Jumping Emerging Substrings in Image Classification

Łukasz Kobyliński and Krzysztof Walczak

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

Abstract. We propose a new image classification scheme based on the idea of mining jumping emerging substrings between classes of images represented by visual features. Jumping emerging substrings (JES) are string patterns, which occur frequently in one set of string data and are absent in another. By representing images in symbolic manner, according to their color and texture characteristics, we enable mining of JESs in sets of visual data and use mined patterns to create efficient and accurate classifiers. In this paper we describe our approach to image representation and provide experimental results of JES-based classification of well-known image datasets.

1 Introduction

Knowledge Discovery in Databases is a process concerning a broad range of types of data that needs to be processed every day. Originally, quantitative and textual data was in the center of interest for developing efficient and effective methods of finding interesting relationships. Today, analysis and understanding of enormous amounts of collected multimedia data seem to be the most pressing problem in the field of KDD.

As many methods for processing non-multimedia data have already been proposed, it is interesting to see how well they perform in the domain of visual data. Mining in such databases requires additional steps to represent visual information in symbolic form that is adequate for existing methods. In this paper we assess the performance of a data mining method, which has been developed focusing on textual data, in the task of image classification. For that purpose we propose an approach to image representation, a method of building a classifier and using it to perform classification of visual data.

Emerging substrings (ESs) [1] are patterns that can be used to differentiate classes of data consisting of sequences of symbols. The idea originates from emerging patterns (EPs) [2], a data mining method of extracting patterns that occur frequently in one class of data and seldom in another. Emerging patterns is an approach to KDD that proved to perform very well in the tasks of classification and prediction of large sets of data, many times much better than classical methods, such as rule- and tree-based classifiers. Emerging substrings allow additionally to reason about sequences of symbols or objects in data, which is an important feature of a visual data mining method. Specifically, we can reason about the spatial arrangement of objects on a particular image. On these grounds we expect an ES-based classifier to perform better in the task of image classification than previously proposed methods based on the idea of emerging patterns. In particular, we suggest using a subset of emerging substrings – jumping emerging substrings – to build classifiers capturing the most distinctive features of two data sets.

In what follows we first outline work conducted previously in the field of pattern-based image classification (Section 2), then give the necessary definitions of jumping emerging substrings (Section 3). Next, we describe image representation used in our experiments (Section 4), the proposed classification method (Section 5) and compare it with other known approaches (Section 6). Finally, we conclude with possibilities of further research (Section 7).

2 Previous Work

The idea of mining emerging substrings as means of capturing interesting relationships in textual data has been proposed in [1]. It was motivated by the earlier concept of emerging patterns, proposed in [2], which have been successfully used in classification of a variety of datasets. While the original algorithm for mining ESs was based on suffix trees, a generalized, linear-time solution has been proposed in [3]. This result, based on suffix arrays and longest common prefix (lcp) tables, has been later improved in [4].

To the best of our knowledge emerging substrings have not been previously studied in the context of image classification, while our own experiments concerning mining jumping emerging patterns in multimedia data have been presented in [5].

3 Jumping Emerging Substrings

Here we cite only the essential definitions of JESs, used in further parts of the paper. Please refer to [1] for complete formal definition.

A sequence is a non-empty string with finite length over an alphabet $\Sigma = \{a_1, a_2, \ldots, a_m\}$. The length of a sequence is the number of symbols contained in it. Having a string $s = s_1 s_2 \ldots s_k$ of length k and a sequence $T = t_1 t_2 \ldots t_l$ of length l, we say that s is a substring of T, denoted as $s \sqsubseteq T$ if $\exists i \in 1 \ldots (l-k+1)$ such that $s_1 s_2 \ldots s_k = t_i t_{i+1} \ldots t_{i+k-1}$. If $s \neq T$, s is a proper substring of T, denoted as $s \sqsubset T$.

A database D is a set of sequences T_i, each associated with a class label $c_{T_i} \in C = \{c_1, c_2, \ldots, c_n\}$, where C is the set of all labels. The support of a string s in a database D is the fraction of sequences in D that s is a substring of: $\mathrm{supp}_D(s) = \frac{|\{T \in D:\ s \sqsubseteq T\}|}{|D|}$. Given two databases $D_1, D_2 \subseteq \mathcal{D}$ we say that a string s is a jumping emerging substring (JES) from D_1 to D_2 if

$\mathrm{supp}_{D_1}(s) = 0 \wedge \mathrm{supp}_{D_2}(s) > 0$. The task of JES mining is to find all strings having a given minimum support θ in D_2, being a JES from D_1 to D_2. We will denote this set of strings as $JES(D_1, D_2, \theta)$. Furthermore, we can distinguish the set of only minimal JESs, that is sequences, for which no frequent substrings exist: $JES_{\mathrm{m}}(D_1, D_2, \theta) = \{T \in JES(D_1, D_2, \theta) : \neg\exists s \in JES(D_1, D_2, \theta)\ s \sqsubset T\}$.

