CLASS ASSOCIATION RULES WITH OCCURRENCE COUNT IN IMAGE CLASSIFICATION

ŁUKASZ KOBYLIŃSKI AND KRZYSZTOF WALCZAK

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
K.Walczak@ii.pw.edu.pl

(Received 30 December 2006; revised manuscript received 31 January 2007)

Abstract: The concept of utilizing association rules for classification has emerged in recent years. This approach has often proved to be more efficient and accurate than traditional techniques. In this paper we extend the existing associative classifier building algorithms and apply them to the problem of image classification. We describe a set of photographs with features calculated on the basis of their color and texture characteristics and experiment with different types of rules which use the information about the existence of a particular feature in an image, its occurrence count and spatial proximity to classify the images accurately. We suggest using association rules more closely tied to the nature of the image data and compare the results with those of classification with simpler rules, taking into consideration only the existence of a particular feature on an image.

Keywords: image, classification, class association rules, associative classifiers

1. Introduction

The large volume of multimedia data collected every day, from satellite and aerial imagery for land planning, agriculture or forestry to that recorded any owners of a digital camera, highlights the problem of automatically extracting meaningful information from such collections of raw image data. Classification is an important part of any knowledge-retrieval system, particularly significant in applications where images are the main source of information in the decision making process.

In this paper, we propose an application of extended Class Association Rules (CARs) to image classification. The approach is suitable for analyzing large sets of photographs, as it has been developed from data-mining techniques dedicated to such databases. As we have shown earlier by experimenting with a similar method [1], it offers particularly good results in classification of images obtained with remote-sensing methods, but may also be used for analysis of any sets of photographs. We have extended the concept of association rules with recurrent items for classification with information specific to image analysis, such as the number of occurrences of a particular feature in an image or the maximum size of a region with uniform feature characteristics. A modified version of the classifier building algorithm (CBA) is used to mine such rules from a training set of images, described by their color and texture features. The rules have the form of an implication between a limited number of features with appropriate weights and a category label. They are pruned and used to create a classifier suitable for efficient classification of unseen examples.

The employed method of classification is a two-stage process in which the classifier is built on the basis of a training set and used to associate category labels with previously unseen examples of images. First, a symbolic representation of images is created to enable the use of data mining methods, by calculating their color and texture features and clustering them into a structure of a dictionary of representative values. A classifier is created on the basis of a reduced set of discovered rules. New photographs are processed in exactly the same way as the training ones, but without the dictionary building and rule mining steps. The existing dictionary is used to label particular blocks of images with identifiers of the dictionary entries, while rules from the classifier are applied to classify the photographs into categories.

The remainder of the paper is organized as follows: Section 2 presents previous work related to the subject of CARs and association rule mining in image databases. In Section 3 we give the details of our approach to image representation, later used in the classification process. In Section 4 we describe the concept of associative classification and propose extended association rule mining and classifier building algorithms. Experimental results of image classification are presented in Section 5. Our conclusion and discussion of possible future improvements is given in Section 6.

2. Previous work

While the concept of mining association rules for classification was first proposed in [2], the first classifier building algorithm (CBA) was introduced in [3], followed by CMAR [4] and ARC [5]. The idea of including recurrent items in association rules was presented as a modification of the Apriori algorithm in [6] and the FP-growth algorithm in [7]. Finally, the possibility of incorporating recurrent items into CARs was presented in [8] by a modification of the ARC-BC algorithm.

Recent applications of data mining to image databases have considered the classification of mammograms [9], mining association rules between regions of paintings [10] or features of aerial images, including their spatial relationships [11].

An image representation approach similar to the method presented here was proposed in [12], where the authors compared various representational models extracting image features from individual blocks.

We have shown in [1] that achieving good classification results of aerial photographs is possible even with simple association rules, having a class label in the consequent. Here, we elaborate further on this subject and compare the results with a classifier more closely related to the problem of image classification.

3. Image representation

A preliminary step of creating a symbolic representation of the source images is required before applying any data mining methods to the database. The images are first normalized by bringing them to a common resolution and performing histogram equalization. This is performed only when necessary (*e.g.* when the available dataset contains images from various sources), using the bicubic resampling method for scaling and histogram equalization for each of the RGB color components. Then, the images are divided into a grid of 32×32 pixels blocks and color and texture features of each of the blocks are calculated to be used in further processing. This initial procedure may be performed well in advance of the actual classification process, for example while adding a new photograph to the database.

