

# **Sélection aléatoire d'espaces de représentation pour la décision binaire en environnement non-stationnaire: application à la segmentation d'images texturées**

Pierre Beauseroy\*, Andre Smolarz\*, Xiyan He\*

\*Institut Charles Delaunay (FRE 2848)

Université de Technologie de Troyes, 12, Rue Marie Curie, BP 2060, 10010 Troyes Cedex  
{pierre.beauseroy , andre.smolarz , xiyan.he}@utt.fr

**Résumé.** Nous proposons dans cet article une méthode de sélection d'espaces de représentation, dans le but d'optimiser ou de préserver les performances d'un système décisionnel en présence de bruit, de perte d'information ou de non-stationnarité. Cette méthode consiste à définir tout d'abord un espace de représentation le plus exhaustif possible, correspondant aux attributs les plus appropriés pour porter les informations utiles au problème traité. Ensuite on sélectionne au hasard des sous-espaces de dimension réduite, obtenus par projection de l'espace initial. La segmentation d'images texturées constitue une application tout à fait appropriée pour illustrer cette méthode et évaluer ses performances. Nous traitons ici un problème à deux classes de textures pour lequel il s'agit de choisir le meilleur espace de représentation en termes de décision aux frontières entre deux classes. Le principe de la méthode consiste à évaluer, par apprentissage, les performances d'un classifieur donné pour chaque espace de représentation sélectionné. Ensuite, l'étape finale de segmentation est effectuée sur une image composée de deux classes de textures. La décision est prise sur la base d'un vote pondéré des décisions prises par le classifieur dans chaque espace de représentation. Nous présentons quelques résultats qui nous semblent justifier la démarche adoptée et nous concluons sur les perspectives qu'ils nous inspirent.

## **1 Introduction**

Pour tout problème d'apprentissage d'une règle de décision à partir d'une base de données contenant des exemples étiquetés, l'espace de représentation définit la perception du système à surveiller.

Un grand nombre de travaux a été consacré à la réduction du nombre d'attributs composant l'espace de représentation (John et al., 1994; Koller et Sahami, 1996; Blum et Langley, 1997; Kohavi et John, 1997; Torkkola, 2003; Guyon et Elisseeff, 2003; Cantu-Paz et al., 2004). L'objectif est alors de concentrer l'information discriminante présente dans l'espace de représentation initial à l'aide d'un nombre restreint d'attributs. La réduction de la dimension de l'espace de représentation peut permettre de supprimer les attributs redondants, de réduire l'impact du

## Sélection aléatoire d'espaces de représentation pour la décision

bruit, de définir un nombre limité d'indicateurs qui ont une valeur explicative pour le problème étudié (Kohavi et John, 1997; Torkkola, 2003; Mutelo et al., 2006). Cette opération peut aussi être une réponse au problème bien connu sous le nom de *malédiction de la dimensionnalité* (Bellman, 1961; Koller et Sahami, 1996). Cette malédiction provient du fait, qu'à précision comparable, le nombre d'exemples nécessaires pour estimer une densité de probabilité croît de façon exponentielle avec la dimension de l'espace de représentation. La nécessité de réduire la dimension de l'espace de représentation peut aussi provenir de motifs plus contingents comme, par exemple, la nécessité de compresser l'information pour satisfaire des contraintes de bandes passantes dans des systèmes distribués (Duarte et Hu, 2004).

La réduction de la dimension peut-être obtenue soit par extraction soit par sélection d'attributs. Dans les deux cas, l'hypothèse de départ est que l'on dispose d'un espace de représentation initial constitué de mesures qui sont regardées comme des variables aléatoires stationnaires. Il existe un grand nombre d'approches différentes permettant de réaliser cette opération. Cet aspect sera détaillé dans la section 2.

L'approche développée dans cette publication est un peu différente. Il s'agit de traiter le cas où certains attributs de l'espace de représentation initial peuvent brutalement changer de comportement, ce qui a pour conséquence d'altérer ou de transformer le modèle marginal qui les représente. Cet aspect sera développé au paragraphe 3.2. Les changements de comportement considérés peuvent provenir de fluctuations brutales du rapport signal à bruit de certaines mesures de l'espace de représentation initial, de panne de certains capteurs... Dans ce cas, une règle de décision apprise dans un contexte de fonctionnement donné n'est plus valide et produira de très mauvaises performances. Deux approches peuvent être envisagées. La première consiste à détecter les changements de distributions des observations de l'espace de représentation, à neutraliser le dispositif de décision lorsque les mesures sortent de leur espace de définition initial puis à émettre une alarme. La seconde approche consiste à tenter de concevoir un dispositif de décision qui, par nature, est robuste vis à vis de ce type de perturbations. Nous nous sommes attelés à l'étude du second problème dans le contexte de la segmentation d'image texturée qui permet de simuler facilement ce type de situations.

Une texture est couramment définie comme une région d'une image dans laquelle l'observation à travers une fenêtre bien choisie se traduit par une perception visuelle invariante par translation à l'intérieur de la région considérée. En accord avec cette définition, on peut conclure que la texture se manifeste donc par une information visuelle fortement redondante. L'état d'un pixel (niveau de gris par exemple) pouvant être considéré comme la réalisation d'un variable aléatoire, une texture sera alors caractérisée, d'un point de vue statistique, par la stationnarité des propriétés locales de l'image. L'échelle minimale pour laquelle ces propriétés de stationnarité sont vérifiées, définit la taille du voisinage à considérer pour caractériser un point de l'image. Ce voisinage produit donc un ensemble de mesures qui forment l'espace de représentation initial. Dans ce contexte une texture sera alors décrite par la densité de probabilité conjointe des variables aléatoires associées aux pixels du voisinage qui la caractérise.

A l'approche d'une frontière entre deux textures, le voisinage d'un point contient des fragments (blocs de pixels de formes variées) provenant de différentes textures. La densité de probabilité conjointe ne correspond donc plus à aucun des modèles caractérisant chacune des textures considérées et l'espace de représentation est donc brutalement perturbé. Ce problème se traduit généralement par une grande difficulté à caractériser de façon précise les frontières des textures. Le choix d'un voisinage spatialement isotrope (typiquement une fenêtre carrée) de

grande taille garantit une bonne description de chacune des textures, mais entraîne en contrepartie une mauvaise localisation des frontières. La réduction de la taille du voisinage permet alors souvent d'améliorer la détection de frontières, mais cette fois, au détriment des performances globales de classification (Beauseroy et Smolarz, 2004).

La solution proposée dans cette publication s'inspire des méthodes de classification de type *random forest* (Breiman, 1996a; Dietterich, 1998, 2000). Ces approches consistent à combiner un grand nombre de classificateurs qui sont individuellement peu performants, mais dont la combinaison permet d'obtenir de bonnes performances.

Nous proposons ici une méthode qui repose sur un ensemble de classificateurs des plus proches voisins dont l'apprentissage est réalisé dans des espaces de représentation constitués aléatoirement à partir d'un voisinage isotrope suffisamment grand et dont la taille est déterminée à partir de la population d'apprentissage. Une stratégie de décision particulière est mise en œuvre pour contrôler l'influence des perturbations provoquées par les frontières.

