Classification Semi-Supervisée basée sur des Algorithmes de CAH multi-métriques

Fabien Carmagnac, Pierre Héroux, Eric Trupin

Classification Semi-Supervisée basée sur des Algorithmes de CAH multi-métriques

Fabien Carmagnac[1,2] – Pierre Héroux[1] – Éric Trupin[1]

[1] Laboratoire PSI - CNRS FRE 2645 - Université de Rouen
UFR des Sciences et Techniques
F-76 821 Mont-Saint-Aignan cedex

[2] A2iA SA
40 bis rue Fabert
75 007 Paris cedex - France

Fabien.Carmagnac@a2ia.com
{Pierre.Heroux,Eric.Trupin}@univ-rouen.fr

Résumé : *Cet article présente un système de classification semi-supervisée d'images de documents destiné à être intégré dans un logiciel commercial de lecture automatique de documents. Ce système se revendique comme une aide à l'annotation de bases d'apprentissage. À partir d'un ensemble d'images inconnues donné par un opérateur, le système établit des hypothèses de regroupement d'images ayant la même structure physique et les propose à l'opérateur. Ce dernier peut les corriger ou les valider, l'objectif pour celui-ci étant d'avoir au final, des groupes homogènes pour l'apprentissage du classifieur supervisée du logiciel. Des arbres de classification ascendante hiérarchique sont construits pour chaque jeu de caractéristiques disponible. Les propositions de regroupements formulées par les différentes CAH sont confrontées et fusionnées. Des résultats, évalués par le nombre de corrections effectuées, sont présentés sur différentes bases d'images.*

Mots-clés : Classification non supervisée, images de document, aide à l'annotation de base d'apprentissage.

1 Introduction

Les récents progrès des techniques de traitements d'images ont conduit à l'émergence d'applications qui automatisent le traitement de documents. À partir de l'image scannée d'un document, de nombreux logiciels sont maintenant capables d'en lire le contenu manuscrit ou typographié ou d'identifier certains symboles ou logos. D'autres peuvent retrouver la catégorie (appelée classe par la suite) à laquelle appartient le document à traiter. Cependant, une phase d'apprentissage préalable est nécessaire. Pour ce faire, un opérateur humain donne pour chaque classe des exemples d'images de même structure physique. Ces images proviennent généralement d'un flux de documents représentatif de celui qu'il faudra trier après l'apprentissage.

Par exemple, le traitement du courrier postal entrant dans les entreprises permet d'acheminer un document entrant vers le bon service ou d'appliquer un traitement particulier en fonction de la classe à laquelle appartient le document [KOC 03] [CLA 00]. Cependant, ces logiciels ne sont pas encore capables d'extraire d'eux-mêmes toute l'information contenue sur les images de documents et un opérateur humain doit définir les tâches à accomplir par le logiciel sur les images en fonction de leur classe.

L'approche que nous proposons dans cet article, étend les fonctionnalités d'une application existante (A2IA FieldReader). À l'origine, cette application permettait la lecture des champs manuscrits ou typographiés sur des documents provenant d'un flux homogène, tous les documents partageant le même modèle de lecture. Une première extension, a permis de traiter des documents provenant de plusieurs classes en dotant le système d'un module de classification supervisé, qui, après une étape d'apprentissage, permet de reconnaître le type d'un document entrant. Le modèle de lecture associé contenant des informations sur la position, la nature et la sémantique des champs à lire, pilote alors la lecture des champs. Le produit étant commercialisé auprès d'utilisateurs n'étant pas experts en analyse de documents, le module de classification doit automatiquement déterminer les caractéristiques les plus discriminantes pour effectuer la classification, et ce, quelle que soit la nature des images, et le nombre de classes. Une autre difficulté provient du fait que l'annotation de bases d'apprentissage est un processus fastidieux voire quasi-impossible si la base est constituée d'images provenant de plusieurs dizaines de classes et d'autant plus si des images de classes différentes ont des variations de structures très légères. Les bases d'apprentissage ne sont alors constituée que de très peu d'exemples. Une autre de nos communications à CIFED'04 propose une méthode de classification supervisée tente de répondre à ces contraintes [CAR 04].

