Sample-dependent feature selection for faster document image categorization

Jérôme Louradour and Christopher Kermorvant
A2iA
40 bis rue Fabert,Paris, France
{jl,ck}@a2ia.com

***Abstract*—In document image classification, some classes of documents can be easily identified using pixel-level features, whereas some distinctions can only be made using semantics, which usually involves a full automatic text transcription. To be as much efficient as possible, the classification system should be able to avoid extracting high-level and time consuming features when they are not necessary to classify with confidence. We introduce here this issue of sample-dependent feature selection, which has not been addressed before as far as we know. We propose a method to tackle this problem, that can be generalized to any classifier that provides a confidence score along with its prediction. Empirical results using AdaBoost on three mail classification problems show that our approach allows to significantly improve classification efficiency (up to 40% CPU time off) without significant loss of accuracy in comparison to the baseline.**

***Keywords*-Image document classification, feature selection, confidence-rated multi-label classification.**

I. Introduction

Mailroom automation is one of the main applications of document image recognition. For large organizations, the volume of paper mail coming in daily can easily reach tens of thousands of documents and should be processed within a couple of days. The routing of the documents can be done automatically with an automatic document classifier embedded in the document management system. The time needed to automatically route a document is directly linked to the cost of processing the document: reducing the classification time allows to reduce the number of computers needed to process the document flow in time. Reducing the processing time is therefore an important goal to increase the competitiveness of an automatic document classification system.

In this paper, we are interested in reducing the document classification time by considering the following aspects:

1) Variable adequacy of features over classes: the type of documents to classify are numerous and can be very different. Official papers, forms, checks, certificates, subscription requests, printed and handwritten letters... Intuitively, some classes of structured documents can be easily recognized with low-level geometric features whereas some others need a text transcription to be well discriminated (*e.g.* semantics are needed to distinguish between letters of complaint, changes of address and other written requests).

2) Variable cost to extract features: the time needed to extract the features on the document can be important, but this time is feature dependent and varies a lot across features. For example extracting a sub-resolution image usually takes 50 ms, a Document Layout Analysis (DLA) [1] takes 500 ms, and text recognition takes 5000 ms. The classification time by it-self (after the features have been computed) is usually lower than 20 ms and can be considered as negligible.

3) Extraction of features by group: another important characteristic is that features are often extracted by groups, and not one by one.

4) Need of a confidence score: For real applications, error rates higher than a few percents are not acceptable. Most of the time, such a low error rate is not reached by state-of-the-art systems and rejection must be done. The usual way to achieve partial but accurate automation is to rely on a confidence score provided along with each prediction: low-confidence system predictions are discarded to be reviewed by a human agent.

Given these aspects and the time constraints of the digital document workflows, we propose here to reduce the overall processing time by avoiding to extract high-level features when it is not necessary, *i.e.* when a high-confidence prediction can be yielded with low-level features. We claim that feature extraction should be sample-dependent, with a view to maintaining the same overall accuracy while gaining significantly in efficiency.

To the best of our knowledge, sample-dependent feature selection does not match any classical machine learning problem. A lot of feature selection methods have been proposed, including wrappers [2], filters [3], gradient descent-based methods [4] and Boosting methods [5]. All these approaches aim at learning the minimal subset of features that are relevant to classify as much accurately as possible. The main motivation is often to have faster and/or more accurate learning procedure. But these feature selection methods disregard the second and third above-mentioned aspects: they assume that all features have the same cost of extraction, and cannot explore the fact that features are extracted group by group.

Some different settings involving feature selection have already been investigated: [6] tackles on-line learning feature selection, for scenarios where features are extracted one at a time (but during learning, not at test time). Group Lasso [7] can be used to select some *groups* of features among a huge number of features. But Lasso approaches assume that all features have the same cost of extraction.

In state-of-the-art feature selection methods, all the selected features are extracted for all the samples at test time. Here we consider the problem of optimizing the overall classification *test* efficiency and we propose a method of sample-dependant feature selection to achieve this.

