Multi-view HAC for Semi-supervised Document Image Classification

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

[1] A2iA SA
40 bis, rue Fabert
75 007 Paris cedex - France
Fabien.Carmagnac@a2ia.com
http://www.A2iA.com
[2] Laboratoire PSI
CNRS FRE 2645 - Université de Rouen
76 821 Mont-Saint-Aignan cedex - France
Pierre.Heroux@univ-rouen.fr, http://www.univ-rouen.fr/psi

Abstract. This paper presents a semi-supervised document image classification system that aims to be integrated into a commercial document reading software.
This system is asserted like an annotation help. From a set of unknown document images given by a human operator, the system computes regrouping hypothesis of same physical layout images and proposes them to the operator. Then he can correct them, validate them, keeping in mind that his objective is to have homogeneous groups of images. These groups will be used for the training of the supervised document image classifier. Our system contains N feature spaces and a metric function for each of them. These allow to compute the similarity between two points of the same space. After projecting each image in these N feature spaces, the system builds N hierarchical agglomerative classification trees (HAC) corresponding to each feature space. The proposals for regroupings formulated by the various HAC are confronted and merged. Results, evaluated by the number of corrections done by the operator are presented on different image sets.

1 Introduction

Recent improvements in pattern recognition and document analysis led to the emergence of applications that automate document processing. From a scanned document, some software are able to read its handwritten or machine printed content or to identify some symbols or logos. Others can retrieve the category (later on called "class") from which it belongs. However, a training step is necessary while a human operator gives image samples with the same layout for each class. Generally these images are representative of the stream to sort.

For example, sorting incoming mails in companies allows to redirect an unknown document to the right department or to apply an appropriate processing

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 191–200, 2004.
© Springer-Verlag Berlin Heidelberg 2004

depending on its document class [1] [2]. However, these softwares are not able to extract all the information on the image yet and a human operator has to define the tasks that have to be accomplished by the software depending on the document class of the image.

The proposed approach improves the functionnalities of an existing software (A2iA FieldReader). At the beginning, this application was able to read handwritten and machine printed fields on documents coming from a homogeneous stream of documents. All of them were sharing the same reading model. Then, a supervised document classifier was added, allowing to process documents from several classes: after a training step, the system was able to find the class of an unknown document. The reading model of each class, containing position, type and semantic of the fields to read drives reading module. The supervised classifier must automatically find the most discriminating feature set for any set of images and any number of classes because the users are not specialists in image analysis. Another difficulty is that a human operator has to give few samples of document image per class to constitute a training database for the supervised classifier. This task becomes quickly difficult or even impossible if the database is composed of images coming from tens of different document classes and all the more if images of different classes have small layout differences. So, the training databases contain usually only a few samples. The classification method presented in [3] proposes a solution for these constraints.

In this article, we propose a semi-supervised classification system inspired by Muslea and al. [4] that aims to be a help for annotation without *a priori* on the number of classes and their characteristics. Then, it is difficult to know which features are discriminating for a given set of images and classes. From a document image set of a heterogeneous stream, the system proposes to the operator some groups of images with the same layout. Thanks to a GUI, the operator can validate or correct these propositions. Few corrections are allowed: semi-automatic merging or splitting of groups, adding or removing documents in proposed groups.

In section 2, we briefly present a few methods of unsupervised classification and justify our choice of the hierarchical agglomerative classification method. We describe our semi-supervised algorithm in section 3. Then, results on five different image databases are presented in section 4. Conclusion and future improvements are mentionned in section 5.

2 Unsupervised Classification Algorithms

A state of the art of unsupervised classification can be found in [5], [6] and [7]. We remind here the main methods.

The K-means algorithm provides the best partition of a set E in k groups of elements well agregated and well separated in the feature space but our system must work without the knowledge of the expected number of class because in most of cases even the operator does not know it.

Self organising maps are based on a neural network with neighbourhood constraints. They do not need the knowledge of the expected number of class

but a big number of samples is necessary to make them converge. We can also notice that this convergence is not guaranteed for feature vectors with a size greater than one [8].

Hierarchical Agglomerative Classification (HAC) is an algorithm allowing to get a hierarchy of sets of the considered data and have the interest to propose a data structure without knowing the number of expected classes. The result is a tree where each node represents a group and the root contains the whole elements. Various existing criteria allow to cut some edges of the tree and to make groups with the elements contained in the descendant-nodes [8].