Dans cet article, nous proposons un système de classification semi-supervisée inpiré de Muslea et al. [MUS 02] dont l'objectif d'offrir une aide à l'annotation de base de documents sans connaissance *a priori* du nombre et de la nature des classes. Il est alors difficile de connaître quelles sont les caractéristiques discriminantes. À partir d'un ensemble d'images de documents provenant d'un flux hétérogène, le système propose à l'opérateur des regroupement d'images

partageant la même structure physique. L'opérateur peut valider ou corriger ces propositions via une interface. Diverses corrections sont proposées : la fusion ou la scission semi-automatique de regroupements, l'ajout ou la suppression de document dans les regroupements proposés.

En section 2, nous présentons brièvement quelques méthodes de classification non-supervisée et justifions notre choix de l'algorithme de classification ascendante hiérarchique. Nous décrivons ensuite notre algorithme de classification semi-supervisée en section 3. Enfin, des résultats sur cinq bases d'images différentes sont présentés en section 4. Les conclusions et perspectives sont développées en section 5.

2 Les algorithmes de classification non supervisée

Un état de l'art sur la classification non supervisée peut être trouvé dans [FUK 90], [JAI 00] et [COR 02]. Nous rappelons ici les principales méthodes.

L'algorithme des nuées dynamiques fournit la meilleure partition possible d'un ensemble E en k groupes d'éléments bien agrégés et bien séparés entre eux. Or notre système doit fonctionner sans la connaissance *a priori* du nombre de classes attendues car l'opérateur ne le connaît pas lui-même.

Les cartes de Kohonen (appelées également SOM pour Self Organizing Map) basées sur un maillage de neurones avec une contrainte de voisinage n'ont pas besoin de l'*a priori* sur le nombre de classes attendues pendant l'apprentissage. Cependant, un grand nombre d'exemples est nécessaire pour faire converger le système. Notons d'ailleurs que cette convergence n'est pas garantie pour des vecteurs de caractéristiques de dimension supérieure à 1.

La classification ascendante hiérarchique (C.A.H.) est un algorithme permettant d'obtenir une hiérarchie de parties sur les données considérées et présente l'intérêt de proposer une structuration des données sans connaissance du nombre de classes attendues [BEN 73] [DID 80]. Le résultat est un arbre où chaque nœud représente un groupe et la racine contient l'ensemble des éléments. Divers critères existants permettent de couper certains arcs de l'arbre et de former ainsi des groupes avec les éléments contenus dans les nœuds fils des arcs coupés [RIB 98].

Parmi ces trois méthodes classiques de classification non-supervisée, la CAH parait être la plus adéquate pour résoudre notre problème. En effet, les cartes de Kohonen manqueront d'exemples pour la convergence et les nuées dynamiques supposent un *a priori* sur le nombre de classes attendues. Cependant, ces trois méthodes fonctionnent à partir de données numériques extraites sur les images de la base d'apprentissage. Ces images, souvent bruitées, vont évidemment introduire une variance dans les caractéristiques. Pour corriger les erreurs dûes à cette variance, l'introduction d'un niveau sémantique serait adéquat comme l'extraction des objets graphiques bien identifiés (cases à cocher, peignes, titres à gros caractères, etc.). Cette solution introduit un biais que l'on s'interdit car cela conduirait à développer une grande base d'extracteurs d'objets graphiques concurrents. Notre idée est donc d'avoir plusieurs espaces de caractéristiques dans lesquels seront projetées les images et de construire un arbre de CAH pour chaque espace. Avoir un grand vecteur de caractéristiques, résultat de la concaténation de plusieurs vecteurs ramène au problème évoqué plus haut. On obtiendra donc autant de CAH que de jeux de caractéristiques. Ces caractéristiques sont différentes : visuelles (détection de zones d'images très blanches ou très sombres, niveau de gris moyen de l'image, etc), structurelles simples (comptage de traits horizontaux ou verticaux, extraction de rectangles quelconque, etc.) et statistiques (variations de tailles de masse de pixels connexes, etc.). Chaque CAH émettra des hypothèses de regroupement qui seront combiner afin de trouver les groupes à présenter à l'opérateur.