Table 1 shows a simple two-class database and its jumping emerging substrings. Based on the above definition, we look at all possible substrings of strings in class A and find these, which are not present in class B. Similarly, we check for JESs from class B to A. The string "ac" would be the only JES, if we were to find only jumping emerging substrings with minimum support of 1. Finally, we reduce the set of discovered patterns to only minimal JESs: $JES_{\mathrm{m}}(D_A, D_B, 1/2) = \{b, e\}$, $JES_{\mathrm{m}}(D_B, D_A, 1/2) = \{ac\}$.

Table 1: Example database and its jumping emerging substrings

class A	class B
acd	cde
ac	ab

JES	support class A	support class B	direction
b	0	1/2	A → B
e	0	1/2	A → B
ab	0	1/2	A → B
ac	1	0	B → A
de	0	1/2	A → B
acd	1/2	0	B → A
cde	0	1/2	A → B

4 Image Representation

We have compared two approaches to calculation of image features: using both a color descriptor and a texture descriptor based on Gabor filters (as in MPEG-7 standard), and a SIFT descriptor. In both cases we divide the images into a rectangular $x \times y$ grid and calculate features in each of the resulting tiles.

In the first approach color and texture features are calculated separately. Image colors are represented by a histogram calculated in the HSV color space, with the hue channel quantized to h discrete ranges, while saturation and value channels to s and v ranges respectively. In effect, the representation takes the form of a $h \times s \times v$ element vector of real values between 0 and 1. For the representation of texture we use a feature vector consisting of mean and standard deviation values calculated from the result of filtering an original image with a bank of Gabor functions. These filters are scaled and rotated versions of the base function, which is a product of a Gaussian and a sine function. By using m orientations and n different scales we get a feature vector consisting of mean (μ) and standard deviation (σ) values of each of the filtered images and thus having a size of $2 \times m \times n$ values. In our experiments a vector size of $2 \times 6 \times 4 = 48$ values has been used for texture and $18 \times 3 \times 3 = 162$ for color representation.

SIFT is a local feature descriptor, proposed in [6], which has been widely used for image representation in classification, recognition and retrieval tasks. Using the VLFeat open implementation [7], we have calculated SIFT features of the center point of each of the image tiles for H, S and V color channels, having a constant scale and orientation set for the descriptor. The feature vector size for every point is thus equal to 3×128 values.

Having calculated features of each of the images in both the training and testing set, we have created a visual dictionary of the most representative color and texture features. The dictionary is built by clustering corresponding feature values into a chosen number of groups. Resulting centroids become the elements of the dictionary and are labeled with unique symbols. These identifiers are then used to describe the images in the database by associating an appropriate label with every tile of each image. This is performed by finding the closest centroid to a feature vector calculated for a given image tile. The same dictionary is used during both the learning and classification phases.

Figure 1 illustrates the used method of image representation. A regular grid of points is used to calculate images features, which are then clustered to create the dictionary. In the case of MPEG-7 features, values representing color and texture are clustered separately and labeled $B_1, B_2, \ldots, B_n$ and $T_1, T_2, \ldots, T_n$ respectively. These labels are then used to describe each of the grid tiles.

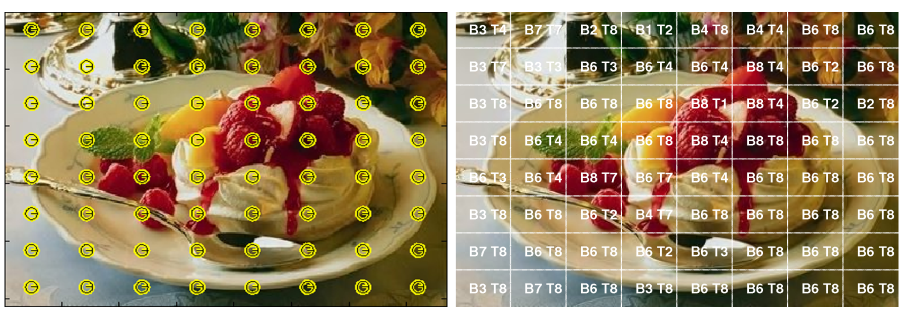

Fig. 1: An example of features calculation and symbolic image representation

5 JES-based Classification

In our approach classification is a two-step process. The first phase consists of building a classifier on the basis of the learning dataset. We use image representation described in the previous section to associate sets of strings to each of the images in the dataset and then mine minimal jumping emerging substrings between respective classes in the database. The strings are formed by taking into account horizontal, vertical and diagonal sequences of symbols of representation of a particular image (see Fig. 2).