Cet article est organisé de la façon suivante : dans la section 2 nous présentons les méthodes de sélection d'attributs dans un cadre général, puis dans le contexte de l'apprentissage par agrégation de modèles. La section suivante introduit la méthode de sélection aléatoire d'espaces de représentation et de décision (SAER) proposée. Les résultats expérimentaux sont présentés et discutés dans la section 4. Finalement, une conclusion et quelques perspectives de travail sont discutées dans la dernière section.

## 2 Sélection d'espaces de représentation

### 2.1 Généralités

Dans le cadre des problèmes de décision ou de caractérisation, la détermination d'espaces de représentation efficaces a pour objectif de permettre de discriminer au mieux les classes du problème traité. Ce processus comporte schématiquement deux étapes principales. La première étape est consacrée au recueil des données et en l'absence d'informations précises sur la nature des meilleures mesures à acquérir, il est fréquent de partir d'un ensemble de mesures relativement exhaustif, donc de grande dimension, pour essayer de capter le plus d'informations possibles. La seconde étape consiste à réduire autant que possible la dimension du problème en préservant le potentiel discriminant des mesures initiales.

Deux approches non exclusives peuvent être considérées pour traiter ce problème de réduction de dimension :

- l'extraction d'attributs, qui consiste à déterminer l'ensemble restreint de mesures discriminantes composant l'espace de représentation par transformation des mesures de départ,
- la sélection d'attributs qui consiste à choisir un nombre limité de mesures ou d'attributs dans l'ensemble de départ.

Dans les deux cas, il s'agit de conserver au mieux l'information discriminante et de supprimer le bruit, les mesures inutiles, redondantes...

Pour ce qui nous concerne, la méthode que nous proposons repose sur la sélection d'espaces de représentation au sein d'un espace initial.

## Sélection aléatoire d'espaces de représentation pour la décision

La réduction de la dimension de l'espace de représentation initial peut être obtenue en sélectionnant un sous-ensemble d'attributs. Cette approche est très largement employée, en particulier lorsque les attributs de l'espace initial ont une valeur explicative que l'on souhaite conserver ou bien quand la disponibilité de toutes les mesures ne peut pas être garantie. La sélection de  $d$  attributs parmi  $m$  en effectuant une recherche exhaustive requiert l'examen de  $\binom{m}{d}$  sous-ensembles d'attributs. Beaucoup de travaux ont été menés pour trouver des algorithmes permettant de déterminer rapidement le meilleur sous-ensemble. Il a malheureusement été démontré (Cover et Campenhout, 1977) qu'un tel algorithme devait pouvoir examiner tous les sous-ensembles pour être certain de sélectionner le meilleur. Trois méthodes sont très souvent employées selon diverses variantes. La méthode descendante qui, partant de l'ensemble de représentation initial complet, consiste à évaluer un critère de performance pour tous les sous-ensembles comportant un attribut de moins puis à répéter cette opération à partir du sous-ensemble ayant donné le meilleur résultat au sens du critère choisi.

Au contraire, avec la méthode ascendante c'est le meilleur attribut, au sens du critère, qui sert de point de départ. Chaque étape consiste ensuite à sélectionner le meilleur sous-ensemble d'attributs obtenu, à construire tous les nouveaux espaces de représentation possibles par ajout d'un attribut à ce dernier, puis à évaluer les nouveaux espaces et à conserver le meilleur. Ces deux méthodes sont rapides mais sous-optimales dans le cas général.

La méthode de *branch-and-bound* (Narendra et Fukunaga, 1977) permet de déterminer la solution optimale du problème et, dans bien des cas, de réduire considérablement l'étendue de la recherche. Elle impose l'emploi d'un critère  $C$  monotone par rapport au nombre d'attributs considérés. Le respect de cette propriété rend inutile l'évaluation de tous les sous-ensembles déduits par suppression d'attributs à partir d'un espace  $\mathcal{X}^k$  quand  $C(\mathcal{X}^k) < C(\mathcal{X}^d)$  avec  $k > d$  et  $\mathcal{X}^d$  étant le meilleur espace de représentation obtenu pendant la recherche jusqu'à l'instant courant.

D'autres méthodes permettent de réaliser la sélection de variables. Les méthodes de type *wrappers* (John et al., 1994) (Kohavi et John, 1997) utilisent directement la méthode d'apprentissage pour mesurer les performances de différents sous-ensembles d'attributs et sélectionner le meilleur. Il existe diverses stratégies pour le choix des sous-ensembles évalués. Le sous-ensemble d'attributs sélectionné est donc adapté au classifieur utilisé. Le *lasso* (Tibshirani, 1996) formule le problème de sélection comme un problème de régression avec une contrainte dépendant d'un paramètre. Ce paramètre permet de garantir la parcimonie et de régler la sélectivité de la solution.

Des méthodes récentes abordent le problème de réduction de dimension de l'espace de représentation sans définir d'objectif en terme de séparabilité des distributions. Il s'agit de représenter les données dans un espace de dimension réduite en partant de l'idée que l'espace des observations est une variété de  $\mathbb{R}^m$ . Ces méthodes s'appuient principalement sur des assemblages de modèles linéaires locaux (*Locally Linear Embedding*) (Roweis et Saul, 2000) ou sur l'utilisation de graphes pour modéliser la variété et tenter de conserver dans l'espace de dimension réduite les relations de distances entre observations mesurées le long des géodésiques de l'espace initial (Tenenbaum, 1998).

## 2.2 Méthodes d'agrégation

Un moyen connu pour améliorer la robustesse de classification consiste à utiliser des méthodes de décision par agrégation. Opitz (1999) indique que le but de la sélection des attributs dans le contexte de l'agrégation est non seulement de trouver les attributs les plus appropriés pour l'algorithme d'apprentissage, mais aussi d'augmenter la diversité des classificateurs. Zenobi et Cunningham (2001) montrent que la diversité au sein d'un ensemble de règles de décision contribue à réduire l'erreur de généralisation. Par similitude avec les méthodes de rééchantillonnage aléatoire (bagging, bootstrap...), Ho (1998b) montre que la sélection aléatoire de sous-ensembles d'attributs est une technique efficace. La méthode qu'il propose, intitulée "random subspace method" (RSM), consiste à construire un ensemble de règles sur la base d'une méthode de classification et d'un choix aléatoire d'espaces de représentation. Dans Ho (1998b,a) la décision finale est obtenue par vote. Pour un ensemble d'apprentissage donné, il montre ainsi expérimentalement que l'utilisation d'un grand nombre de classificateurs définis dans des espaces de petite dimension peut être préférable à l'apprentissage d'un classificateur dans l'espace initial de grande dimension.

Plusieurs travaux ont été menés sur la base de ce principe. Opitz (1999) propose une méthode de sélection d'attributs au moyen d'un algorithme génétique pour un ensemble de réseaux de neurones. La méthode des forêts aléatoires proposée par Breiman (2001) est une combinaison d'arbres de décision. Breiman montre que l'erreur de la généralisation d'une forêt converge presque sûrement lorsque le nombre d'arbres est très grand. Tsymbal et al. (2002) présentent un algorithme analogue bâti sur des classificateurs Bayesiens.