3 Algorithme de CAH multi-métrique

3.1 Quelques définitions

Soient $FeatureSet$ l'ensemble des jeux de caractéristiques et $ImageSet$ l'ensemble des images qui constitueront la base d'apprentissage. Pour chaque espace de caractéristiques E, une fonction F_E qui projette une image dans l'espace E est définie par :

$$E \in FeatureSet$$
$$F_E : ImageSet \rightarrow E$$

Pour chaque espace de caractéristiques E, une fonction M_E qui mesure la distance entre deux points de E est définie par :

$$E \in FeatureSet$$
$$M_E : E \times E \rightarrow \mathbb{R}^+$$

Pour chaque espace de caractéristiques E, une fonction D_E qui mesure la distance entre deux images $ImageSet$ est définie par :

$$E \in FeatureSet$$
$$(I_1, I_2) \in ImageSet \times ImageSet$$
$$D_E : ImageSet \times ImageSet \rightarrow \mathbb{R}^+$$
$$D_E(I_1, I_2) = M_E(F_E(I_1), F_E(I_2))$$

La fonction F_E projette une image dans l'espace E. La fonction M_E calcule la distance entre deux points de l'espace E. Pour simplifier l'écriture, nous noterons D_E la fonction qui calcule la distance dans l'espace E entre deux images de document.

3.2 Construction d'un arbre de CAH

Voici l'algorithme de construction d'un arbre de CAH pour un espace de caractéristiques E donné :

1. Initialiser une liste L en formant un groupe par image de $ImageSet$
2. Calculer la distance entre tous couple d'images de $ImageSet$
3. Réunir en un groupe G, les deux groupes les plus proches A et B
4. Éliminer A et B de L et ajouter G à L
5. Calculer la distance entre G et tous les groupes de L
6. Si L contient plus d'un groupe, retourner à l'étape 3

Cet algorithme nécessite donc de définir deux distances. L'une doit mesurer la distance entre deux images (étape 2) : c'est la distance D_E définie en 3.1. L'autre doit mesurer la distance entre deux groupes d'images (étape 5).
La fonction mesurant dans un espace de caractéristique $E \in FeatSpace$ la dissimilarité entre deux groupes d'images G et G' de $ImageSet$ est une distance. Elle est définie par

$$\max_{I \in G, I' \in G'} (D_E(I, I'))$$

Cette distance est aussi appelée diamètre de l'ensemble $G \cup G'$. Le choix de cette distance permet d'avoir une mesure de la dispersion des données du groupe $G \cup G'$. Cette information ne peut être obtenue avec l'écart-type car cela n'induit pas une fonction distance. D'autres fonctions distance entre groupes sont cependant possibles (saut minimum, saut moyen, indice de Ward).
Lorsqu'un groupe G est créé (étape 3) avec les deux groupes A et B les plus proches, la distance entre A et B est aussi la hauteur du groupe G. C'est pourquoi cette structure arborescente est souvent représentée par un dendogramme.
L'algorithme se termine lorsque lorsqu'il ne reste plus qu'un seul groupe contenant tous les éléments. Ce groupe est appelé la racine de l'arbre de CAH.
Notre algorithme construit donc l'arbre de CAH correspondant à chaque espace de caractéristiques disponible dans le système.

3.3 Extraction des nœuds communs

Le système dispose maintenant plusieurs arbres de CAH qui représentent chacun une structuration différente des mêmes données. Pour chaque couple d'arbres de CAH, on extrait les groupes composés des mêmes éléments dans les 2 arbres. Ces groupes peuvent être vus comme des hypothèses de regroupement partagées par plusieurs points de vue. Nous regroupons dans la suite $Select$ l'ensemble des groupes apparaissant dans au moins 2 CAH. Le système dispose maintenant d'un ensemble de groupes qui sont retrouvés dans plusieurs arbres, donc *a priori* les groupes les plus fiables par construction.

