Incremental learning for interactive sketch recognition

Achraf Ghorbel, Abdullah Almaksour, Aurélie Lemaitre, Eric Anquetil

Incremental learning for interactive sketch recognition

Achraf Ghorbel[1], Abdullah Almaksour[2], Aurélie Lemaitre[2], and Eric Anquetil[1]

[1] INSA de Rennes
[2] Université Européenne de Bretagne, France
UMR IRISA, Campus de Beaulieu, F-35042 Rennes
Université Européenne de Bretagne, France
{achraf.ghorbel, abdullah.almaksour, aurelie.lemaitre, eric.anquetil}@irisa.fr

Abstract. In this paper, we present the integration of a classifier, based on an incremental learning method, in an interactive sketch analyzer. The classifier recognizes the symbol with a degree of confidence. Sometimes the analyzer considers that the response is insufficient to make the right decision. The decision process then solicits the user to explicitly validate the right decision. The user associates the symbol to an existing class, to a newly created class or ignores this recognition. The classifier learns during the interpretation phase. We can thus have a method for auto-evolutionary interpretation of sketches. In fact, the user participation has a great impact to avoid error accumulation during the analysis. This paper demonstrates this integration in an interactive method based on a competitive breadth-first exploration of the analysis tree for interpreting the 2D architectural floor plans.

1 Introduction

In this paper, we are working on mapping technical paper documents, like architectural floor plans, to numerical ones. We aim at offering a complete, interactive and auto-evolving solution to unify paper document recognition and pen-based sketch interpretation (for instance: with Tablet PC).

At present, structured documents can be very complex. Faced with this complexity, the various existing methods [1] [2] [3] [4] keep a margin of error. Therefore, very often, an a posteriori verification phase will be necessary to ensure there is no recognition error. In this phase the user browses the document to correct the errors due to the interpretation.

To avoid the verification phase on the one hand, and avoid error accumulation during the analysis step on the other hand, we proposed an interactive method of analysis of off-line structured document where the decision process solicits the user if necessary. In our previous work [5], the role of user was limited to validate the right hypothesis and then unlock a situation where the decision process is not sure to make the right decision. In summary, the process can solicit the user to be sure to make the correct decision.

Now, we want to exploit the solicitation of the user during the analysis not only to unlock a situation but also to improve the analysis process. In this context, we focus in this paper on improving the capacity of symbol recognition. In the sketch interpretation method, the symbol recognition is made by the classifier.

The classification systems can be generally categorized into two types: static and evolving systems. Static systems are trained in batch mode using a predefined learning dataset, while incremental learning algorithms are used to train evolving classifiers, like for our symbol recognition system. In incremental learning algorithms, new instances from existing classes can be progressively introduced to the system to improve its performance. Moreover, new unseen classes can be added to the system at any time by the incoming data.

In this work, we present the advantage of soliciting the user to improve the recognition capacity of the classifier by incremental learning. It is also able to dynamically add new classes. The remaining of the paper is organized as follows. In the section 2, we introduce our existing interactive analysis method. Section 3 describes principles of the incremental classifier. The coupling of this incremental classifier with our interactive analysis of sketches is described in section 4. The rejection mechanism is explained in section 5. Experimental results are reported in section 6 and finally, section 7 concludes the paper.

2 Interactive Breadth-First Exploration

In this section, we summarize our interactive method of structured document interpretation (referred as IMISketch) [5] in which we propose to integrate our incremental classifier. This analyzer is based on the following characteristics:

- a priori structural knowledge of the document are expressed through a visual language based on production rules;
- a two-dimensional descending breadth-first analysis;
- a spatial contextual focus of the exploration to limit the combinatory;
- the uncertainty is formalized by the attribution of scores to each hypothesis represented by the tree analysis branch;
- if the ambiguities can not be resolved in the local context in an automatic manner, the user will be solicited by the analyzer to resolve the ambiguity.

