



HAL
open science

Discrimination chiffre/rejet pour l'extraction de champs numriques dans des documents manuscrits

Clément Chatelain, Laurent Heutte, Thierry Paquet

► **To cite this version:**

Clément Chatelain, Laurent Heutte, Thierry Paquet. Discrimination chiffre/rejet pour l'extraction de champs numriques dans des documents manuscrits. CIFED 2010, Sep 2006, France. pp.55-60. hal-00113991

HAL Id: hal-00113991

<https://hal.science/hal-00113991>

Submitted on 15 Nov 2006

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.

Discrimination chiffre/rejet pour l'extraction de champs numériques dans des documents manuscrits

Clément Chatelain – Laurent Heutte – Thierry Paquet

Laboratoire LITIS, Université de Rouen
76800 Saint Etienne du Rouvray, FRANCE

{clement.chatelain, laurent.heutte, thierry.paquet}@univ-rouen.fr

Résumé :

Dans cet article, nous proposons une méthode pour l'extraction automatique de champs numériques dans des documents manuscrits non contraints. Le système repose sur une stratégie de segmentation-reconnaissance suivie d'une analyse syntaxique des lignes de texte afin de détecter les séquences constituant un champ numérique. Nous nous intéressons plus particulièrement au classifieur chiffre sur lequel repose la stratégie de segmentation-reconnaissance, et à ses capacités de rejet vis-à-vis des composantes non numériques : mots, fragments de mots, bruit, etc. Pour cela, une méthode de rejet en deux étapes est proposée : une première étape rapide est chargée de rejeter ce que nous appelons les "rejets évidents" ; la seconde étape repose sur l'analyse des confiances d'un classifieur chiffre "classique" pour distinguer les composantes numériques du reste du document. Nous montrons l'influence de la stratégie de rejet sur les résultats du système.

1 Introduction

Ces dernières années, la lecture automatique de documents manuscrits a bénéficié de nombreux travaux. Cependant, la variabilité inhérente à l'écriture manuscrite a limité les travaux à des applications spécifiques destinées à la lecture de documents très contraints tels que les chèques bancaires, les adresses postales ou encore les formulaires contenant des champs manuscrits. Plus récemment, de nouvelles problématiques sont apparues avec le traitement automatique de documents manuscrits non contraints : les systèmes de lecture sont alors confrontés à l'absence de connaissances *a priori* sur le document, ce qui rend la lecture intégrale avec une fiabilité suffisante impossible dans des temps de traitement raisonnables.

Néanmoins, il est possible de considérer des problèmes d'extraction d'informations dans ces documents complexes, où la connaissance *a priori* apportée par le système est relative à l'information recherchée plutôt qu'au document. L'extraction de champs numériques (numéro de téléphone, code client, numéro de fichier, code postal, etc.) dans des courriers manuscrits de type *courrier entrant* (voir figure 1) en est un exemple particulier. Il s'agit d'un problème difficile car les champs numériques peuvent être situés dans n'importe quelle zone du document : en-tête et/ou corps de texte. De plus, les champs numériques n'appartiennent pas à un dictionnaire fermé : la succession de chiffres est aléatoire. Ces

champs contiennent également souvent des chiffres liés difficiles à détecter (voir figure 2).

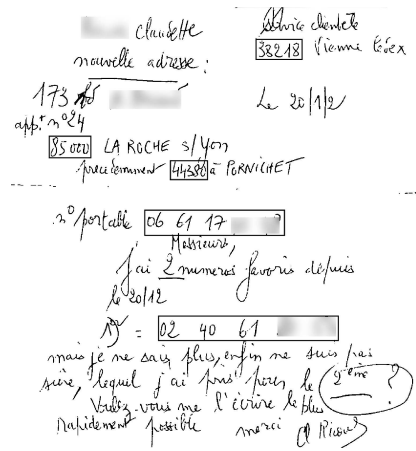


FIG. 1 – Exemple de courrier entrant



FIG. 2 – Exemples de champs numériques.