3.4 Création de la forêt d'inclusion minimale

Le système établit des liens hiérarchiques entre les nœuds de la liste $Select$ de la façon suivante. Un nœud a pour père le plus petit nœud l'incluant. On obtient ainsi une forêt (ensemble d'arbres) notée F.
Les figures 1 et 2 présentent deux forêts d'inclusion. Chaque groupe contient la liste de ses images sous la forme [C]_[N] ou C est le numéro de la classe d'image de l'image et N son numéro à l'intérieur de la classe C. Les nœuds colorés sont homogènes (contenant des images de la même classe), les nœuds au fond blanc contiennent des images de classes différentes.
La forêt de la figure 1 contient deux arbres, non homogènes. De plus, 2 groupes de la même classe (classe 1) n'ont pas été regroupés : {01_01, 01_04} et {01_02, 01_00}. Pour les autres classes, il existe un nœud qui regroupe tous les éléments de la même classe. La forêt de la figure 2 contient 13 arbres dont seulement un n'est pas homogène, deux classes y étant représentées. Par ailleurs, la classe 11 est scindée sur deux arbres.

3.5 Présentation de la forêt à l'opérateur

Pour chaque arbre de la forêt, les images constituantes d'un groupe sont présentées à l'opérateur sous forme d'imagettes dans un tableau. Face à un groupe G, l'opérateur peut :
- Valider G si les images sont de la même classe. Le groupe est alors laissé tel quel pour une éventuelle fusion avec un autre groupe.
- Rejeter G s'il contient des images de différentes structures physiques. Dans ce cas, le système supprime G et présente à l'opérateur les groupes des nœuds fils de G. Expérimentalement, ce cas est fréquent car la structuration des groupes se fait sur des heuristiques numériques. Ainsi la probabilité qu'un groupe soit homogène baisse lorsque sa taille augmente.
- Fusionner G avec un autre groupe G' si les images de G et G' proviennent d'une même classe sursegmentée. Ces groupes doivent avoir été préalablement validés. Le système remplace alors G et G' par un groupe G'', union des images constituantes G et G'. C'est le cas notamment lorsque qu'une partie seulement des images d'une classe ont le même défaut. Par exemple, un logo noir pourra ou non être blanchi par une binarisation adaptative. Il paraît alors normal que l'algorithme sépare en deux sous-classes les images si le logo est blanchi ou non.

4 Résultats

Les résultats pour cinq bases d'images différentes et pour un regroupement de quatre de ces bases sont présentés dans le tableau 1. Des images extraites de ces bases sont présentées sur la figure 3 et sur les figures 1, 3, 4 et 5 de [CAR 04] publié dans ces mêmes actes. Les classes d'images de tests sont constituées d'un nombre aléatoire d'images (de 3 a 10) tirées au hasard dans une base contenant plusieurs milliers d'images.
On appelle "Images classées", la proportion d'images présentes dans la forêt. Par exemple, pour la base DB4, "81% des 70 images classées" signifie que 13 images ne sont pas présentes dans la forêt d'inclusion. Expérimentalement nous avons constaté que ces images comportaient de gros défauts par rapport aux autres images de la même classe.
On appelle "Classes bien formées", la proportion de classes retrouvées par le système. Ainsi, pour la base DB4, "93% des 15 classes bien formées" signifie qu'une classe n'a pas été retrouvée.
À la fin des corrections, le système va apprendre les images des groupes validés par l'opérateur. Il rappellera cependant à l'opérateur que des images n'ont pas été incluses dans l'apprentissage puisque présentes dans aucun groupe et fera une proposition de classification que l'opérateur devra valider ou rejeter. L'opérateur peut alors terminer la configuration des classes d'apprentissage en plaçant lui-même les images non classées.