II. Proposed method

The principle of the system we propose is to embed a classification learning method in a cascade that incrementally analyzes classification confidence estimates at the same time as features are extracted, with a view to possibly stoping extracting features prematurely. The principle is general provided that the embedded classifiers output a target class $\widehat{y_i}$ along with a confidence score

$$f^{(i)}_{\widehat{y_i}}(x) = f^{(i)}\left(\widehat{y_i}\,|\,\{\mathbf{x}_0, \cdots, \mathbf{x}_i\}\right) \tag{1}$$

where x is an input sample, and $\mathbf{x}_k$ are the corresponding subsets of features used to train the classifier. This confidence score can be a posterior probability estimate, but it does not have to be bounded nor normalized.

Algorithm 1 Sample-dependent feature selection for fast confidence-rated classification

```
• Inputs
    - One test sample, $x$
    - N "group-of-features" extractors $\mathbf{x}_i$, ranked by increasing CPU needs.
• Set of free parameters
    - N confidence-rated classifiers $f^{(i)}$, trained on all first groups of features $\{\mathbf{x}_0, \cdots, \mathbf{x}_i\}$,
    - $\mathrm{N} - 1$ confidence thresholds $\tau_i$,
    - N confidence classifier calibration functions $\varphi_i$.
• Outputs
    - Predicted class $\hat{Y}(x)$,
    - Back-end confidence $F_{\hat{Y}}(x)$.

  for $i := 1$ to N do
    Compute the group of features $\mathbf{x}_i$.
    Compute outputs of the $i^{th}$ classifier$f^{(i)}$ (1).
    if $i = \mathrm{N}$ or $f^{(i)}_{\widehat{y_i}}(x) \geq \tau_i$ then
      terminal step $I := i$
      Return
        $\hat{Y}(x) \;:=\; \widehat{y_I}$                                  (2)
        $F_{\hat{Y}}(x) \;:=\; \varphi_I \circ f^{(I)}_{\widehat{y_I}}(x)$   (3)
    end if
  end for
```

The proposed test procedure is summarized in algorithm 1. The model handles growing sets of features by making a cascade out of several confidence-rated classifiers trained on fixed sets of features. At each step of the cascade, a decision is made: either to continue extracting higher-level features, or to trust the current classification prediction and terminate. This choice is made by comparing the confidence score (1) to a threshold τ_i. Let $I(x)$ denote the step at which Algo.1 terminates and answers on sample x. The expected behaviour is to take a decision on easiest samples using only the first subsets of features ($I(x) \approx 1$), and to rather rely on the classifiers trained on all features to answer on the hardest samples ($I(x) \approx \mathrm{N}$).

An important point of Algo.1 is the computation of a back-end confidence based on the embedded classifiers confidence estimates. If we imagine that score (1) gauges the posterior probability of the class prediction to be true, no correction would be needed (φ_i would just be identity). But for most of classifiers families, nothing guarantees that confidence produced with different features will be comparable. Hence the possible presence of back-end classifier calibration function φ_i.

Note that Algo.1 can be seen as a decision tree where each node of the tree has only one descendant. It is justified in our experimental setup where feature extraction times vary a lot over groups of features (with factors of 10): we know in advance the order in which features should be computed and processed to achieve optimal overall efficiency, and groups of features $\mathbf{x}_0, \cdots, \mathbf{x}_\mathrm{N}$ are simply ranked by increasing cost. If several groups of features have similar costs, the approach can steadily be generalized by learning a tree with a wider topology.

By itself, our sample-dependent feature selection method relies on confidence scores provided by a classification method. We discuss the desirable properties for the embedded classifiers $f^{(i)}$ in section II-A. Threshold parameters τ_i and calibration functions φ_i are typically tuned on a validation set, separate from the training set used to learn embedded classifiers. Given our classification setting, we actually divide the overall optimization problem into two sub-problems: the trade-off between accuracy and efficiency by optimizing thresholds τ_i (section II-B), and the back-end confidence estimation which involves calibration φ_i (section II-C).

A. Choice of embedded classification learning method

Algorithm 1 relies on the quality of the confidence scores estimated by embedded classifiers $f^{(i)}$. See [8] for a good overview on confidence estimation. Here, the confidence scores do not need to be posterior probability estimates, but they must be relevant to rank predictions by uncertainty.

