



HAL
open science

Boosting vs maximum de vraisemblance

Michel Arnaud, Jean-Stéphane Bailly, Christian Puech

► **To cite this version:**

Michel Arnaud, Jean-Stéphane Bailly, Christian Puech. Boosting vs maximum de vraisemblance. Revue Française de Photogrammétrie et de Télédétection, 2002, 12 (4), pp.399-420. hal-01470956

HAL Id: hal-01470956

<https://hal.science/hal-01470956>

Submitted on 18 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Boosting vs maximum de vraisemblance

Michel Arnaud* — Jean Stéphane Bailly** — Christian Puech**

* CIRAD Tera Campus International de Baillarguet TA 60/15 F-34398 Montpellier cedex 5 - michel.arnaud@cirad.fr

** UMR3S CEMAGREF ENGREF Maison de Télédétection, 500 rue J. F Breton F-34093 Montpellier cedex 5 - {[bailly](mailto:bailly@teledetection.fr),[puech](mailto:puech@teledetection.fr)}@teledetection.fr

RÉSUMÉ. L'objectif de cet article est de présenter le Boosting en tant que méthode de classification supervisée et de comparer ses résultats à ceux obtenus par le maximum de vraisemblance. La comparaison se fait à partir d'un exemple théorique issu de télédétection dans lequel on fait varier plusieurs critères : nombre de pixels par classes, chevauchement ou non des radiométries entre classes, avec ou sans structure spatiale des classes dans l'espace géographique. On montre que 1) le maximum de vraisemblance donne de meilleurs résultats que le Boosting lorsque les valeurs radiométriques de chaque classe sont bien séparées. Cet avantage se perd lorsque le nombre de pixels par classe augmente ; 2) le Boosting l'emporte toujours sur le maximum de vraisemblance lorsque les classes radiométriques se chevauchent, qu'il y ait une structure spatiale ou non.

ABSTRACT. This article sets out to demonstrate how boosting can serve as a supervised classification method, and to compare its results with those of conventional methods. The comparison begins with a theoretical example drawn from remote sensing classification in which several criteria are varied: number of pixels per class, overlapping (or not) of radiometric values between classes, with and without spatial structuring of classes within the geographical space. It is seen that 1) maximum likelihood give better results than boosting when the radiometric values for each class are clearly separated. This advantage is lost as the number of pixels per class increases; 2) boosting is systematically better than maximum likelihood in the event of overlapping radiometric variable classes, whether or not there is a spatial structure.

MOTS-CLÉS : classification, boosting, discrimination, maximum de vraisemblance.

KEYWORDS: classification, boosting, discrimination, maximum likelihood.

1. Introduction

Dans de nombreuses applications on cherche à classer un objet dans une classe définie *a priori*. Cette opération s'effectue à partir d'un ensemble d'observations multivariées et la méthodologie utilisée est assez générale : elle consiste à construire une règle de décision qui s'adapte au mieux à cet ensemble de données. L'obtention des règles de décisions peut faire appel à un grand nombre de techniques (Hastie *et al.*, 2001) issues de disciplines très diverses comme la statistique ou l'intelligence artificielle (Ripley 1996). Bien que ces méthodes soient susceptibles d'être utilisées dans un grand nombre de secteurs d'applications (économie, agronomie, sociologie, géographie,...) on n'envisage ici que des applications liées à la télédétection.

Ces dernières années, face à la grande masse des données à analyser et à leur complexité, de nouvelles méthodes ont été développées dans le secteur de la fouille des données (Data Mining) et dans celui de l'apprentissage assisté par ordinateur (Computer Learning). De cette recherche féconde un certain nombre d'algorithmes de classification supervisée ont émergé. Nous allons présenter le 'Boosting' qui nous paraît bien adapté aux problèmes de classification posés en télédétection. Proposé à l'origine par Freund *et al.*, (1996), le 'Boosting' est une procédure permettant d'obtenir une règle de prédiction très précise constituée d'une combinaison linéaire de règles de décisions relativement simples et de précisions modérées.

