificateurs probabilistes sont aussi employés avec les ensembles de classificateurs tels que les méthodes dites de *boosting* (Kotsiantis and Pintelas, 2005), ou pour les classificateurs itératifs (Neville and Jensen, 2000). En comparaison, nous proposons d'apprendre des arbres de décision à partir d'un jeu de données faiblement labélisé. Les arbres de décision et les forêts aléatoires sont des techniques de classification supervisée très performantes et très souples (Breiman, 2001). Cependant, à notre connaissance, il n'existe pas de méthodes qui permettent d'apprendre un arbre de décision avec des données faiblement labélisées, i.e. des arbres qui prennent des probabilités en entrée. En effet, si les forêts aléatoires produisent des probabilités en sortie, l'apprentissage des arbres est toujours effectué pour des données dont les labels sont connus. La seconde contribution de ce papier concerne le développement d'une procédure itérative d'apprentissage faiblement supervisé. L'objectif est

de modifier itérativement les labels des données d'apprentissage afin que les *a priori* faibles convergent vers des *a priori* forts et afin que les *a priori* forts restent des *a priori* forts. De cette façon, le classificateur final est appris sur des données quasiment supervisées. Nous étudions le comportement des méthodes proposées sur des jeux de données provenant de la base de données UCI, et une application à l'acoustique halieutique est proposée. Les techniques proposées sont comparées à des modèles de classification connus : un modèle génératif et un modèle discriminant.

Le papier est organisé comme suit. Nous présentons l'apprentissage faiblement supervisé des arbres de décision dans la section 2, puis la procédure itérative est détaillée dans la section 3. Enfin, une validation expérimentale est effectuée dans la section 4, avant de conclure dans la section 5.

2 Arbres de decision et forêts aléatoires

Dans cette section, nous présentons l'apprentissage faiblement supervisé des arbres de décision et des forêts aléatoires. Pour cela, nous commençons par un bref état de l'art sur les arbres de décision.

2.1 Arbres de décision et forêts aléatoires : état de l'art

Les arbres de décision sont usuellement construits à partir de données labélisées. La méthode consiste à sous-échantillonner l'espace des descripteurs de manière dichotomique, sous la contrainte que les sous-échantillons de l'espace soient le plus homogène en classes. La scission d'un sous-espace s'arrête en fonction d'un critère d'homogénéité des classes (typiquement la variance, l'entropie, etc). Il existe un grand nombre de méthodes pour construire un arbre de décision. Chaque méthode se différencie par son critère de scission. Des méthodes emploient le critère de Gini (Breiman et al., 1984), d'autres l'entropie de Shannon (Quinlan, 1986) (Quinlan, 1993), ou encore des tests statistiques comme ANOVA (Loh and Shih, 1997) ou le test du χ^2 (Kass, 1980). Toutes ces méthodes proposent des résultats sensiblement équivalents, la différence de performance se faisant sur le jeu de données testé.

Ici, nous focalisons notre attention sur la méthode C4.5 (Quinlan, 1993) qui est l'une des méthodes les plus utilisées pour construire un arbre de décision. Lors de l'apprentissage, à un noeud donné de l'arbre (i.e. en l'un des sous-échantillons de l'espace jusque là créés), il faut choisir la valeur de scission S_d associée au descripteur d qui maximise le gain d'informations G :

$$\arg \max_{\{d,S_d\}} G(S_d) \tag{1}$$

tel que G représente le gain d'entropie des classes :

$$\begin{cases} G = \left(\sum_m E^m \right) - E^0 \\ E^m = -\sum_i p_{mi} log(p_{mi}) \end{cases} . \tag{2}$$

où E^0 est l'entropie des classes au noeud considéré, E^m l'entropie au noeud fils m, et p_{mi} la probabilité de la classe i au noeud fils m. Une fois l'arbre construit, un exemple de test le

parcourt jusqu'au noeud final en fonction des différentes règles qui dépendent de S_d en chaque noeud. Dans le noeud final, l'exemple test se voit attribuer la classe correspondant à la classe majoritaire parmi les données d'apprentissage.

Les forêts aléatoires combinent une procédure de " bagging " (Breiman, 1996) (génération de plusieurs arbres de décision à partir de sous-échantillons d'exemples d'apprentissage, puis vote sur les propositions de chacun des arbres), et le choix aléatoire d'un sous-échantillon de descripteurs en chaque noeud (Ho, 1995). L'idée est de générer un grand nombre de classificateurs simples et peu performants, mais qui, une fois réunis ensemble, proposent une frontière de séparation moyenne très efficace. Grâce à la distribution *a posteriori* des classes obtenue à l'aide du vote sur l'ensemble des arbres de la forêt, les forêts aléatoires (Breiman, 2001) peuvent fournir des probabilités de classification en sortie. Il existe d'autres variantes pour la construction des arbres, notamment concernant le choix aléatoire du descripteur en un noeud donné (Geurts et al., 2006). Dans le reste du papier, nous ne considérons que l'approche standard (Breiman, 2001).

2.2 Apprentissage faiblement supervisé des arbres de décision

Dans cette section, nous proposons une méthode pour apprendre un arbre de décision au moyen d'un ensemble d'apprentissage faiblement supervisé $\{x_n, \pi_n\}$.

En comparaison des travaux antérieurs, chaque noeud de l'arbre est associé à un vecteur qui contient l'*a priori* des classes. L'idée principale est de propager ce vecteur à travers l'ensemble des noeuds de l'arbre. En conséquence, étant donné un arbre de décision, un exemple test parcourt cet arbre tel qu'en chaque noeud un vecteur d'*a priori* des classes lui est associé. Ainsi, au noeud terminal, plutôt que de se voir associer une classe définitive comme en classification supervisée, l'exemple test reçoit pour label un vecteur de probabilités.

Soit p_{mi} la probabilité *a priori* de la classe i dans le noeud m. Le point central de l'apprentissage faiblement supervisé des arbres de décision réside dans le calcul de p_{mi}, ensuite le critère de scission (2) reste inchangé. En classification supervisée, la classe des exemples d'apprentissage est connue, la probabilité de la classe i au noeud m (p_{mi}) se déduit donc facilement. En revanche, en classification faiblement supervisée, les classes réelles des exemples d'apprentissage sont inconnues, ce qui engendre des difficultés dans le calcul de p_{mi}. Nous proposons de calculer p_{mi} comme une somme pondérée des *a priori* initiaux $\{\pi_{ni}\}$ des exemples concernés au noeud m. Soit le descripteur d, et x_n^d la valeur du descripteur d pour l'exemple x_n. Alors, pour le noeud fils m_1 qui regroupe ensemble les données telles que $\{x_n^d\} < S_d$, la règle de fusion suivante est proposée :

$$p_{m_1 i} \propto \sum_{\{n\}|\{x_n^d\}<S_d} (\pi_{ni})^\alpha \tag{3}$$

Pour le second noeud fils m_2 qui regroupe les exemples tels que $\{x_n^d\} > S_d$, une règle équivalente de fusion est proposée :

$$p_{m_2 i} \propto \sum_{\{n\}|\{x_n^d\}>S_d} (\pi_{ni})^\alpha \tag{4}$$

Les règles de fusion (3) et (4), relatives aux deux noeuds fils respectivement, sont des estimateurs des probabilités *a priori* des classes dans les noeuds fils. Elles ont pour but de

produire un vecteur de probabilité qui indique le poids de chaque classe dans les noeuds fils. De manière intuitive, ce critère permet de fusionner des données imprécises en conservant leurs degrés de certitude. Ainsi, la fusion de données dont la connaissance sur les labels est très faible a peu d'impact sur les probabilités finales de classification de l'ensemble des données d'un noeud. En revanche, si le degré de certitude est élevé pour les labels, cela impacte fortement la fusion des données et la probabilité des classes d'un noeud donné. Par exemple, pour un ensemble de données faiblement labélisée, si la classe i se détache des autres par un ensemble de probabilités *a priori* fortes, alors p_{mi} est maximale pour la classe i. De là, on se ramène dans l'espace des probabilités en normalisant le vecteur $\{p_{mi}\}_i$. Ainsi, le poids p_{mi} est la probabilité de la classe i dans le noeud fils m.