La stratégie générale proposée pour aborder ce problème d'extraction des champs numériques repose sur l'exploitation de la syntaxe connue du champ numérique recherché pour le localiser : nombre de chiffres et position éventuelle de séparateurs. Par exemple, l'extraction d'un numéro de téléphone français dans une ligne de texte manuscrite consiste à détecter une séquence de 10 chiffres comprenant éventuellement des séparateurs (points ou tirets) entre chaque paire de chiffres. Ceci est réalisé par une étape de reconnaissance de chiffres suivie d'une analyse syntaxique des hypothèses de reconnaissance, qui filtre les séquences valides au sens des syntaxes de champs connues du système. Ainsi, une des clefs du système est la capacité du classifieur chiffre à distinguer les composantes numériques du reste du document : mots, fragments de mots, bruit, etc. Dans cet article, nous proposons une méthode simple en deux étapes pour le rejet de ces composantes non numériques.

L'article est organisé de la manière suivante : la section 2 présente une vue globale du processus d'extraction des

champs numériques avec une brève description des différentes étapes. La section 3 décrit la stratégie de rejet mise en œuvre pour le rejet des composantes non numériques. Nous présentons dans la partie 4 nos résultats expérimentaux obtenus sur une base de documents manuscrits réels.

2 Extraction des champs numériques par une stratégie de segmentation/reconnaissance

La stratégie d'extraction des champs numériques repose sur une analyse syntaxique des lignes de texte afin de filtrer les séquences vérifiant une syntaxe particulière connue du système. Une étape de reconnaissance est donc nécessaire afin d'identifier les chiffres (isolés ou liés), les séparateurs et les informations non numériques que nous cherchons à rejeter. Ceci est réalisé par une stratégie de segmentation-reconnaissance reposant sur un classifieur chiffre qui possède des capacités de rejet vis-à-vis des composantes textuelles. Le processus de reconnaissance lancé sur toutes les composantes d'une ligne produit un treillis d'hypothèses de reconnaissance (voir figure 3, où 'X' représente une confiance) dans lequel nous allons rechercher une hypothèse de reconnaissance globale pour toute la ligne. Les trois étapes de notre système sont les suivantes :

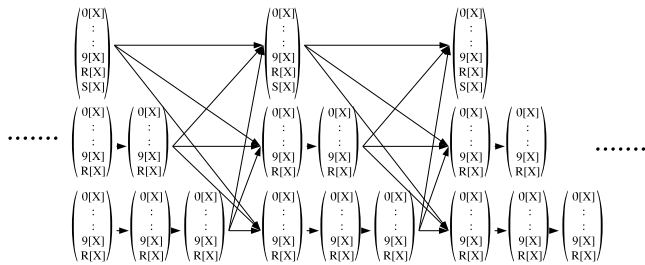


FIG. 3 – Treillis de reconnaissance d'une ligne de texte.

Segmentation en lignes : les composantes connexes sont extraites du document puis regroupées en lignes selon une méthode relativement classique [LIK 95] non détaillée dans cet article.

Segmentation-reconnaissance : le but de cette étape de reconnaissance est de détecter les composantes appartenant à un champs numérique : chiffre isolé ou chiffres liés, et séparateurs. Toutes les autres composantes sont considérées comme du rejet. La présence de chiffres liés nous a conduit à considérer successivement les composantes comme des composantes numériques, des séparateurs, et du rejet, à l'aide des 3 approches suivantes :

Composantes numériques : les composantes sont soumises à un processus de segmentation-reconnaissance qui considère la composante comme un simple, un double et un triple chiffre. La figure 4 donne un exemple du principe de segmentation-reconnaissance pour la reconnaissance d'un double chiffre : un algorithme de segmentation propose plusieurs chemins de coupure, et les hypothèses de chiffres segmentés sont soumises à un classifieur chiffre. Nous conser-

vons la solution qui maximise le produit des confiances du classifieur (ici la première hypothèse). Pour la reconnaissance de triple chiffres, cette étape est réitérée sur la composante la "moins bien reconnue" (celle qui dispose de la confiance la plus basse).

Drop fall	ascendant gauche	ascendant droit	descendant gauche	descendant droit
chemin de coupure				
classifieur chiffre	0[98] 8[82]	2[27] 8[35]	0[73] 8[36]	0[92] 8[34]
produit des confiances	81	09	26	32

FIG. 4 – Exemple de reconnaissance d'une composante en double digit

Séparateurs : la reconnaissance des séparateurs est effectuée à l'aide d'un classifieur spécifique basé sur un vecteur de caractéristiques contextuelles [CHA 04]. Une classe "Séparateur" et sa confiance associée est ainsi ajoutée sur le premier niveau du treillis seulement ; en effet nous faisons l'hypothèse que les séparateurs ne peuvent être liés à d'autres composantes.