Bien que récent, cet outil a déjà été utilisé dans un certain nombre d'applications en géomatique : carte d'occupation du sol dans le projet MODIS MOD12 Land Cover¹, et travaux de Briem *et al.*, (2001) pour le classement d'images satellites multi-capteurs. D'autre part, l'étude du comportement du Boosting a été réalisée dans le cas des diverses configurations de deux groupes : frontière linéaire en diagonale avec et sans intrus, bandes parallèles alternées, tore (Arnaud *et al.*, 2002).

Le but de cet article est de présenter les notions de base de l'algorithme du Boosting ; d'étudier le fonctionnement du Boosting sur des exemples théoriques et de comparer les résultats obtenus par le Boosting avec le maximum de vraisemblance.

2. Le Boosting et les méthodes classiques de classification

2.1. Problème, Notations et Algorithme

Problème : On désire classer une image en télédétection. Chaque pixel de l'image est caractérisé par ses valeurs radiométriques, sa position dans l'espace. On suppose qu'on a défini *a priori* K classes d'occupation du sol. Après une reconnaissance terrain qui permet de mettre en relation un échantillon de pixels avec

¹ <http://geography.bu.edu/landcover/userguidelc/lc.html> et http://modis.gsfc.nasa.gov/sci_team/pubs/abstracts/MST-A0426.html

l'une des K classes, il s'agit de trouver une règle de décision qui affecte au mieux les pixels dans l'une de ces classes.

Ensemble de base, test et anonyme : On aura trois sous ensembles : (1) l'ensemble des pixels de **base** avec lesquels on **construit** la règle de décision ; (2) l'ensemble des pixels **test** qui servent à **tester** la règle de décision ; (3) l'ensemble des pixels **anonymes** que l'on désire **classer** dans l'un des K groupes en utilisant la règle de décision trouvée en (1). La présence de l'ensemble test est indispensable : elle permet de valider le modèle de classification.

Notations : On note $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ l'ensemble de base (respectivement $\{(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_n, y'_n)\}$ l'ensemble test) où $\mathbf{x}_i = (x_i^1, x_i^2, \dots, x_i^p) \in \mathbf{R}^p$; x_i^j la valeur observée sur le pixel i pour la variable j ; x^j la variable ; $y_i \in Y = \{1, \dots, K\}$ le numéro du groupe ; K le nombre de groupes.

Définition 1 : on appelle 'apprentisseur' faible une fonction à valeurs réelles :

$$h : \mathbf{R}^p \times Y \rightarrow \mathbf{R} \text{ telle que } (\mathbf{x}, \lambda) \rightarrow h(\mathbf{x}, \lambda) \in \mathbf{R}$$

Les valeurs de $h(\mathbf{x}, \lambda)$ peuvent s'interpréter : (1) si $h(\mathbf{x}, \lambda)$ est positif alors \mathbf{x} est proche du groupe λ et inversement, elle est éloignée du groupe λ lorsque $h(\mathbf{x}, \lambda)$ est négatif ; (2) plus la valeur absolue $|h(\mathbf{x}, \lambda)|$ est forte plus la prédiction est "sûre".

Définition 2 : Etant donné h_1, \dots, h_T un ensemble de T 'apprentisseurs' faibles, on

appelle 'classifieur' associé à l'apprentisseur faible $\sum_{t=1}^T \alpha_t h_t$ la fonction H à valeurs

dans $Y = \{1, \dots, K\}$: $H : \mathbf{R}^p \rightarrow Y = \{1, \dots, K\}$.

$$\mathbf{x} \rightarrow H(\mathbf{x}) = \{\lambda \in Y = \{1, \dots, K\} \text{ tel que } \sum_{t=1}^T \alpha_t h_t(\mathbf{x}, \lambda) \text{ maximum}\}$$

Algorithme du Boosting. L'algorithme initial AdaBoost (Freund *et al.*, 1996) ne prenait en compte que deux groupes. Il a été généralisé par AdaBoost.MH (Schapire *et al.*, 1999) pour répondre à des problèmes de classement plus généraux et notamment multi-classes et multi-labels. C'est cet algorithme que nous avons utilisé dans cet article. Il est présenté en détail dans le tableau 1 ci-après.