Bay (1999) propose une approche appelée "multiple feature subsets" (MFS) qui combine plusieurs classificateurs des plus proches voisins basé, comme pour RSM, sur une définition aléatoire des sous-espaces de représentation. Il montre expérimentalement que sa méthode est concurrentielle avec les méthodes "arcing" (Breiman, 1996b). Cette méthode sera employée à titre de comparaison avec celle que nous proposons.

## 3 Présentation de la méthode SAER

### 3.1 Principe général

Soit une base d'apprentissage composée de couples  $(\mathbf{x}_i, y_i)$  où  $\mathbf{x}_i$  est une observation et  $y_i$  son label. La méthode proposée consiste dans un premier temps à tirer au sort un nombre fini d'espaces de représentation noté  $\mathcal{X}_\ell^d$  constitués de  $d$  attributs pris parmi  $m$  ( $\ell$  correspond au numéro de l'espace considéré). La valeur de  $d$  est choisie arbitrairement dans ce cas, mais tout autre scénario est envisageable. Une base d'apprentissage est ensuite constituée pour chaque espace de représentation  $\mathcal{X}_\ell^d$  et une règle de décision correspondante  $h_\ell$  définie. Chaque observation  $\mathbf{x}$  est classée une première fois suite à un vote de toutes les règles  $h_\ell(\mathbf{x})$ . Un label temporaire  $y_{temp}(\mathbf{x})$  lui est attribué après comptabilisation des scores  $q_k(\mathbf{x})$  de chacune des classes  $\omega_k$ . Le score d'une classe est déterminé en faisant la somme pondérée des voix des règles qui se sont prononcées en faveur de cette classe. Le poids  $w_k^\ell$  associé à la classe  $\omega_k$  et à la règle de décision  $h_\ell$  est défini en fonction des performances de la règle de décision évaluées à l'aide d'un ensemble de test.

A l'issue de cette première classification, les observations (tous les pixels de l'image) sont examinées à nouveau par un sous-ensemble de règles de décisions, sélectionnées en fonction

## Sélection aléatoire d'espaces de représentation pour la décision

de la qualité inférée par l'espace de représentation qui leur est associé. Le principe est le suivant. Si dans le  $\ell$ ème espace  $\mathcal{X}_\ell^d$  l'observation  $x$  obéit à l'une des lois conditionnelles  $P(x|\omega_i)$  alors, la règle de décision définie dans cet espace est conservée. En revanche, si certaines composantes de  $x$  sont perturbées, le modèle probabiliste décrivant  $X$  n'appartient pas à l'ensemble  $\{P(x|\omega_1), P(x|\omega_2) \dots\}$ . La règle de décision associée à cet espace est alors écartée.

Le test permettant de décider si un espace de représentation est perturbé ou non est défini en fonction des décisions prises pour chaque observation lors de leur première classification et il est effectué pour chaque espace. Le test consiste alors à observer si toutes les composantes de  $X$  ont reçu le même label lors de la classification initiale. Si c'est le cas, on considère qu'il n'y a aucune composante perturbée et la règle correspondante est conservée. Dans le cas contraire, la règle est éliminée. Seul le résultat de la seconde classification  $y_{final}(x)$  est conservé. Dans le cas de la segmentation d'image, ce test présente certaines spécificités qui sont décrites ci-dessous. La figure 1 présente schématiquement le principe de la méthode proposée.

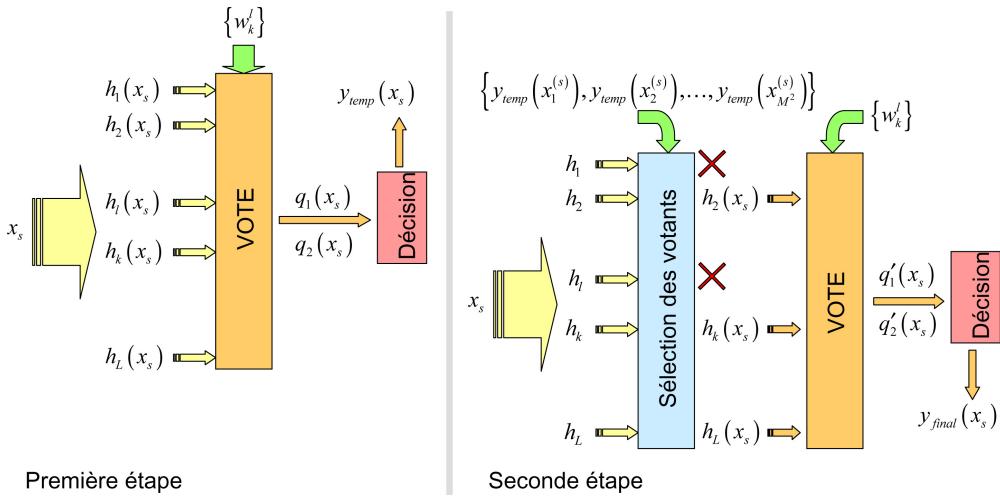


FIG. 1 – *Principe de la méthode SAER. Sur cette figure, les  $x_i^{(s)}$   $i = 1, M^2$  représentent tous les voisins de  $x_s$  définis dans l'espace de représentation initial.*

## 3.2 Application à la segmentation d'images texturées

Dans le cadre de la segmentation d'images texturées, les observations sont les niveaux de gris de chaque pixel et la décision concerne la classe de texture à laquelle appartient le pixel. Nous ferons l'hypothèse que l'information utile à la représentation et à la caractérisation d'un pixel dans une texture est contenue dans un voisinage de ce pixel et nous nous attachons donc à la recherche d'attributs locaux. Formellement, nous allons donc attacher une variable aléatoire  $X_s$  à chaque pixel  $s$ , qui prendra ses valeurs dans l'ensemble  $\Omega$  des niveaux de gris et chaque variable "voisine" de  $X_s$  représentera un attribut de l'espace de représentation (précisons que  $X_s$  appartient à l'espace de représentation).

Si l'on considère un voisinage carré  $M \times M$  centré sur chaque pixel, le pixel  $s$  sera ainsi représenté dans l'espace  $\mathcal{X} = \Omega^{M \times M}$ , par  $\mathbf{X}^{(s)} = [X_1^{(s)}, X_2^{(s)}, \dots, X_{M^2}^{(s)}]^t$ , un vecteur aléatoire d'attributs. Notre méthode sélectionne alors plusieurs espaces de représentation qui seront obtenus par des projections aléatoires de  $\mathcal{X}$  sur des sous-espaces  $\mathcal{X}_\ell^d$  de dimension  $d < M^2$ . Dans l'espace  $\mathcal{X}_\ell^d$ , le pixel  $s$  sera représenté par le vecteur d'attributs  $\mathbf{X}_\ell^{(s)} = [X_{\ell,1}^{(s)}, X_{\ell,2}^{(s)}, \dots, X_{\ell,d}^{(s)}]^t$  comme l'illustrent les exemples de la figure 2. Ensuite, l'algorithme construit autant de classificateurs, notés  $h_\ell$  (correspondant à chaque  $\mathcal{X}_\ell^d$ ), à partir de l'ensemble d'apprentissage et nous noterons  $L$  le nombre total de classificateurs.