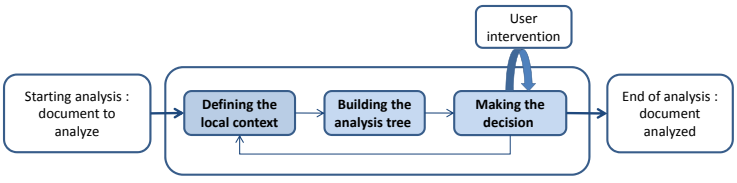


Fig. 1. Analysis process

These characteristics were chosen in order to ensure the best interactivity with the analysis system. This interactivity allows in particular to avoid a posteriori verification phase, which can become fastidious on complex documents. Indeed, the user participation, on the critical phases of the analysis of the document,

has a great impact to avoid error accumulation during the analysis step and overcomes the combinatory due to the sketch complexity. Figure 1 shows the complete process of analysis and the relationship between the three parts of the analyzer.

The first step consists of extracting the necessary information from the structured document. This phase is generic and off-line and does not depend on the type of document to interpret. We have chosen to work only with line-segments, which represents the primitives of our analysis. The primitive recognition, depends on its neighbourhood in structured documents. The analyzer begins by defining a spatial contextual focus that aims to limit the combinatory exploration due to the breadth-first exploration of analysis tree. The structured document requires a two-dimensional context. This two-dimensional local context is defined for an analysis tree as the maximum distance between the elements of the root and the elements of any leaves.

Once the context is well defined, the analyzer goes to the second stage. In this stage, the analyzer explores all possible hypotheses of interpretation in the spatial context using a set of two-dimensional rules that describe the structure of the document. These production rules are described by the context-driven constraint multiset grammars (CD-CMG) [6]. Each primitive can be interpreted in several ways. Each node or leaf is the application of a production rule deduced from the previous node. Every leaf or node of the tree has a score calculated from both its local score and the score obtained from the preceding nodes. Every score determines the adequacy degree to validate a production. The score calculated by each production is due to preconditions and constraints of the rule production (Equation 1). The use of the square root is a normalization using a geometric average. The production score can also be deduced from a classifier. A score is associated with each branch (hypothesis). Equation 2 determines the degree of adequacy (score) of a hypothesis. |PS| is the number of production in the considered branch (referred as PS).

$$\rho_P = \sqrt{\mu_{preconditions} \cdot \mu_{constraints}} \tag{1}$$

$$\rho_{PS} = \Big(\prod_{P_i \in PS} \rho_{P_i} \Big)^{\frac{1}{|PS|}} \tag{2}$$

Each analysis tree characterize the element to interpret in the define local context. Each root is the production rule that would consume this primitive. The number of analysis trees corresponds to the number of possible interpretations for the current primitive. The construction of the tree based on a breadth-first exploration allows to have several competitive hypotheses.

Once the tree is well constructed, we start the decision phase. The role of the decision process is to validate the right hypothesis among a set of competing hypotheses generated with a descending breadth first analysis.

Sometimes the decision process is not sure to make the right decision. In this case, it solicits the user. In practice, if the difference of scores between the top two branches is below a threshold of confidence and if these two branches are contradictory (at least one joint primitive is not consumed by the same rule production), the user intervention is required.

When the correct root is validated, other roots are put on hold and the new roots are either the sons of this root if exists, or the waiting roots otherwise and the analyzer go back to the first step (defining the local context step). The analysis is complete when no more production rule is applicable.

In the current state, the information provided by the user is only used to unlock situations. In this paper, we want to benefit more widely this information by learning continuously during the analysis. In this context, we propose to integrate an incremental classifier which uses information supplied by the user to improve its capabilities during the analysis.