Among these three classical methods of unsupervised classification, the HAC seems to be the most interesting one to resolve our problem. Indeed, the drawback of the computiong complexity in compensed because only few samples are used. As HACs only need the definition of a distance they may be built with numerical, syntactic or structural data. On the other hand SOM will lack samples to garantee the convergence and the K-means method needs the number of expected classes. However, all of them work from numerical data extracted from the images of the training set. These images, often noisy, will introduce variability in the features. To correct these errors the introduction of a semantic level would be appropriate like extracting well identified graphical objects (such as boxes, titles, combs, etc.). This solution introduces a bias we forbid because it will lead to develop a big database of concurrent extractors. Our idea is to have few feature spaces in which we will project the images and build a HAC tree for each space. Having a big feature vector, result of the concatenation of some vectors bring us back to the problem just evoked. So we will get as many HAC trees as feature spaces. These features are different: visual (seeking white or black zones of the image, average grey value, etc.), structural (lines counting, rectangle extracting, etc.) and statistics (differential of connected components size, etc.). Each HAC will voice regrouping hypothesis that will be all exploited in parallel to finally find out the groups that must be submitted to the operator.

3 Multi-view HAC Algorithm

3.1 Few Definitions

Let $ImageSet$ be the training set. Let $FeaturesSet$ be the available feature space set. For any feature space E, a function F_E that projects an image in E is defined by:

$$E \in FeaturesSet, F_E : ImageSet \rightarrow E$$

For any feature space E, a function M_E that computes the distance between 2 points of E is defined by:

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

For any feature space E, a function D_E that computes the distance between 2 images of $ImageSet$ is defined by:

$$E \in FeaturesSet, D_E : ImageSet \times ImageSet \rightarrow \mathbb{R}^+$$

$$E \in FeaturesSet, (I_1, I_2) \in ImageSet^2, D_E(I_1, I_2) = M_E(F_E(I_1), F_E(I_2))$$

The function denoted as F_E of the feature space E projects an image in the space E. The function denoted as M_E computes the distance between two points of the feature space E. To simplify the notation, we will note D_E the function that computes the distance between two images in the feature space E.

3.2 Building a HAC Tree

Here is the building algorithm of an HAC tree for a given feature space E:

1. Initialize a list L with one group per image of $ImageSet$
2. Compute the distance between all images of $ImageSet$
3. Merge in a group G, the two closest groups A and B
4. Remove A and B from L and add G to L
5. Compute the distance between G and all groups of L
6. If L contains more that one group, go back step 3

This algorithm needs to define two distances. The first one has to compute the distance between two images (step 2): it is the distance D_E defined in 3.1. The other one has to compute the distance between two groups of images (step 5). It can be defined by:

- Diameter of the $G \cup G'$ set. The choice of this distance allows to have a measure of the variability of the $G \cup G'$ group: $Max_{I \in G, I' \in G'}(D_E(I, I'))$
- Minimal distance between points of each group: $Min_{I \in G, I' \in G'}(D_E(I, I'))$
- Average distance between the points of the union of the two groups:

$$\frac{\sum_{I,J \in G \cup G', I \neq J} D_E(I, J)}{||G \cup G'||}$$

When a G group is created (step 3) with the two closest groups A and B, the distance between A and B is also the heigth of the G group. That is why this tree structure is often represented by a dendogram.

The algorithm stops when only one group is remaining. This group is called the root of the HAC tree and contains all the images.

So, our algorithm builds as many HAC trees as available feature spaces in the system.

3.3 Extraction of Grouping Hypothesis Common to Different HAC

The system has now several HAC trees that represents different structures of the same data. For any pair of HAC trees, we extract every groups (nodes) containing the same images in two trees. These groups can be considered as regrouping hypothesis shared by different points of view. We will denote *Select*, the set of the nodes appearing in at least two HAC. The system has now a set of groups shared by several HAC trees, so *a priori* the most reliable groups.

3.4 Building the Minimal Inclusion Forest

The system establish hierarchical links between the nodes of the *Select* list as following: the father of a given node N is the smaller node containing N. The result is a forest F (set of trees).

Figures 1 and 2 presents two inclusion forests. Each group (node) contains its image list with the following syntax: $[C]\_[N]$ with C as the identifier of the class and N the identifier of the image inside the class. The coloured nodes are homogeneous (images of the same class) and the nodes with white background contain images from different classes.