Rejet : la plupart des composantes segmentées appartenant à la classe rejet, les hypothèses de reconnaissance chiffre et séparateur doivent être soumises à une stratégie de rejet capable de produire un score de confiance de la composante à la classe "rejet". Nous décrivons cette étape dans la section 3.

Finalement, l'étape de reconnaissance des composantes fournit pour chaque composante une hypothèse de reconnaissance à 3 niveaux. Sur une ligne de texte entière, ces hypothèses sont concaténées pour former un treillis de reconnaissance (voir figure 3).

Analyse syntaxique des lignes de texte : un modèle de ligne de texte est défini pour chaque type de champs à extraire. Ces modèles fournissent au système les contraintes syntaxiques d'une ligne de texte pouvant contenir un champs numérique. Les modèles sont constitués d'états décrivant 12 symboles : 10 classes de chiffres + séparateur (S) + Rejet (R) (voir figure 5). Les transitions autorisées entre les états ont été apprises sur une base de documents. L'exploration du treillis des hypothèses de reconnaissance est réalisée par programmation dynamique en fonction des scores de confiance, sous les contraintes des modèles.

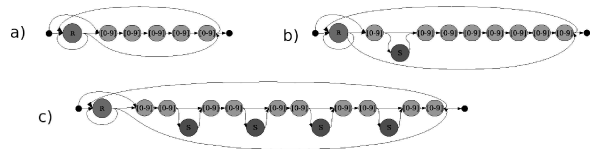


FIG. 5 – Modèles d'une ligne de texte contenant a) un code postal, b) un code client, c) un numéro de téléphone

3 Stratégie de rejet appliquée à un classifieur chiffre

Si la discrimination entre les classes de chiffres est un problème bien résolu, le rejet des caractères non numériques

est toujours un épineux problème. Ceci est dû à l'extrême variabilité de ces formes. Nous avons vu dans la partie précédente la nécessité d'obtenir 12 mesures de confiance pour les 10 classes de chiffre, la classe rejet et la classe séparateur au premier niveau ; 11 mesures (sans les séparateurs) dans les niveaux suivants. Dans cette section, nous nous intéressons à la conception d'une stratégie pour le rejet de formes non numériques basée sur un classifieur chiffre.

L'analyse d'une base de "rejet" nous a incité à considérer deux classes de rejet (voir figure 6) : **(i) les rejets évidents** possédant une forme très différente des chiffres isolés : bruit, fragment de mots ou mots entiers, traits, point, etc. **(ii) les rejets ambigus** ayant une forme proche des chiffres isolés : lettre, groupe de lettres ou fragment de lettres principalement. Evidemment, cette deuxième catégorie est plus difficile à distinguer des chiffres.



FIG. 6 – Exemples de rejets évidents (en haut) et ambigus (en bas).

Cette observation nous a conduit à considérer une stratégie en deux étapes pour rejeter ces formes (voir figure 7) :

Une première étape vise à rejeter les "rejets évidents" : comme la distinction est assez facile, nous proposons un classifieur à deux classes basé sur un vecteur de caractéristiques réduit. L'idée est de filtrer un maximum de rejet en ne rejetant aucun chiffre.

Dans un second temps, nous chercherons à distinguer parmi les composantes acceptées par la première étape les chiffres des rejets ambigus. Ce problème étant plus difficile, nous proposons une décision pondérée, basée sur l'analyse des scores de confiance d'un classifieur chiffre.

Nous détaillons maintenant ces deux étapes.

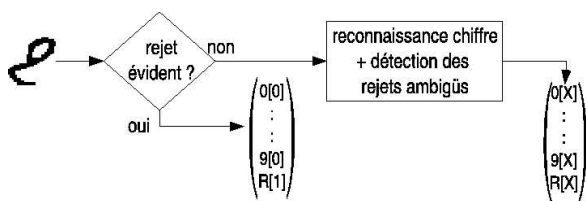


FIG. 7 – Une stratégie de rejet en deux étapes.

3.1 Filtrage des rejets évidents

Cette première étape agit comme un filtre qui décide si une forme est un rejet évident ou non. Le but est de réaliser un classifieur binaire rapide qui ne rejette aucun chiffre tout en rejetant un maximum de composantes rejet.