3 Incremental Learning of a Fuzzy Inference System

The incremental learning algorithm is supposed to be supervised. The recognition of each data sample must be followed by a validation or a correction action in order to learn it. If the system answer is validated, the data sample will reinforce the system knowledge associated to its class. If an external correction signal is sent, the confusion between the (wrong) winner class and the true class is solved by the incremental learning algorithm. A third scenario may take place when the input data sample is declared as the first sample from a new unseen class. Our classification system is based on first-order Takagi-Sugeno (TS) fuzzy inference system[7]. It consists of a set of fuzzy rules of the following form:

$$\mathbf{Rule}_i : \mathbf{IF}\ \boldsymbol{x}\ \textit{is close to}\ P_i \mathbf{THEN}\ y_i^1 = l_i^1(\boldsymbol{x}), ...,\ y_i^k = l_i^k(\boldsymbol{x}) \tag{3}$$

where $\boldsymbol{x} = \{x_1, x_2, ..., x_n\}$ is the input vector. It consists of a set of features values extracted from the symbol image. These features describe the symbol shape from different viewpoints. The estimation of these features is not the aim of the current paper; it will be the subject of a future paper. $l_i^m(\boldsymbol{x})$ is the linear consequent function of the rule i for the class m:

$$l_i^m(\boldsymbol{x}) = \boldsymbol{\pi}_i^m \boldsymbol{x} = a_{i0}^m + a_{i1}^m x_1 + a_{i2}^m x_2 + ... + a_{in}^m x_n \tag{4}$$

where n is the size of the input vector. a_{ij}^m is a coefficient value in the rule i between the score of class m and the feature j of the input vector. The Prototype P is defined by a center and a fuzzy zone of influence. To find the class of $\boldsymbol{x}$, its membership degree $\beta_i(\boldsymbol{x})$ to each fuzzy prototype is first computed. After normalizing these membership degrees, the sum-product inference is used to compute the system output for each class:

$$y^m(\boldsymbol{x}) = \sum_{i=1}^{r} \bar{\beta}_i(\boldsymbol{x})\ l_i^m(\boldsymbol{x}) \tag{5}$$

where r is the number of fuzzy rules in the system. The score of each class y^m is between 0 and 1. The higher is the score, the higher is the degree of confidence in that class to be associated to the given input. The winning class label is given by finding the maximal output and taking the corresponding class label as response:

$$class(\boldsymbol{x}) = y = argmax\ y^m(\boldsymbol{x}) \qquad m = 1, .., k \tag{6}$$

The membership degree is computed by the prototype center $\boldsymbol{\mu_i}$ and its variance-covariance matrix A_i using the multivariate Cauchy probability distribution:

$$\beta_i(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{|A_i|}}\left[1 + (\boldsymbol{x} - \boldsymbol{\mu_i})^t A_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i})\right]^{-\frac{n+1}{2}} \tag{7}$$

The incremental learning algorithm of our model consists of three different tasks: the creation of new rules, the adaptation of the existing rule's premises, and the tuning of the linear consequent parameters. These three tasks must be done in an online incremental mode and all the needed calculation must be completely recursive.

3.1 Incremental clustering

When introducing a new training sample in an online learning mode, it will either reinforce the information contained in the previous data and represented by the current clustering, or bring enough information to form a new cluster or modify an existing one. The importance of a given sample in the clustering process can be evaluated by its *potential* value. The potential of a sample is defined as inverse of the sum of distances between a data sample and all the other data samples:

$$Pot_k(\boldsymbol{x}(k)) = \frac{1}{1 + \sum_{i=1}^{k-1} \|x(k) - x(i)\|^2} \tag{8}$$

A recursive method for the calculation of the potential of a new sample was introduced in [8], which made this technique a promised solution for any incremental clustering problem. The recursive formula avoids memorizing the whole previous data but keeps - using few variables - the density distribution in the feature space based on the previous data:

$$P_k(\boldsymbol{x}(k)) = \frac{k-1}{(k-1)\alpha(k) + \gamma(k) - 2\zeta(k) + k - 1} \tag{9}$$

where

$$\alpha(k) = \sum_{j=1}^{n} x_j^2(k) \tag{10}$$

$$\gamma(k) = \gamma(k-1) + \alpha(k-1), \quad \gamma(1) = 0 \tag{11}$$

$$\zeta(k) = \sum_{j=1}^{n} x_j(k)\eta_j(k), \quad \eta_j(k) = \eta_j(k-1) + x_j(k-1), \quad \eta_j(1) = 0 \tag{12}$$