Quelques remarques concernant l'algorithme (tableau 1) :

- A chaque itération t, AdaBoost.MH recherche un apprentisseur faible h_t à partir des observations de l'ensemble de base. Le Boosting peut utiliser tout type d'apprentisseur faible tel que les arbres de décision ou les réseaux de neurones. De nombreux essais (Friedman *et al.*, 1998) ont montré que les apprentisseurs faibles reposant sur des arbres de classification à un seul niveau (Breiman *et al.*, 1984) donnaient de bons résultats. L'algorithme AdaBoost.MH repose sur cette méthode : à chaque pas t, on obtient une fonction h_t identifiant à la fois une variable x^j et une valeur réelle s_t^j , appelée seuil, relative à cette variable. Cette valeur permet de construire une partition de l'espace \mathbf{R}^p des variables :

$$\mathcal{D}_0^t = \{\mathbf{x}=(x_1, \dots, x_p) \text{ tel que } x^j < s_t^j\} \text{ et } \mathcal{D}_1^t = \{\mathbf{x}=(x_1, \dots, x_p) \text{ tel que } x^j \geq s_t^j\}$$

Les valeurs prises par l'apprentisseur faible h_t se présente sous la forme :

	Classe 1	K	Classe λ	K	Classe K	
\mathcal{D}_0^t	c_{01}	K	$c_{0\lambda}$	K	c_{0K}	avec $h_t(\mathbf{x}, \lambda) = \begin{cases} c_{0\lambda} & \text{si } \mathbf{x} \in \mathcal{D}_0^t \\ c_{1\lambda} & \text{si } \mathbf{x} \in \mathcal{D}_1^t \end{cases}$
\mathcal{D}_1^t	c_{11}	K	$c_{1\lambda}$	K	c_{1K}	

1. On se donne :

- $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ un ensemble d'entraînement avec :

$\forall i=1, \dots, m : \mathbf{x}_i \in \mathbf{R}^p$ et $y_i \in Y = \{1, \dots, K\}$;

- $\{(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_n, y'_n)\}$ un ensemble test avec :

$\forall i=1, \dots, n : \mathbf{x}'_i \in \mathbf{R}^p$ et $y'_i \in Y = \{1, \dots, K\}$;

- T le nombre d'essais ;

- $D_t(i, \lambda)$ le poids de l'observation i pour le groupe λ au pas t .

- Pour $t=1$ et $\forall i=1, \dots, m$ on pose : $D_1(i, \lambda) = 1/mK$ (poids initiaux).

2. Répéter pour $t = 1$ à T :

(a) En utilisant les poids $D_t(i, \lambda)$, trouver sur l'ensemble d'entraînement l'apprentisseur faible h_t approprié :

$$h_t : \mathbf{R}^p \times Y \rightarrow \mathbf{R} ;$$

(b) Choisir le coefficient $\alpha_t \in \mathbf{R}$;

(c) Calculer :

$$\forall i=1, \dots, m \text{ et } \forall \lambda=1, \dots, K : D_{t+1}(i, \lambda) = \frac{D_t(i, \lambda) \exp\{-\alpha_t y_i(\lambda) h_t(\mathbf{x}_i, \lambda)\}}{Z_t}$$

où : Z_t est un facteur de normalisation choisi de telle manière que D_{t+1} soit une distribution ;

$$\text{et } y_i(\lambda) = \begin{cases} +1 & \text{si } y_i = \lambda \\ -1 & \text{si } y_i \neq \lambda \end{cases}$$

(d) calculer le % de bien classés sur l'échantillon de base et sur l'échantillon

test à l'aide du classifieur associé à $\sum_{i=1}^t \alpha_i h_i$.

3. On obtient le classifieur final $H(\mathbf{x})$ associé au classifieur $\sum_{i=1}^T \alpha_i h_i$:

$$H(\mathbf{x}) = \{\lambda \in Y = \{1, \dots, K\} \text{ tel que } \sum_{t=1}^T \alpha_t h_t(\mathbf{x}, \lambda) \text{ maximum}\}$$

Tableau 1. Algorithme AdaBoost.MH (Schapire et al., 1999)

