Clustering Document Images using a Bag of Symbols Representation

Eugen Barbu, Pierre Héroux, Sébastien Adam, Eric Trupin

HAL Id: hal-00601832

https://hal.archives-ouvertes.fr/hal-00601832

Submitted on 20 Jun 2011

Clustering document images using a bag of symbols representation

Eugen Barbu   Pierre Héroux   Sébastien Adam   Éric Trupin
Laboratoire PSI
CNRS FRE 2645 - Université de Rouen
UFR des Sciences et Techniques
76821 Mont-Saint-Aignan cedex - France
Eugen.Barbu@univ-rouen.fr

Abstract

Document image classification is an important step in document image analysis. Based on classification results we can tackle other tasks such as indexation, understanding or navigation in document collections. Using a document representation and an unsupervised classification method, we may group documents that from the user point of view constitute valid clusters. The semantic gap between a domain independent document representation and the user implicit representation can lead to unsatisfactory results.

In this paper we describe document images based on frequent occurring symbols. This document description is created in an unsupervised manner and can be related to the domain knowledge. Using data mining techniques applied to a graph based document representation we find frequent and maximal subgraphs. For each document image, we construct a bag containing the frequent subgraphs found in it. This bag of "symbols" represents the description of a document.

We present results obtained on a corpus of 60 graphical document images.

1. Introduction

A document image analysis (DIA) system transforms a document image into a description of the set of objects that constitutes the information on the document and which are in a format that can be further processed and interpreted by a computer [1]. Documents can be classified in mostly graphical or mostly textual documents [11]. The mostly textual documents also known as structured documents respect a certain layout and powerful relations exist between components. Examples of such documents are technical papers, simple text, newspapers, program, listing, forms,... Mostly graphical documents do not have strong layout restrictions but usually relations exist between different document parts.

Examples of this type of documents are maps, electronic schemas, architectural plans...

For these two categories of documents, graph based representations can be used to describe the image content (e.g. region adjacency graph [12] for graphical and Voronoi-based neighborhood graph [2] for textual document images).

In this paper we present an approach similar with the "bag of words" method from Information Retrieval (IR) field. We describe a document using a bag of symbols found automatically using graph mining [16] techniques. In other words, we consider the frequent subgraphs of a graph-based document representation as "symbols" and we investigate whether the description of a document as a bag of "symbols" can be profitably used in a clustering task.

The approach has the ability to process document images without knowledge of, or models for, document content. In the literature one can find papers dealing with clustering of textual documents using frequent items [6] and description of XML documents using frequent trees [15] but we do not know any similar approach in the DIA field.

The motivation for our study is the fact that unsupervised classification can represent the starting point for semi-supervised classification or indexation and retrieval from document collections. Also, the existing clustering solutions for document images are usually domain dependent and can not be used in an "incoming document flow" (fax, business mail...) setting, where supervised techniques are not at hand. The outline of this paper is as follows. In section 2 we present a graph representation and how we create this representation from a document image. Section 3 presents the graph-mining method used, in section 4 we describe how we create clusters based on dissimilarities between bags of symbols. Section 5 shows some experimental results. We conclude the paper and outline perspectives in section 6.

2. Document graph based representations

Eight levels of representation in document images are proposed in [4]. These levels are ordered in accordance with their aggregation relations. Data array level, primitive, lexical, primitive region, functional region, page, document, and corpus level are the representation levels proposed.

Without loosing generality, in the following paragraphs we focus our attention on a graph-based representation build from the primitive level. The primitive level contains objects such as connected components (set of adjacent pixels with the same color) and the relations between them.

Let I be an image and $C(I)$ the connected components from I, $c \in C(I)$ is described as $c = (id, P)$, where id is a unique identifier and P the set of pixels the component contains. Based on this set P, we can compute the center for the connected component bounding box and also we can associate a feature vector to it. Considering that, $c = (id, x, y, v), v \in R^n$. Subsequently using a clustering procedure on the feature vectors we can label the connected component and reach the description $C = (id, x, y, l)$ where l is a nominal label. The graph $G(I)$ representing the image is $G = G(V(I), E(I))$. Vertices $V(I)$ correspond to connected components and are labeled with component labels. An edge (u, w) between vertex u and vertex w exists $iff \sqrt{(u.x - w.x)^2 + (u.y - w.y)^2} < t$, where t is a threshold that depends on the image I global characteristics (size, number of connected components...).

The exact methodology employed to construct the graph representation is subsequently presented. From a binary (black and white) document image we extract connected components (black and white). The connected components will be the graph nodes. For each connected component we extract features. In the actual implementation the extracted characteristics are rotation and translation invariant features based on Zernike moments [9]. The invariants represent the magnitudes of a set of orthogonal complex moments of a normalized image.