5 Conclusion

Nous avons présenté une technique performante de classification semi-supervisée. Nous avons essayé d'introduire une

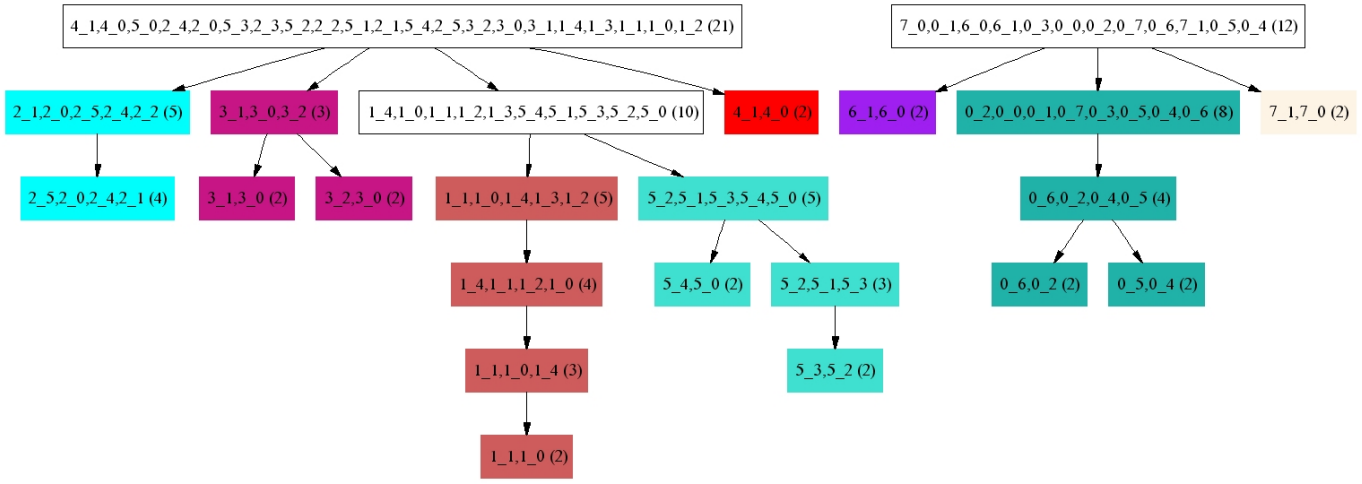

FIG. 1 – Forêt de la base DB1

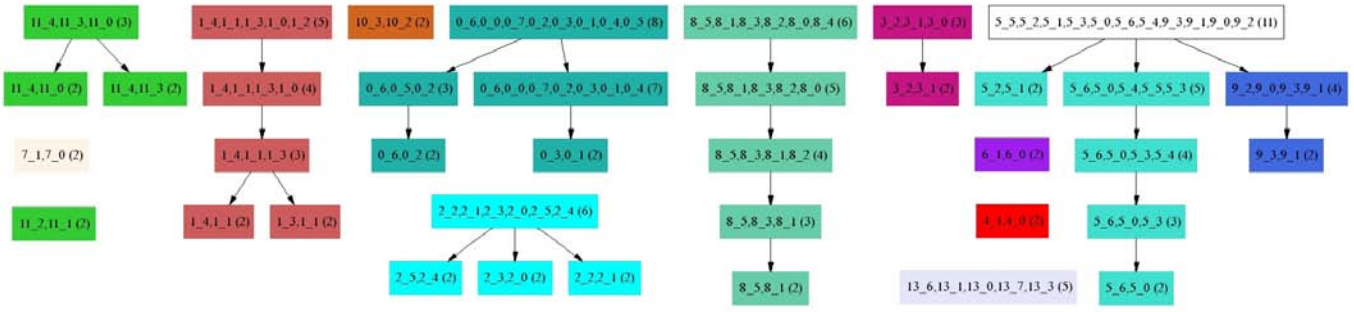

FIG. 2 – Forêt de la base DB4

	DB 0	DB 1	DB 2	DB 3	DB 4	DB 1,2,3,4
Nombre d'images (total)	15	33	31	31	70	165
Nombre de classes	2	8	6	6	15	35
Rejets	0	3	5	5	1	6
Fusions (après validation)	0	0	1	0	1	3
Images classées	100%	100%	100%	100%	81%	99%
Classes bien formées	100%	100%	100%	100%	93%	100%