Besides, note that the meta-classifier of Algo.1 chooses the best candidate class (2) and estimates the back-end confidence score (3) by considering *only* the expertise of the last classifier $f^{(I)}$, and ignores the outputs of previously used classifiers $\{f^{(k)}\}_{k<I}$. In fact, we assume here that the classification model is able to learn how to optimally combine all available features. It means that if $k < i$, then not only classifier $f^{(i)}$ performs at least as well as the classifier $f^{(k)}$ with less features, but also no significant gain in accuracy could be achieved by combining the scores $f^{(I)}_{\widehat{y}_i}(x)$ with the previously computed scores $\{f^{(k)}_{\widehat{y}_i}(x)\}_{k<I}$. In practice, improving accuracy by adding input features into a single model is not guaranteed, for instance because of the *curse of dimensionality*. For the cases of embedded classifiers that do not necessarily improve accuracy as the set of input features grows, Cascade Generalization [9] can be a solution to fulfill this property.

In this work, we chose Adaptive Boosting (AdaBoost) [5] because it is designed to incrementally select and combine features that are de facto heterogeneous[1]. Output confidence scores of AdaBoost classifiers are learned so as to minimize an exponential loss (upper bound of the Hamming loss). It has already been observed that AdaBoost not only achieves good classification accuracy, but is also well suited to optimize the area under the ROC curve [10] which is an indicator of the confidence scoring quality. AdaBoost has already been integrated into cascade systems with success in the past, *e.g* in [11] where the goal is to reduce false positive rates.

B. *Optimization of overall efficiency*

Table I: Description of costs used to choose thresholds τ_i

Case description for Algo.1		
Back-end answer is	**Could the answer be true with more/less features?**	**Associated sample cost**
CORRECT	NO	0
	yes, with LESS features	c_0
	yes, with MORE features	0
WRONG	NO	0
	yes, with LESS features	c^-
	yes, with MORE features	c^+

In this section, we discuss how to optimize threshold values τ_i of Algo.1 to improve the system efficiency (with respect to the common approach which consists in using the classifier $f^{(\mathrm{N})}$ straightforwardly). To define the corresponding loss function, we introduce three types of cost parameters c_0, c^-, and c^+ described in table I. The sample cost can take five values (either 0, c_0, c^-, c^+ or $c^- + c^+$) and the overall objective loss function is the average of these sample costs:

$$\begin{aligned}\mathcal{L}_{\mathbb{X}}(\tau_0, \cdots, \tau_{\mathrm{N}-1}) \;&=\; c_0 \left| \mathbb{X}_1 \cap \mathbb{X}_1^- \right| \\ &+c^- \left| \mathbb{X}_0 \cap \mathbb{X}_1^- \right| + c^+ \left| \mathbb{X}_0 \cap \mathbb{X}_1^+ \right|\end{aligned} \tag{4}$$

where $\left| \cdot \right|$ denotes the cardinality of a set and

- $\mathbb{X}_1$ is the set of correctly classified samples,
- $\mathbb{X}_0$ is the set of errors,
- $\mathbb{X}_1^-$ is the set of samples that could have been well classified with less features, and
- $\mathbb{X}_1^+$ is the set of samples that could have been well classified with more features.

Note that cost parameter c_0 concerns efficiency only, c^- concerns both efficiency and accuracy, and c^+ concerns accuracy only. As only the relative ratio of these parameters affects the optimum of the loss function, we can arbitrarily set $c^+ = 1$, and it remains two hyper-parameters ($c_0, c^- \geq 0$). Tuning these two parameters amounts to choosing a trade-off between accuracy and efficiency.

The loss function (4) is not convex neither differentiable with respect to threshold parameters τ_i. In our application, we optimize it using a grid search, which can be afforded given that there are a few groups of features (N is small).