An additional step of creating a dictionary of typical feature values is necessary before training a new classifier. This is performed by clustering the values to find a chosen number of group centroids, which then become the elements of a dictionary. Individual image blocks are then labeled with identifiers of the most similar entries present in the dictionary. An image's representation consists of a list of all identifiers associated with its blocks.

3.1. Calculating color features

Color features are represented by a histogram calculated in the HSV color space, with the H channel quantized to 18 values, and S and V channels – to 3 values each. In effect, the representation assumes the form of a 162-element vector of real values between 0 and 1. Histogram intersection measure is used to compare two feature vectors, h and g, given below in the form of a distance measure:

$$d_I(h, g) = 1 - \sum_{i=0}^{N-1} \min(h[i], g[i]). \tag{1}$$

3.2. Calculating texture features

The statistical approach presented in [13], utilizing Gabor filtering, is used to represent important information about the texture visible on the photographs. The feature vector consists of mean and standard deviation values calculated from images resulting from filtering the original pixels with a bank of Gabor functions, which are a product of a Gaussian and a sine function. These filters are scaled and rotated versions of the base function, given by the following formula:

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right], \tag{2}$$

where σ_x and σ_y are parameters of the Gaussian, while W denotes the frequency of the sinusoidal component.

Six different orientations and four scales of the base function are used to filter every photograph. Thus, images resulting from consecutive passes assume the form of:

$$W_{mn}(x, y) = |I(x, y) * g_{mn}(x, y)|, \tag{3}$$

where $*$ denotes spatial convolution, m – filter orientation and n – scale. The final feature vector consisting of mean, μ, and standard deviation, σ, values assumes the form of:

$$\overline{f} = [\mu_{00}\, \sigma_{00}\, \cdots\, \mu_{M-1\,N-1}\, \sigma_{M-1\,N-1}]. \tag{4}$$

Comparison of two feature vectors, $\overline{f}^{(i)}$ and $\overline{f}^{(j)}$, is accomplished by the distance measure given below:

$$d(i, j) = \sum_m \sum_n d_{mn}(i,j), \tag{5}$$

$$d_{mn}(i,j)=\left|\frac{\mu_{mn}^{(i)}-\mu_{mn}^{(j)}}{\alpha(\mu_{mn})}\right|+\left|\frac{\sigma_{mn}^{(i)}-\sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})}\right|, \tag{6}$$

where $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the values' standard deviations over the entire database.

3.3. Further image processing

The proposed method of representing images as the color and texture feature values of their tiles enables further image processing that relies only on that representation and does not involve analysis of the originals. One such technique is simple segmentation roughly detecting the images' main regions and reducing the number of processed tiles to a much smaller number of segments. A simplified version of the EdgeFlow algorithm, presented in [14], makes use of the texture information to iteratively approximate boundaries between segments of an image.

The EdgeFlow algorithm

First, the differences in texture feature values are calculated between every block, s, and its eight neighbors:

$$\begin{aligned} E(s,0)&=d(\overline{f}(x+1,y),\overline{f}(x,y)),\\ E(s,\pi/4)&=d(\overline{f}(x+1,y+1),\overline{f}(x,y)),\\ &\vdots\\ E(s,7\pi/4)&=d(\overline{f}(x+1,y-1),\overline{f}(x,y)). \end{aligned} \tag{7}$$

Next, probabilities of reaching a segment boundary in every direction, θ, are calculated:

$$P(s,\theta)=\frac{E(s,\theta)}{E(s,\theta)+E(s,\theta+\pi)}. \tag{8}$$

The information is then used to determine the most probable direction of finding a segment boundary from every image block. The angle, Θ, maximizing the sum of probabilities in a continuous range of four directions is found. The next step is to calculate the "edge-flow" vector, having the combined energy and direction of texture feature value differences in the most probable half-circle:

$$\boldsymbol{F}(s)=\sum_{\Theta(s)\le\theta<\Theta(s)+\pi} E(s,\theta)\cdot\exp(j\theta). \tag{9}$$

The vectors are then iteratively propagated onto their neighbors if the angle between them is acute, In which case the source vector is added to the destination and removed from the original location. Otherwise, it remains in its original location. When the propagation has stopped and there are no similarly oriented neighboring vectors, the boundaries between segments can be detected by locating vectors that point at each other (neighboring vectors of opposite directions).

