

Encoding of Modified X-Y trees for document classification

Francesca Cesarini, Marco Lastrì, Simone Marinai, and Giovanni Soda
Dipartimento di Sistemi e Informatica - Università di Firenze
Via S.Marta, 3 - 50139 Firenze - Italy
tel: +39 055 4796361. fax: +39 055 4796363
e-mail: {cesarini,simone,lastrì,giovanni}@dsi.unifi.it

Abstract

In this paper we describe a method for classifying document images on the basis of their physical layout. The layout is described by means of a hierarchical description, the Modified X-Y tree, that is derived by the classical X-Y tree segmentation algorithm taking into account cuts along lines in addition to cuts along white spaces between blocks. In order to reduce problems due to noise and skew of the input image, the Modified X-Y tree is built on top of regions extracted by a commercial OCR. The tree is afterwards coded into a fixed-size representation that takes into account occurrences of tree-patterns in the tree representing the page. Lastly, this feature vector is fed to an artificial neural network that is trained to classify document images.

The system is applied to the classification of documents belonging to Digital Libraries, examples of classes taken into account are "title page", "index", "regular page". Many tests have been carried out on a data-set of more than 600 pages belonging to an on-line Digital Library. These tests allowed us to conclude that the use of MXY trees is advantageous with respect to the classical XY decomposition for this classification task.

Keywords: *Digital Library, Page classification, Modified X-Y tree, Tree encoding, Artificial Neural Networks.*

1. Introduction

Document page classification aims at assigning an appropriate label to a document image, and is helpful in many application domains, for instance in form processing systems and in Digital Libraries. Page classification can be considered as a sub-task of page layout analysis, since a physical/logical meaning is assigned to each page. In layout analysis, learning methods (e.g. neural networks) can be used for classification purposes at three levels: pixel classification (employed for binarization and region segmentation), region classification, and page classification.

Whereas labels for pixel and region classification are quite standard (e.g. we can distinguish among text, graphics, and line drawing regions) a broad range of targets can be taken into account for page classification. For instance, the *type of document* can concern "technical paper" or "commercial letter"; the *editor* can be "IEEE" or "ACM"; the *layout* can be based on 2 columns or 1 column; the *type of page* can be "title page", "regular page", or "index page".

Different representations can be considered as input to the process of document page classification. *Sub-symbolic* features, like the density of black pixels in a region, are computed directly from the image. *Symbolic* features, for instance the number of horizontal lines, are extracted from a segmentation of the image. *Structural* features (e.g. relationships between objects in the page) can be computed from a hierarchical description of the document. *Textual* features, for instance presence of some keywords, are obtained from the text in the image recognized by an OCR.

Page classification has been addressed with different objectives and methods. Most work concerned form classification that is aimed at selecting an appropriate reading method for each form to be processed. Form classification often take into account the presence of ruling lines in the pre-printed form layout [9, 11, 15]. Other approaches address the problem of grouping together similar documents in business environments, for instance separating business letters from technical papers [5, 14]. In the last few years the classification of pages in journals and books received more attention [8, 10, 13].

In this paper we describe a page classification system aimed at splitting pages (belonging to journals or monographs in Digital Libraries) on the basis of the type of page, considering a structural representation of the layout of the page. This page classifier is under development in the domain of the METAe European project¹. METAe is focused on the semi-automatic extraction of meta-data from scanned documents of historical books and journals, in or-

¹METAe: the Metadata engine. <http://meta-e.uibk.ac.at>

der to make the digital conversion of printed material more reliable in terms of digital preservation. In the context of METAe, the role of page classification is crucial for both extracting page-level meta-data (e.g. identification of table of contents page) and narrowing the set of pages where to look for some specific meta-data (e.g. identifying the title page helps to retrieve the title of a book).

The paper is organized as follows, in Section 2 we summarize previous work on the encoding of graphs into fixed-size feature vectors, in Section 3 we describe the proposed classification method, experimental results are reported in Section 4, while conclusions are drawn in Section 5.

2. Encoding of structural representations

Structural representations of objects are very appropriate in the domain of document processing. For instance, printed and handwritten characters can be described by analyzing the contour (or the skeleton) of their strokes. Likewise, the page layout can be represented by using graphs [16] or trees [12]. Most trainable classifiers (e.g. decision trees and neural networks) require a fixed-size feature vector as input to the decision algorithm. Some approaches (briefly outlined in the following) have been considered for the mapping of a graph-based representation into a fixed-size vector to be used by a neural classifier.