L'emploi d'un produit dans les équations (3) et (4) est dommageable du fait qu'il suffit d'un seul *a priori* nul $\pi_{ni} = 0$ pour annuler la probabilité p_{mi}, or, une classe peut dominer dans un groupe qui contient des exemples dont les probabilités de classification *a priori* sont nulles pour cette classe. Cela justifie l'utilisation de la somme.

Le paramètre α a deux rôles conjugués. Premièrement, il permet de fixer la dynamique du vecteur $\{p_{mi}\}_i$. En effet, pour un même ensemble de données faiblement labélisées, le paramètres α modifie les poids $\{p_{mi}\}_i$ tout en conservant leur ordre hiérarchique. Par exemple, dans le cas de deux classes, pour une valeur de α donnée, la fusion des informations peut engendrer un vecteur de dynamique très faible ($\{p_{mi}\}_i = [0,4\ 0,6]$), ou au contraire, un vecteur de dynamique très forte ($\{p_{mi}\}_i = [0,1\ 0,9]$), tout en conservant la même prédominance d'une classe sur l'autre. Deuxièmement, α permet de pondérer les exemples dont l'*a priori* est faible, l'objectif étant d'amoindrir la contribution de ces exemples porteurs de bruits dans le calcul de l'entropie (2). Ainsi, pour $\alpha = 0$, l'effet des probabilités *a priori* est annulé (dans ce cas, $\pi_{ni}^{\alpha} = 1, \forall n$), pour $0 < \alpha < 1$ les probabilités *a priori* proches de 0 ont beaucoup moins de poids que les autres dans le calcul de p_{mi}, pour $\alpha = 1$ les probabilités *a priori* ne subissent aucune modification, et enfin, pour $1 < \alpha$ seules les probabilités *a priori* proches de 1 contribuent au calcul de p_{mi}. Une étude de paramètres est effectuée dans la section 4.3.

Notons que nous présentons le cas de deux noeuds fils, mais cette méthode s'étend facilement au cas multi noeuds fils en considérant, par exemple, une valeur de scission S_d pour chaque classe.

L'extension aux forêts aléatoires s'effectue très simplement. Pour un exemple test x, la sortie de chaque arbre t de la forêt est un vecteur de probabilités $p_t = \{p_{ti}\}$. p_{ti} est la probabilité *a priori* de la classe i au noeud terminal atteint de l'arbre t. La probabilité finale que l'exemple x soit de la classe i, i.e. la probabilité de classification *a posteriori* $p(y = i|x)$, est donnée par la moyenne :

$$p(y = i|x) = \frac{1}{T} \sum_{t=1}^{T} p_{ti} \tag{5}$$

où $y = i$ désigne le fait que l'exemple x est de la classe i et T désigne le nombre d'arbres de la forêt. La classification finale est effectuée en attribuant la classe la plus probable au sens de la probabilité *a posteriori* (5).

3 Classification itérative

Dans cette section, une procédure itérative est proposée. Celle-ci s'applique uniquement sur l'ensemble d'apprentissage. Une version simple et naïve est proposée, puis nous présentons une version plus robuste qui élimine les effets de sur-apprentissage.

3.1 Procédure itérative simple

L'ensemble d'apprentissage est constitué d'exemples auxquels sont associés les *a priori* pour chaque classe. Ces vecteurs de probabilités provoquent un grand nombre de situations : les *a priori* peuvent être forts (on s'approche alors du cas de l'apprentissage supervisé) ou ils peuvent être faibles, les classes pouvant être équiprobables pour un exemple donné (c'est le cas de l'apprentissage non supervisé). Plus les *a priori* sont faibles, plus les labels sont bruités. L'idée principale du processus itératif est de modifier les *a priori* de l'ensemble d'apprentissage tels qu'ils convergent vers des valeurs plus fortes, et s'ils sont déjà forts, ils doivent le rester. Le classificateur final est donc appris après le processus itératif qui peut être vu comme une étape de filtrage : le classificateur d'une itération donnée peut être vu comme un filtre qui diminue le bruit des labels. Cette approche est similaire au *self training* (Rosenberg et al., 2005) et au *co-training* (Blum and Mitchel, 1998) en classification semi-supervisée (Chapelle et al., 2006) : des labels sont attribués aux données non labélisées au fur et à mesure des itérations tel que l'ensemble d'apprentissage croît avec le nombre d'itérations. Dans ce papier, les différences sont d'une part que nous traitons l'ensemble des données d'apprentissage à chaque itération, d'autre part que nous disposons d'un *a priori* initial sur les exemples d'apprentissage.

De telles procédures itératives ont été étudiées auparavant dans différents contextes, mais particulièrement avec des modèles génératifs (Neville and Jensen, 2000). Malgré de bonnes performances (Macskassy and Provost, 2003), ces modèles itératifs sont empiriques et l'étude de la convergence reste complexe (Culp and Michailidis, 2008) (Haffari and Sarkar, 2007). Les principaux inconvénients sont les effets de sur-apprentissage et la possible propagation des erreurs dès les premières itérations.

L'implémentation de cette procédure itérative naïve est effectuée comme suit. A l'itération m, étant donné l'ensemble d'apprentissage faiblement supervisé $\{x_n, \pi_n^m\}$, un classificateur C_m faiblement supervisé est appris. Dans ce papier, C_m est la forêt aléatoire proposée dans la section 2. C_m est alors utilisé pour mettre à jour les *a priori* des données d'apprentissage. Les nouveaux *a priori* sont notés π^{m+1}. Cette mise à jour doit exploiter à la fois la sortie du classificateur C_m et l'*a priori* initial π^1. Ici, l'expression de la mise à jour des *a priori* est donnée par : $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$ où $y_n = i$ désigne la classe de l'exemple n.

Cette mise à jour permet de fusionner toutes les informations de classification liée à l'exemple d'apprentissage. Ainsi, la probabilité de classification *a priori* fournie par l'ensemble d'apprentissage initial est fusionnée avec la probabilité de classification *a posteriori* de l'itération m qui, sous réserve de classification correcte, tend vers une valeur plus probable et plus précise. Cette fusion entraine deux avantages. Premièrement, en cas de classification erronée, i.e. dans le cas d'une probabilité *a posteriori* $p(x_n|y_n = i, C_m)$ qui spécifie la mauvaise classe, l'apport de l'*a priori* initial peut modifier les probabilités de classification finales π_n^{m+1} vers une valeur plus correcte. Cela prend de l'importance pour les cas compliqués qui impliquent des exemples qui se situent sur les frontières inter-classes et qui impliquent des probabilités *a posteriori* très voisines d'une classe à l'autre. En revanche, pour des exemples aisément séparables, les proba-

Soit l'ensemble d'apprentissage initial $T_1 = \{x_n, \pi_n^1\}$ et M itérations,

1. Pour m allant de 1 à M
   - Apprendre le classificateur C_m à partir de T_m.
   - Classer T_m à partir de C_m.
   - Mettre à jour $T_{m+1} = \{x_n, \pi_n^{m+1}\}$ avec $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$.
2. Apprendre le classificateur final avec T_{M+1}.

TAB. 1 – *Procédure iterative simple (Iter1).*

bilités de classification *a posteriori* sont très élevées pour une classe et la fusion ne change pas le résultat de classification. Deuxièmement, cette fusion a pour effet de conserver les *a priori* nuls. Ayant fait le choix d'un produit, si $\pi_n^m = 0$, alors $\pi_n^{m+1} = 0$. Ainsi, cela diminue les degrés de libertés dans l'espace des classes, et par conséquent, les erreurs possibles.

L'algorithme complet est donné dans le tableau 1. Par la suite, cette procédure simple est référencée sous le nom de Iter1.

3.2 Procédure itérative améliorée

L'inconvénient majeur de la procédure Iter1 est qu'à l'itération m, les données classées par le classificateur C_m sont celles qui ont appris le classificateur C_m. Cela produit un fort effet de sur-apprentissage qui doit être évité (voir figure 1).