- Au début du processus, les poids des observations de base sont tous égaux. A la fin de chaque pas, de nouveaux poids sont calculés afin d'augmenter celui des observations mal classées. Le choix du coefficient α_t dépend des valeurs prises par l'apprentisseur faible h_t . Dans le cas où il peut prendre toute valeur réelle on prend $\alpha_t = 1$. Lorsqu'on contraint h_t à prendre des valeurs discrètes (-1 ou +1) on choisit α_t de manière à minimiser le facteur de normalisation Z_t (Schapire *et al.*, 2000) ;
- *Avantages et limites d'AdaBoost.MH.* Il a nombreux avantages : il est rapide en temps calcul, facile à programmer et à utiliser. Le seul paramètre à fixer est T, le nombre total d'itérations. Les poids du Boosting permettent de détecter les données aberrantes (erreur de groupe d'affectation, observation difficile à classer dans l'un des groupes) (Schapire 1999). En revanche, AdaBoost est dépendant des données et peut avoir des résultats peu satisfaisants lorsque on a peu d'observations ou lorsque les données sont 'bruitées' (Dietterich 1999b).
- *Test et comparaison avec d'autres algorithmes :* Adaboost a été testé par de nombreux chercheurs (Dietterich 1999a, Quilan 1996) sur un ensemble de plusieurs jeux de données (Blake *et al.*, 1998) et comparé à d'autres méthodes d'apprentissage comme C4.5 (Quinlan 1993). Il donne généralement les meilleurs résultats.

2.2. La mise en œuvre du Boosting

Elle consiste à examiner certains critères fournis par l'algorithme. (1) Le *pourcentage de bien classés* pour l'échantillon test est un indicateur de la qualité de classement. (2) La *valeur de T* sera celle qui maximise le pourcentage de bien classés de l'échantillon test. (3) On peut étudier l'évolution du *poids de Boosting* $D_t(i, \lambda)$, représentés dans la matrice ci-dessous et d'identifier les observations ayant de fortes valeurs (observations aberrantes, erreur de classement,...) qui exigeront un retour aux données.

$$D_t = \left(\begin{array}{c|cc} & \text{Classe 1} & \text{K} & \text{Classe K} \\ \hline 1 & D_t(1,1) & K & D_t(1, K) \\ M & M & K & M \\ m & D_t(m,1) & K & D_t(m, K) \end{array} \right) \text{ avec } \sum_{i=1}^m \sum_{\lambda=1}^K D_t(i, \lambda) = 1$$

(4) La *valeur de l'apprentisseur faible* $\sum_{t=1}^T \alpha_t h_t$ permet de dresser une carte de

chaque classe $\lambda=1 \dots K$ à partir des colonnes de la matrice ci-dessous. Ces cartes aident à interpréter l'appartenance de chaque pixel dans l'un des K groupes en

tenant compte à la fois du signe de $\sum_{t=1}^T \alpha_t h_t$ mais également de sa valeur absolue.

Pour une classe donnée λ , on pourra ainsi distinguer des pixels ayant une forte chance, une faible chance ou aucune certitude d'être affecté dans cette classe.

$$h_T = \begin{pmatrix} \begin{array}{c|cc} & \text{Classe 1} & \text{Classe K} \\ \hline 1 & \sum_{t=1}^T \alpha_t h_t(1,1) & \sum_{t=1}^T \alpha_t h_t(1,K) \\ \hline M & M & M \\ \hline m & \sum_{t=1}^T \alpha_t h_t(m,1) & \sum_{t=1}^T \alpha_t h_t(m,K) \end{array} \end{pmatrix}$$

(5) Lorsqu'on ne considère que deux variables pour classer les pixels on peut visionner l'*espace des décisions*. (6) L'*importance des variables* est mesurée par la fréquence d'apparition de chacune d'entre elles au pas T.

3. Application du Boosting sur un exemple théorique

Nous avons testé l'algorithme sur plusieurs exemples théoriques. Pour six classes d'occupation du sol on a simulé les valeurs des deux variables XS2 et XS3 en combinant à la fois 1) l'effort d'échantillonnage plus ou moins grand (N=12, 24 et 36 pixels par classe), 2) le chevauchement ou non des classes radiométriques, 3) la prise en compte ou non des coordonnées géographiques, 4) dans deux situations : avec ou sans structure spatiale. L'échantillonnage réalisé est emboîté : les petits échantillons sont contenus dans les grands. Les résultats de ces différentes situations ont été comparés au maximum de vraisemblance.

3.1. Sans coordonnées géographiques

Nous allons présenter les résultats des deux exemples suivants : le premier va examiner la situation dans laquelle les polygones englobants les six classes radiométriques n'ont aucune intersection. Le deuxième considère le cas d'un recouvrement radiométrique des classes. Dans les deux cas on étudie l'effet de l'augmentation du nombre de pixels dans les classes.