Introducing a new sample affects the potential values of the centers of the existing clusters, which can be recursively updated by:

$$P_k(\mu_i) = \frac{(k-1)P_{k-1}(\mu_i)}{k - 2 + P_{k-1}(\mu_i) + P_{k-1}(\mu_i)\sum_{j=1}^{n} \|\mu_i - x(k-1)\|_j^2} \tag{13}$$

If the potential of the new sample is higher than the potential of the existing centers then this sample will be a center of a new cluster and a new fuzzy rule will be formed in the case of our neuro-fuzzy model. So, the center of the new prototype $\boldsymbol{\mu}_{r+1} = \boldsymbol{x}_k$ and its covariance matrix $A_{r+1} = \epsilon I$, where I is the identity matrix of size n and ϵ is a problem-independent parameter and can generally be set to 10^{-2}.

3.2 Premise adaptation

This adaptation process allows to incrementally update the prototype centers coordinates according to each new available learning data, and to recursively compute the prototype covariance matrices in order to give them the rotated hyper-elliptical form. For each new sample $\boldsymbol{x}_k$, the center and the covariance matrix of the prototype that has the highest activation degree are updated.

The center coordination of the selected prototype is recalculated as follows:

$$\boldsymbol{\mu}_i = (1 - \frac{1}{s_i + 2})\boldsymbol{\mu}_i + \frac{1}{s_i + 2}(\boldsymbol{x}_k - \boldsymbol{\mu}_i) \tag{14}$$

where s_i represents the number of updates that have been already applied on this prototype. The covariance matrix is recursively computed as follows:

$$A_i = (1 - \frac{1}{s_i + 2})A_i + \frac{1}{s_i + 1}(\boldsymbol{x}_k - \boldsymbol{\mu}_i)(\boldsymbol{x}_k - \boldsymbol{\mu}_i)^t \tag{15}$$

For practical issues, since the membership degree can be calculated using only $A^{-1}(|A| = \frac{1}{|A^{-1}|})$, and in order to avoid any matrix inversion, we use an updating rule for A^{-1} directly:

$$A_i^{-1} = \frac{A_i^{-1}}{1 - \alpha} - \frac{\alpha}{1 - \alpha} \cdot \frac{(A_i^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_i)) \cdot (A_i^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_i))^t}{1 + \alpha((\boldsymbol{x}_k - \boldsymbol{\mu}_i)^t A_i^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_i))} \tag{16}$$

where $a = \frac{1}{s_i + 1}$.

3.3 Linear consequent tuning

The tuning of the linear consequent parameters in a first-order TS model can be done by the weighted Recursive Least Square method (wRLS). Let Π_i be the linear consequent parameters of the rule i:

$$\Pi_i = \left[\boldsymbol{\pi}_i^1 \;\; \boldsymbol{\pi}_i^2 \;\; ... \;\; \boldsymbol{\pi}_i^m\right]^t \tag{17}$$

where $\boldsymbol{\pi}_i^c = [a_{i0}^c \;\; a_{i1}^c \;\; ...a_{in}^c]$. It can be recursively estimated as follows:

$$\Pi_i = \Pi_i + C_i \bar{\beta}_i(\boldsymbol{x}_k)\boldsymbol{x}_k(Y_k - \boldsymbol{x}_k\Pi_i), \;\; \Pi_{init} = \mathbf{0} \tag{18}$$

$$C_i = C_i - \frac{\bar{\beta}_i(\boldsymbol{x}_k)C_i\boldsymbol{x}_k\boldsymbol{x}_k^t C_i}{1 + \bar{\beta}_i(\boldsymbol{x}_k)\boldsymbol{x}_k^t C_i\boldsymbol{x}_k}, \;\; C_{init} = \Omega I \tag{19}$$

where Ω is a large positive number, and I is the identity matrix.