Ainsi, nous proposons une seconde procédure itérative. Afin que les données classées ne soient pas celles qui apprennent le classificateur, il est nécessaire, à chaque itération, de séparer aléatoirement l'ensemble d'apprentissage en deux sous-ensembles d'apprentissage et de test. Plus précisément, à l'itération m, l'ensemble d'apprentissage $T_m = \{x_n, \pi_n^m\}$ est scindé en deux selon une proportion β. Un sous-ensemble d'apprentissage Tr_m et un ensemble de test Tt_m sont créés. Tr_m sert à apprendre le classificateur faiblement supervisé C_m. Les exemples de test Tt_m sont classés à l'aide de C_m, puis les *a priori* de Tt_m sont mis à jour selon la même règle de mise à jour que Iter1 : $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$. β donne la proportion d'exemple d'apprentissage Tr_m tandis que les $(1-\beta)$ exemples restant constituent l'ensemble de test Tt_m. Le choix de β conduit à compromis : pour une bonne estimation du classificateur C_m, la quantité d'individus du sous-ensemble d'apprentissage doit être suffisamment élevée, i.e. β doit être grand. Mais si β est trop grand, seuls quelques échantillons de l'ensemble de test verront leur *a priori* mis à jour, ce qui conduira à un temps de convergence relativement long. Le choix de la valeur de β est détaillée dans la section 4.3.

L'algorithme complet est donné dans le tableau 2. Dans la suite cette procédure améliorée est notée Iter2.

4 Performances

Les modèles proposés sont testés sur des jeux de données provenant de la base de données UCI, puis sur un jeu de données réelles de bancs de poissons. Plusieurs modèles de

Soit l'ensemble d'apprentissage initial $T_1 = \{x_n, \pi_n^1\}$ et M itérations,

1. pour m allant de 1 à M
   - Scinder aléatoirement T_m en deux groupes : $Tr_m = \{x_n, \pi_n^m\}$ et $Tt_m = \{x_n, \pi_n^m\}$ selon la proportion β.
   - Apprendre le classificateur C_m à partir de Tr_m.
   - Classer Tt_m à l'aide de C_m.
   - Mettre à jour $Tt_{m+1} = \{x_n, \pi_n^{m+1}\}$ avec $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$.
   - Mettre à jour l'ensemble d'apprentissage global T_{m+1} tel que : $T_{m+1} = \{Tr_m, Tt_{m+1}\}$.
2. Apprendre le classificateur final avec T_{M+1}.

TAB. 2 – *Procédure iterative améliorée (Iter2).*

classification sont évalués : Iter1 et Iter2 associés aux forêts aléatoires probabilistes (respectivement notés FA+Iter1 et FA+Iter2), les forêts aléatoires probabilistes seules (FA), le modèle génératif (EM) (Ulusoy and Bishop, 2005) et le modèle discriminant non linéaire (Fisher+Kernel) (Fablet et al., 2008). Pour les données de bancs de poissons, nous utiliserons le modèle FA+Fisher+Kernel qui est l'utilisation du processus itératif Iter1 sur deux itérations, l'une utilisant le classificateur discriminant non linéaire (Fisher +Kernel), l'autre les forêts aléatoires (FA).

Nous cherchons le meilleur modèle de classification, i.e. celui qui fournit le meilleur taux de bonnes classifications. Dans ce sens, les performances des classificateurs sont évaluées relativement aux taux de bonnes classifications.

4.1 Protocole de simulation

Afin de maitriser le niveau de bruit des labels et ainsi de mesurer la réponse des modèles de classification vis-à-vis de la complexité des *a priori*, les ensembles d'apprentissage pour la classification faiblement supervisée sont générés artificiellement à partir de jeux de données supervisés. Le protocole de construction des images est comme suit. Les exemples d'apprentissage sont distribués dans plusieurs groupes pour lesquels la proportion des classes est fixée préalablement. Tous les exemples d'un groupe se voient attribuer comme *a priori* les proportions des classes du groupe considéré.

La première catégorie d'expérience (section 4.4.1) mesure la robustesse des classificateurs relativement au nombre de classes compris dans les mélanges. Dans ce cas, tout type de mélange est considéré, allant des cas de labellisations quasi supervisés, i.e. une classe est fortement plus probable que les autres dans le mélange, aux cas où les classes sont quasi équiprobables. Dans le tableau 3, nous montrons des exemples de proportions cibles dans les groupes. Dans cet exemple, pour un jeu de données qui contient trois classes, nous créons des groupes dont les proportions incluent une seule classe (cas supervisé), deux classes, ou trois classes. La complexité des *a priori* augmente donc avec le nombre de classes dans le mélange. Pour chacun des trois cas de figure, différentes complexités de mélange sont proposées, allant du cas quasi équiprobable, au cas où une classe domine dans le mélange.

La seconde catégorie d'expérience (section 4.4.2) évalue la robustesse des classificateurs relativement à la complexité des mélanges. Lors de la création des ensembles d'apprentissage, chacune des proportions cibles est identiques de telle sorte que chaque classe domine au moins une fois dans un mélange. Nous créons ainsi différents types de complexité de probabilité *a priori*, allant du cas de l'apprentissage supervisé, à l'apprentissage non supervisé. Dans le tableau 4, nous exposons les proportions cibles choisies dans le cas d'un jeux de données qui comprend 3 classes. Analysons les composantes de ce tableau. Pour l'apprentissage supervisé une seule classe est possible dans chaque mélange. Pour l'apprentissage faiblement supervisé, nous avons créé deux niveaux de complexité : " apprentissage faiblement supervisé (1) " pour lequel toutes les classes sont probables mais avec l'une des classes qui domine largement dans le mélange, et " apprentissage faiblement supervisé (2) " pour lequel les probabilités *a priori* sont plus proches. Enfin, pour l'apprentissage non supervisé, les classes sont équiprobables.

Une validation croisée sur 100 tests permet d'extraire un taux de bonne classification moyen. Pour chaque test, 90% des exemples du jeu de données considéré sont utilisés pour apprendre un classificateur, les 10% restants sont classés.

Mélanges à 1 classe :
$\begin{pmatrix}1\\0\\0\end{pmatrix}$ $\begin{pmatrix}0\\1\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\\1\end{pmatrix}$ (apprentissage supervisé)
Mélanges à 2 classes :
$\begin{pmatrix}0.8\\0.2\\0\end{pmatrix}$ $\begin{pmatrix}0.2\\0.8\\0\end{pmatrix}$ $\begin{pmatrix}0.6\\0.4\\0\end{pmatrix}$ $\begin{pmatrix}0.4\\0.6\\0\end{pmatrix}$ $\begin{pmatrix}0.8\\0\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0\\0.8\end{pmatrix}$ $\begin{pmatrix}0.6\\0\\0.4\end{pmatrix}$ $\begin{pmatrix}0.4\\0\\0.6\end{pmatrix}$ $\begin{pmatrix}0\\0.8\\0.2\end{pmatrix}$ $\begin{pmatrix}0\\0.2\\0.8\end{pmatrix}$ $\begin{pmatrix}0\\0.6\\0.4\end{pmatrix}$ $\begin{pmatrix}0\\0.4\\0.6\end{pmatrix}$
Mélanges à 3 classes :
$\begin{pmatrix}0.8\\0.1\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.8\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.1\\0.8\end{pmatrix}$ $\begin{pmatrix}0.4\\0.2\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.4\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.2\\0.4\end{pmatrix}$

TAB. 3 – *Pour un jeux de données à 3 classes, exemples de probabilités a priori pour les données d'apprentissage. Dans cet exemple, la complexité du jeu de données dépend du nombre de classes possibles dans les mélanges.*

Apprentissage supervisé :
$\begin{pmatrix}1\\0\\0\end{pmatrix}$ $\begin{pmatrix}0\\1\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\\1\end{pmatrix}$
Apprentisage faiblement supervisé (1) :
$\begin{pmatrix}0.8\\0.1\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.8\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.1\\0.8\end{pmatrix}$
Apprentisage faiblement supervisé (2) :
$\begin{pmatrix}0.5\\0.3\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.5\\0.3\end{pmatrix}$ $\begin{pmatrix}0.3\\0.2\\0.5\end{pmatrix}$ $\begin{pmatrix}0.5\\0.2\\0.3\end{pmatrix}$ $\begin{pmatrix}0.3\\0.5\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.3\\0.5\end{pmatrix}$
Apprentisage non supervisé :
$\begin{pmatrix}0.33\\0.33\\0.33\end{pmatrix}$