3.1.1. Sans chevauchement radiométrique

Dans les 6 classes théoriques considérées (bois, végétations forte, moyenne et faible, sol nu et eau) représentées dans la figure 1, on a tiré au hasard (avec remise) les valeurs radiométriques XS2 et XS3 de respectivement N=12, 24 et 36 pixels pour constituer un échantillon de base et un échantillon test. Lorsque le nombre de pixels par classe N=12, le *pourcentage de bien classés* de l'échantillon de base atteint 100% au pas 8 et celui pour l'échantillon test la valeur maximale de 87,5% au pas 16 (figure 2). Lorsque N est respectivement égal à 24, 36 le taux de bien classés pour l'échantillon test est égal à 89,6 et 92,1 %. Dans l'*espace des décisions* formé par le couple (XS2, XS3) on visualise la partition des classes construite à l'aide des hypothèses faibles (figure 3). Ce graphique permet d'identifier les zones

radiométriques dans lesquelles les pixels tests ou anonymes seront mal classés. On remarque que ce découpage est ajusté très étroitement aux valeurs de l'échantillon de base, c'est ce qui rend la méthode du Boosting très dépendante des données de base. *Les poids de Boosting* : lorsque $N = 12$, on a représenté les valeurs maximum et minimum des poids des pixels calculés à chaque pas (inférieur ou égal à 8) pour la classe sol nu ('e') (figure 4). On distingue le pixel 52 qui a un poids élevé à partir de $t=6$. Il est du à sa situation dans l'espace radiométrique (extrémité gauche de la classe 'e' (figure 1) et très proche de la classe 'a'). L'algorithme va ainsi rechercher des apprennisseurs faibles afin d'affecter le pixel 52 dans le groupe 'e'. Dans les autres classes, on a constaté que les poids des pixels sont relativement peu élevés.

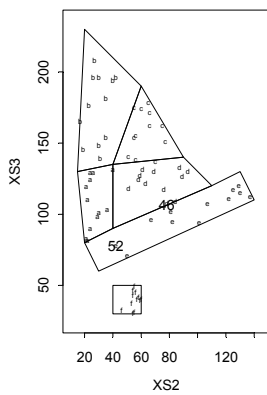


Figure 1. Position des pixels dans l'espace des radiométries ($N = 12$) : échantillon de base

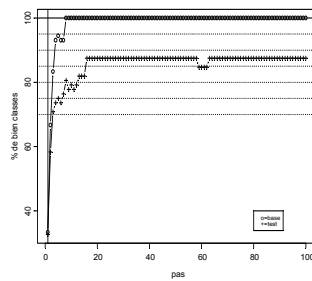


Figure 2. Evolution du pourcentage de bien classés pour l'échantillon de base et l'échantillon test ($N = 12$)

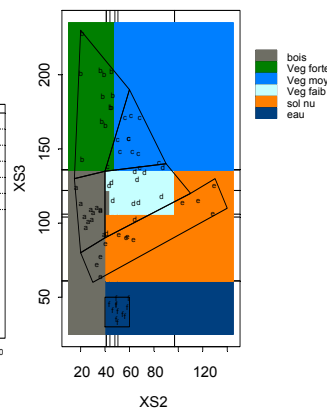


Figure 3. Espace des décisions après le pas 16 ($N = 12$) et pixels de l'échantillon test

3.1.2. Avec chevauchement radiométrique

Lorsque les six classes ont une intersection non vide et se chevauchent dans l'espace radiométrique (figure 5), les résultats du Boosting se dégradent fortement : les taux de bien classés pour l'échantillon test, oscillent autour de 50% : 50%, 50,7% et 48,6% pour $N=12, 24, 32$. En situation de fort chevauchement entre les classes et lorsque les coordonnées géographiques ne sont pas utilisées, l'algorithme du Boosting a de la difficulté à bien classer les observations.

On dispose toujours, outre des valeurs radiométriques attachées à chaque pixel, de sa position dans l'espace géographique. Contrairement au maximum de vraisemblance, le Boosting peut prendre en compte cette information.