4 User Intervention in the Interactive Analysis Process

In this section, we present the possibilities offered by the introduction of a classifier based on incremental learning in our interactive sketch recognizer. In particular we detail when and how the user can interact with the incremental classifier. During the analysis, each time the classifier is solicited to identify a symbol, the decision process uses the confident degree given by the classifier to make its decision. If the decision process considers that the confidence degree is sufficiently high to make the right decision, it validates the recognition. Otherwise, The decision process will solicit the user. The user is then in front of four possibilities (cf. Figure 2):

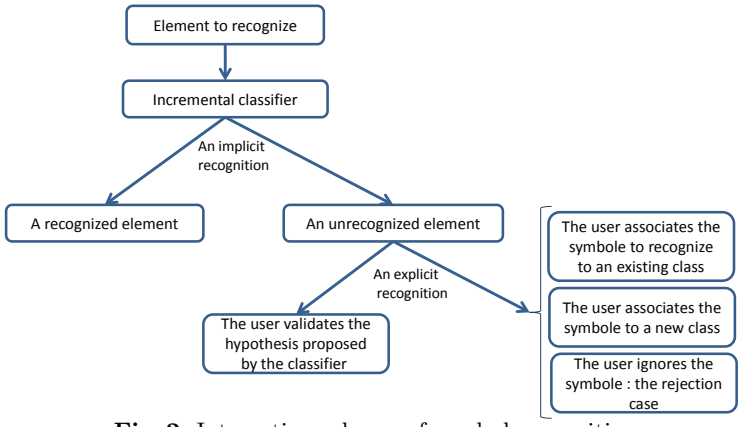


Fig. 2. Interaction scheme of symbol recognition

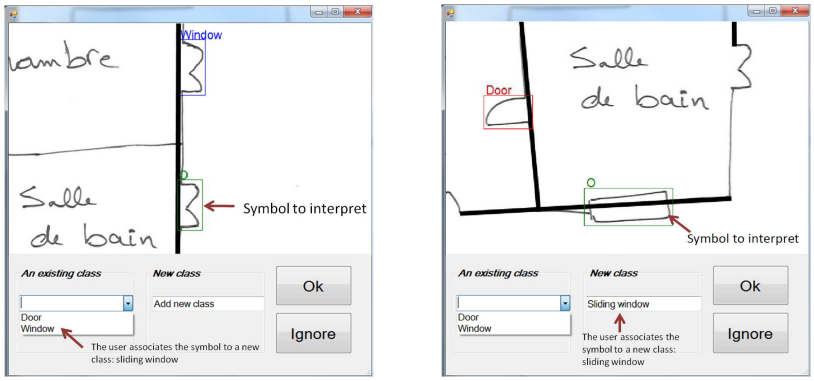
- The user validates the hypothesis proposed by the classifier in spite of the low degree of confidence given by the classifier. The classifier will enhance the model of this class.
- The user associates the symbol to recognize to other existing class in the classifier. The classifier will reduce the confusion between two classes.
- The user associates the symbol to a new class: the user considers that the symbol does not belong to an existing class. With this new information, the classifier will start to learn a new class of symbols.
- The user ignores the symbol to recognize: the rejection case. The user considers that the recognized symbol is an outlier (noise in the image). No action is done by the classifier.

With this interaction process, the classifier continuously learns to improve its interpretations. The more the analysis is going on, the more the classifier is accurate, the less the user is solicited. This incremental learning is able to deal with the recognition of new classes of symbols. It is a key point to absorb the great variability of symbols that can occur in a sketch.

Figure 3 shows a case where the user solicitation is judged necessary to interpret a handwritten architectural floor plan (Figure 4(a)). In this intervention the user is in front of the four possible actions described in section 4. The system presents an interface that contains the hypothesis given by the classifier, the other available classes of the classifier and a field where the user can add a new class. Figure 3(a) shows a case in which the user indicates that the symbol to recognize is a classical window. Figure 3(b) shows a case in which the user associates the symbol to a new class of windows (a sliding windows). The user participation has a great impact to avoid error accumulation during the analysis step. This solicitation allows the classifier to learn from confused recognized symbols. Adding a new symbol causes the creation of a new class in our classifier.