The detected boundaries are closed by a simple algorithm connecting each open contour with its nearest neighbor located in an approximate contour's direction. The number of resulting segments is then reduced to a user-specified one by joining the most similar regions (having the closest texture feature values).

The final operation is to calculate the color and texture features of the newly created segments, which now contain many original tiles. This may be easily achieved by calculating the mean value of color features of the tiles constituting a particular segment. In the case of texture, the resulting vector, as given by Equation (4), consists of values calculated on the basis of N building tiles with the following formulas:

$$\mu_{mn} = \frac{1}{N}\sum_{k=1}^{N}\mu_{mn}^{(k)}, \tag{10}$$

$$\sigma_{mn} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left((\mu_{mn}^{(k)})^2 + (\sigma_{mn}^{(k)})^2\right) - \mu_{mn}^2}. \tag{11}$$

3.4. Creating the feature dictionary

The dictionary consists of the most typical color and texture features of individual blocks of photographs in the training set. It is created by clustering corresponding feature values into a chosen number of groups. The clustering is performed using a k-Means algorithm with a histogram intersection measure for comparing color features and a Gabor feature distance for comparing texture features.

Centroids resulting from the clustering operation become elements of the dictionary and are labeled with consecutive natural numbers. These identifiers are then used to describe blocks of images in the database. During the classification phase, the previously created dictionary is queried with color and texture feature values and responds with labels of the most similar entries. An example of images converted to their symbolic representation by a dictionary lookup is shown in Figure 1.

image	features
1	B_8, T_{14}
2	B_1, B_{12}, T_{14}
3	$B_{12}, B_{13}, T_{11}, T_{16}$
4	$B_3, B_6, B_9, B_{13}, T_{14}$
5	B_6, B_{14}, T_{16}
6	B_{13}, T_2, T_{15}
7	B_{12}, B_{13}, T_{11}
8	B_{13}, T_{11}
⋮	
m	$B_7, B_9, B_{12}, T_{13}, T_{14}$

Figure 1. Converting a set of images to their symbolic representation. From left to right: the dictionary content (n values of color and texture features after clustering), a set of original images, and their symbolic representation

4. Associative classification

Associative classifiers are a recent, two-stage approach to classification, in which a set of association rules between attribute values and category labels is first discovered and then a compact classifier is created by selecting the most important rules for classification.

4.1. Association rules for classification

Formally, an association rule used for classification is an implication of the form of $X \rightarrow c$, where item set X is a non-empty subset of all possible items in the database, $X \subseteq I$, $I = \{i_1, i_2, \ldots, i_n\}$, and c is a class identifier, $c \in \{c_1, c_2, \ldots, c_n\}$. Let thus a rule item set be a pair $< X, c >$, containing item set X and class label c. The rules are discovered in a training set of transactions D_t. Each transaction is a triple of the $< tid, Y, c >$ form, containing a transaction identifier, tid, itemset $Y \subseteq I$ and a class label, c.

In our approach we discover the most interesting association rules between images of the training set, described by dictionary entries, and their category labels. This is a slight modification of the classic association rule mining problem, as the consequent implication is always limited to a class label. The aim of mining is then to discover the rules constituting a subset of general association rules and having the following form:

$$R_c : \{color_1, \ldots, color_n, texture_1, \ldots, texture_m\} \Rightarrow class\ \ label \tag{12}$$

We have adapted the existing methods of association rule mining to create a classifier suitable for categorization of image data. Direct application of any rule mining algorithm to a transactional database containing images represented by feature values in their particular locations would result in a large number of irrelevant associations. Therefore, we consider only the existence, occurrence count and spatial proximity of features in order to create rules that are sufficiently general to classify previously unseen examples.

The initial set of discovered rules is usually very large, so it is necessary to limit the number of associations by specifying the minimum support and confidence values and employing various pruning techniques. We have used the CBA approach proposed in [3] to mine the rules along with frequent item sets and then apply a pruning strategy to limit their number.