The following step is to associate each connected component a label.

The two main categories of clustering methods are partitional and hierarchical. Partitional methods can deal with large sets of objects ("small" in this context means less than 300) but needs the expected number of clusters in input. Hierarchical methods can overcome the problem of number of clusters by using a stopping criterion [7] but are not applicable on large sets due to their time and memory consumption. In our case the number of connected components that are to be labeled can be larger than the limit of applicability for hierarchical clustering methods . In the same time we cannot use a partitional method because we do not know the expected number of clusters. Based on the hypothesis that a "small" sample can be informative for the geometry of data,

we obtain in a first step an estimation for the number of clusters in data. This estimation is made using an ascendant clustering algorithm with a stopping criterion. The number of clusters found in the sample is used as input for a partitional algorithm applied on all data. We tested this "number of cluster estimation" approach using a hierarchical ascendant clustering algorithm [7] that uses Euclidean distance to compute the dissimilarity matrix, complete-linkage to compute between-clusters distances, and Calinsky-Harabasz index [10] as a stopping criterion. The datasets (T_1, T_2, T_3) (see Table 1) are synthetically generated and contains well separated (not necessary convex) clusters.

T	$|T|$	number of clusters
T_1	24830	5
T_2	32882	15
T_3	37346	24

Table 1. Data sets description

Considering S the sample extracted at random from a test set, in Table 2 we present predicted cluster numbers obtained for different sample sizes. If the test set is T and $|S| = 50$, after repeating ten times the sampling procedure we obtain a set of estimations for the number of clusters. We can see that by using a majority voting decision rule we can find the good number of clusters in most of the cases and even when the sample size is very small (50 or 100) compared with the data set size.

We employed our sampling approach combined with the k-medoids clustering algorithm [8] on the connected components data set from images in our corpus (see section 5). The k-medoids clustering algorithm is a more robust version of the well known kmeans algorithm. The images from our corpus contains 6730 connected components. The proposed number of clusters using ten samples of size 600 is [16,14,17,16,16,19,7,17,15,16] and by considering the majority we use 16 clusters as input to the partitional clustering algorithm.

After labeling the connected components (nodes in the graph) subsequently we describe the way we add edges to the graph. The edges can be labeled or not (if unlabeled the significance is Boolean : we have or not a relation between two connected components) and can be relations of spatial proximity, based on "forces" [14], orientation or another criterion. In our actual implementation the distance between centers of connected components is used (see Fig. 1). If the distance between two connected components centers is smaller than a threshold, then an edge will link the two components (nodes).

$|S|$	50	100	300	500	600	700
T_1	$[6, 8, 7,$ $6, 5, 6,$ $6, 6, 5,$ $5]$ **6**	$[5, 7, 9,$ $7, 5, 5,$ $7, 5, 5,$ $7]$ **5**	$[7, 5, 7,$ $8, 7, 5,$ $5, 5, 7,$ $7]$ **7**	$[8, 7, 5,$ $5, 5, 5,$ $5, 5, 5,$ $5]$ **5**	$[5, 5, 5,$ $5, 5, 7,$ $7, 7, 7,$ $5]$ **5**	$[5, 5, 7,$ $5, 7, 5,$ $5, 7, 5,$ $5]$ **5**
T_2	$[9, 15, 15,$ $14, 13, 15,$ $13, 13, 14,$ $15]$ **15**	$[15, 15, 13,$ $15, 15, 15,$ $15, 15, 15,$ $15]$ **15**	$[15, 15, 15,$ $15, 15, 15,$ $15, 15, 15,$ $14]$ **15**	$[15, 15, 15,$ $15, 15, 15,$ $15, 15, 15,$ $15]$ **15**	$[15, 15, 15,$ $15, 15, 15,$ $15, 15, 15,$ $15]$ **15**	$[15, 15, 15,$ $15, 15, 15,$ $15, 15, 14,$ $15]$ **15**
T_3	$[11, 7, 9,$ $18, 7, 7,$ $6, 4, 14,$ $8]$ **7**	$[6, 14, 23,$ $21, 7, 17,$ $23, 16, 12,$ $11]$ **23**	$[22, 24, 23,$ $19, 23, 24,$ $21, 21, 24,$ $24]$ **24**	$[21, 25, 25,$ $24, 22, 24,$ $23, 24, 24,$ $24]$ **24**	$[20, 25, 21,$ $24, 19, 23,$ $24, 25, 24,$ $22]$ **24**	$[23, 20, 21,$ $20, 25, 24,$ $24, 21, 25,$ $24]$ **24**

Table 2. Proposed cluster numbers

Figure 1. An image and its associated graph

3. Graph Mining

"The main objective of graph mining is to provide new principles and efficient algorithms to mine topological substructures embedded in graph data" [16].