One approach (e.g. [1]) is based on the assignment of some pre-defined slots of the feature vector to each node and edge of the graph. In order to represent features of nodes and edges more slots correspond to each item. This approach is appropriate when the maximum size of the graph is bounded, and when a robust ordering algorithm of nodes and edges is available. One advantage is the property that the mapping from the graph to the vector is without loss of information, and that real attributes in nodes and arcs can be taken into account.

A different approach for dealing with structural representations relies on the encoding of the graph structure into the architecture of the network, instead of representing the graph (or the tree) into a fixed-size vector. This approach is based on recursive neural networks [7] which are extensions of recurrent networks, and have been used for logo recognition [6]. In this case attributes of the nodes can have real values, however a limit of the approach is that the maximum number of children for each node must be known and pre-fixed.

Another method, that is appropriate when node and arc attributes have discrete values, is based on the computation of the occurrences of each possible combination of attribute values of nodes and edges of the graph. For instance, in [4] the skeleton of handwritten characters is represented by a set of circular arcs and described with an ARG (Attributed Relational Graph), whose nodes (described by 3 features)

represent the circular arcs, whereas edges represent relationships between pairs of connected arcs (described by a 2-tuple). In the proposed encoding each combination of values of the 3-tuple, and each combination of values of the 2-tuple, corresponds to one item of the feature vector. The value of each item is the number of occurrences of the corresponding tuple in the description of the character. With this method variable size graphs can be taken into account, however the structure of the graph is lost during the encoding, and only discrete attributes can be considered for both nodes and arcs of the graph.

A related approach for statistical document modeling is based on generalized N-grams [2]. In this paper, the tree structure of logical documents is represented by probabilities of local tree node patterns similar to mono-dimensional N-grams, which are generalized in order to deal with trees. The generalization proposed in [2] is obtained by considering “vertical” N-grams (describing ancestor and child relations) in addition to the more usual “horizontal” N-grams (corresponding to sibling relations). With this approach, the local structure of the tree is partially preserved, since the relationships among nodes are taken into account.

3. Page classification

In this paper, page classification is performed with a sequence of operations that is aimed at encoding the hierarchical structure of the page into a fixed-size feature vector. The structure of the page is described with a Modified X-Y tree (MXY tree in the following) [3] that is built considering the regions extracted by a commercial OCR, instead of dealing directly with the document image. MXY trees are encoded into a fixed-size representation that takes into account the occurrences of some specific tree-patterns in the tree corresponding to each document image. Lastly, this vector of features is fed to an MLP that is trained to classify document images according to the labels assigned to training data.

3.1. Symbolic building of Modified X-Y tree

The X-Y tree [12] is a top-down data-driven method for page layout analysis. The basic assumption behind the X-Y tree segmentation is the property that elements of the page (columns, paragraphs, figures) are generally laid out in rectangular blocks. Furthermore, the blocks can usually be grouped in such a way that blocks that are adjacent to one another within a group have one dimension in common. In the basic X-Y tree algorithm the root of the tree corresponds to the whole document, that is then split into regions separated by horizontal or vertical white spaces, which are found by looking for “valleys” in the horizontal or vertical projection profile computed from the image. Each region

corresponds to a child of the root, and the algorithm is recursively applied on each subregion.

The MXY tree extends the basic X-Y tree approach by taking into account splitting of regions into sub-parts by means of cuts along horizontal and vertical lines, in addition to the classical cuts along white spaces. Each node of an MXY tree is associated either to a region of the page or to a horizontal or vertical line. In particular, internal nodes can have four labels (corresponding to two cutting directions, and two cutting ways), and leaves can have four labels.