4.2. Considering occurrence count

Extending association rules to include information about item occurrence count in multimedia applications was first proposed in [6]. We use this general idea to mine classification rules with recurrent items and apply a selected number of such associations to the problem of image classification. A slight modification in calculating the support of such rules is necessary, as a single transaction may increase the support of an item set by more than one. The support of an item set, X, may thus be calculated as follows (see [7]):

$$\mathrm{supp}(X) = \sum_{k=0}^{|D|} \frac{\phi(X, t_k)}{|D|}, \tag{13}$$

where ϕ is a function returning the ratio at which transaction t_k of database D supports item set X, defined as follows:

$$\phi(X, t) = \min\left(\frac{\alpha_j}{\beta_j}\right), \quad j = 1 \ldots n, \tag{14}$$

$t_k = \{\alpha_1 i_1, \alpha_2 i_2, \ldots, \alpha_n i_n\}$, $X = \{\beta_1 i_1, \beta_2 i_2, \ldots, \beta_n i_n\}$, $\alpha_i \neq 0$, $\beta_i \neq 0$.

The support of a rule with recurrent items is calculated similarly as when considering simple association rules, by counting the support of a set consisting of the rule's antecedent and consequent. The definition of confidence also remains unchanged and may be calculated as $\mathrm{supp}(X \cup Y)/\mathrm{supp}(X)$. The definition of a frequent item set may be extended by including an additional condition of maximum support, $\mathrm{supp}(X) < \Sigma$, apart from its minimum value, $\mathrm{supp}(X) > \sigma$, which helps minimize the number of uninteresting rules.

A modified version of the CBA algorithm, presented as Algorithm 1 below, is used to mine either all possible rules or only those having a certain maximum number of items in the antecedent.

Algorithm 1. CBA-RG with recurrent items

Input D_t (training set), σ (min. support), Σ (max support), δ (min. confidence)
Output CAR (class association rules with recurrent items)
1: $F_1 \leftarrow \{\text{frequent 1 rule itemsets}\}$
2: $M \leftarrow \{\text{maximum occurrence of frequent 1 itemsets in } D_t\}$
3: $CAR_1 \leftarrow \{f \in F_1 \mid \mathrm{supp}_{\mathrm{rule}}(f) < \Sigma \wedge \mathrm{conf}(f) > \delta\}$
4: $k \leftarrow 1$
5: **while** $F_k \neq \emptyset$ **do**
6: $\quad C_{k+1} \leftarrow (F_k \otimes F_k) \cup \{f \in F_k \oplus x \in F_1 \mid \mathrm{count}(x, f) < M[x]\}$
7: $\quad$ **for all** $t \in D$ **do**
8: $\qquad$ **for all** $c \in C_{k+1}$ **do**
9: $\qquad\quad \mathrm{supp}_X(c) = \mathrm{supp}_X(c) + \phi(c,t)$
10: $\qquad\quad \mathrm{supp}_{\mathrm{rule}}(c) = \mathrm{supp}_{\mathrm{rule}}(c) + \phi(c, t) \mid \mathrm{class}(c) = \mathrm{class}(t)$
11: $\qquad$ **end for**
12: $\quad$ **end for**
13: $\quad F_{k+1} \leftarrow \{c \in C_{k+1} \mid \mathrm{supp}_{\mathrm{rule}}(c) > \sigma\}$
14: $\quad CAR_{k+1} \leftarrow \{f \in F_{k+1} \mid \mathrm{supp}_{\mathrm{rule}}(f) < \Sigma \wedge \mathrm{conf}(f) > \delta\}$
15: **end while**
16: **return** $\bigcup_k CAR_k$

In lines 1−3 of Algorithm 1, a first pass over the database is made to find all sufficiently frequent item sets which can be used to build rules with a single value in the antecedent. The maximum number of occurrences of every item in the database's transactions is also counted, as per the MaxOccur algorithm [6], in order to limit the number of item recurrences while generating candidates. Line 6 generates candidates using the Apriori method and includes another occurrence of an existing item as long as the current count remains below the maximum value. The *count* function returns the item's current number of occurrences in an item set, while the $\otimes$ and $\oplus$ symbols respectively denote item-set merging and item concatenation operations. Lines 7−12 are used to independently calculate the support of each rule and the support of its antecedent. These values are then used to calculate the rules' confidence in line 14.