Pour cela, nous avons défini un vecteur à 8 caractéristiques : (f_1) ratio hauteur/largeur, (f_2) densité de pixels noirs, (f_3) nombre de "water reservoirs" (les water reservoirs sont une métaphore pour illustrer les vallées d'une composante,

voir [PAL 01] pour plus de détails), (f_{4-6}) nombre d'intersections avec deux sondes horizontales et une sonde verticale, (f_7) nombre de fin de traits, (f_8) nombre d'occlusions.

Plusieurs classifieurs à deux classes ont été testés sur ce jeu de caractéristiques. Ils ont été retenus selon deux critères : rapidité et performances.

- Le RIPPERk est un classifieur très rapide à base de règles qui donne de bons résultats sur des données bruitées [COH 95].
- Les réseaux de neurones, discriminant MultiLayer Perceptron (MLP) et modélisant Radial Basis Function (RBF) [BIS 95], connus pour leur bon compromis rapidité/performance.
- Les machines à vecteurs de support (SVM) [VAP 95] sont connues pour être des classifieurs très performants, bien que généralement plus lents que les autres classifieurs.

Une particularité de notre problème de classification est qu'il comporte des coûts de mauvaise classification très différents : les conséquences d'un faux rejet (FR : rejet d'un chiffre) sont beaucoup plus graves qu'une fausse acceptation (FA : acceptation d'un rejet). En effet, un FR ne peut être rattrapé par la suite, alors qu'une FA pourra être rejetée à un niveau supérieur, soit lors de la seconde étape de rejet, soit lors de l'analyse syntaxique. Nous introduisons donc un paramètre de coût pénalisant un FR par rapport à une FA.

Nous avons cherché dans un premier temps les meilleurs paramètres intrinsèques de chaque classifieur (nombre de neurones dans la couche cachée pour le MLP, nombre de bases radiales pour le RBF, γ et C pour le SVM), le paramètre coût étant déterminé dans un second temps expérimentalement. Concernant les paramètres intrinsèques, il n'existe pas de méthode analytique permettant de déterminer des paramètres optimaux, ceux-ci ont donc été ajustés sur une base de test. Concernant le paramètre coût, nous montrerons son influence sur les performances du système en section 4.

Les classifieurs ont été entraînés sur une base de 7500 exemples (2/3 rejet, 1/3 chiffre). La capacité de rejet des classifieurs est évaluée à l'aide de la courbe ROC (Receiver-Operating Characteristic) qui est une représentation graphique du compromis entre les taux FR et FA, pour différentes valeurs du paramètre coût. Les courbes sont présentées en figure 8.

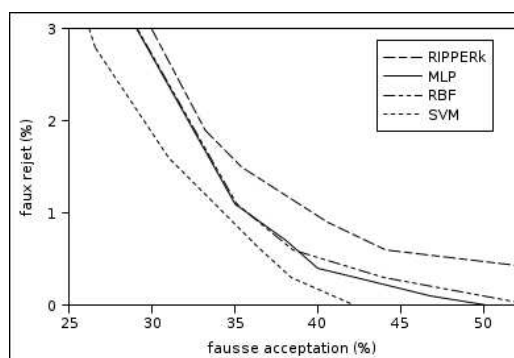


FIG. 8 – Courbe ROC des classifieurs RIPPERk, MLP, RBF, SVM.

Ces courbes montrent que le SVM donne la meilleure courbe ROC. Les classifieurs SVM, MLP et RBF atteignent des taux de FR nuls en rejetant respectivement 57,8%, 49,8% et 47% des formes non numériques. Signalons que le RIPPERk n'atteint pas de FR nul sauf lorsqu'il accepte toutes les formes (FA=100%). Cependant, les temps de traitement d'un SVM sont généralement assez importants. Ils dépendent du nombre de vecteurs de support (SV) conservés à l'issue de l'apprentissage. Dans notre cas, nous conservons en moyenne 2500 SV. La table 1 montre que même si le SVM est plus lent que les autres classifieurs, les temps de traitement sont tout à fait raisonnables (moins de 6 secondes pour traiter 7500 formes, beaucoup plus que le nombre de formes présentes dans un document manuscrit). Nous conservons donc le classifieur SVM par la suite pour filtrer les rejets évidents.