Mining frequent patterns in graphs is the problem of finding in a set of graphs those subgraphs that occur more times than a threshold (minimum support). Because the number of patterns can be exponential this problem complexity can also be exponential. An approach to solve this problem is to start with finding all frequent patterns with one element, then all patterns with two elements, etc. in a level-by-level setting . In order to reduce the complexity different constraints are used: the minimum support, the subgraphs are connected, and not overlapped.

An important concept is that of maximal subgraph. A graph is said to be maximal if it does not have a frequent super-graph.

In our document image analysis context we are interested in finding maximal frequent subgraphs because we want to find symbols but to ignore their parts.

A system that is used to find frequent patterns in graphs is FSG (Frequent Subgraph Discovery) that "finds patterns corresponding to connected undirected subgraphs in an undirected graph database" [13].

The input for the FSG program is a list of graphs and a minimum support threshold. Each graph represents a transaction. We present subsequently how we construct the transaction list starting from a set of document images. Using the procedure presented in section 2 we create for each document an undirected labeled graph. Every connected component of this graph represents a transaction. Using FSG we extract the frequent subgraphs and we construct a bag of frequent subgraphs occurring in each document.

Figure 2. Occurences of a frequent subgraph in an image

In the following paragraphs we consider that the frequency condition is sufficient for a group of connected components to form a symbol and we will conventionally make an equivalence between the frequent subgraphs found and symbols. As we can see in the example (Fig. 2) the proposed symbols are far from being perfect due to the image noise, connected components clustering procedure imperfections,. . . however we can remark the correlation between this artificial symbol and the domain symbols.

4. Dissimilarity between document descriptions

A collection of documents is represented by a symbol-by-document matrix A, where each entry represents the occurrences of a symbol in a document image, $A = (a_{ik})$ where a_{ik} is the weight of symbol i in document k. Let f_{ik} be the number of occurrences of symbol i in document k, N the number of documents in the collection, and n_i the total number of times symbol i occurs in the whole collection. In this setting conform with [5] one of the most effective weighting scheme is entropy-weighting. The weight for symbol i in document k is given by :

$$a_{ik} = \log\left(1 + f_{ik}\right).\left(1 + \frac{1}{\log N}\sum_{j=1}^{n}\frac{f_{ij}}{n_i}\log\frac{f_{ij}}{n_i}\right)$$

Now, considering two documents A, B with the associated weights $A = (a_1, a_2, \ldots, a_t)$, $B = (b_1, b_2, \ldots, b_t)$ where t is the total number of symbols, then

$$d(A, B) = 1 - \frac{\sum_{i=1}^{t} a_i.b_i}{\sqrt{\sum_{i=1}^{t} a_i^2 . \sum_{i=1}^{t} b_i^2}}$$

represents a dissimilarity measure based on the cosine correlation.

5. Experiments

A comparison between results obtained using the proposed document representation and three other representations is made in the following paragraphs. On a corpus of graphical document images we have extracted different sets of features. Each document image is described with one of the following types of features : Zernike moments for the whole image (a vector with 16 components), pixel densities (the feature vector considered is composed of the 85 (1+4+16+64) gray levels of a 4-level-resolution pyramid [3], see Fig 3), connected components label list, and symbol label list.
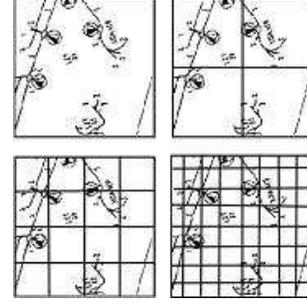

Figure 3. Four level resolution pyramid

Using a hierarchical ascendant clustering procedure on the dissimilarities between document representations (as Zernike moments, pixels densities,. . .) combined with Calinsky-Harabasz stopping criterion we obtain four partitions that were compared with the groundtruth partition of the corpus.

In order to evaluate the partitions proposed by the clustering algorithm, we employ the overall F-measure index. Let D represent the set of documents and let $C = \{C_1, \ldots, C_k\}$ be a clustering of D. Also let $C' = \{C'_1, \ldots, C'_l\}$ the reference (ground truth) classification. Then the recall of cluster j with respect to class i is

$$rec(i, j) = \frac{|C_j \cap C'_i|}{C'_i}$$

the precision

$$prec(i, j) = \frac{|C_j \cap C'_i|}{C_j}$$

and

$$F_{ij} = \frac{2.prec(i, j).rec(i, j)}{prec(i, j) + rec(i, j)}$$

represents the F-Measure. The overall F-Measure of a clustering is

$$F = \sum_{i=1}^{l} \frac{|C'_i|}{|D|} \max_{j=1\ldots k} F_{ij}$$

F-measure is 1.0 if the matching between the two partitions (ground truth and the one proposed by the clustering algorithm) is perfect.