TAB. 4 – *Pour un jeux de données à 3 classes, exemples de probabilités a priori pour les données d'apprentissage. Dans cet exemple, la complexité des données d'apprentissage dépend des valeurs des probabilités de classification a priori, allant du cas de l'apprentissage supervisé, à celui de l'apprentissage non supervisé.*

4.2 Jeux de données

4 jeux de données ont été sélectionnés dans la base de données UCI. D1 est un jeu de données provenant de la communauté de vision par ordinateur. Il contient 7 classes de texture d'images, qui sont représentées par 19 descripteurs de texture. L'intérêt de ce jeu de données est la possibilité de créer des mélanges complexes à 7 classes. D2 est une base de données contenant les dimensions de 3 classes d'Iris. Chaque fleur est représentée dans un espace de 4 descripteurs. D3 décrit les tendances de certains graphiques (normaux, périodiques, etc). 5 descripteurs quantifiés ou continus décrivent les courbes. Comme pour D1, l'intérêt de ce jeu de données se situe dans le grand nombre de classes proposé : 6. Enfin, D4 est un jeu de données formé des descripteurs de formes d'ondes (Breiman et al., 1984). 19 descripteurs continus décrivent 3 classes d'ondes formées de la combinaison de plusieurs bases d'ondes auxquelles s'ajoute du bruit Gaussien.

Ces jeux de données ont été choisis car ils correspondent à certains critères. Tout d'abord, ils doivent contenir plusieurs classes afin de créer des *a priori* complexes impliquant un grand nombre de classes. Typiquement, nous avons choisi des jeux de données contenant plus de 2 classes. Ensuite, la base de données doit contenir suffisamment d'exemples par classe pour créer un grand nombre de mélanges et une grande variété de types de proportion. Enfin, les jeux de données doivent être équilibrés en classes pour que les mélanges obtenus suivent les proportions imposées par les proportions cibles (cf. tableaux 3 et 4). Les expériences relatives à ces 4 jeux de données sont reportées dans la section 4.4.

Le cinquième jeu de données D5 constitue l'application de ces travaux : l'acoustique halieutique (section 4.5). D5 est composé des descripteurs de bancs de poissons dans une image. Il est constitué de 4 classes d'espèces de poissons, tel que chaque banc de poissons est décrit dans un espace de 20 descripteurs, et tel que les classes sont déséquilibrées.

Les caractéristiques essentielles des jeux de données D1, D2, D3, D4, et D5 sont récapitulées dans le tableau 5.

Base de données	**Nature**	**Nombre de classes**	**Exemples par classe**	**Descripteurs**
D1	Texture	7	330	19 (Continus)
D2	Végétal	3	50	4 (Continus)
D3	Graphique	6	100	5 (Quantifiés+Continus)
D4	Forme d'onde	3	200	19 (Continus)
D5	Bancs de poissons	4	179-478-667-95	20 (Continus)

TAB. 5 – *Caractéristiques des jeux de données avec leur nature (thème de classification), I (le nombre de classes), le nombre d'exemples par classe, et le nombre de descripteurs.*

4.3 Choix des paramètres de simulation

Dans cette section, nous discutons du choix des paramètres des modèles de classification en partant du postulat suivant : les performances de classification en apprentissage faiblement supervisée ne peuvent pas être meilleures que celles obtenues en classification supervisée.

Cette hypothèse est raisonnable dans le sens où, en classification faiblement supervisée, les données d'apprentissage sont bruitées, et donc elles affectent la qualité de l'estimation des paramètres des modèles. Cette hypothèse est vérifiée à travers les simulations proposées dans la section 4.4. Ajoutons que le processus itératif que nous proposons, a pour objectif de nous rapprocher du cas de l'apprentissage supervisé. Ainsi, dans un premier temps, nous cherchons les valeurs des paramètres dans le cas de l'apprentissage supervisé, puis, dans un second temps, nous reportons les valeurs obtenues dans les modèles de classification faiblement supervisée.

Nous souhaitons que nos modèles de classification soit génériques, donc les paramètres recherchés doivent satisfaire tous les jeux de données à la fois. Ainsi, le protocole d'évaluation des paramètres consiste à choisir celui qui produit le meilleur taux de bonne classification moyen. Cela est effectué en testant les modèles de classification pour plusieurs valeurs des paramètres.

Le premier paramètre est la proportion d'exemples de l'ensemble d'apprentissage utilisée pour la construction d'un arbre d'une forêt aléatoire (cf. section 2.1). Pour un ensemble de valeur possible de cette proportion, et pour chaque jeu de données, les performances de classification sont reportées dans le tableau 6. Pour cette expérience, le nombre d'arbres de la forêt est fixé à $T = 100$. Au vu des résultats, la proportion choisie est 0,8. Notons que ce paramètre a peu d'impact sur les résultats, tant qu'il est différent de 1. En effet, dans ce cas, tous les arbres de la forêt sont sensiblement équivalents, ce qui va à l'encontre de l'essence même des forêts aléatoires : la création d'incertitudes et de classificateurs dits " faibles " qui apportent des vues et des propositions de classification éloignées (Breiman, 1996).

Proportion d'exemples d'apprentissage pour un arbre d'une forêt	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
D1	0.95	0.95	0.95	0.96	0.95	0.92
D2	0.92	0.92	0.93	0.97	0.92	0.92
D3	1	1	1	1	1	1
D4	0.86	0.81	0.82	0.79	0.80	0.68
D5	0.89	0.89	0.90	0.90	0.89	0.84
Moyenne	0.92	0.91	0.92	0.92	0.91	0.87

TAB. 6 – *Performance de classification supervisée en fonction de la proportion d'exemples utilisés pour l'apprentissage d'un arbre de décision d'une forêt aléatoire (cf. section 2.1). Pour cette expérience,* $T = 100$.

Le second paramètre est le nombre d'arbres dans la forêt : T (équation (5)). Pour un ensemble de valeurs possibles, et pour chaque jeu de données, les performances de classification sont affichées dans le tableau 7. Au vu des résultats, nous choisissons T =100.

Le troisième paramètre est l'exposant α qui pondère les probabilités *a priori* dans les critères de fusion (3) et (4). Cette fois, nous nous plaçons dans le cas de l'apprentissage faiblement supervisé (α n'est d'aucune utilité en classification supervisée) tel que des mélanges à 3 classes sont générés (cf. " mélange à 3 classes " dans le tableau 3). Comme précédemment, pour un ensemble de paramètres et pour tous les jeux de données, nous calculons les perfor-

mances de classification. Les résultats sont reportés dans le tableau 8. Au vu des résultats, nous choisissons $\alpha = 1$.

Enfin, le choix du paramètre β (section 3.2) est déduit par raisonnement logique. Dans le processus itératif Iter2, il faut suffisamment de données d'apprentissage pour une estimation correcte du classificateur d'une itération donnée, mais β ne doit pas être trop grand afin que le temps de simulation soit raisonnable. Pour les expériences à suivre, nous choisissons de mettre en avant la précision de la classification au détriment du temps de calcul, alors nous fixons $\beta = 0,75$. Ce choix se justifie principalement en fonction du jeu de données D2 qui ne contient que 50 exemples par classe (cf. tableau 5), ainsi, avec $\beta = 0,75$, les classificateurs seront constitués à partir d'un jeu de données qui contient environ 37 exemples par classe. Si nous avions choisi $\beta = 0.5$, alors le jeu de données partiel d'une itération donnée ne contiendrait que 25 exemples par classe. Ce choix est en fait une manière évidente d'obtenir à chaque fois des taux de bonne classification quasi-optimaux pour le classificateur donné. Cependant, nous verrons dans la section 4.4.1 que ce choix implique un temps de calcul considérable.