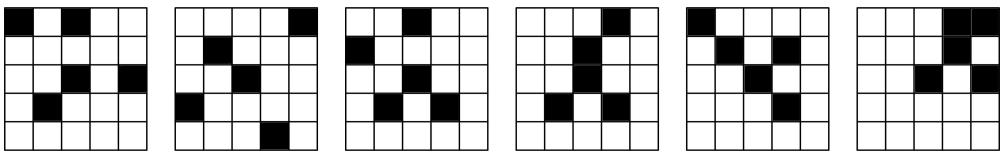


FIG. 2 – Pour un voisinage donné représenté par une grille ( $M = 5$ ), différents espaces de représentation obtenus par projection aléatoire - les attributs sélectionnés sont figurés par des cases noires.

Pour nos expériences, nous avons utilisé la règle des plus proches voisins et nous nous sommes limités à un problème de classification à deux classes de texture.

Par analogie avec les méthodes agrégatives de type *random forest*, la décision de classement est donc prise à la suite d'un vote pondéré des différentes règles de décision selon le principe suivant. Pour chaque règle  $h_\ell$ , les probabilités de bonnes décisions  $w_k^\ell$ , conditionnellement à chaque classe  $k$  sont utilisées pour définir les poids du vote pour chacune des hypothèses, selon la formule suivante :

$$q_k(x_s) = \sum_{l=1}^L w_k^\ell \mathbb{1}_{h_\ell(x_s)=k} \text{ où } k \in 1, 2 \text{ représente l'indice de la classe} \quad (1)$$

Les probabilités  $w_k^\ell$  sont estimées au préalable par le taux de pixels bien classés, calculé à l'aide d'un ensemble test. L'ensemble test est bien entendu constitué d'images contenant une seule classe de texture.

Dans le cas d'une image à segmenter, la décision initiale en chaque point de l'image est prise en faveur de la classe qui réalise le meilleur score défini en (1) en considérant toutes les règles de décision. Un résultat illustrant cette étape est présenté sur la figure 3 (c).

Sur la base de cette segmentation initiale, on sélectionne les règles de décision selon le principe décrit au paragraphe 3.1.

Typiquement, pour une observation  $\mathbf{x}_\ell^{(s)} = [x_{\ell,1}^{(s)}, x_{\ell,2}^{(s)}, \dots, x_{\ell,d}^{(s)}]^t$ ,

Sélection aléatoire d'espaces de représentation pour la décision

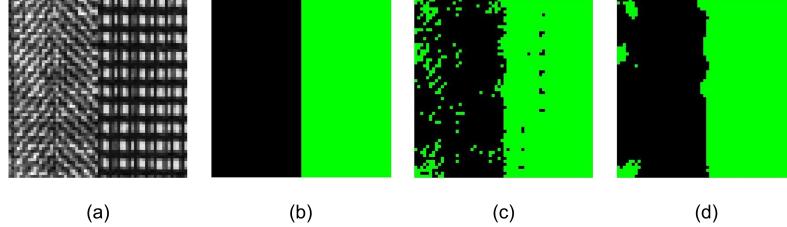


FIG. 3 – *Image test (a) - segmentation vraie (b) - classification initiale (c) - classification finale (d).*

$$\begin{cases} \text{si } h_\ell(x_{\ell,i}^{(s)}) = k \forall i = 1, d \text{ et } i \neq s \text{ (puisque la classe du pixel } s \text{ est inconnue).} \\ \text{La règle } h_\ell \text{ est conservée (donc l'espace } \mathcal{X}_\ell^d \text{)} \\ \text{sinon } \text{elle est éliminée du processus de décision concernant } \mathbf{x}_\ell^{(s)} \end{cases}$$

Notre démarche repose sur l'hypothèse qu'au cœur des régions, la texture est homogène ou stationnaire et que, par conséquent, les pixels d'un voisinage ont une très forte probabilité de recevoir le même label lors de la segmentation initiale. Dans les zones proches des frontières, on peut en revanche s'attendre à observer une distribution beaucoup moins homogène des labels sur un voisinage.

Pour évaluer les résultats, nous sommes donc amenés à distinguer deux zones caractérisant les régions homogènes ou "cœur de région" et les frontières selon que :

**Le voisinage du pixel à classer ( $s$ ) est constitué de pixels de la même classe (cœur de région).**

Dans le cas de deux classes  $\omega_1$  et  $\omega_2$  le modèle du vecteur d'attributs sera donné par :

$$f_{\mathbf{X}^{(s)}}(\mathbf{x}^{(s)} | \omega_1) \quad \text{ou bien} \quad f_{\mathbf{X}^{(s)}}(\mathbf{x}^{(s)} | \omega_2) \quad (2)$$

**Le pixel à classer ( $s$ ) est au voisinage d'une frontière.** Dans ce cas, il y a rupture de modèle selon la représentation ci-dessous :

$$f_{\mathbf{X}^{(s)}}(\mathbf{x}^{(s)} | \omega_1 \cup \omega_2) = f_{\mathbf{X}^{(s)}}([\mathbf{x}_1^{t(s)}, \mathbf{x}_2^{t(s)}]^t | \omega_1 \cup \omega_2) \quad (3)$$

Avec  $\mathbf{x}_1^{t(s)} \in \omega_1$  et  $\mathbf{x}_2^{t(s)} \in \omega_2$

### 3.3 Algorithme de la méthode

**Algorithme SEAR**

**Variables :**

$\mathbf{X}_{\text{app}}$ ,  $\mathbf{X}_{\text{test}}$ ,  $\mathbf{X}_{\text{Im}}$  : des observations d'apprentissage, de test et les ob-

servations de l'image à segmenter dans l'espace de représentation  $\mathcal{X}$

$\mathbf{X}_{\text{app}}(\mathbf{p})$  par exemple désigne l'observation  $\mathbf{x}_p$  de l'ensemble d'appren-

tissage,

$\mathbf{L}$  : le nombre d'espaces de représentations,

**K** : le nombre de classes,  
**d** : la dimension des espaces sélectionnés,  
**Y<sub>app</sub>, Y<sub>test</sub>** les labels des ensembles d'apprentissage et de test,  
**V<sub>Im</sub>** : V<sub>Im</sub>(s) donne les numéros des voisins du pixel central de  $x^{(s)}$ ,  
**LX<sub>app</sub>, LX<sub>test</sub>, LX<sub>Im</sub>** : des observations d'apprentissage, de test et les observations de l'image à segmenter dans les espaces de représentation  $\mathcal{X}_\ell^d \forall \ell = 1 \dots L$   
**LX<sub>app</sub>(ℓ)** désigne par exemple les individus de l'ensemble d'apprentissage vus dans l'espace de représentation  $\mathcal{X}_\ell^d$ ,  
**LX<sub>app</sub>(ℓ, p)** représente la  $p^{ième}$  observation  $\mathbf{x}_\ell^{(p)}$  de l'ensemble d'apprentissage vue dans l'espace de représentation  $\mathcal{X}_\ell^d$ ,  
**Attrib\_Indice(ℓ)** : les indices des attributs de  $\mathcal{X}$  présents dans  $\mathcal{X}_\ell^d$ ,  
**Y<sub>temp</sub>, Y<sub>final</sub>** : la classification temporaire et finale