5 Confusion reject

The purpose of the confusion rejection is to assess the reliability of the classifier by detecting patterns for which the classifier is likely to misclassify. These errors are near the decision boundaries because scores of at least two classes are nearly



(a) The user associates the symbol to a 'Window'

(b) The user associates the symbol to a new class (sliding window).

Fig. 3. User interventions. Four possibilities exist. The user associates the set of primitives located in the bounding box 'O' to a right class.

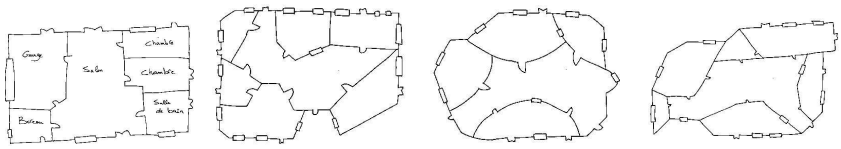
equal. Confusion reject can be realized by defining a reject zone on each side of decision boundaries. Each pattern within one of these zones is considered as potential error and is therefore rejected.

To formalize the confusion reject we use the notion of reliability functions. Here, the reliability function $\psi(X)$ represents the degree of confusion in classifying an sample X :

$$\psi(X) = (Sc_1(X) - Sc_2(X)/Sc_1(X) \tag{20}$$

where $Sc_1(X)$ is the score obtained for the best class and $Sc_2(X)$ is the score obtained for the second best class. A sample X is then rejected when the degree of confusion is below a specific threshold λ. The threshold value represents the width of the reject zones around decision boundaries.

6 Experimental Results

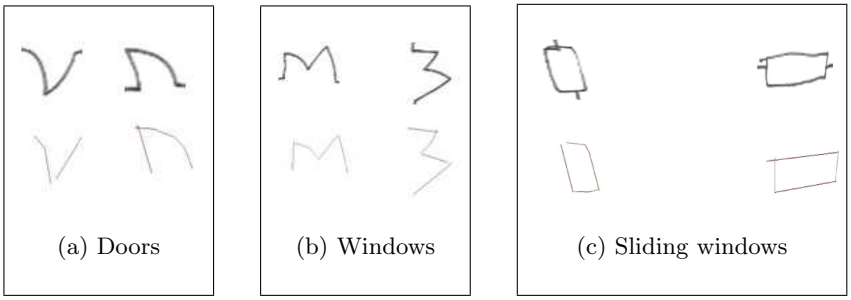


(a) (b) (c) (d)

Fig. 4. Examples of architectural plans

We analyze in this section the effect of our incremental learning approach on the evolution of the classification performance. This performance is measured by two values: error rate and rejection rate. We aim at reducing the recognition errors and minimizing as much as possible the number of user interventions. The entire dataset used in these experiments contains 1500 samples from three different classes (door, window, sliding window). Some examples of symbols are illustrated in Figure 5. We divide the dataset into three subsets:

– Initial learning subset: used to train the classifier in full-supervised manner, i.e. the label of each sample is given by the user. This subset contains 113 samples in our experiments.
– Evaluation subset: used to evaluate the classifier performance by measuring error and rejection rates. 378 samples are used in this subset.
– Incremental learning subset: used to improve the classifier performance by soliciting user intervention when a confusion reject is detected. It contains 1019 samples.