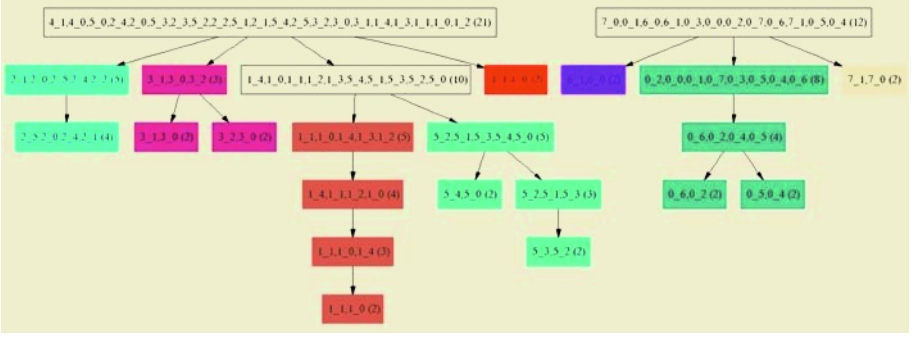


Fig. 1. Forest of DB1. HAC built with the Min distance.

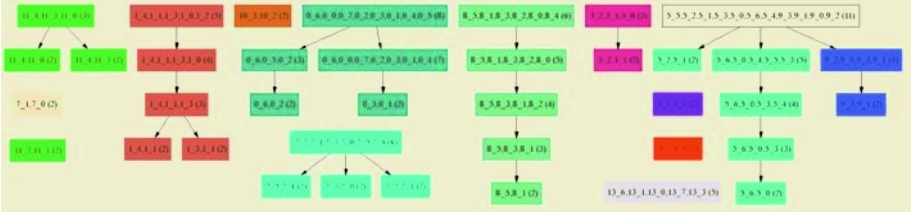


Fig. 2. Forest of the DB4. HAC built with the Max distance.

The forest of figure 1 contains two trees, non homogeneous. For each class, we can find a node containing every images of the class.

The forest of figure 2 contains thirteen trees but only one is not homogeneous. This node contains images of two different classes. One class has given two roots without link between them (class 11, left of the image).

3.5 Presenting the Forest to the Human Operator

For each tree of the forest, the contained images of a group are presented to the operator in an array of thumbnails. In front of a G group, the operator can:

- Validate G if the images are from the same class. The group is ready for a possible merge with another group.
- Reject G if it contains images different layout structure. In that case, the system removes G and presents the groups of the descendant nodes of G to the operator. Experimentally, that case is frequent because the structure of the groups is done with numerical heuristics so the probability that a group is homogeneous decreases when its size increases.
- Merge G with another group G' if the images of G and G' are of the same over-segmented class. Beforehand these groups have to be validated. The system replace G and G' by a G'' group, union of the images contained in G and G'. It is the case when only a part of the images of a class has the same default. For example, a black logo can be whitened or not by an adaptative binarization. It seems natural that the algorithm separates the images in two sub-classes if the logo is whitened or not.

4 Results

The results for five image sets and for one database regrouping four sets are presented in Table 1. Sample images are presented in Fig. 3, 4, 5, 6 and 7. The image classes are composed by a random number of images (from 3 to 10) randomly drawn in a database containing thousands of images.

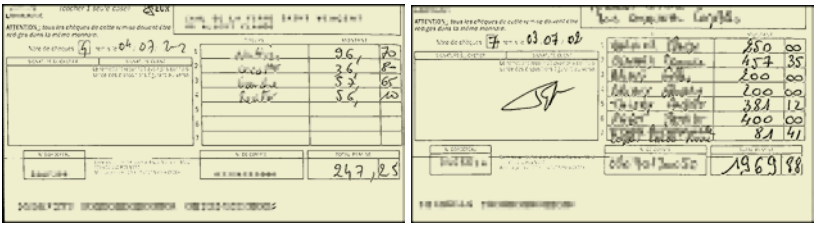
Table 1. Operator cost to get homogeneous groups.

	DB 0	DB 1	DB 2	DB 3	DB 4	DB 1,2,3,4
#Images (total)	15	33	31	31	70	165
#Classes	2	8	6	6	15	35
Rejects	0	3	5	5	1	6
Merges (after validation)	0	0	1	0	1	3
Classified Images	100%	100%	100%	100%	81%	99%
Well Formed Classes	100%	100%	100%	100%	93%	100%

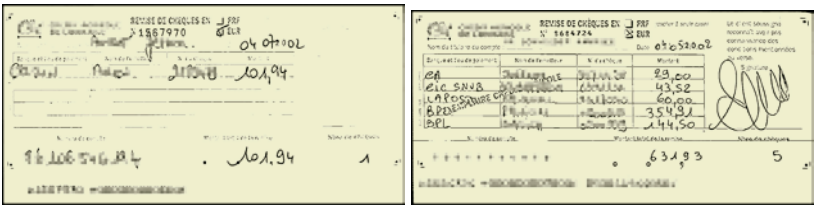
We call "Classified Images", the part of images inside the inclusion forest. For example, for the database 4, "81% of the 70 images have been classified" means 13 images are not inside the inclusion forest. Experimentally we have noticed that these images have significant default compared to the other images of the class.