TAB. 1 – Coût opérateur pour reconstituer des groupes homogènes

notion multi points de vue avec des métriques différentes pour éviter l'effet "d'aveuglement" dû aux considérations purement numériques qu'induisent les CAH. Cependant, ces arbres de CAH nous affranchissent du problème de la forme des nuages et de leur nombre, valeur inconnue de l'opérateur lui-même. Les performances étant mesurées *in fine* par le nombre d'actions nécessaire pour obtenir des classes homogènes et uniques, nous devons considérer avec attention la façon de présenter à l'opérateur les résultats de notre algorithme.

Comme la plupart des systèmes de classification non-supervisée, après le calcul des distances entre les images, nous ne revenons par sur ces images alors qu'au final ce sont elles qui seront montrées à l'opérateur. Il pourrait alors être judicieux de concevoir un algorithme qui extrairait automatiquement un ensemble d'objets graphiques ainsi que les relations qu'ils entretiennent sur une même image. Il justifierait la présentation d'un groupe à l'opérateur par la présence de ces objets ainsi que la validation de leurs relations de voisinage sur toutes les images du groupe. L'extraction des objets graphiques se ferait sans *a priori* sur le type d'objet pour ne pas retomber dans la problématique evoquée en section 2 mais avec de simples critères géometriques sévères afin de limiter les erreurs.

D'autre part, il serait intéressant d'essayer de couper l'arbre d'inclusion construit afin de présenter directement des classes homogènes à l'opérateur. Cependant, des tests ont été effectués sur ces bases avec différents critères de coupure mais tous ont donné des coûts pour l'opérateur bien plus élévés qu'en présentant l'arbre d'inclusion. On peut alors penser qu'il est illusoire de chercher à former automatiquement des groupes homogènes. En effet, rappelons que cette aide à l'annotation permet à l'opérateur de former rapidement des classes d'images pour l'apprentissage du logiciel de tri. Donc si l'automatisation crée des erreurs sur les classes apprises, les conséquences seront sérieuses sur l'efficacité du système de tri de document.

Références

[BEN 73] BENZÉCRI J.-P., *L'Analyse de Données : la Taxinomie*, Dunod, Paris, 1973.

[CAR 04] CARMAGNAC F., HÉROUX P., TRUPIN E., Classification supervisée d'images de document et contraintes industrielles, *Actes du Huitième Colloque International Francophone sur l'Écrit et le Document* CIFED'04, 2004.

[CLA 00] CLAVIER E., Stratégies de tri : un système de tri des formulaires, Thèse de Doctorat, Université de Caen, 2000.

[COR 02] CORNUÉJOLS A., MICLET L., *Apprentissage artificiel - concepts et algorithmes*, Eyrolles, 2002.

[DID 80] DIDAY E., *Optimisation en Classification Automatique*, tomes 1 et 2, INRIA, 1980.

[FUK 90] FUKUNAGA K., *Introduction to Statistical Pattern Recognition*, Academic Press Inc., 2^{nd} édition, 1990.

[JAI 00] JAIN A. K., DUIN R. P. W., MAO J., Statistical Pattern Recongition : A Review, *IEEE Transaction on Pattern Recognition and Machine Intelligence*, vol. 22, n° 1, 2000, pp. 4-37.

[KOC 03] KOCH G., HEUTTE L., PAQUET T., Numerical sequence extraction in handwritten incoming mail documents, *Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR'2003*, 2003, pp. 369-373.

[MUS 02] MUSLEA I., MINTON S., KNOBLOCK C., Active + Semi-Supervised Learning = Robust Multi-View Learning, *Proceedings of the 19th International Conference on Machine Learning (ICML 2002)*, 2002, pp. 435-442.

[RIB 98] RIBERT A., Structuration évolutive de données : Application à la construction de classifieurs distribués, Thèse de Doctorat, Université de Rouen, 1998.