4.3. Considering spatial proximity

Apart from the association rules between the number of particular features present in the images and their category labels, we also consider rules that include information about spatial proximity of features. While mining for the association rules, we check for spatial relationships between recurring features and include them in the rules multiple times only when they constitute a single area of an image. This approach may be used only when no segmentation is applied to the original tiles, as it reduces all similarly textured blocks to a single region.

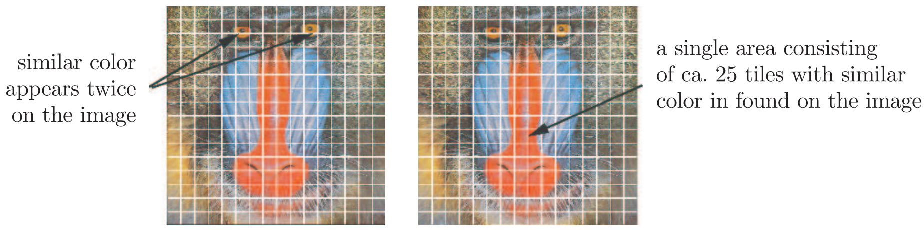
It is possible to mine such rules without any change to the above-mentioned algorithm by slightly changing the images' representation. Each transaction is scanned for every element of the dictionary to find the largest area covered by a single feature. The original number of occurrences of every item is then reduced to that maximum value before the association rule mining algorithm is applied. Table 1 illustrates the difference between the two approaches to image representation and Figure 2 demonstrates the difference in classification rules.

Table 1. An example of image representation when considering spatial proximity of features

B_1, T_1	B_1, T_2	B_2, T_1	B_1, T_2
B_2, T_1	B_1, T_2	B_2, T_2	B_1, T_1
B_1, T_2	B_2, T_1	B_1, T_1	B_1, T_2
B_2, T_2	B_2, T_1	B_1, T_1	B_2, T_2

$\rightarrow$

Direct representation:
$9B_1$, $7B_2$, $8T_1$, $8T_2$
Considering spatial proximity of features:
$5B_1$, $3B_2$, $4T_1$, $3T_2$



$R_i: 2 \cdot color_{beige}, \ldots \Rightarrow class_{chimp}$ $\quad$ $R_i: 25 \cdot color^{area}_{red}, \ldots \Rightarrow class_{chimp}$

Figure 2. Difference in rules when considering the occurence count only and when including spatial proximity

4.4. Building the classifier

Having found all the rules with minimum and maximum support, as well as the minimum confidence, we face the problem of creating a classifier to be used when associating category labels with previously unseen images. The final classifier is created by first sorting the rules in the descending order of their confidence and support and in ascending order of the number of items in their antecedents. Next, for every rule in the sorted list, all elements of the training set matching that rule are found and removed from further processing. A rule is then added to the classifier if it matches at least one element of the set. A default class of the classifier is selected at each step of the iteration that minimizes the error of classification of the remaining data. Finally, when the rule or data set is empty, the final classifier is reduced to the first number of rules reducing the general error rate of classification.

4.5. Classification

Classification is performed by applying the first matching rule from the classifier to a given image described by dictionary entries. A default class label is given to images for which there are no matching rules. An image matches a rule when it contains each of the items of the rule's antecedent with at least equal occurrence count.

An exemplary representation of a few photographs without considering spatial relationships between features, possible classifier content and classification results are shown in Table 2. The classifier was created using the CBA approach to limit the number of rules. Dictionary entries were identified by B_i (color) and T_i (texture) labels. The dictionary size in this example was 8 entries each for color and texture. The first two images were respectively matched by the first and second rule of the classifier and associated with the C_1 category label. The other two images were classified using the *default_class* value, as they remained unmatched by any rule.