C. *Confidence estimation*

We remind that the typical use of this back-end confidence score is to reject predictions that are too uncertain, by comparing the score to a threshold. Such classification with reject can be seen as a ranking problem. We can consider the following loss function:

$$\mathcal{L}_{\mathbb{X}}(F_{\hat{Y}}(\cdot)) = \sum_{x \in \mathbb{X}} \left\{ \begin{array}{l} 0 \text{ if } x \text{ is well classified} \\ rank_{\mathbb{X}}(F_{\hat{Y}}(x)) \text{ otherwise} \end{array} \right. \tag{5}$$

where $rank_{\mathbb{X}} \geq 1$ is the rank of the scores in increasing order over $\mathbb{X}$.

Calibration functions φ_i can be chosen among non-parametric or parametric sets of functions. Non parametric methods may over-fit the validation set. So we choose here to focus on parametric methods, and define the calibration function as a classifier-dependent sigmoidal function:

$$\varphi_i \left(F_{\hat{Y}}(x) \right) = \mathrm{sigm} \left(\alpha_i F_{\hat{Y}}(x) + \beta_i \right) \qquad , \alpha_i > 0 \tag{6}$$

Determining the classifier-dependent parameters α_i, β_i is critical to have reliable back-end confidence scores. We learn them using an optimized grid search to minimize (5).

III. Experiments

A. *Databases*

1) ISRI-OCRtk: This public database was originally collected to evaluate the OCR performance [12]. But the 2889 image documents are labeled with 7 different classes that we use to make a classification task: *Department of Energy* reports, annual reports, legal documents, business letters, magazines, US newspapers and Spanish newspapers. Results presented on this database are obtained by 10-times repeated random validation, with 60% of training data, 20% of validation data and 20% of test data.

[1] In the application of interest, features are numerical data with different distribution shapes, as well as symbolic and variable-length data (text).

2) A2iA-Ima: This database is composed of images of paper mails addressed to a bank. There are 15 different classes of documents: ID cards, passports, several classes of invoices, receipts and tax notification, and several types of handwritten letters. The database has been split into train / validation / test sets (resp. 20752 / 5186 / 6182 samples). Both training and validation datasets contain more than 10% label errors whereas the test set has been entirely double-checked.

3) A2iA-Tepa: This database is composed of images of paper mails addressed to an insurance company. These documents are very diverse and there are 127 classes with a very unbalanced distribution (7 classes fill 60% of the images). Some classes are very close and not always well defined, such as "consumer letters" and "accompanying letters". The database has been split into a train / validation / test sets (resp. 57215 / 14304 / 8399 samples).

B. Document Front-end processing and classification

We used the following set of features for training the document classifier (approximate CPU times per document are indicated in parentheses):

1) Image sub-resolution (50 ms)*:* This first set of features is a simple sub-resolution image of the original document. It is computed as the average pixel values and we used a 6×8 regular grid.

2) Document layout analysis (DLA) [1] (500 ms)*:* The document image is segmented in zones corresponding to geometrical lines, printed text and handwritten text. We then compute some statistics on each kind of zone, such as the number and the total surface.

3) Pre-defined document type detectors (pre-detect) (1 sec)*:* For *A2iA-Ima* we also use predefined detectors specialized to recognize bank checks, cursive letters, printed letters and different types of official papers which are part of the dataflow. For a given document image, each detector provides a score used as a single numerical feature (as in a cascade [9]).

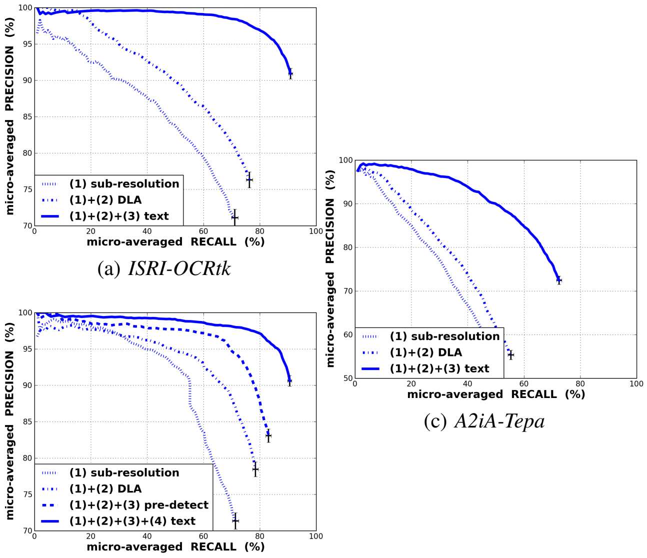
4) Full automatic text transcription (5 sec)*:* The automatic transcription provides additional inputs: a bag-of-words with some errors/noise (the word error rate lies between 15% and 50%). For efficiency concern, only the 10 000 most frequent words in the training set are used for classification.