Classifieur	RIPPERk	MLP	RBF	SVM
Temps de traitement	<1s	<2s	<2s	6s

TAB. 1 – Temps de traitements (en secondes) pour traiter 7500 formes.

3.2 Identification des rejets ambigus par un classifieur chiffres

La deuxième étape de notre approche traite les formes acceptées par la première étape : chiffres et rejets ambigus. Cette étape est effectuée de manière conjointe avec la reconnaissance chiffre. Nous avons donc besoin d'un classifieur chiffre capable de rejeter, c'est-à-dire de produire 11 scores de confiance : 10 classes de chiffre + Rejet. Ce problème difficile demande un ensemble de caractéristiques assez conséquent et donc une base d'exemples d'apprentissage important.

Il existe plusieurs techniques pour rejeter à partir d'un processus de classification : entraîner un classifieur avec des données rejet [LIU 02], modéliser les classes à accepter et utiliser une stratégie de rejet à partir d'une mesure de distance [MIL 04], utiliser des classifieurs à une classe [TAX 01], ou encore rejeter à partir des sorties d'un classifieur "classique" [PIT 03]. Nous avons choisi cette solution appliquée à un MLP, pour les raisons suivantes :

- Le classifieur étant lancé plusieurs fois sur chaque composante d'une page entière d'écriture, et ce en grande dimension, le classifieur doit être assez rapide. Cette contrainte élimine les classifieurs SVM multiclassés et les classifieurs à une classe. Du fait de leur rapidité en décision, les MLP semblent au contraire parfaitement adaptés.
- Si l'utilisation des classifieurs "modélisants" (tels que RBF ou classifieurs à une classe) permettent un rejet de distance, ils ne résistent généralement pas au problème de la grande dimension, et ne sont pas très performants en discrimination. En revanche, les MLP sont bien adaptés aux hautes dimensions (sous la contrainte d'un nombre d'exemple suffisamment élevé) et donnent de bonnes performances en discrimination [BIS 95, LIU 02].

Nous avons donc mis en place une combinaison de deux MLP, entraînés sur une base de 130000 chiffres avec deux vecteurs de caractéristiques : un vecteur à 117 caractéristiques structurelles et statistiques développé dans nos travaux précédents [HEU 98], et un vecteur de 128 caractéristiques basé sur les directions du contour [KIM 94]. Une règle de combinaison de type produit est ensuite appliquée entre les deux MLP. Sur une base de test de 60000 chiffres, la combinaison donne des taux de reconnaissance de 98,44%, 99,48% et 99,75% en TOP 1, 2 et 3 respectivement.

La règle de rejet est alors la suivante : une valeur de confiance pour la classe rejet est estimée à partir du score de confiance de la première proposition du classifieur chiffre, à l'aide d'une table de correspondance (ou LUT pour Look Up Table). La LUT a été générée en analysant le comportement du classifieur chiffre sur une base de 4500 éléments contenant des chiffres et des rejets. Les statistiques sur le score de la première proposition ont permis de donner la LUT présentée en figure 9.

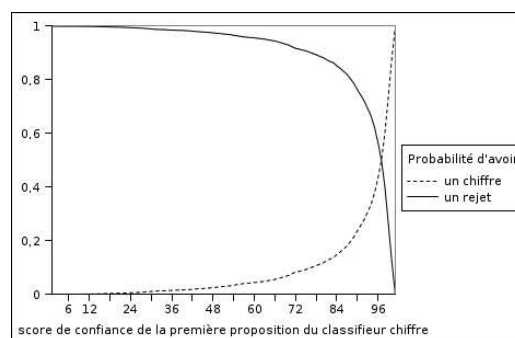


FIG. 9 – LUT fournissant le score de la classe rejet à partir du score de la première proposition du classifieur chiffre.

À l'issue de cette estimation, le système fournit 11 scores de confiance (10 scores du MLP + score rejet), sur lesquels nous appliquons la fonction *softmax* [BRI 90] pour donner des estimations de probabilités *a posteriori*.

3.3 Evaluation de la stratégie de rejet

Pour évaluer notre stratégie de rejet, nous avons constitué une base de données contenant des chiffres et des rejets : lettres, mots, fragments de mots, points, bruit, etc. Nous soumettons cette base au classifieur chiffre couplé à la LUT qui estime le score de la classe rejet, et à ce même système augmenté du SVM filtrant les rejets évidents. La figure 10 donne la courbe ROC de ces deux systèmes.