T	**1**	**100**	**200**	**300**	**400**
D1	0.90	0.96	0.96	0.96	0.96
D2	0.94	0.93	0.97	0.93	0.93
D3	1	1	1	1	1
D4	0.65	0.82	0.79	0.78	0.80
D5	0.81	0.90	0.90	0.89	0.89
Moyenne	0.86	0.92	0.92	0.91	0.91

TAB. 7 – *Performance de classification supervisée en fonction du paramètre T (équation (5)), le nombre d'arbres de décision considérés dans une forêt aléatoire. Pour cette expérience, la proportion d'exemples utilisés pour l'apprentissage d'un arbre d'une forêt aléatoire, relativement à l'ensemble d'apprentissage initiale (cf. section 2.1), est fixé à* 0.8.

α **(3 classes)**	**0.1**	**0.4**	**1**	**3**	**8**
D1	0.92	0.91	0.91	0.90	0.91
D2	0.79	0.80	0.81	0.82	0.82
D3	0.86	0.87	0.89	0.94	0.91
D4	0.75	0.73	0.81	0.77	0.77
D5	0.72	0.69	0.68	0.71	0.73
Moyenne	0.80	0.80	0.82	0.82	0.82

TAB. 8 – *Performance de classification faiblement supervisée en fonction du paramètre α (équations (3) et (4)), le coefficient de pondération pour le calcul de l'entropie en chaque noeud des arbres de décision. Pour chaque observation de l'ensemble d'apprentissage faiblement annoté, trois classes sont probables (cf. " mélanges à 3 classes " dans le tableau 3).*

4.4 Simulation avec les données UCI

Dans cette section, nous effectuons des expériences de simulations sur les jeux de données D1, D2, D3, et D4.

4.4.1 Performances en fonction du nombre de classes probables

Pour cette expérience, nous évaluons les performances de classification en fonction du nombre de classes possibles qui définit aussi le niveau de complexité des probabilités *a priori*. Cette fois, une grande variété de proportions cibles est créée de telle sorte qu'il existe des situations de labels faiblement bruités et fortement bruités à la fois, la seule variable étant le nombre de classes présentes dans chaque mélange. Les mélanges cibles sont affichés dans la tableau 3 pour un jeu de données qui contient 3 classes (D2).

Les résultats sont affichés dans le tableau 9. Le taux moyen de bonne classification est reporté pour chaque jeu de données en fonction du nombre de classes probables pour chaque exemple de l'ensemble d'apprentissage. Pour chacun des modèles de classification testés (FA, FA+Iter1, FA+Iter2, Fisher+Kernel, et EM), la moyenne des taux de réussite sur l'ensemble des niveaux de complexité, ainsi que l'écart type des taux de réussite sont reportés. Les résultats sont positifs si cette moyenne est élevée, ce qui indique que les performances globales sont bonnes, et si l'écart type des taux de réussite est faible, ce qui signifie que le classificateur est robuste vis-à-vis de la complexité des données d'apprentissage.

	Nombre de classes dans le mélange	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Moyennes / Ecart type**
D1	FA+Iter1	**0.96**	0.90	0.88	0.88	0.85	0.75	0.55	0.80 - 0.13
	FA+Iter2	**0.96**	**0.96**	**0.96**	**0.94**	**0.94**	**0.92**	0.81	**0.92** - 0.05
	FA	**0.96**	0.92	0.91	0.88	0.88	0.84	0.62	0.86 - 0.11
	Fisher+Kernel	0.90	0.89	0.89	0.89	0.89	0.89	**0.84**	0.88 - **0.01**
	EM	0.83	0.83	0.84	0.83	0.83	0.83	0.75	0.82 - 0.03
D2	FA+Iter1	**0.97**	**0.97**	0.84					0.90 - 0.09
	FA+Iter2	**0.97**	**0.97**	**0.92**					**0.94** - **0.03**
	FA	**0.97**	0.90	0.81					0.89 - 0.08
	Fisher+Kernel	0.89	0.80	0.69					0.79 - 0.10
	EM	0.94	0.95	0.85					0.90 - 0.05
D3	FA+Iter1	**1**	0.90	0.91	0.82	0.74	0.74		0.86 - 0.07
	FA+Iter2	**1**	**1**	**0.99**	**0.98**	**0.97**	**0.98**		**0.98** - **0.01**
	FA	**1**	0.9	0.89	0.75	0.82	0.88		0.87 - 0.08
	Fisher+Kernel	0.78	0.72	0.68	0.62	0.62	0.73		0.69 - 0.06
	EM	0.77	0.62	0.62	0.45	0.47	0.58		0.58 - 0.11
D4	FA+Iter1	0.79	0.74	0.35					0.54 - 0.27
	FA+Iter2	0.79	**0.83**	**0.81**					**0.82** - **0.01**
	FA	0.79	**0.83**	**0.81**					0.81 - 0.02
	Fisher+Kernel	0.85	0.81	0.77					0.81 - 0.04
	EM	0.82	0.8	0.74					0.79 - 0.04

TAB. 9 – *Pour D1, D2, D3, et D4, les taux moyens de classification sont reportés en fonction du nombre de classes dans chaque mélange. Des jeux de mélanges cibles sont créés, allant du cas supervisé au cas où toutes les classes sont probables (cf. tableau 3). Les modèles testés sont les forêts aléatoires avec la procédure itérative 1 (FA+Iter1), les forêts aléatoires avec la procédure itérative 2 (FA+Iter2), les forêts aléatoires seules (FA), le modèle discriminant non linéaire (Fisher+Kernel) (Fablet et al., 2008), et le modèle génératif (EM) (Ulusoy and Bishop, 2005).*

Globalement, concernant la moyenne des taux de réussite, la méthode FA+Iter2 est la plus performante pour tous les jeux de données. En termes de robustesse relativement à la com-

plexité des données d'apprentissage, la méthode FA+Iter2 est aussi la plus performante, sauf pour le jeu de données D1 pour lequel le modèle discriminant produit l'écart type des taux de bonne classification le plus faible (0,01).

L'analyse des résultats obtenus à l'aide des classificateurs élémentaires (FA, Fisher+Kernel, et EM) montre que le modèle discriminant et les forêts aléatoires présentent des performances équivalentes. Par exemple, pour le jeu de données D1, le modèle discriminant est meilleur avec une moyenne des taux de réussite valant 0,88 (contre 0,86 pour les forêts aléatoires), et un écart type des taux de réussite qui vaut 0,01 (contre 0,11 pour les forêts aléatoires). Inversement, pour le jeu de données D2, la moyenne des taux de bonne classification atteint 89% pour les forêts aléatoires, contre 79% pour le modèle discriminant, et l'écart type des taux de bonne classification est de 8% pour FA, contre 10% pour le modèle Fisher+Kernel. Cela s'explique par l'organisation intrinsèque des données qui requière l'emploi d'un classificateur particulier. Ainsi, dans le domaine de la classification automatique, il est admis, qu'à un jeu de données, correspond un type de classificateur. Par exemple, pour le modèle discriminant non linéaire, le choix du noyau est essentiel, si les performances de classification sont moins bonnes avec le modèle discriminant, cela peut venir du fait que les similarités spatiales induites par le noyau Gaussien ne correspondent pas à la distribution spatiale des données. En revanche, pour ces jeux de données, les performances obtenues à l'aide du modèle génératif sont moins bonnes, en moyenne, que celles obtenues à l'aide des autres modèles. Cela peut s'expliquer par la distribution spatiale des données qui ne correspond pas à une organisation de mélange de Gaussiennes. Cependant, pour le jeu de données D2, les performances de classification sont meilleures que celles des modèles de classification élémentaires FA et Fisher+Kernel, atteignant 90% de taux de réussite moyen et un écart type des taux de réussite de 5%.