Our corpus contains 30 images from the class of a French telephony operator (FT) maps, 25 electronic schemas, and 5 architectural plans. These images are scanned images that contain real and artificial noise.

We can see that the connected component list approach obtains good results compared with the simple approaches
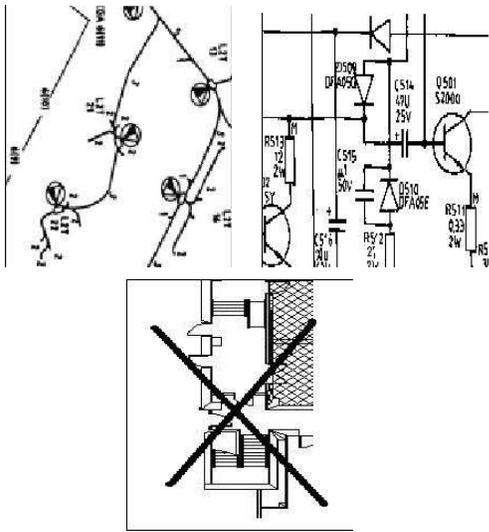
Figure 4. Images from the corpus

	ZM		Densities		Connected Component list			Symbol list		
F-Measure	0.58		0.69		0.89			0.90		
Confusion Matrix	1	29	30	0	26	4	0	26	4	0
	0	25	25	0	2	1	22	0	3	22
	0	5	0	5	0	5	0	0	5	0

Table 3. Experimental results

(Zernike moments and densities). In the same time the symbol list approach representation is more compact than the connected component list and also obtains better results.

6. Conclusions

The research undertaken represents a novel approach for clustering document images. The approach uses data mining tools for knowledge extraction. It automatically finds frequent symbols. These frequent patterns are part of the document model and can be put in relation with the domain knowledge. The exposed method can be applied to other graph representations of a document. In the near future, we will apply this approach to layout structures of textual document images. Another follow up activity is to quantify the

	Cluster 1	Cluster 2
FT maps	1	29
Electronic Schemas	0	25
Architectural drawings	0	5

Table 4. Confusion matrix details

way noise can affect the connected components labeling, and the manner in which an incorrect number of clusters can affect the graph mining procedure. Based on this error propagation study we can ameliorate our method. Other possible improvements can be obtained if we would employ a graph-based technique that can deal with error tolerant graph matching.

References

[1] A. Antonacopoulos. *Introduction to Document Image Analysis*. 1996.

[2] A. D. Bagdanov and M. Worring. Fine-grained document genre classification using first order random graphs. In *Proc. of the sixth International Conference on Document Analysis and Recognition*, pages 79–83, 2001.

[3] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.

[4] D. Blostein, R. Zanibbi, G. Nagy, and R. Harrap. Document representations. In *Proc. of the IAPR Workshop on Graphic Recognition*, 2003.

[5] S. T. Dumais. Improving the retrieval information from external ressources, behaviour research methods. *Instrument and Computers*, 23(2):229–236, 1991.

[6] B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent items. In *Proc. of the SIAM Conference on Data Maining*, 2003.

[7] A. D. Gordon. *Classification*. Chapman & Hall, 2nd edition edition, 1999.

[8] L. Kaufmann and P. J. Rousseeuw. *Statistical Data Analysis based on the L1 Norm and Related Methods*, chapter Clustering by Means of Medoids, pages 405–416. Elsevier Science, 1987.

[9] A. Khotazad and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Trans. on Pattern Recogntion and Machine Analysis*, 12(5), 1990.

[10] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 58(2):159–179, 1985.

[11] G. Nagy. Twenty years of document analysis in pami. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1):38–62, 2000.

[12] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, 1982.

[13] M. Seno, M. Kuramochi, and G. Karypis. Pafi, a pattern finding toolkit. http://www.cs.umn.edu/ karypis/, 2003.

[14] S. Tabbone, L. Wendling, and K. Tombre. Matching of graphical symbols in line-drawing images using angular signature information. *International Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.

[15] A. Termier, M. Rousset, and M. Sebag. Mining xml data with frequent trees. In *Proc. of DBFusion Workshop*, pages 87–96, 2002.

[16] T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsletter*, 5(1):59–68, 2003.