(a) Doors (b) Windows (c) Sliding windows

Fig. 5. Examples of symbols

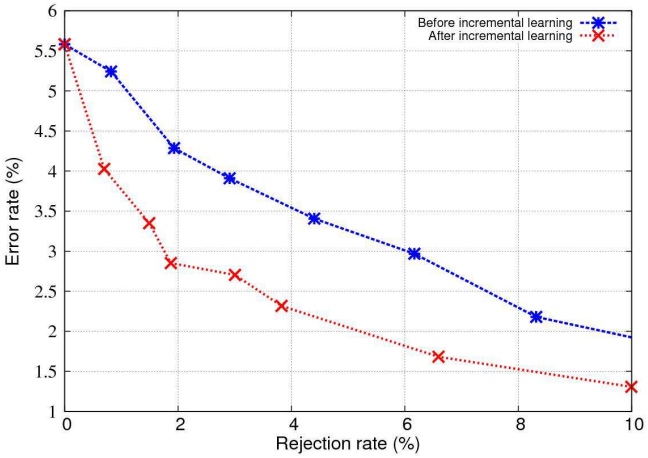
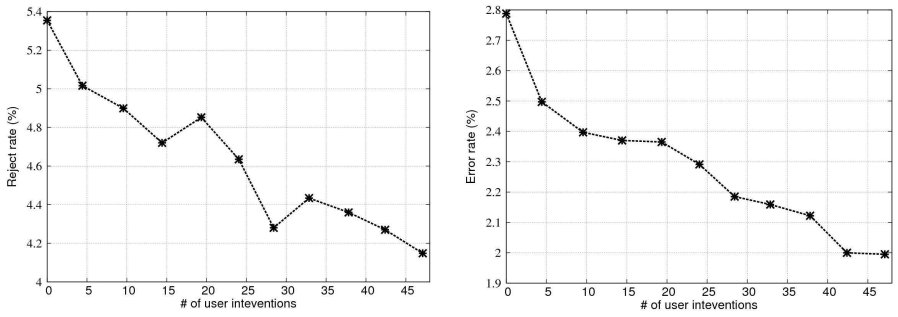


Fig. 6. Error/reject rates before and after the incremental learning process

The experiments have been repeated for different rejection threshold. We can see in Figure 6 the different performance points (error, reject) only using the initial learning subset in the first curve , and then using the incremental learning subset in the second curve. We note the incremental learning process improves the classifier performance thanks to user interventions for rejected samples. In order to give an idea about the number of interventions required by the incremental learning process, we show in Figure 7 the evolution of error rate and rejection rate according to intervention numbers, for a specific rejection threshold value $\lambda = 0.5$. We notice that the estimated classifier error rate has been reduced by about 30% after 40 user interventions, and the estimated rejection rate has also been reduced by about 22%.

X



(a) (b)

Fig. 7. Evolution of Error/Reject rates during the incremental learning process ($\lambda = 0.5$)

7 Conclusion

In this paper, we have presented the integration of a classifier based on an incremental learning method, in an interactive method for interpreting the 2D architectural floor plans. The role of classifier is to recognize the symbol with a degree of confidence. If this degree of confidence is considered insufficient by the decision process to take the right decision, the analyzer solicits the user to validate the right hypothesis. The user is then in front of four possibilities. He can either confirm the recognition proposed by the classifier, or associates the symbol to an existing class, a new class, or ignores this recognition. The classifier is incrementally learned during the analysis phase. This strategy offers an auto-evolutionary method for sketch interpretation.

Acknowledgment

The authors would like to thank all the people who took part in the experiments. This work benefits from the financial support of the ANR Project Mobisketch.

References

1. K. Chan and D. Yeung, "An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions," *Pattern Recognition*, vol. 33, no. 3, pp. 375–384, 2000.
2. J. Fitzgerald, F. Geiselbrechtinger, and T. Kechadi, "Mathpad: A fuzzy logic-based recognition system for handwritten mathematics," in *ICDAR 2007*, 2007.
3. S. Mao, A. Rosenfeld, and T. Kanungo, "Document structure analysis algorithms: a literature survey," in *Proc. SPIE Electronic Imaging*, vol. 5010, 2003, pp. 197–207.
4. B. Coüasnon, "Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way," *IJDAR 2006*, vol. 8, no. 2.
5. A. Ghorbel, S. Macé, A. Lemaitre, and E. Anquetil, "Interactive competitive breadth-first exploration for sketch interpretation," in *ICDAR*, 2011, pp. 1195–1199.
6. S. Macé and E. Anquetil, "Eager interpretation of on-line hand-drawn structured documents: The dali methodology," *Pattern Recognition*, pp. 3202–3214, 2009.
7. A. Almaksour and E. Anquetil, "Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers," *Evolving Systems*, vol. 2, pp. 25–33, 2011.
8. P. Angelov and D. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 34, pp. 484–498, 2004.