Table 2. An example of image representation, classifier content and classification results

I	image features
I_1	$7B_2, 38B_3, 51B_4, 88T_1, 7T_2, 1T_3$
I_2	$2B_1, 3B_2, 23B_3, 68B_4, 65T_2, 15T_3, 6T_4$
I_3	$23B_1, 72B_2, 1B_3, 4T_1, 57T_2, 21T_3, 14T_4$
I_4	$48B_1, 14B_2, 34B_3, 1T_1, 60T_2, 24T_3, 11T_4$

rules
$2B_3, 1B_4, 1T_1 \Rightarrow C_1$
$1B_2, 3B_3, 1B_4 \Rightarrow C_1$
$default\_class = C_0$

I	class
I_1	C_1
I_2	C_1
I_3	C_0
I_4	C_0

5. Experimental results

We have verified the results of image classification of the proposed method with a test dataset made available by the authors of the SIMPLIcity CBIR system [15]. We chose 400 photographs, having a resolution of 384×256 pixels and associated with four different category labels, namely *buses*, *flowers*, *horses* and *mountains*. The accuracy of the approach described above and the method of classification with simple class association rules proposed earlier in [1] was compared by performing classification of the same set of images belonging to two different categories. We have used ten-fold cross-validation to reduce any influence due to the selection of training and test images from the available dataset.

The results of these experiments are presented in Table 3. Classification accuracy of both methods is shown for each dictionary size k (the number of different color and different texture entries). In the first experiment, referred to Exp. 1, we considered classification between *horse* and *flower* sets of photographs, in the second – between *bus* and *mountain*, and in the third – between *bus* and *horse*. The rules were mined with the minimum support of 0.01, the minimum confidence of 0.50, the maximum support of 1.00, and the antecedent length limited to 5 items.

The influence of including the segmentation step in the process is shown in Figure 3, while the relationship between the dictionary size, the chosen method of classification and the number of rules found and included in the classifier is presented in Figure 4. Clearly, including recurrent items when discovering the rules significantly increases their number. Considering spatial proximity of features helps to reduce the number of found associations, as well as the number of rules in the classifier. The segmentation step does not increase the method's accuracy but reduces the number of analyzed tiles and, consequently, the processing time.

The presented results are proof that extending class association rules to include recurrence of items and information about spatial proximity of features may improve the accuracy of classifying photographs. Most of the results obtained using the newly proposed method are more accurate than those presented earlier. As other authors proved that associative classifiers are better than C4.5 or other similar methods (in experiments conducted on a set of 34 benchmark problems from the UCI machine learning repository), we have not performed a comparison of classifiers ourselves.

While association rules with recurrent objects may be thought of as generalizations of simple rules with binary information about item existence, there remains the problem of selecting the most effective ones for classification. This is why not every experiment turned out better results when considering extended rules. There are cases for which the applied method of rule selection produces better results with simple class association rules. Considering spatial proximity of features present in an image does not seem to improve the accuracy of classification considerably, but helps limit the number of discovered associations.

Table 3. Classification accuracy of the four test datasets

k	simple rules			with recurrence			with spatial proximity		
	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3
4	93.02	87.73	93.26	94.12	91.74	94.24	93.35	89.86	93.86
8	90,70	94.48	99.44	93.48	93.27	98.96	91.32	94.31	99.17
12	96,51	95.70	96.07	97.53	96.48	97.45	95.85	96.35	96.24
16	95,93	95.70	97.75	95.26	95.19	96.89	95.32	96.18	97.89
20	94.77	93.25	98.32	94.34	94.24	98.43	94.80	94.11	98.39

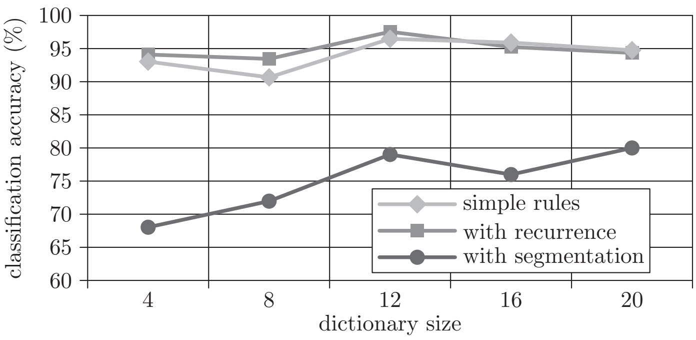


Figure 3. The influence of segmentation step on classification accuracy; the light gray line corresponds to the classifier utilizing simple rules, the medium gray – to the classifier with recurrent items, the dark gray – to the segmentation step included in classification with such rules