La combinaison de classificateurs, à l'aide du processus itératif, permet d'améliorer nettement les performances de classification des classificateurs élémentaires. En effet, si les performances des forêts aléatoires décroissent avec la complexité de l'ensemble d'apprentissage, le processus itératif induit davantage de robustesse, diminuant l'écart type des taux de réussite, et par conséquent, augmentant le taux de réussite moyen. Cela est dû au fait que les probabilités *a priori* sont corrigées de manière itérative pour les modèles Iter1 et Iter2. Par exemple, pour le jeu de données D1, l'écart type des taux de réussite diminue de 6% pour FA+Iter2 (par rapport à FA), bénéficiant ainsi des corrections itératives des *a priori* des données d'apprentissage. Notons que les combinaisons de modèles discriminants ou de modèles génératifs, via des procédures itératives n'ont pas donné de résultats convaincants, étant données les performances relativement moyennes obtenues en classification supervisée (les résultats de classification faiblement supervisée sont logiquement inférieurs à ceux obtenus en classification supervisée).

Afin d'illustrer le comportement des procédures itératives Iter1 et Iter2, nous reportons dans la figure 1 l'évolution du taux moyen de bonne classification en fonction du nombre d'itération pour les jeux de données D1 et D3, et plusieurs types de mélange. Cette figure démontre bien l'apport de Iter2 par rapport à Iter1. Alors que la procédure itérative Iter1 n'améliore absolument pas les performances de classification d'une itération à l'autre, Iter2 permet de diminuer nettement le taux d'erreurs après quelques itérations. C'est le cas du jeu de données D3, en considérant des mélanges à 2 classes, pour lequel le taux d'erreurs est diminué de plus de 10% après 10 itérations. Ces résultats sont expliqués par le fait que, à chaque itération, Iter2 sépare l'ensemble d'apprentissage en un sous-ensemble d'apprentissage et un sous-ensemble de test. A l'inverse, Iter1 ne fait pas cette scission, ce qui conduit à un fort phénomène de

sur-apprentissage.

Cependant, comme nous privilégions les performances de classification au détriment du temps de calcul (en choisissant $\beta = 0,75$, cf. section 4.3), alors les simulations sont très longues. Par exemple, en considérant une validation croisée et une itération du processus itératif qui dure environs 12h, alors les 15 points de la figure 1 sont obtenus en une semaine. L'utilisation de ces algorithmes dépend de la rapidité souhaitée pour l'application. Toutefois, remarquons que les simulations sont effectuées sous Matlab et que ce temps de calcul peut être réduit en optimisant les programmes dans un langage de programmation plus pertinent.

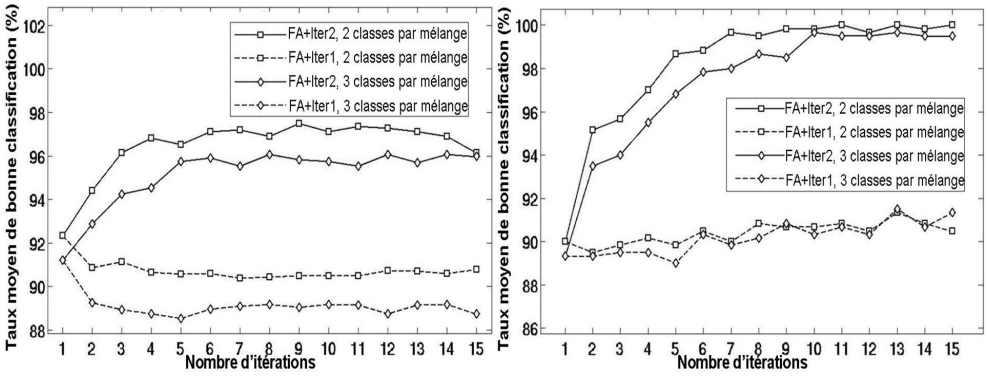

FIG. 1 – *Evolution des performances de classification des procédures itératives Iter1 et Iter2 pour les jeux de données D1 (à gauche) et D3 (à droite).*

4.4.2 Performances en fonction du niveau de bruit des labels

Dans un second temps, nous étudions les performances de classification en fonction du niveau de bruit des labels de l'ensemble d'apprentissage. 4 niveaux de complexité sont définis, allant du cas de l'apprentissage supervisé au cas de l'apprentissage non supervisé. Dans le tableau 4, les proportions cibles des groupes de données d'apprentissage qui deviendrons les labels des données d'apprentissage, sont représentées pour un jeu de données qui contient 3 classes. Dans le cas de l'apprentissage supervisé, une seule classe est présente dans le mélange, elle indique la classe de l'exemple considéré. Dans celui de l'apprentissage faiblement supervisé (1), une classe domine dans les mélanges mais toutes les classes sont probables. Dans le cas de l'apprentissage faiblement supervisé (2), le niveau de bruit est supérieur à celui de l'apprentissage faiblement supervisé (1), i.e. les valeurs des probabilités *a priori* sont plus faibles (cf. tableau 4). Pour éviter un quelconque déséquilibre, chaque classe domine au moins une fois dans un ensemble de données. Enfin, pour le cas de l'apprentissage non supervisé, les classes sont équiprobables. Cette expérience montre bien comment ce formalisme de l'apprentissage faiblement supervisé généralise tous les autres types d'apprentissage.

Les résultats sont affichés dans le tableau 10. De manière prévisible, les performances de classification chutent quand le niveau de complexité des labels augmente, passant de 90% de taux de réussite moyen dans le cas de l'apprentissage supervisé, à 24% de taux de réussite moyen dans le cas de l'apprentissage non supervisé. Cela montre l'importance des valeurs des

probabilités *a priori* des classes. Fort logiquement, sans information *a priori* sur les classes, i.e. dans le cas de l'apprentissage non supervisé, les modèles répondent très difficilement. Seule la connaissance d'un *a priori* permet d'améliorer nettement les performances de classification par rapport au cas de l'apprentissage non supervisé. Ainsi, pour des probabilités *a priori* assez fortes (" apprentissage faiblement supervisé (1) ", cf. tableau 4), les performances de classification ne sont dégradées que de 11% en moyenne par rapport au cas de l'apprentissage supervisé. Notons que cette tendance concerne tous les modèles de classification.

Les commentaires sur le comportement spécifique des classificateurs sont les mêmes que pour la section 4.4.1 : les forêts aléatoires seules (FA) produisent des résultats équivalents au modèle discriminant non linéaire (Fisher+Kernel), mais l'utilisation des forêts aléatoires dans un processus itératifs (FA+Iter2) permet d'améliorer nettement les performances de classification, enfin, le modèle génératif (EM) engendre les plus mauvais résultats en raison de l'organisation intrinsèque des données qui ne correspond pas à celle d'un mélange de Gaussienne.

Type d'apprentissage		**Supervisé**	**Faiblement supervisé (1)**	**Faiblement supervisé (2)**	**Non supervisé**
D1	RF+Iter1	**0.96**	0.85	0.72	0.14
	RF+Iter2	**0.96**	**0.91**	**0.89**	0.14
	RF	**0.96**	0.85	0.73	0.14
	Fisher+Kernel	0.90	0.87	0.86	0.14
	EM	0.83	0.83	0.82	**0.19**
D2	RF+Iter1	**0.97**	0.80	0.64	0.33
	RF+Iter2	**0.97**	**0.92**	**0.81**	0.33
	RF	**0.97**	0,78	0.60	0,33
	Fisher+Kernel	0,89	0,82	0.54	0,33
	EM	0,94	0,72	0.36	**0,38**
D3	RF+Iter1	**1**	0.74	0.63	0.16
	RF+Iter2	**1**	**0.95**	**0.90**	0.16
	RF	**1**	0.76	0.63	0.16
	Fisher+Kernel	0.78	0.63	0.57	0.17
	EM	0.77	0.48	0.38	**0.18**
D4	RF+Iter1	0.79	0.81	0.33	**0.33**
	RF+Iter2	0.79	**0.82**	**0.78**	**0.33**
	RF	0.79	0.81	0.69	**0.33**
	Fisher+Kernel	**0.85**	**0.82**	0.64	**0.33**
	EM	0.82	0.48	0.63	0.23
Moyennes		0.90	0.79	0.65	0.24