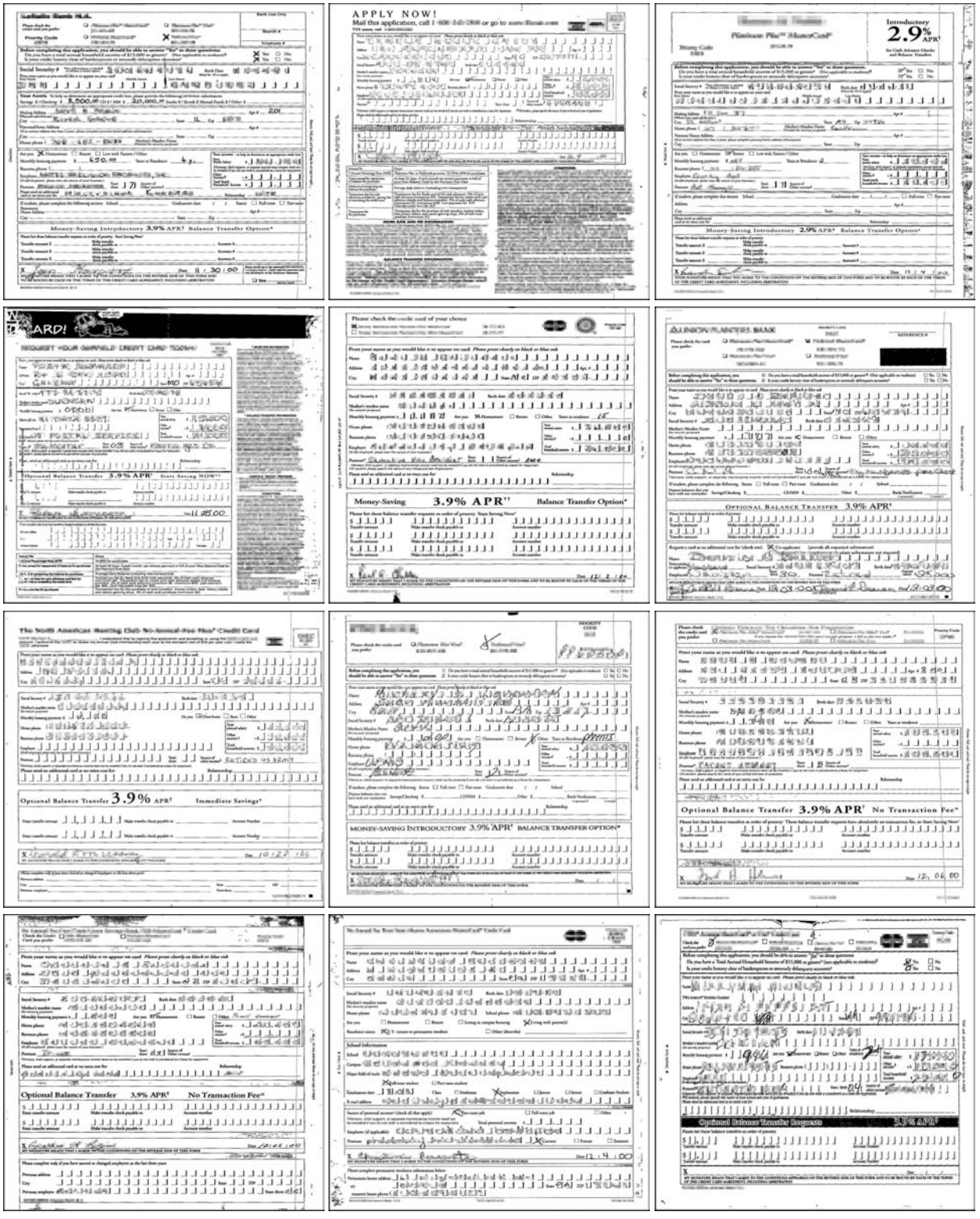

FIG. 3 – Un exemple pour 12 classes des 15 classes de DB4