Le compromis entre faux rejet et fausse acceptation est nettement meilleur avec l'étape de filtrage des rejets évidents par le SVM. À l'équilibre entre FA et FR, le système est capable de rejeter correctement 91% des rejets en acceptant 91% des chiffres ("Break Even Point" = 9%).

4 Performances du système d'extraction des champs

Cette section présente les performances du système concernant l'extraction des champs. Il a été évalué sur une

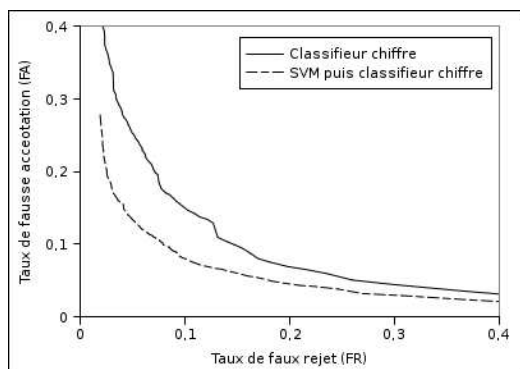


FIG. 10 – Courbe ROC pour les deux stratégies de rejet.

base de 293 courriers entrants réels provenant d’une entreprise. Ces documents contenant des informations privées (nom, adresse, numéro de téléphone, etc.), ils ne peuvent malheureusement pas être diffusés pour une comparaison des résultats.

Comme nous proposons un système d’extraction d’informations, le critère de performance utilisé est le compromis rappel-précision, ces deux quantités sont définies comme suit :

$$\text{rappel} = \frac{\text{nb de champs bien reconnus}}{\text{nb de champs à extraire}}$$

$$\text{précision} = \frac{\text{nb de champs bien reconnus}}{\text{nb de champs proposés par le système}}$$

Un champ est considéré comme “bien reconnu” si et seulement il a été parfaitement localisé et parfaitement reconnu, c’est-à-dire si toutes les composantes constituant le champ sont extraites (et seulement ces composantes), et chacune de ces composantes a été bien reconnue (la bonne étiquette chiffre ou séparateur).

L’analyse syntaxique est effectuée par l’algorithme forward, qui donne les n meilleurs alignements. Un champ détecté en $TOPn$ signifie que la bonne hypothèse de reconnaissance d’un champ est contenue dans les n meilleures propositions de l’analyseur syntaxique.

La figure 11 montre la courbe rappel-précision du système et l’influence du paramètre coût décrit dans la section 3.1. Pour chaque valeur du paramètre coût, on obtient une courbe constituée de 5 points qui représentent les performances du TOP1 au TOP5 (du coin supérieur gauche au coin inférieur droit respectivement). Le réglage du paramètre coût est effectué de la manière suivante : partant de la valeur appelée C_{zeroFR} (valeur qui fournit un $FR=0$, voir figure 8), nous faisons varier cette valeur autour de C_{zeroFR} . Remarquons que quand le coût tend vers l’infini, toutes les formes sont acceptées lors de la première étape de rejet ; ce qui est équivalent à une absence de filtrage des rejets évidents.

Sur la figure 11, on peut remarquer que le meilleur compromis rappel-précision est obtenu pour un coût égal à C_{zeroFR} . L’interprétation de ce résultat est la suivante : les coûts dont la valeur est inférieure à C_{zeroFR} fournissent toujours un FR nul, mais avec en rejetant moins de composantes.

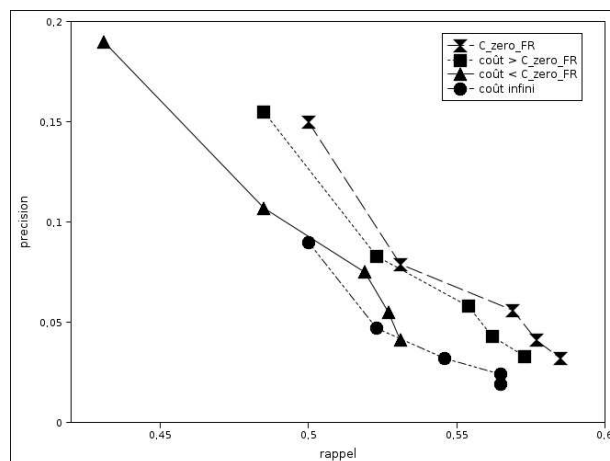


FIG. 11 – Reconnaissance des champs : compromis rappel-précision pour différentes valeurs du coût.