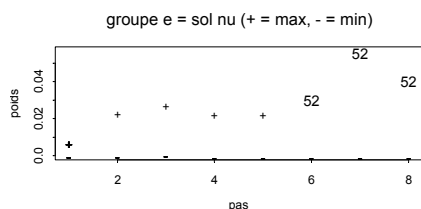


Figure 4. Etude de l'évolution des poids du boosting dans le groupe 'e' jusqu'au pas 8 (N = 12)

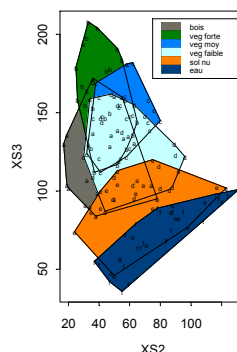


Figure 5. Enveloppe convexe des valeurs radiométriques de chaque classe, pour l'échantillon de base et test (N = 12)

3.2. Avec coordonnées géographiques et structure

Nous allons examiner le comportement du Boosting dans deux situations géographiques bien différenciées : celle d'une structure spatiale très forte et celle d'une situation de répartition quasi-aléatoire des classes.

3.2.1. Sans chevauchement radiométrique

Connaissant les valeurs radiométriques des pixels pour les deux canaux XS2 et XS3, nous avons tiré au hasard (sans remise) la valeur des coordonnées des N=12, 24 et 36 pixels des deux échantillons de base et test dans un espace géographique ayant une structure spatiale bien marquée. Dans la figure 6 ci-dessous on visualise les pixels de l'échantillon de base pour N=12.

Pour l'échantillon test, pour respectivement N=12, 24 et 36, le *pourcentage de bien classés* est égal à 87.5%, 93.8% et 97.2%. Pour N=12, nous avons représenté les valeurs maximum et minimum des *poids des pixels* pour les classes 'd' et 'e' pour lesquelles la valeur maximum est élevée (figure 7). On identifie les deux pixels : 52 et 46. Le pixel 52 a un poids élevé dans la classe sol nu ('e'). Quant au pixel 46, il a un poids élevé aussi bien dans la classe 'd' que 'e'. On voit ici apparaître à travers le pixel 46 un autre effet des poids du Boosting : ils contraignent le pixel 46 à intégrer la classe 'd' et à ne pas appartenir la classe 'e'. Ces deux pixels sont tous deux situés à la frontière des classes aussi bien dans l'espace radiométrique que géographique (figures 1 et 6).

La *contribution* des coordonnées géographiques est relativement faible (20%) quand N=12 ou 24. Elle augmente (37%) lorsque N=36. Donc, en situation de structure spatiale forte et de non chevauchement des classes radiométriques,

l'utilisation des coordonnées géographiques des pixels améliore les résultats de la classification lorsque le nombre de pixels par classe n'est pas trop faible.

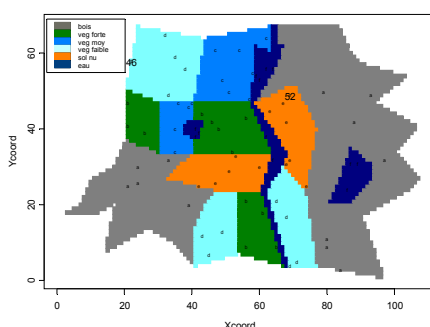


Figure 6. Position des pixels de l'échantillon de base dans l'espace géographique ($N=12$)

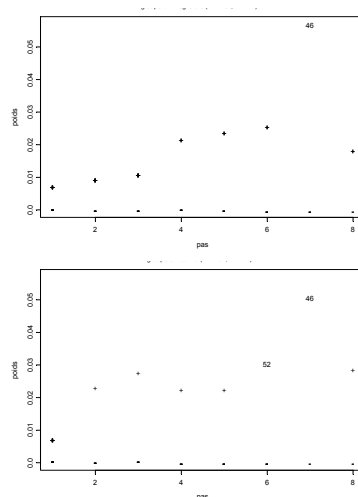


Figure 7. Evolution des poids du boosting jusqu'au pas 8. En haut classe 'd' ; en bas classe 'e' ($N=12$) (max=+, min=-)

3.2.2. Avec chevauchement radiométrique

Avec une structure spatiale, les *pourcentages de bien classés* sur l'échantillon test sont moins bons avec chevauchement que sans chevauchement mais bien meilleurs que ceux obtenus avec chevauchement sans l'utilisation des coordonnées géographiques (cf tableau 2). L'algorithme du Boosting compense relativement bien le chevauchement des valeurs radiométriques en utilisant les coordonnées géographiques. Lorsqu'il y a chevauchement des valeurs radiométriques, les coordonnées géographiques *contribuent* pratiquement autant que les valeurs radiométriques. En situation de structure spatiale forte et de chevauchement des classes radiométriques, l'utilisation des coordonnées géographiques des pixels améliore les résultats de la classification quel que soit le nombre de pixels par classe.