**Instructions :**

**Pour ℓ allant de 1 à L faire**

Tirer au hasard les indices de  $d$  attributs pour former l'espace de représentation  $\mathcal{X}_\ell^d$   
Mémoriser les indices dans *Attrib\_Indice(ℓ)*  
Construire  $LX_{app}(\ell)$ ,  $LX_{test}(\ell)$  et  $LX_{Im}(\ell)$

**FinPour**

**Pour ℓ allant de 1 à L faire**

Apprendre la règle  $h_\ell$  à l'aide de  $LX_{app}(\ell)$   
Appliquer  $h_\ell$  à  $LX_{test}(\ell)$  et mémoriser le résultat dans  $Y$   
Determiner  $w_\ell^k$  pour  $k = 1 \dots K$  en comparant  $Y$  et  $Y_{test}$

**FinPour**

[Début de la première classification

**Pour** chaque point  $p$  de l'image **allant de 1 à P faire**

Initialiser  $q_k$  à zéro  $\forall k = 1 \dots K$

**Pour ℓ allant de 1 à L faire**

Appliquer  $h_\ell$  à  $LX_{Im}(\ell, p)$   
Ajouter  $w_\ell^k$  à  $q_k$  où  $k$  est l'indice de la classe choisie par  $h_\ell$

**FinPour**

$Y_{temp}(p) = \operatorname{argmax}_k \{q_k\}$

**FinPour**

[Fin de la première classification et début de la seconde

**Pour** chaque point  $p$  de l'image **allant de 1 à P faire**

Initialiser  $q_k$  à zéro  $\forall k = 1 \dots K$

**Pour ℓ allant de 1 à L faire**

**Si**  $Y_{temp}(V_{Im}(p, \operatorname{Attrib\_Indice}(\ell)))$  sont tous identiques **Alors**

Appliquer  $h_\ell$  à  $LX_{Im}(\ell, p)$   
Ajouter  $w_\ell^k$  à  $q_k$  où  $k$  est l'indice de la classe choisie par  $h_\ell$

**Finsi**

Sélection aléatoire d'espaces de représentation pour la décision

```

FinPour

$$Y_{final}(p) = \operatorname{argmax}_k \{q_k\}$$

FinPour
[Fin de la seconde étape de classification]
Fin Algorithme SEAR

```

## 4 Etude expérimentale

### 4.1 Présentation du problème

Pour évaluer la méthode proposée, nous traitons plusieurs problèmes de segmentation d'images texturées. Les images sont extraites de la base de textures de l'album de Brodatz (Brodatz, 1966) d'une part et de synthèses de textures obtenues par simulation d'un champ de Markov (Smolarz, 1997) (figure 4 et figure 5). Le tableau 1 décrit l'origine des images.

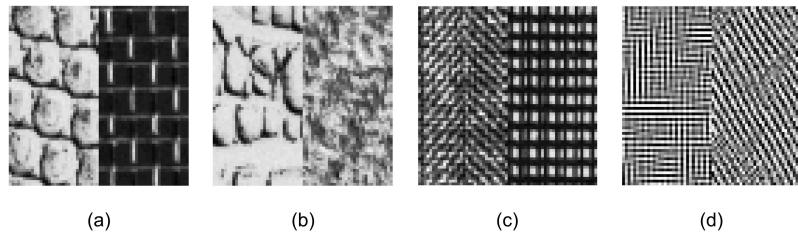


FIG. 4 – Images texturées utilisées dans les tests avec frontière simple.

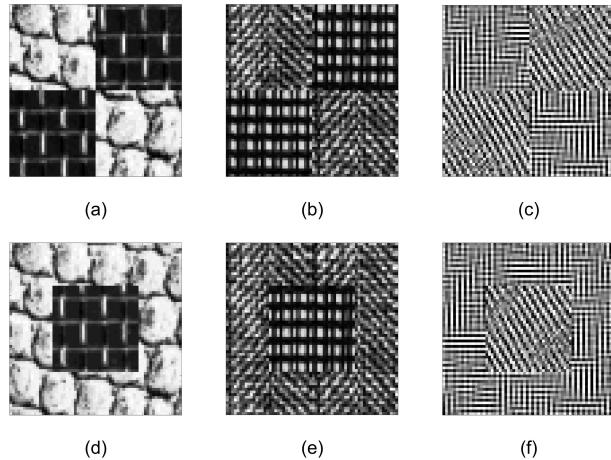


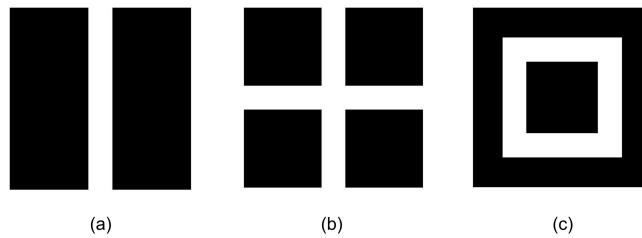
FIG. 5 – Images texturées utilisées dans les tests avec frontière complexe.

Image	source
FIG. 4(a), FIG. 5(a) et 5(d)	D 3 et D 6 de Brodatz (1966)
FIG. 4(b)	D10 et D11 de Brodatz (1966)
FIG. 4(c), FIG. 5(b) et 5(e)	D16 et D21 de Brodatz (1966)
FIG. 4(d), FIG. 5(c) et 5(f)	Textures markoviennes (Smolarz, 1997)

TAB. 1 – *Les images contenant deux textures utilisée dans les expériences.*

Toutes les textures sont monochromes et quantifiées sur 256 niveaux de gris. Les images sont de taille de  $640 \times 640$  pour l’album de Brodatz et de  $512 \times 512$  pour les images synthétisées. Les ensembles d’apprentissage et d’évaluation sont construits par partition de ces images en fenêtres de taille  $64 \times 64$ . Les images de test ont la même taille que celles d’apprentissage. Les images à segmenter contiennent toujours 2 classes de textures différentes représentées par des régions aux frontières variées.

En accord avec la présentation du problème faite à la section 3.2, deux types de régions sont considérés. Les différentes formes de régions que nous avons choisies ainsi que les zones frontières sont représentées sur la figure 6. Les taux d’erreur sont estimés conditionnellement à chaque type de zone. Nous noterons  $Errh$ , le taux d’erreur dans les cœurs de régions et  $Errf$  le taux d’erreur dans la zone frontière.