Inversement, les valeurs de coût supérieures à C_{zeroFR} améliorent la capacité de rejet, mais induisent également un rejet trop important de chiffres ce qui, lorsqu’ils appartiennent à un champs numérique, interdit une localisation correcte du champ. Nous avons ainsi montré l’importance du paramètre coût sur les performances du système, et donné une méthode pour fixer celui-ci à partir de la courbe ROC.

5 Conclusion et perspectives

A l’aide d’un vecteur de caractéristiques réduit et d’un classifieur SVM utilisé comme filtre précédant un classifieur chiffre, nous avons amélioré les capacités de rejet de celui-ci. Nous avons montré l’influence d’une telle amélioration lorsque le classifieur est embarqué dans un système de segmentation-reconnaissance. Nos futurs travaux porteront sur l’apprentissage du SVM : nous souhaitons mettre en oeuvre un critère prenant en compte l’aire sous la courbe ROC lors de l’apprentissage [RAK 04], plutôt que le taux de reconnaissance. Une autre perspective est la combinaison du système présenté dans cet article avec une approche alternative développée dans nos travaux précédents [CHA 04] afin d’augmenter les performances du système.

Références

- [BIS 95] BISHOP C., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [BRI 90] BRIDLE J. S., Probabilistic Interpretation of Feed-forward Classification Network Outputs, with Relationships to Statistical Pattern Recognition, SOULIE F. F., HERAULT J., Eds., *Neurocomputing : Algorithms, Architectures and Applications*, NATO ASI, pp. 227–236, 1990.
- [CHA 04] CHATELAIN C., HEUTTE L., PAQUET T., A syntax-directed method for numerical field extraction using classifier combination, *9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan, , 2004*, pp. 93–98.

- [COH 95] COHEN W., Fast Effective Rule Induction, *Machine Learning*, 1995, pp. 115–123.
- [HEU 98] HEUTTE L., PAQUET T., MOREAU J., LECOURTIER Y., OLIVIER C., A structural/statistical feature based vector for handwritten character recognition, *Pattern Recognition Letters*, vol. 19, 1998, pp. 629–641.
- [KIM 94] KIMURA F., TSURUOKA S., MIYAKE Y., SHRIDHAR M., A Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words, *IEICE Trans. on Information & Syst.*, vol. E77-D, n° 7, 1994, pp. 785–793.
- [LIK 95] LIKFORMAN-SULEM L., FAURE C., Une methode de resolution des conflits d’alignements pour la segmentation des documents manuscrits, *Traitement du Signal*, vol. 12, 1995, pp. 541–549.
- [LIU 02] LIU J., GADER P., Neural Networks with Enhanced Outlier Rejection Ability for Off-Line Handwritten Word Recognition Pattern Recognition, *Pattern Recognition*, vol. 35, 2002, pp. 2061–2071.
- [MIL 04] MILGRAM J., SABOURIN R., CHERIET M., An hybrid classification system which combines model-based and discriminative approaches, *ICPR’04, Cambridge, UK*, vol. 1, 2004, pp. 155–162.
- [PAL 01] PAL U., BELAÏD A., CHOISY C., Water Reservoir Based Approach for Touching Numeral Segmentation, *ICDAR*, , 2001, pp. 892–897.
- [PIT 03] PITRELLI J., PERRONE M., Confidence-Scoring Post-Processing for Off-Line Handwritten-Character Recognition Verification, *ICDAR’03*, vol. 1, 2003, pp. 278–282.
- [RAK 04] RAKOTOMAMONJY A., Optimizing AUC with Support Vector Machine, *European Conference on Artificial Intelligence Workshop on ROC Curve and AI*, , 2004, pp. 469–478.
- [TAX 01] TAX D., DUIN R. P. W., Combining One-Class Classifiers, *MCS ’01*, 2001, pp. 299–308.
- [VAP 95] VAPNIK V., The Nature of Statistical Learning Theory, *Springer*, , 1995.