We call "Well Formed Classes", the part of classes found by the classifier. For example, for the database 4, "93% of the 15 classes have been well formed" means that one class has not been retrieved.

At the end of the corrections done by the operator, the system will learn the images of the validated groups. It will remind the operator that some images have not been learned because they were not included in any of the trees and will try to classify them with the approval of the operator. Then, the operator can finish the configuration of the learning classes.



(a) Class 1 Images (2 kinds of documents)



(b) Class 2 Images (2 kinds of documents)

Fig. 3. Image Samples of DB0.

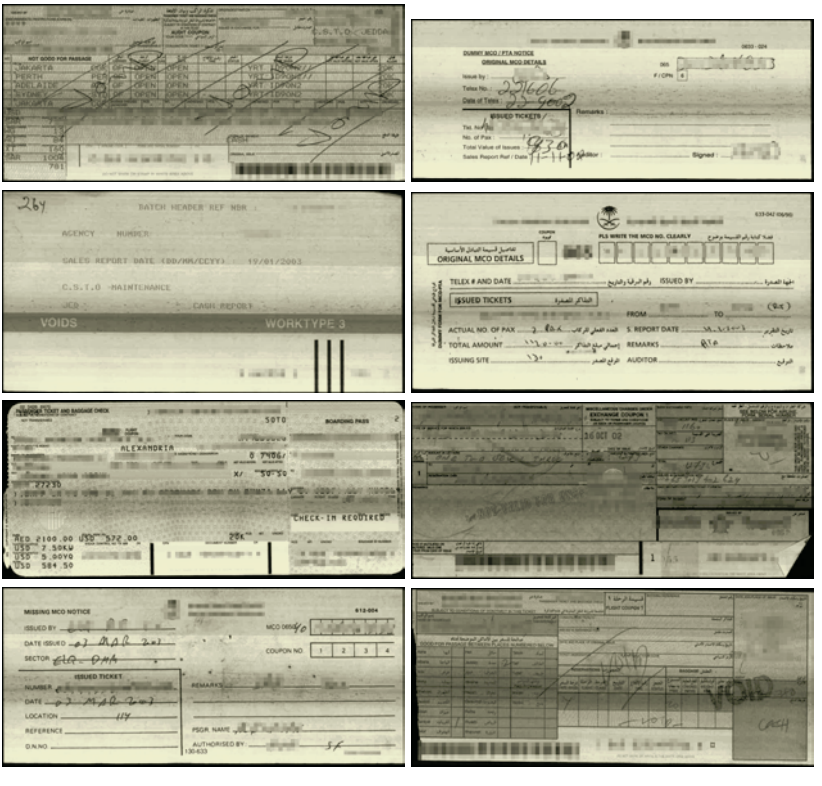


Fig. 4. Image Samples for the 8 classes of DB1.

Fig. 5. Image Samples for the 6 classes of DB2.

Fig. 6. Image Samples for the 6 classes of DB3.

5 Conclusion

This article presented an effective technique of semi-supervised classification. We tried to introduce a multi-view notion with different feature spaces to prevent the blindness due to purely numerical considerations induced by the HAC trees. On the other hand, the HAC trees free us from the problem of the form of the clusters in the different feature spaces and their number, information that even the operator does not know. The performance criteria depending *in fine* of the number of corrections the operator has to make to get homogeneous classes, we have to consider carefully the way to present the results of our algorithm to the operator.

As most of the unsupervised classification systems, after computing the distances between the images, we do not exploit the images anymore whereas they are shown to the operator. It could then be judicious to design an algorithm which would automatically extract a set of graphical objects as well as their neighbourhood relationships. The system would justify the presentation of a group to the operator by the presence of these objects as well as the validation of their neighbourhood relationship on all images of the group. Graphical objects could be extracted without *a priori* knowledge not to bring back to the problems evoked in section 2 but with a simple geometric severe criteria in order to limit errors.