FIG. 6 – *Représentation des "cœurs de régions" (en noir) et des zones frontières (en blanc) correspondant aux images des figure 4 et 5.*

## 4.2 Définition de l'espace de représentation initial

Selon les caractéristiques d’échelle d’une texture, la taille minimum du voisinage qui décrit la stationnarité locale, joue un rôle crucial dans la définition du modèle. En l’absence d’information a priori, on choisit souvent un voisinage isotrope (carré  $M \times M$ ). Par conséquent, afin de déterminer le voisinage définissant l’espace de représentation initial, nous avons fait plusieurs essais de classification pour des tailles de voisinage  $M = 3, 5, 7, 9$  sur les images présentées à la figure 4. Pour juger de la qualité de l’espace de représentation initial de dimension  $M^2$ , on en extrait au hasard  $L = 50$  sous-espaces de dimension  $M$ . La dimension retenue sera celle qui tend à minimiser le taux d’erreur  $Errh$  dans les cœurs de régions. Les résultats sont présentés dans le tableau 2 et illustrés par un exemple (figure 7). On peut constater que

## Sélection aléatoire d'espaces de représentation pour la décision

les dimensions  $M = 5$  et  $M = 7$  se partagent les meilleures résultats. On utilisera donc pour la suite un espace initial de dimension 25 ( $M = 5$ ).

M	3	5	7	9
FIG. 4(a)	1.72	0.064	0.375	0.76
FIG. 4(b)	14.5	15.2	17.7	18.4
FIG. 4(c)	6.9	3.88	1.87	6.52
FIG. 4(d)	0.44	0.13	0.56	1.16

TAB. 2 – Taux  $Errh$  estimés (%) pour différentes dimensions d'espace de représentation.

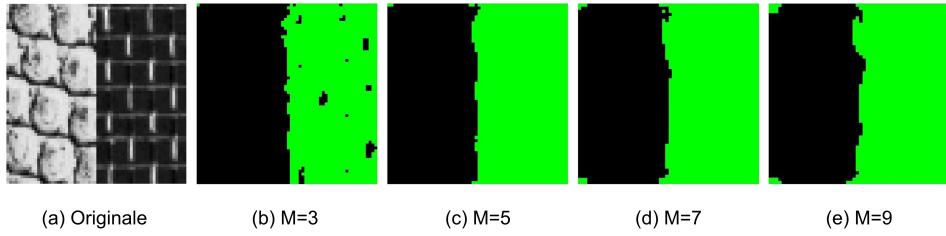


FIG. 7 – Exemple de segmentation pour différentes valeurs de  $M$ .

### 4.3 Choix du nombre de sous-espaces de représentation extraits aléatoirement

Le nombre d'espaces de représentation  $L$  est un paramètre qu'il nous faut fixer. Là encore, en l'absence d'a priori, nous avons procédé à quelques essais avec différentes valeurs de  $L$  (25,50,75,100,125,150,200). La figure 8 présente l'évolution des taux d'erreur  $Errh$  et  $Errf$  en fonction de  $L$  pour l'image test de la figure 4(a). Le comportement observable sur cette figure est commun à tous les couples images testés. Le taux d'erreur dans les coeurs de région est toujours très faible indépendamment des valeurs testées pour  $L$  (avec  $L \geq 25$ ). En revanche, le taux d'erreur dans la zone frontière diminue fortement lorsque  $L$  augmente au delà de 50. A partir de 75, le taux d'erreur reste relativement stable. Afin de choisir un bon compromis entre les performances souhaitées et le temps d'exécution, nous avons retenu la valeur  $L = 100$ .

### 4.4 Analyse des résultats

Le tableau 3 présente les résultats obtenus sur les images de test dans la figure 4 et la figure 5 en utilisant trois méthodes différentes :

- La méthode du plus proche voisin appliquée dans l'espace de représentation initial de dimension 25 (taille de voisinage  $5 \times 5$ ),
- La méthode MFS (Bay, 1999)
- Notre méthode SAER.

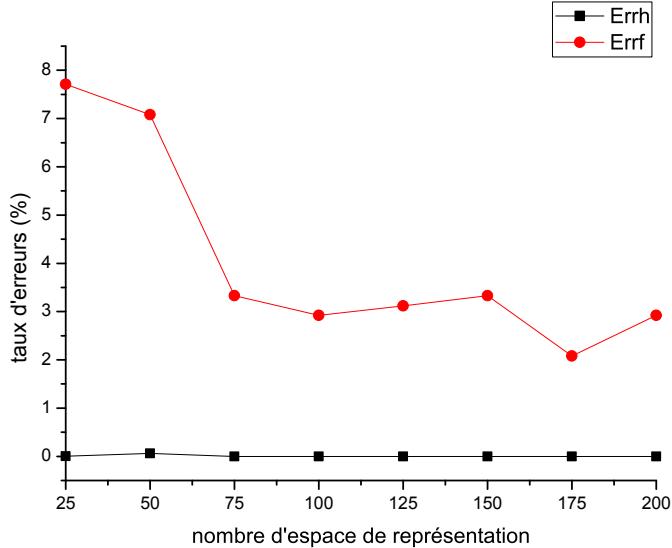


FIG. 8 – Les taux d’erreur pour différents nombres d’espace de représentation (test sur l’image de la figure 4(a)).

Dans le tableau 3, nous présentons les 2 taux d’erreur, le taux d’erreur dans les coeurs de régions et le taux d’erreur dans la zone frontière. Les meilleurs résultats en termes de taux d’erreurs dans la zone frontière sont présentés en gras. Nous présentons aussi graphiquement le taux d’erreur *Errf* pour les trois méthodes afin de mieux illustrer l’apport de notre méthode pour la décision aux frontières (figure 9)

Dans le tableau 3, ainsi que sur la figure 9, on peut constater que pour 8 des 10 tests, la méthode SAER a le taux d’erreur *Errf* le plus faible. L’amélioration la plus marquée est de 66.8% par rapport au résultat des ppv et est obtenue pour l’image de la figure 5(a). Cela illustre bien l’intérêt de disposer d’espaces de représentation adaptés aux situations de changement de modèle. On peut noter également que la méthode MFS donne des performances qui se situent, en moyenne, entre celles de notre approche et celles des ppv. Par ailleurs, la méthode SAER donne globalement de meilleurs résultats que les deux autres au niveau des taux d’erreur dans les coeurs de régions, comme on peut le voir en analysant les résultats reportés dans le tableau 3 et en observant les figures 10 et 11.

Concernant le temps de calcul, en prenant le temps de traitement d’une image par la méthode des ppv comme unité de mesure, le temps nécessaire pour la méthode MFS est de 8,8. Le temps d’apprentissage des poids  $w_\ell^k$  de la méthode SAER est de 17,4 et le temps de classement à proprement parler est de 11,9 ce qui est comparable à la méthode MFS. Ces résultats ont été établis à partir de programmes exécutés en matlab. Ils montrent que le temps de traitement n’est pas un obstacle à la mise en œuvre de la méthode proposée.