TAB. 10 – *Pour D1, D2, D3, et D4, les taux moyens de classification sont reportés en fonction du niveau de bruit des labels. Des jeux de mélanges cibles sont créés, allant du cas de l'apprentissage supervisé au cas de l'apprentissage non supervisé (cf. tableau 4). Les modèles testés sont les forêts aléatoires avec la procédure itérative 1 (FA+Iter1), les forêts aléatoires avec la procédure itérative 2 (FA+Iter2), les forêts aléatoires seules (FA), le modèle discriminant non linéaire (Fisher+Kernel) (Fablet et al., 2008), et le modèle génératif (EM) (Ulusoy and Bishop, 2005).*

4.5 Application à l'acoustique halieutique

L'apprentissage faiblement supervisé est appliqué à des données provenant de l'acoustique halieutique (Fablet et al., 2008) (Lefort et al., 2009) (cf. introduction). Les images acquises à l'aide d'un sondeur acoustique contiennent des bancs de poissons qui sont extraits automatiquement à l'aide d'un algorithme de seuillage. Chaque banc est caractérisé par des descripteurs morphologiques (longueur, hauteur, etc), bathymétriques (profondeur dans la colonne d'eau, sonde, etc), et énergétiques (moments d'ordre 1 et 2 des échos rétrodiffusés, énergie rétrodiffusée par volume, etc). En pratique, pour identifier la classe des bancs observés, il faut se référer

aux chalutages (pêches) qui renseignent sur les classes présentes dans l'image au moment du chalutage. Le chalutage étant plurispécifique, il donne la proportion des classes d'espèce dans les images de bancs de poissons. En ramenant cette proportion relative aux bancs individuellement, on obtient un ensemble d'apprentissage faiblement supervisé du type $\{x_{kn}, \pi_k\}$ où k indice les images (ou les chalutages) et n indice les bancs de l'image k. Il est impossible de connaître la classe réelle des bancs observés. Cependant, une base de données labélisée a été constituée par des experts qui ont fait le lien entre des bancs de poissons et des chalutages monospécifiques. Ainsi, comme dans la section 4.4, nous recréons des ensembles d'apprentissage faiblement supervisés à partir de données supervisées.

	Nombre de classes dans le mélange	**1**	**2**	**3**	**4**	**Moyennes / Ecart type**
D5	FA+Iter1	**0.89**	0.72	0.62	0.45	0.67 - 0.18
	FA+Iter2	**0.89**	0.79	0.71	0.42	0.70 - 0.20
	FA	**0.89**	0.71	0.68	0.58	0.71 - 0.12
	FA+Fisher+Kernel	**0.89**	**0.86**	**0.86**	**0.77**	**0.84 - 0.05**
	Fisher+Kernel	0.70	0.71	0.65	0.56	0.65 - 0.06
	EM	0.66	0.52	0.51	0.47	0.54 - 0.08

TAB. 11 – *Evolution du taux moyen de classification du jeu de données D5 en fonction du nombre de classes dans chaque mélange. Des jeux de proportions sont créés, allant du cas supervisé au cas où toutes les classes sont probables (cf. tableau 3). Les résultats sont reportés pour les modèles FA+Iter1, FA+Iter2, FA, FA+Fisher+Kernel, Fisher+Kernel, et EM.*

La première expérience concerne les performances de classification relativement au nombre de classes dans le mélange. Les proportions cibles générées comportent tous les niveaux de bruit comme dans le tableau 3. Notons que ces mélanges représentent la diversité des mélanges obtenus lors des campagnes de pêches acoustiques. Les résultats sont reportés dans le tableau 11 pour le jeu de données D5, et pour 6 modèles de classification (FA+Iter1, FA+Iter2, FA, FA+Fisher+Kernel, Fisher+Kernel, EM). Le modèle FA+Fisher+Kernel consiste en l'utilisation du processus itératif Iter1 (cf. section 3.1) sur 2 itérations, telles que l'une adopte le classificateur discriminant non linéaire (Fisher+Kernel), et l'autre utilise les forêts aléatoires probabilistes (FA). En termes d'analyse des performances, la même tendance que pour les autres jeux de données (D1, D2, D3, D4) est constatée : l'association des forêts aléatoires avec le processus itératif Iter2, produits les meilleurs résultats, excepté pour le cas qui contient 4 classes par mélanges, i.e. toutes les classes sont probables, pour lequel les forêts aléatoires probabilistes sont les plus performantes. Cependant, les résultats sont moins convaincants que pour les jeux de données D1, D2, D3, et D4, en effet, les modèles semblent moins robustes, surtout pour FA+Iter2 dont l'écart type des performances, qui représente la robustesse du modèle relativement aux variations de complexités de l'ensemble d'apprentissage, atteint 0,2. En revanche, nous constatons que le modèle discriminant non linéaire (Fisher+Kernel), malgré de faibles performances, est très robuste vis-à-vis de la complexité du label (l'écart type vaut 0,05). Cela justifie l'utilisation du modèle FA+Fisher+Kernel. L'idée est de combiner la robustesse du modèle discriminant avec les forts taux de réussites atteints par les forêts aléatoires en classification supervisée. Les résultats sont très convaincants, le modèle FA+Fisher+Kernel étant largement plus performant que les autres modèles à la fois en termes de moyennes des taux de classification, et en termes de robustesse. L'explication du déclin du modèle FA+Iter2 est donnée après la seconde expérience.

Pour la seconde expérience, nous évaluons les performances de classification en fonction du niveau de bruit des labels des exemples d'apprentissage, allant du cas supervisé, au cas non supervisé, en passant par 2 niveaux de complexité d'apprentissage faiblement supervisé (cf. tableau 4). Les performances de classification sont reportées dans le tableau 12 pour le jeu de données D5. Comme pour l'expérience précédente, les performances de classification chutent avec l'augmentation de la complexité, surtout à partir du niveau de complexité " apprentissage faiblement supervisé (2) ". Cependant, malgré de très bons résultats généraux pour les modèles basés sur les forêts aléatoires dans le cas de l'apprentissage supervisé, le modèle FA+Iter2 produit les plus mauvais résultats, après le modèle génératif. Cela va à l'encontre des résultats précédemment obtenus dans la section 4.4.

Pour le jeu de données D5, nous constatons que le modèle proposé FA+Iter2 n'est plus aussi performant que pour les données D1, D2, D3, et D4. La première explication vient du déséquilibre entre les classes. Dans le tableau 5, seul le jeu de données D5 est déséquilibré en classes, produisant des données faiblement labellisées qui ne correspondent pas vraiment à la proportion cible fixée. Par exemple, la classe qui contient 95 exemples sera sous représentée par rapport à la classe qui en contient 667. Si le déséquilibre est trop fort, la classe sous représentée ne domine jamais dans aucun des mélanges possibles, et donc l'apprentissage d'un modèle est très difficile pour la classe considérée. Une idée serait de forcer l'équilibrage des classes, à chaque expérience de la validation croisée, en supprimant des exemples aléatoirement. Cela explique la baisse générale des performances dans le tableau 12. La seconde explication vient de l'organisation intrinsèque des données dans l'espace des descripteurs. En comparaison des jeux de données D1, D2, D3, et D4, cette organisation est très complexe pour le jeu de données D5. En effet, les données ont été acquises dans des zones spatiales et temporelles différentes, ce qui impacte fortement l'ensemble des descripteurs morphologiques et énergétiques des bancs de poissons, mais aussi l'organisation structurelle des ensembles d'agrégations. Ces échantillonnages successifs font que les distributions des agrégations dans l'espace des descripteurs sont fortement multi modales. Les classificateurs doivent donc prendre la totalité des données d'apprentissage pour être suffisamment performant et pour décrire au mieux cette distribution complexe, or le second processus itératif (Iter2) scinde les données d'apprentissage en deux paquets, atténuant ainsi la précision du classificateur. Cela explique les mauvaises performances du processus itératif Iter2, obtenues dans le tableau 11 relativement à celles du tableau 9, et cela explique pourquoi le processus itératif Iter1, qui considère l'ensemble des données d'apprentissage, est plus performant que le processus Iter2 dans le tableau 12.

5 Conclusion

Dans ce papier, nous avons présenté des contributions méthodologiques pour l'apprentissage faiblement supervisé des paramètres des modèles de classification. Pour ce type d'apprentissage, les classes des données d'apprentissage ne sont pas connues, seules des probabilités de classification *a priori* sont disponibles. Ce type d'apprentissage généralise l'apprentissage supervisé, l'apprentissage semi-supervisé, le cas des données labélisées de manière binaire dans le domaine de la vision par ordinateur, et l'apprentissage non supervisé.