Segmentation capabilities of the X-Y tree algorithm (and also of the MXY tree) depend on the level of noise in the image, and are heavily affected by the presence of skew. In order to overcome these problems of the X-Y segmentation algorithm, but preserve its powerful hierarchical representation, we combined the MXY tree algorithm with the segmentation provided by a commercial OCR. To this purpose we developed a symbolic algorithm that builds the MXY tree starting from homogeneous regions extracted by a commercial OCR, instead of dealing directly with the image. Since horizontal and vertical lines are not provided by the OCR, we look for them in zones of the image not covered by regions found by the OCR. Segmentation points corresponding to horizontal and vertical white spaces, are located by computing an approximate projection profile. This profile is obtained by considering an uniform contribution from each region either in the horizontal or vertical direction. The amount of contribution to the profile depends on the average number of black pixels in each region, and this value can be either extracted directly from the image or estimated on the basis of the number of characters in the region, and their average size. A side effect of this approach is that noise in the image (not included in segmented regions) does not affect the building of the MXY tree.

3.2. Tree encoding and classification

In this paper, we propose an encoding method that is used for the classification of trees describing the page layout. The idea that is behind this encoding is the observation that similar layout structures often have similar sub-trees in the corresponding MXY representation (for instance see Figure 3).

In real cases, however, we cannot expect to find exactly the same sub-tree in all the trees of a given class. For instance, a block of text can be sometimes split into two or more sub-parts, for other documents. Due to this variability in the size of the common sub-trees, we describe each tree by counting the occurrences of some tree-patterns composed by three nodes. This approach is somehow similar to the encoding described in [4] and to the generalized N-grams [2]. The main difference with respect to generalized N-grams is that the tree-patterns taken into account are com-

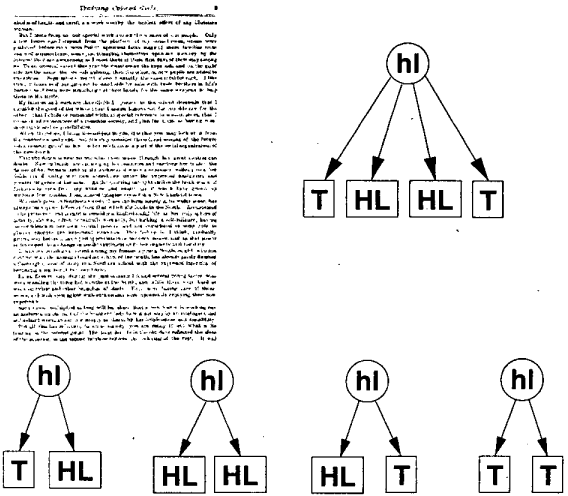


Figure 1. A simple page with the corresponding MXY tree, and the four balanced tree-patterns found in the tree. Non adjacent nodes are considered in the last pattern.

posed by three nodes connected one to the other by a path in the tree. On the contrary, generalized N-grams include also patterns made by three siblings without taking into account their parent. Trees composed by three nodes can have two basic structures: one composed by a root and two children (referred to as balanced tree-pattern), and one composed by a root, a child, and a child of the second node. Four labels can be assigned to each internal node: hs, vs (for cuts along spaces), hl, vl (for cuts along lines). Each leaf can have four labels: HL (horizontal line), VL (vertical line), T (text region), and I (image). Leaves of tree-patterns can correspond either to a leaf of the MXY tree or to an internal node, consequently internal nodes of a tree pattern can get four values, and leaves can have eight values. Taking into account all the combinations of labels, 384 possible tree patterns can be defined.

A special care is required by the balanced tree-patterns. Since siblings in the MXY tree can be ordered according to their horizontal or vertical position (depending on the cutting direction described in their parent), the relative position between contiguous blocks is preserved in this description. However, due to noise (or simply variable layouts of the documents), one sub-tree can differ from the reference one only for a node that is inserted between two representative siblings. In order to overcome this problem, when computing the tree-patterns appearing in a MXY tree, we look also for non-adjacent children (Figure 1), and this is another difference with respect to generalized N-grams.

The encoding just described takes into account only discrete attributes in the nodes of the tree. In order to consider also some information about the size of regions considered in the nodes, we added four values in the feature vector, that take into account the size of textual blocks belonging to the same tree-pattern. Textual blocks are labeled as “small” or “big” depending on the ratio of their area with respect to the area of the page. Blocks with area lower than a fixed threshold are labeled as “small”, whereas larger blocks are labeled as “big”. Therefore each tree-pattern containing textual leaves can belong to one of four classes according to the possible combinations of size labels of the leaves. The four features bringing size information are obtained by computing the relative distribution of each of the four combinations in the encoded MXY tree.