We chose Real AdaBoost MH [5] as the reference classifier (*cf.* justification in section II-A), and the boosted weak learners are:

- *numerical stumps*, whose outputs depend on the comparison of a numerical feature value to a threshold.
- *word stumps*, whose outputs depend on the presence or absence of a given word within the text transcription.

C. Results

For the three databases, Fig.1 presents classification results of AdaBoost *without* sample-dependent feature selection: the baseline (classifier $f^{(\mathrm{N})}$ in Algo.1) along with intermediate results obtained by selecting only a subset of features (classifiers $f^{(i)}$, $i < \mathrm{N}$). The curves plot *micro-averaged* recall and precision [13] for different values of the rejection threshold: the more concave the curve, the better the quality of the confidence scoring. These results corroborate the hypothesis of section II-A: the more input features there are, the more accurate the classification is. *A2iA-Ima* presents the most favorable characteristics for sample-dependent feature selection: for the four sets of features, the classification results range nicely from sub-resolution to full feature extraction, and confidence scores are reliable. *A2iA-Tepa* is less favorable since there is a big gap between using text transcription or not. Further statistical analysis reveals that some classes can be accurately detected with only sub-resolution, for example the ID cards in *A2iA-Ima*, whereas some other classes need the semantic information from the text transcription to be recognized, for example the business letters in *ISRI-OCRtk* and letter classes in *A2iA-Tepa*.



(a) *ISRI-OCRtk*

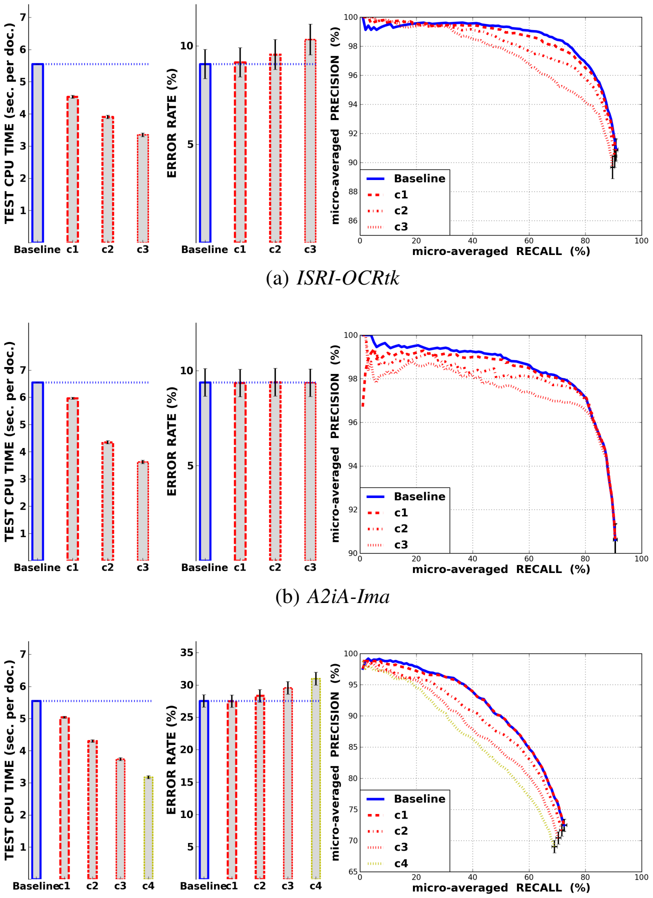
(b) *A2iA-Ima*

(c) *A2iA-Tepa*

Figure 1: Baseline AdaBoost classification results, with several (growing) sets of features.