La première contribution concerne l'apprentissage des arbres de décision et des forêts aléatoires. Une méthode est proposée pour que les arbres soient construits à partir des probabilités des classes et telle que les forêts aléatoires fournissent en sortie une probabilité de classifica-

Type d'apprentissage		Supervisé	Faiblement supervisé (1)	Faiblement supervisé (2)	Non supervisé
D5	FA+Iter1	**0.89**	**0.81**	0.38	0.25
	FA+Iter2	**0.89**	0.47	0.32	0.25
	FA	**0.89**	0.59	0.35	0.25
	FA+Fisher+Kernel	**0.89**	0.75	**0.62**	0.24
	Fisher+Kernel	0.70	0.72	0.61	0.27
	EM	0.66	0.47	0.46	**0.28**
Moyennes		0.82	0.63	0.45	0.25

TAB. 12 – *Evolution du taux moyen de classification du jeu de données D5 en fonction du niveau de bruit des labels des exemples d'apprentissage. La complexité des données d'apprentissage évolue du cas de l'apprentissage supervisé au cas de l'apprentissage non supervisé, en passant par des cas d'apprentissage faiblement supervisé plus ou moins complexes (cf. tableau 4). Les résultats sont reportés pour les modèles FA+Iter1, FA+Iter2, FA, FA+Fisher+Kernel, Fisher+Kernel, et EM.*

tion. La seconde contribution concerne une méthode itérative appliquée aux données d'apprentissage, dont le but est de modifier les *a priori* des classes afin de renforcer une classe et de proposer un ensemble d'apprentissage moins bruité.

De manière générale, sur des données issues de la base de données UCI, les résultats expérimentaux mettent en valeur les méthodes proposées. En effet, en se comparant à des travaux antérieurs (Ulusoy and Bishop, 2005) (Lefort et al., 2009), un gain significatif est obtenu en termes de performance de classification. De plus, une application à des données issues de l'acoustique halieutique, permet de confirmer ces résultats prometteurs.

Des futurs travaux peuvent être envisagés. Premièrement, l'approche reste empirique et qualitative. En effet, il convient d'approfondir le développement théorique, notamment concernant la convergence des processus itératifs. Deuxièmement, nous avons vu que les modèles proposés dans ce papier ne sont pas adaptés aux cas complexes, proches de l'apprentissage non supervisé, il faudrait donc s'inspirer des méthodes graphiques proposées en apprentissage semi-supervisé (Chapelle et al., 2006) pour améliorer les performances. Troisièmement, nous avons vu que les méthodes itératives proposées sont moins performantes si le jeu de données n'est pas homogène en classes (section 4.5). Il est alors envisageable d'appliquer les méthodes d'apprentissage des arbres de décision qui s'appuient sur l'entropie décentrée (Lenca et al., 2010) en conservant les critères de fusion (3) et (4). Enfin, étant donné que la classification faiblement supervisée est une généralisation de la classification semi-supervisée, il serait intéressant de comparer les méthodes proposées avec celles de l'état de l'art des méthodes d'apprentissage semi-supervisé (Chapelle et al., 2006).

Références

Blum, A. and Mitchel, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the eleventh annual Conference on Computational Learning Theory*, pages 92-100.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26(2) :123-140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45 :5-32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. *Chapman & Hall*.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). Semi-supervised learning. *MIT Press*.

Crandall, D. J. and Huttenlocher, D. P. (2006). Weakly supervised learning of part-based spatial models for visual object recognition. *European Conference on Computer Vision*, pages 16-29.

Culp, M. and Michailidis, G. (2008). An iterative algorithm for extending learners to a semi-supervised setting. *Journal of Computational and Graphical Statistics*, 17(3) :545-571.

Fablet, R., Lefort, R., Scalabrin, C., Massé, J., and Boucher, J.-M. (2008). Weakly supervised learning using proportion based information : an application to fisheries acoustic. *International Conference on Pattern Recognition*, pages 1-4.

Fergus, R., Perona, P., and Zisserman, A. (2006). Weakly supervised scaled-invariant learning of models for visual recognition. *International Journal of Computer Vision*, 71(3) :273-303.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 36(1) :3-42.

Haffari, G. and Sarkar, A. (2007). Analysis of semi-supervised learning with the yarowsky algorithm. *23rd Conference on Uncertainty in Artificial Intelligence*.

Ho, T. K. (1995). Random decision forest. *International Conference on Document Analysis and Recognition*.

Hongeng, S., Nevatia, R., and Bremond, F. (2004). Video-based event recognition : activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2) :129-162.

Kass, G. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of applied statistics*, 29(2) :119-127.

Kotsiantis, P. and Pintelas, P. (2005). Logitboost of simple bayesian classifier. *Informatica Journal*, 29(1) :53-59.

Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transaction on PAMI*, 27 :1265-1278.

Lefort, R., Fablet, R., and Boucher, J.-M. (2009). Combining image-level and object-level inference for weakly supervised object recognition. Application to fisheries acoustics. *International Conference on Image Processing*, pages 293-296.

Lenca, P., Lallich, S., and Vaillant, B. (2010). Construction of an off-centered entropy for the supervised learning of imbalanced classed : some first results. *Communications in Statistics - Theory and methods*, 39(3).

Loh, W.-Y. and Shih, Y.-Y. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7 :815-840.

Maccormick, J. and Blake, A. (2000). A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1) :57-71.

Macskassy, S. A. and Provost, F. (2003). A simple relational classifier. *Proceedings of the second workshop on multi-relational data mining*, pages 64-76.

Mc Lachlan, G. and Krishnan, T. (1997). The EM algorithm and extensions. Wiley.

Neal, R. and Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse and other variants. *Kluwer Academic Publishers*.

Neville, J. and Jensen, D. (2000). Iterative classification in relational data. *AAAI workshop on learning statistical models from relational data*, pages 42-49.

Ponce, J., Hebert, M., Schmid, C., and Ziserman, A. (2006). Toward category-level object recognition. *Lecture Notes in Computer Science, Springer*.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1) :81-106.

Quinlan, J. (1993). C4.5 : Programs for machine learning. *Morgan Kaufmann Publishers*.

Rosenberg, C., Hebert, M., and Schneidermann, H. (2005). Semi-supervised self-training of object detection models. *Workshop on Application of Computer Vision*, pages 29-36.

Rossiter, J. and Mukai, T. (2007). Bio-mimetic learning from images using imprecise expert information. *Fuzzy Sets and Systems*, 158(3) :295-311.

Schmid, C. (2004). Weakly supervised learning of visual models and its application to content-based retrieval. *International Journal of Computer Vision*, 56 :7-16.

Torralba, A. (2003). Contextual priming for object detection. *International Journal of Computer Vision*, 53(2) :169-191.

Ulusoy, I. and Bishop, C. (2005). Generative versus discriminative methods for object recognition. *International Conference on Computer Vision and Pattern Recognition*, 2 :258-265.

van de Vlag, D. and Stein, A. (2007). Incorporing uncertainty via hierarchical classification using fuzzy decision trees. *IEEE Transaction on GRS*, 45(1) :237-245.

Weber, M., Welling, M., and Perona, P. (2000). Unsupervised learning of models for object recognition. *European Conference on Computer Vision*, 1 :18-32.

Summary

In the field of data mining, depending on the training data complexity, several kinds of classification scheme exist. This paper deals with weakly supervised learning that generalizes the supervised and semi-supervised learning. In weakly supervised learning training data are given as the priors of each class for each sample. We first propose a weakly supervised strategy for learning soft decision trees. Besides, the introduction of class priors for training samples instead of hard class labels makes natural the formulation of an iterative learning procedure. The iterative procedure makes prior refined every iteration. We report experiments for UCI object recognition datasets. These experiments show that recognition performance close to the supervised learning can be expected using the proposed framework. We further discuss the relevance of weakly supervised learning for fisheries acoustics applications.