## Sélection aléatoire d'espaces de représentation pour la décision

Image	ppv		MFS		SAER	
	Errh	Errf	Errh	Errf	Errh	Errf
FIG. 4(a)	0.224	5.42	0	4.79	0	<b>2.92</b>
FIG. 4(b)	23.1	23.1	17.3	21.7	14.5	<b>20.8</b>
FIG. 4(c)	2.34	12.5	5.16	10	2.34	<b>7.08</b>
FIG. 4(d)	0.353	<b>7.71</b>	0.481	<b>7.71</b>	0	8.13
FIG. 5(a)	0.513	29.6	0.0961	25.6	0.074	<b>9.82</b>
FIG. 5(b)	1.73	<b>11.96</b>	4.36	13.8	1.48	14.4
FIG. 5(c)	0.444	7.59	0.666	5.92	0.074	<b>4.35</b>
FIG. 5(d)	0.427	11.5	0.116	10.5	0.116	<b>8.69</b>
FIG. 5(e)	5.59	10.4	9.01	12.1	6.02	<b>9.67</b>
FIG. 5(f)	0.66	8.79	0.74	7.52	0.116	<b>6.15</b>

TAB. 3 – Comparaison des taux d'erreur (%) obtenus avec différentes méthodes.

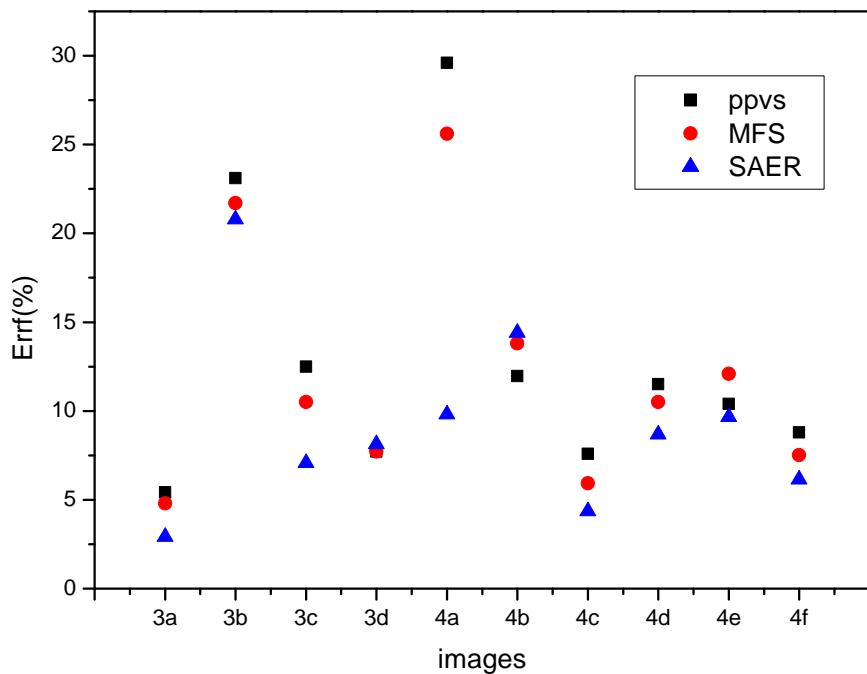


FIG. 9 – Représentation, pour chaque méthode, du taux d'erreur dans les zones frontières sur l'ensemble des images testées.

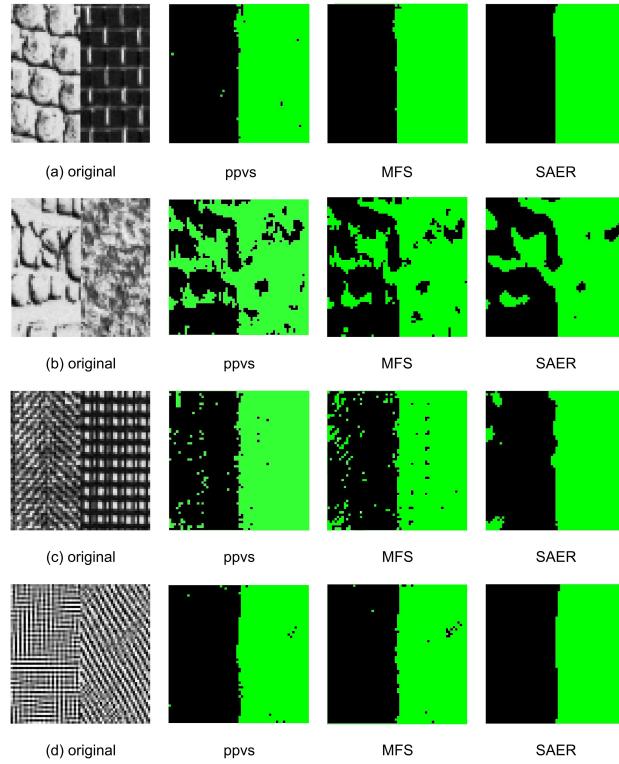


FIG. 10 – Résultats de la classification par les méthodes des ppvs, MFS et SAER pour les images de la figure 4.

Sélection aléatoire d'espaces de représentation pour la décision

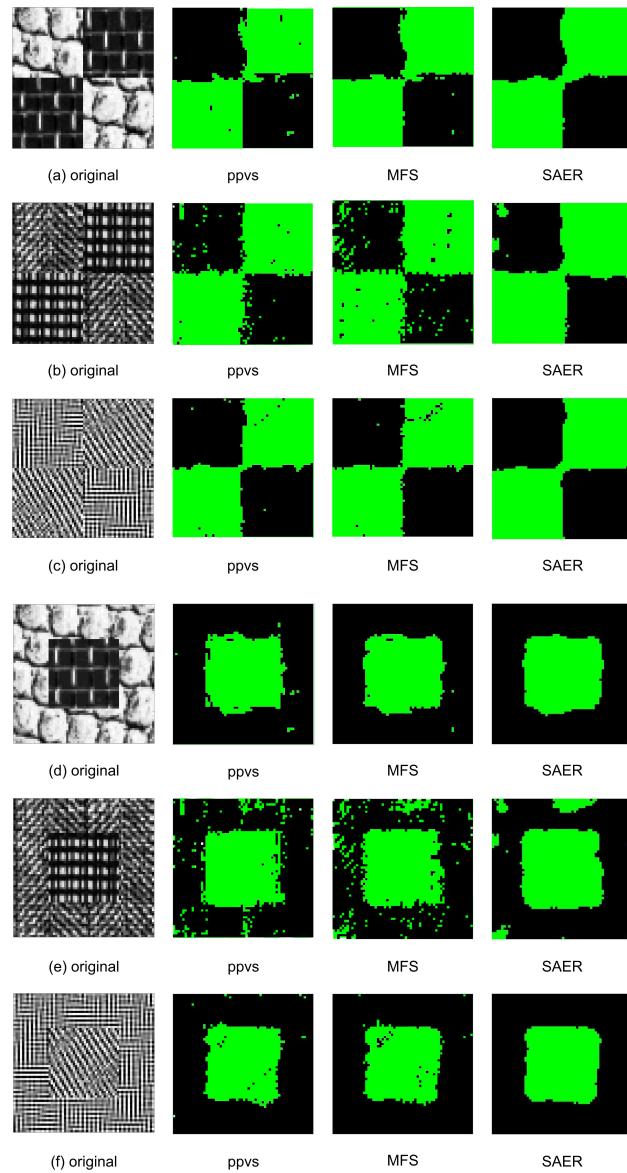


FIG. 11 – Résultats de la classification par les méthodes des ppv, MFS et SAER pour les images de la figure 5.