The classification results of the proposed sample-dependent feature selection method are presented in Fig.2 beside the results of the baseline. The different versions of the proposed approach (c1, c2, . . .) only differ by the values of cost parameters c_0/c^+ and c^-/c^+ in (4). Bar charts on the left show the overall classification test CPU times and error rates.

If we accept only a very small performance degradation, we can save 20% of CPU time for *ISRI-OCRtk* (c1), 40% of CPU time for *A2iA-Ima* (c3) and 10% of CPU time for *A2iA-Tepa* (c1). Of course, this gain depends on the class distribution (ratio of formatted documents,. . .).



(a) *ISRI-OCRtk*

(b) *A2iA-Ima*

(c) *A2iA-Tepa*

Figure 2: (left) Overall test CPU times, (center) misclassification rates and (right) recall/precision curves, for the baseline and for several levels of sample-dependent feature selection (varying c_0 and c^-). 95% confidence interval are indicated.

Two aspects may explain why we can save more CPU time on the *A2iA-Ima* database compared to the other databases. First, as noticed previously, each set of features contributes almost equally to the increase of the classification rate. For some kind of documents, a good classification rates can be obtained with low level features. Second, the precision/recall curves of each set of features is also more favorable : the precision rate decreases slowly with an increasing recall rate, which means that the confidence estimation is good. The situation is less favorable for the two other databases.

Note that even if the overall accuracy can be preserved with a more efficient system based on sample-dependent feature selection, recall-precision curves indicate that confidence scores can be significantly less robust, especially in the region of low and medium confidence, despite the approach described in section II-C.

IV. Conclusions

We presented a novel classification problem setting motivated by industrial constraints in image document categorization. Due to the time constraints on the classification process, we argue that it is be essential to avoid extracting costly features when they are not necessary. This is a feature selection problem, but contrarily to methods in the literature, the selection is done at test time and is dependent on the current sample to be classified. We proposed a method which permits to reduce the CPU time from 10% up to 40% without degrading the accuracy, on three databases of real document images. By providing baseline results on a public document database, we would like to foster the research of new algorithms in this new classification setting.

References

[1] S. Mao, A. Rosenfeld, and T. Kanungo, “Document structure analysis algorithms: A literature survey,” 2003.

[2] R. Kohavi and G. John, “Wrappers for feature subset selection,” *Artificial Intelligence*, vol. 97, 1997.

[3] M. Hall, “Correlation-based feature selection for discrete and numeric class machine learning,” in *Proc. of the Int. Conf. on Machine Learning*, 2000.

[4] S. Perkins and Lacker, “Grafting: fast, incremental feature selection by gradient descent in function space,” *Journal of Machine Learning Research*, vol. 3, 2003.

[5] R. Schapire and Y. Singer, “Improved boosting algorithms using confidence-rated predictions,” *Machine Learning*, vol. 37, no. 3, 1999.

[6] S. Perkins and J. Theiler, “Online feature selection using grafting,” in *Proc. of the Int. Conf. on Machine Learning*, 2003.

[7] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society, Series B*, vol. 68, 2006.

[8] Z. Bosnić and I. Kononenko, “An overview of advances in reliability estimation of individual predictions in machine learning,” *Intell. Data Anal.*, vol. 13, no. 2, 2009.

[9] J. Gama, “Local cascade generalization,” in *Proc. of the Int. Conf. on Machine Learning*, 1998.

[10] C. Cortes and M. Mohri, “Auc optimization vs. error rate minimization,” in *Advances in Neural Information Processing Systems*, 2003.

[11] P. A. Viola and M. J. Jones, “Fast and robust classification using asymmetric adaboost and a detector cascade,” in *Advances in Neural Information Processing Systems*, 2001.

[12] T. A. Nartker, S. Rice, and S. Lumos, “Software tools and test data for research and testing of page-reading ocr systems,” in *Proc. Int. Symp. on Electronic Imaging Science and Technology*, 2005.

[13] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Computing Surveys*, vol. 34, no. 1, 2002.