6. Conclusions

We have proposed an extension of associative classifiers with recurrent items and experimented with applications of association rules in classification of photographs. We have used class association rules with recurrent items and considered the spatial proximity of features of an image to accurately classify a set of photographs. We have applied this method to a dataset containing photographs associated with four different categories and presented results of their classification. The described approach has proved to perform better than the previously tested classifier utilizing only simple rules with no item occurrence information. Associative classification of

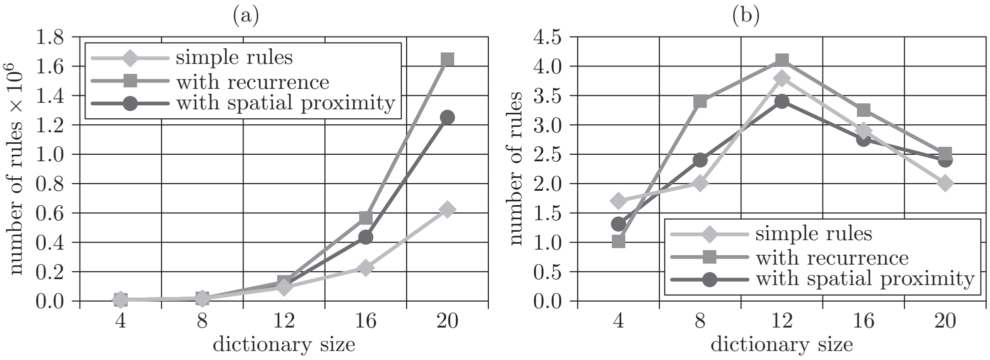


Figure 4. Average number of rules with respect to dictionary size; the light gray line corresponds to the classifier utilizing simple rules, the medium gray – to the classifier with recurrent items, the dark gray – to the classifier considering spatial proximity of features;
(a) the number of discovered rules with respect to dictionary size;
(b) the number of rules in the classifier with respect to dictionary size

images is a promising area of research, as many different approaches to image representation and association rule mining and pruning may be proposed to improve the process' accuracy.

Acknowledgements

The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

References

[1] Kobyliński Ł and Walczak K 2006 *Proc. Int. IIS: Intelligent Information Processing and Web Mining Conf. Advances in Soft Computing*, Ustron, Poland, pp. 479–487

[2] Bayardo R 1997 *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining*, Newport Beach, CA, USA, pp. 123–126

[3] Liu B, Hsu W and Ma Y 1998 *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*, New York, USA, pp. 80–86

[4] Li W, Han J and Pei J 2001 *IEEE Int. Conf. on Data Mining*, Washington, D.C., USA, pp. 369–376

[5] Zaïane O R and Antonie M-L 2002 *Proc. 13th Australasian Database Conference*, Melbourne, Australia, pp. 215–222

[6] Zaïane O R, Han J and Zhu H 2000 *Proc. 16th Int. Conf. on Data Eng.*, San Diego, USA, pp. 461–470

[7] Ong K-L, Ng W-K and Lim E-P 2001 *Proc. 3rd Int. Conf. on Information, Communications & Signal Processing (ICICS'2001)*, Singapore, 5 pages

[8] Rak R, Stach W, Zaïane and Antonie M-L 2005 *Proc. 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, Hanoi, Vietnam, pp. 240–248

[9] Antonie M-L, Zaïane O R and Coman A 2002 *Revised Papers from MDM/KDD and PAKDD/KDMCD*, pp. 68–83

[10] Ardizzone E, Daurel T, Maniscalco U and Rigotti C 2001 *Proc. 1st Int. Workshop on Multimedia Data and Document Eng.*, Lyon, France, 6 pages

[11] Tešić J, Newsam S and Manjunath B S 2003 *Proc. SIAM Int. Conf. on Data Mining, 6th Workshop on Mining Scientific and Engineering Datasets*, San Francisco, USA, pp. 71–77

[12] Zhu L, Rao A and Zhang A 2002 *ACM Trans. Inf. Syst.* **20** 224

[13] Manjunath B S and Ma W 1996 *IEEE Trans. on Patt. Anal. and Machine Intell.* **18** 837

[14] Ma W Y and Manjunath B S 2000 *IEEE Trans. on Image Processing* **9** 1375

[15] Wang J Z, Li J and Wiederhold G 2001 *IEEE Trans. on Patt. Anal. and Machine Intell.* **23** 947