3.3. Avec coordonnées géographiques et sans structure spatiale

Nous avons tiré au hasard (sans remise) les coordonnées des $N=12, 24$ et 36 pixels des deux échantillons de base et test emboîtés dans un espace géographique n'ayant pas de structure spatiale. Cet espace est constitué par la répartition aléatoire de 36 grandes parcelles contiguës représentant chacune des 6 classes d'occupation du sol ayant 6 répétitions (figure 8).

3.3.1. Sans chevauchement radiométrique

On retrouve pratiquement les mêmes *taux de bien classés* (cf tableau 2) que ceux obtenus dans le cas où on ne prend pas en compte l'information géographique et lorsque les classes radiométriques sont sans chevauchement (cf. §3.1.1). Les *poids de Boosting* donnent aussi les mêmes résultats qu'au §3.1.1. Les coordonnées géographiques n'interviennent pas lorsque $N=12$ et ont une contribution respectivement égale à 14 % et 27,7% lorsque $N=24, 36$.

En situation d'absence de structure spatiale et de non chevauchement des classes radiométriques, l'utilisation des coordonnées géographiques des pixels n'améliore pas les résultats de la classification.

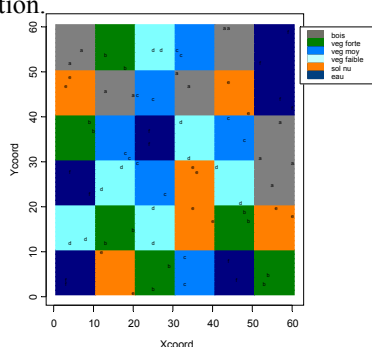


Figure 8. Représentation des pixels de chaque classe dans le cas d'une structure aléatoire ($N=12$)

3.3.2. Avec chevauchement radiométrique

Les *pourcentages de bien classés* sont améliorés (cf. tableau 2) lorsqu'on prend en compte l'information spatiale (cf. § 3.1.2). Lorsqu'il y a chevauchement des valeurs radiométriques, les coordonnées géographiques *contribuent* en relation directe avec N avec respectivement 45%, 54,5% et 55,1% quand $N=12, 24, 36$. En situation d'absence de structure spatiale et de chevauchement des classes radiométriques, l'utilisation des coordonnées géographiques des pixels améliore les résultats de la classification. Cependant l'amélioration est bien meilleure lorsqu'il y a une structure spatiale.

Dans les diverses situations rencontrées précédemment, la méthode du maximum de vraisemblance a été utilisée. Le tableau 2 donne les résultats comparatifs des deux techniques. On constate que, du point de vue du pourcentage de bien classés pour l'échantillon test, 1) dans l'ensemble, le Boosting donne généralement de bien meilleurs résultats lorsqu'on utilise l'information spatiale (coordonnées des pixels) sauf dans le cas où, simultanément, la structure spatiale est absente et les classes radiométriques ne se chevauchent pas ; 2) le maximum de vraisemblance donne de meilleurs résultats que le Boosting dans le cas où les valeurs radiométriques de chaque classe sont bien séparées. Cet avantage se perd lorsque le nombre de pixels par classe augmente ; 3) le Boosting l'emporte toujours sur le maximum de

vraisemblance lorsque les classes radiométriques se chevauchent, qu'il y ait une structure spatiale ou non ;

		N	Boosting		Maximum de vraisemblance (Sans utilisation des coordonnées géographiques)	
			Sans utilisation des coordonnées géographiques	Avec utilisation des coordonnées géographiques		
				Structure spatiale		Structure aléatoire
radiométrie	sans chevauchement	12	87.5	87.5	87.5	95.8
		24	89.6	93.8	89.6	93.8
		36	92.1	97.2	93.5	95.4
	avec chevauchement	12	50.0	69.4	62.5	41.7
		24	50.7	81.9	69.4	48.6
		36	48.6	82.9	76.9	47.2