Morever, it would be interesting to try to cut the trees of the forest in order to directly present homogeneous image groups to the operator. However, tests were carried out on these bases with various cut criteria but all of them were more

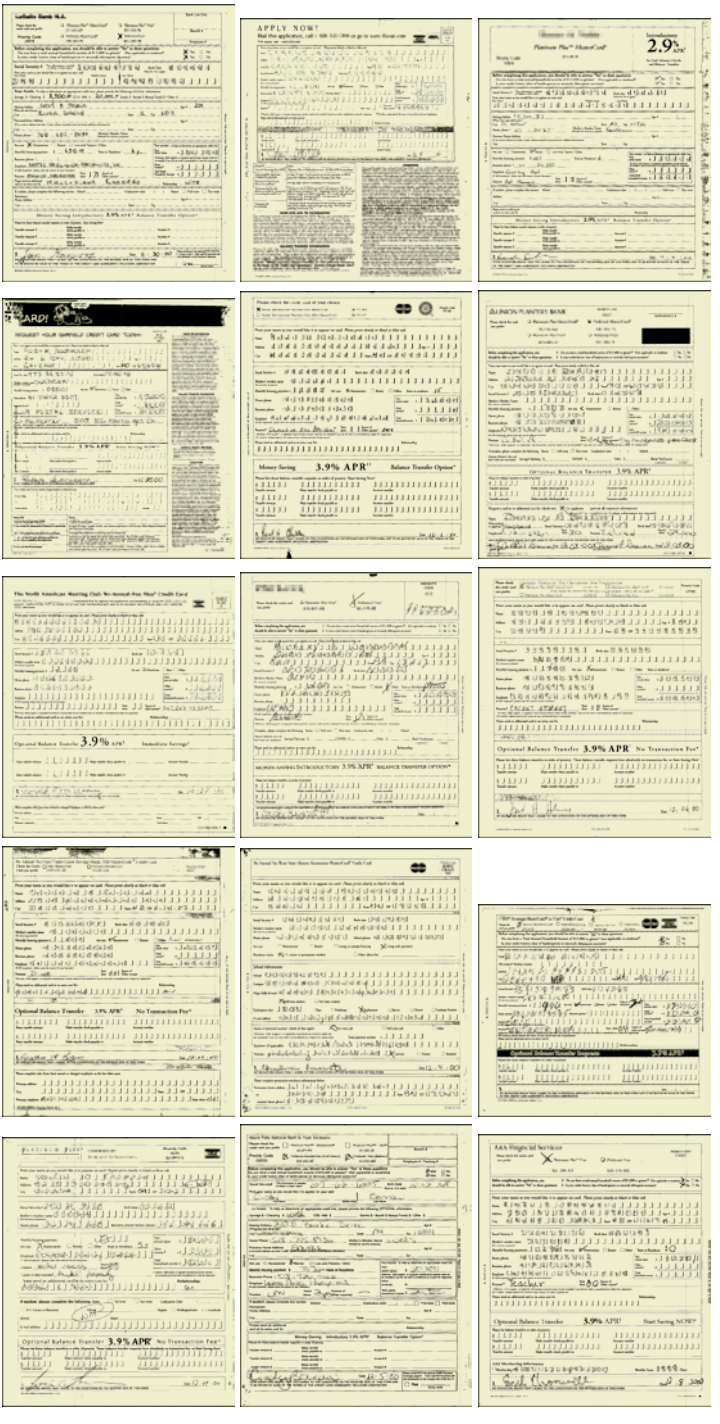


Fig. 7. Image Samples the 15 classes of DB4.

expensive for the operator than by presenting the inclusion forest. We can then think that it is illusory to try to make homogeneous groups automatically. Indeed let us recall that this system help the operator to quickly form classes of images for the training of the document sorting software. Thus if automation creates errors on the learned classes, the consequences are serious on the effectiveness of the document classification system.

References

1. Koch, G., Heutte, L., Paquet, T.: Numerical sequence extraction in handwritten incoming mail documents. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR'2003. (2003) 369–373
2. Clavier, E.: Stratégies de tri: un système de tri des formulaires. Thèse de doctorat, Université de Caen (2000)
3. Carmagnac, F., Héroux, P., Trupin, E.: Distance Based Strategy for Document Image Classification. Lecture Notes in Computer Science. In: Advances in Pattern Recognition. Springer-Verlag (2004) to be published.
4. Muslea, I., Minton, S., Knoblock, C.: Active + semi-supervised learning = robust multi-view learning. In: Proceedings of the 19th International Conference on Machine Learning (ICML 2002). (2002) 435–442
5. Fukunaga, K.: Introduction to Statistical Pattern Recognition. 2^{nd} edn. Academic Press Inc. (1990)
6. Cornuéjols, A., Miclet, L.: Apprentissage artificiel - concepts et algorithmes. Eyrolles (2002)
7. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys **31** (1999) 264–323
8. Ribert, A.: Structuration évolutive de données: Application à la construction de classifieurs distribués. Thèse de doctorat, Université de Rouen (1998)