After extracting the vectorial representation of the MXY tree corresponding to a page, various algorithms can be used for the classification. In this paper we addressed the problem with a classical MLP-based classifier (trained with the Backpropagation algorithm), which takes the normalized feature vector as input, whereas outputs describe, with an one-hot coding, the membership of each example. One problem with such an approach (that is common to other classification methods) is the large size of the feature vector, since many combinations of node labels can be considered. From a practical point of view, we can easily find out that not many tree-patterns are found in actual documents of a given data set, as we will analyze in the next section on experimental results.

4. Experimental results

We carried out a set of experiments in order to evaluate the classification results that can be achieved by taking into account the information about the relative size of textual blocks, by using cuts along lines in addition to the classical cuts along white spaces, and by considering also non-adjacent leaves when computing occurrences of balanced tree-patterns. The experiments are made with a data-set of pages belonging to a historical journal: the *American Missionary* journal, that is available in the on-line Digital Library *Making of America*². We considered five classes having different layout, and appearing in each issue of the Journal. Samples of the 5 classes are shown (together with the corresponding MXY trees) in Figure 3. Some classes have a very stable layout (e.g. the “first page” and the “index page”), whereas other classes have a more variable layout (e.g. the “advertisement” class) and give rise to most errors. For each experiment the documents are split into two classes: one is used for training the neural network, and the other is considered for testing purposes. The training set

²Document images can be downloaded from the web site of the collection: <http://cdl.library.cornell.edu/moa/>.

was furtherly divided into three sub-sets in order to perform a three-folder cross validation that allowed us to find the optimal number of epochs required for MLP training. In order to reduce the size of the MLP input, a feature selection step was performed by removing from the feature vectors all the items that never appear in the training set. In this way we used a feature vector containing no more than 274 tree-patterns, instead of the 384 possible combinations of labels assigned to nodes. The classification results obtained with the basic features (by means of an MLP having 274 inputs 20 hidden nodes, and 5 outputs) are summarized in Table 1 B). A pattern is rejected when the difference between the highest MLP output and the next one is lower than 0.2 (the outputs are in the range [0,1]).

| Method | Number of features | Test pages | | |
|-----------------------------|--------------------|------------|-------|--------|
| | | Right | Wrong | Reject |
| A) Basic and size features | 278 | 281 | 17 | 7 |
| B) Basic features only | 274 | 281 | 19 | 5 |
| C) Without cuts along lines | 63 | 186 | 33 | 86 |
| D) Only contiguous siblings | 168 | 277 | 14 | 14 |

Table 1. Comparison of the results obtained with various approaches. See the text for more details.

As described in Section 3, in order to take into account the size of textual blocks, we added to the basic features four block size features. We selected the most appropriate threshold (discriminating among “small” and “big” blocks), by evaluating the results with different threshold values. The results of this analysis are summarized in Figure 2, and from this experiment we selected a threshold of 10 % as an optimal one (the corresponding confusion matrix is reported in Table 2). Analyzing Table 1 A) we can see that few errors are made when introducing the information about the area of blocks.

| True class | Output class | | | | | |
|------------|--------------|-------|-------|----------|---------|--------|
| | advert | first | index | receipts | regular | Reject |
| advert | 38 | 0 | 1 | 2 | 5 | 3 |
| first | 0 | 56 | 0 | 0 | 2 | 1 |
| index | 1 | 0 | 52 | 0 | 0 | 0 |
| receipts | 1 | 0 | 1 | 57 | 0 | 1 |
| regular | 0 | 2 | 0 | 2 | 78 | 2 |

Table 2. Confusion table of the test set, when adding the textual block size features considering a threshold of 10 %.

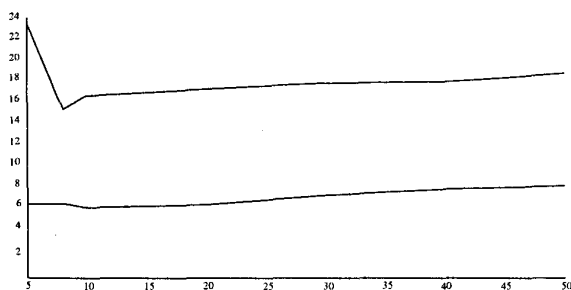


Figure 2. Classification error (%) with different threshold values, when using XY trees (top line), and MXY trees (lower line).