Tableau 2. Comparaison des trois méthodes sur l'échantillon test

5. Conclusion

Les résultats montrent que le boosting est équivalent au maximum de vraisemblance lorsque les coordonnées géographiques ne sont pas prises en compte. En revanche, le Boosting présente un avantage sauf dans le cas de non chevauchement des classes radiométriques. Cet avantage est dû, en grande partie, à la valorisation des coordonnées géographiques par le Boosting lorsqu'il existe une structure spatiale. La souplesse de l'algorithme permet en outre de prendre en compte non seulement la radiométrie et les coordonnées géographiques mais aussi d'autres informations susceptibles d'améliorer la classification (carte des sols,...). Il permet de dresser la cartographie de la qualité de la classification de chaque pixel. On peut identifier, pour chaque groupe λ , trois types de pixels : ceux pour lesquels on a une forte **certitude de classer** dans le groupe λ , ceux pour lesquels on a une **forte certitude de ne pas classer** dans le groupe λ , enfin ceux pour lesquels **l'appartenance au groupe λ est incertaine**. Si les résultats du Boosting s'avèrent supérieurs, ils demandent néanmoins à l'utilisateur une certaine prudence dans son utilisation : recours obligatoire à l'échantillon test pour valider la règle de décision obtenue (car le boosting est très dépendant de l'échantillon de base) ; examen des poids pour détecter les éventuelles données aberrantes ou intruses. Une faiblesse de la méthode vient du fait qu'elle ne permet pas une démarche explicative : elle ne cherche pas à trouver les variables permettant d'expliquer la différence des groupes. Néanmoins, on peut utiliser la contribution de chaque variable au pas T. Il reste à comparer les résultats sur une application réelle, travail en cours.

Bibliographie

- Arnaud M., Bailly J.S., Bourgeon G., Puech C., 2002, Le Boosting : une méthode de classification non paramétrique. *Revue internationale de géomatique*. Volume 12 – n° 4/2002. (sous presse).
- Blake C.L., Merz C.J., 1998, UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mlearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
- Breiman L., Friedman J. H., Olshen R. A., Stone C. J., 1984, CART. Classification and regression trees, Wadsworth and Brooks/Cole (Advanced books and Software, Monterey, California).
- Briem G. J., Benediktsson J. A., Sveinsson J. R., 2001, Boosting, bagging and consensus based classification of multisensor remote sensing data. 2nd Workshop on Multiple Classifier Systems. Cambridge Juillet 2001.
- Dietterich T. G., 1999a, Machine learning research: Four current directions. *AI Magazine*, 18(4), pp. 97-136.
- Dietterich T. G., 1999b, An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, Boosting and randomization.. *Machine learning*. pp. 1-22.
- Freund Y., Schapire R. E., 1996, Experiments with a new Boosting algorithm. In Saitta, L. (Ed.), *Proceedings of the Thirteenth International Conference on Machine Learning*, San Francisco, CA. Morgan Kaufmann. pp. 148-156.
- Friedman J., Hastie T., Tibshirani R., 1998, Additive Logistic Regression : a Statistical View of Boosting. Technical Report.
- Hastie T., Tibshirani R., Friedman J., 2001, *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer-Verlag.
- Quinlan J. R., 1993, *C4.5 : Programs for Empirical Learning*. Morgan Kaufmann. San Francisco, CA.
- Quinlan J. R., 1996, Bagging, Boosting and C4.5 : In *Proceedings of Thirteenth National Conference on Artificial Intelligence*. Cambridge, MA. AAAI Press/MIT Press. pp. 725-730.
- Ripley B.D., 1996, *Pattern Recognition and Neural Networks*, Cambridge University Press.
- Schapire R.E., 1999, Theoretical views of boosting and applications. In *Tenth International Conference on Algorithmic Learning Theory*.
- Schapire R. E., Singer Y., 1999, Improved boosting algorithms using confidence-rated predictions, *Machine Learning*, 37, 297-336.
- Schapire R. E., Singer Y., 2000, BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39, 135-168.