## 5 Conclusion et perspectives

Nous avons proposé et testé une méthode de sélection aléatoire d'espaces de représentation (SAER). Comme nous l'avons précisé en préambule, notre démarche n'a pas pour objet de proposer une nouvelle méthode de segmentation d'images, mais plutôt d'étudier l'intérêt d'une sélection adéquate d'espaces de représentation dans le but d'améliorer les performances d'un système décisionnel face à une rupture ou à une dégradation du modèle décrivant les observations conditionnellement aux classes. Le problème de la segmentation d'images texturées nous a paru constituer un exemple tout à fait approprié pour illustrer notre démarche et tester sa pertinence. Pour cette raison, les résultats que nous présentons ne sont pas comparés à ceux que l'on obtiendrait avec d'autres méthodes de segmentation, mais à ceux que l'on obtient avec d'autres systèmes de sélection d'espaces de représentation. Bien entendu, comme pour toute méthode de sélection d'attributs, les espaces sélectionnés sont de dimension raisonnable. Nous nous sommes limités au cas de 2 classes de texture, afin de ne pas introduire des difficultés supplémentaires inhérentes au traitement des problèmes multiclass (population des ensembles d'apprentissage, taille des images pour estimer les performances dans de bonnes conditions...).

Le principe de la méthode SAER consiste à prendre la décision relative à une observation en deux temps. Ses performances ont été comparées à celles obtenues avec la méthode du plus proche voisin, associée à l'espace contenant tous les attributs initiaux et la méthode MFS qui combine plusieurs sous-ensembles d'attributs aléatoires. Nous avons pu observer que la méthode SAER donnait des résultats bien meilleurs que les deux autres en terme de taux d'erreur dans les zones frontières entre deux classes, c'est-à-dire là où il y a rupture de modèle. Ces résultats sont très encourageants et permettent de proposer quelques pistes à explorer.

Dans l'approche proposée, les sous-ensembles d'attributs sont sélectionnés de façon aléatoire dans l'espace de représentation initial et environ 100 espaces sont nécessaires pour obtenir de bonnes performances avec les exemples traités. Les travaux en cours concernent la construction d'un ensemble de sous-ensembles d'espaces bien choisis de façon à mieux maîtriser le processus de généralisation mis en œuvre dans cette approche. L'utilisation de différents types de classifieurs (Support Vector Machine, Kernel Fisher Discriminant...) fera aussi l'objet d'une étude approfondie.

## Références

- Bay, S. D. (1999). Nearest neighbor classification from multiple feature subsets. *Intelligent Data Analysis* 3(3), 191–209.
- Beauseroy, P. et A. Smolarz (2004). Optimisation de la géométrie du voisinage pour la segmentation d'images texturées. Montpellier, France. XXXVIèmes Journées de statistiques.
- Bellman, R. (1961). *Adaptive Control Processes : a guided tour*. Princeton : Princeton University Press.
- Blum, A. et P. Langley (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence* 97(1-2), 245–271.
- Breiman, L. (1996a). Bagging predictors. *Machine Learning* 24(2), 123–140.

## Sélection aléatoire d'espaces de représentation pour la décision

- Breiman, L. (1996b). Bias, variance, and arcing classifiers. Technical report, Statistics Department, University of California.
- Breiman, L. (2001). Random forests. *Machine Learning* 45(1), 5–32.
- Brodatz, P. (1966). *Textures : A Photographic Album for Artists and Designers*. New York : Dover Publications.
- Cantu-Paz, E., S. Newsam, et C. Kamath (2004). Feature selection in scientific applications. In *ACM International Conference on Knowledge Discovery and Data Mining*, pp. 788–793.
- Cover, T. et J. V. Campenhoult (1977). On the possible orderings in the measurement selection problem. *IEEE Transaction on Systems, Man, and Cybernetics* 7(9), 657–661.
- Dietterich, T. G. (1998). An experimental comparison of three methods for constructing ensembles of decision trees : Bagging, boosting, and randomization. *Machine Learning* 28, 1–22.
- Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science* 1857, 1–15.
- Duarte, M. F. et Y. H. Hu (2004). Distance-based decision fusion in a distributed wireless sensor network. *Telecommunication Systems* 26(2-4), 339–350.
- Guyon, I. et A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182.
- Ho, T. K. (1998a). Nearest neighbors in random subspaces. In *SSPR '98/SPR '98 : Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*.
- Ho, T. K. (1998b). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8), 832–844.
- John, G. H., R. Kohavi, et K. Pfleger (1994). Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121–129.
- Kohavi, R. et G. H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence* 97(1-2), 273–324.
- Koller, D. et M. Sahami (1996). Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 284–292.
- Mutelo, R. M., L. C. Khor, W. L. Woo, et S. S. Dlay (2006). Two-dimensional reduction pca : a novel approach for feature extraction, representation, and recognition. In *Proc. of the SPIE-IS and T Electronic Imaging, Visualization and Data Analysis*, Volume 6060, pp. 60600E.
- Narendra, P. et K. Fukunaga (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers* 26(9), 917–922.
- Opitz, D. W. (1999). Feature selection for ensembles. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pp. 379–384.
- Roweis, S. et L. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(22), 2323–2326.

- Smolarz, A. (1997). Etude qualitative du modèle auto-binomial appliquée à la synthèse de texture. In *Proceedings of the XXIXèmes Journées de Statistique*, pp. 712–715.
- Tenenbaum, J. B. (1998). Mapping a manifold of perceptual observations. In *Advances in neural information processing systems 10*, pp. 682–688.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* 58(1), 267–288.
- Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research* 3, 1415–1438.
- Tsymbal, A., S. Puuronen, et D. W.Patterson (2002). Ensemble feature selection with the simple bayesian classification. In *Foundations of Intelligent Systems :ISMIS*, Volume 2366, pp. 592–600.
- Zenobi, G. et P. Cunningham (2001). Using diversity in preparing ensembles of classifiers based on differnt feature subsets to minimize generalization error. In *Proc : ECML 2001 12th European Conf. on Machine Learning, LNCS*, Volume 2167, pp. 576–587. Springer.

## Summary

A method for feature subsets selection is presented. It tends to optimize or preserve the performances of a decisional system in case of noise, loss of information or nonstationary perturbations. The method consists first in defining a representation space which is as exhaustive as possible, corresponding to the features that carry out useful information to discriminate classes. Second, random projections are used to define a set of subspaces of reduced size. A classifier is trained for each representation space. The decision is made on the basis of a weighted vote. A two steps process is used. The initial decision is obtained while all classifiers are participating to the vote. The final decision is taken according to the vote of selected voters. To illustrate this approach, the context of textured images segmentation is considered. Problems of image segmentation in the case of two classes of texture are studied. Our attention is focused on the question of choosing the best set of representation space in terms of decision at the borders between two classes. Compared to two other methods, the obtained results seem to justify the proposed approach. Finally some future working directions are pointed out.