In order to evaluate the benefit of using cutting lines (MXY trees) in addition to the classical XY tree representation, we made another experiment, representing the document images in the data set with the classical XY tree decomposition algorithm. In order to obtain a uniform comparison, also in this case we extracted the tree starting from the same blocks located from the commercial OCR in the case of MXY tree building. A new test with a variable threshold value was considered for encoding XY trees. The results are compared with the standard one in Figure 2, and clearly show that better results are obtained when taking into account cuts along lines (see also Table 1 C)).

A further experiment was performed to evaluate the gain that can be achieved when considering tree-patterns of the MXY tree generated from non-adjacent siblings. In this experiment we generated feature vectors considering only adjacent siblings, and still taking into account the four textual block size features. The results are summarized in Table 1 D), and more trees are rejected with the other approaches.

5. Conclusions

We propose a method for classification of document images belonging to digital collections of monographs and journals. The method is based on a vectorial encoding of the MXY tree representing the document image. Each item in the feature vector describes the occurrences of some tree-patterns in the tree corresponding to the document. Some conclusions can be drawn after an extensive test on a database of more than 600 pages belonging to 5 classes. First, the use of cutting lines in addition to the classical cutting along spaces made in XY segmentation allows us to improve the classifier performance. Second, an encoding that takes into account non-contiguous siblings, and that uses information on the relative size of textual siblings, is appropriate. Future work is related to the use of less naive feature selection approaches, and on a massive experimen-

tation on larger data sets. Moreover, other classifiers will be taken into account in place of the MLP-based classifier considered in this paper.

We would like to thank Oya Y. Rieger from Cornell University for her unique help in collecting data taken into account for our experiments.

References

- [1] A. Amin, H. Alsadoun, and S. Fischer. Hand-printed arabic character recognition system using an artificial network. *Pattern Recognition*, 29(4):663–675, 1996.
- [2] R. Brugger, A. Zramdini, and R. Ingold. Modeling documents for structure recognition using generalized N-grams. In *Proc. of ICDAR '97*, pages 56–60, 1997.
- [3] F. Cesarini, M. Gori, S. Marinai, and G. Soda. Structured document segmentation and representation by the modified X-Y tree. In *Proc. of ICDAR '99*, pages 563–566, 1999.
- [4] L. Cordella, C. DeStefano, and M. Vento. A neural network classifier for OCR using structural descriptions. *Machine Vision and Applications*, 8(5):336–342, 1995.
- [5] A. Dengel and F. Dubiel. Clustering and classification of document structure - a machine learning approach. In *Proc. of ICDAR '95*, pages 587–591, 1995.
- [6] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Sheng, G. Soda, and A. Sperduti. Logo recognition by recursive neural networks. In *Proceedings of GREC97*, pages 144–151, 1997.
- [7] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 9(5):768–786, September 1998.
- [8] J. Hu, R. Kashi, and G. Wilfong. Document image layout comparison and classification. In *Proc. of ICDAR '99*, pages 285–288, 1999.
- [9] Y. Ishitani. Flexible and robust model matching based on association graph for form image understanding. *Pattern Analysis and Applications*, 3(2):104–119, 2000.
- [10] T. Kochi and T. Saitoh. User-defined template for identifying document type and extracting information from documents. In *Proc. of ICDAR '99*, pages 127–130, 1999.
- [11] J. Lin, C.-W. Lee, and Z. Chen. Identification of business forms using relationships between adjacency frames. *Machine Vision and Applications*, 9:56–64, 1996.
- [12] G. Nagy and S. Seth. Hierarchical representation of optically scanned documents. In *Proc. Int. Conf. Pattern Recognition*, pages 347–349, 1984.
- [13] C. Shin and D. Doermann. Classification of document page images based on visual similarity of layout structures. In *SPIE 2000*, pages 182–190, 2000.
- [14] S. Taylor, M. Lipshutz, and R. Nilson. Classification and functional decomposition of business documents. In *Proc. of ICDAR '95*, pages 563–566, 1995.
- [15] S. L. Taylor, R. Fritzson, and J. Pastor. Extraction of data from preprinted forms. *Machine Vision and Applications*, 5(5):211–222, 1992.
- [16] J. Yuan, Y. Y. Tang, and C. Y. Suen. Four directional adjacency graphs (fdag) and their application in locating fields in forms. In *Proc. of ICDAR '95*, pages 752–755, 1995.