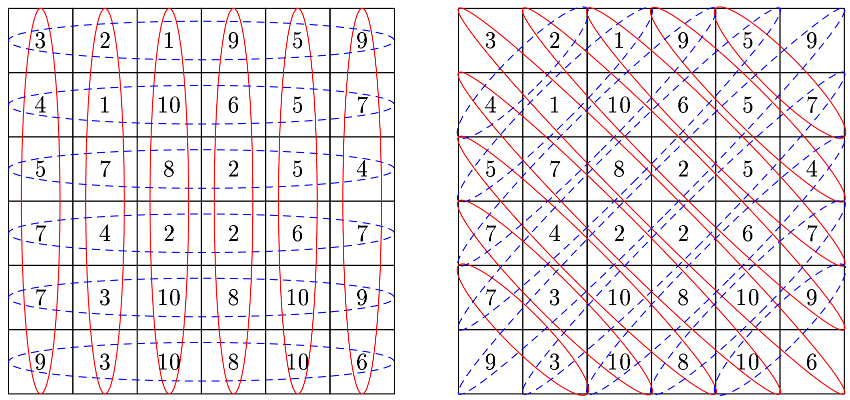

Fig. 2: Representation used to mine JESs between classes of images. Strings are formed by considering horizontal, vertical and diagonal sequences of symbols.

In the second phase, we use the created classifier to assign images from the testing set to respective categories. This is done by aggregating all minimal JESs that match the representation of a particular image and determining the majority class of the patterns. The winning category is then assigned to the example.

Formally, for a multi-class set of images, represented as a learning database of strings with associated class labels $\mathcal{D}_L = \bigcup_c C_c$, where C_c is a database containing images of class c and C'_c is its complementary database, and a test set $\mathcal{D}_T$, we can formally write the algorithm as follows:

1. For each $c \in C$:
   (a) Discover minimal $JES_{\mathrm{m}}(C'_c, C_c, \theta)$
   (b) For each test image $T \in \mathcal{D}_T$: calculate $\mathrm{score}(T, c) = \sum_X \mathrm{supp}_{C_c}(X)$, where $X \in JES(C'_c, C_c)$ such that $X \sqsubseteq T$.
2. Assign image T to a class c, which has the maximum score.

6 Experimental Results

We have used two different datasets to assess the performance of the proposed JES-based image classification approach. Firstly, we have prepared a synthetic two-class set of images, which consists of photographs containing the same object, positioned randomly on a static background. On the images of class A the object is oriented vertically, while in class B – horizontally (see Fig. 3). Each of the classes contains ca. 60 images.

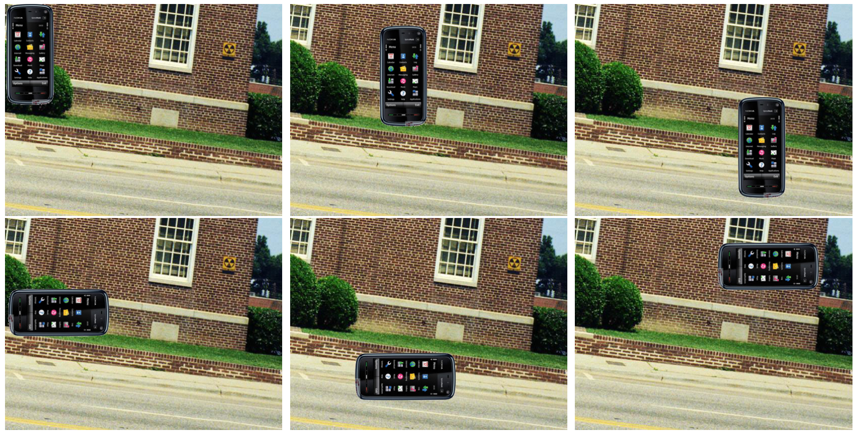

Fig. 3: A synthetic test database with two classes of images

Table 2: Classification accuracy of the synthetic dataset

method	minimum support	MPEG-7 features accuracy (%)	MPEG-7 features patterns no.	SIFT features accuracy (%)	SIFT features patterns no.
JES	0.250	95.33	92	69.33	80
	0.200	96.67	156	79.67	140
	0.150	97.50	229	**87.00**	198
	0.100	98.33	352	86.00	574
	0.050	**99.17**	1175	85.67	1830
	0.025	99.17	3304	85.33	10934
	0.010	99.17	20797	85.33	10934
	0.005	99.17	20797	85.33	10934
C4.5	-	93.46	-	57.00	-
SVM	-	96.67	-	65.00	-

The database has been prepared to validate the idea behind using JESs for image classification and the chosen image representation method. While the object and background are exactly the same in each of the classes, in our approach we are able to capture more data about their relationship than using regular methods, which do not take spatial information into consideration. As presented in Table 2 JES-based classifier performs much better, regardless of used feature descriptor.

All experiments have been performed as a ten-fold cross validation, where the feature dictionary is recreated in every iteration. The images have been divided into 8×8 tiles and the dictionary size has been limited to 16 values. In the case of this synthetic database, SIFT features have resulted in worse classification performance than the color and Gabor-based texture features, mostly because of the sparse grid used to calculate values in particular points. For comparison purposes, we have used the same locations for calculation of SIFT and MPEG-7 features. Classification with other methods than emerging substrings and emerging patterns has been carried out using the Weka package [8] and the LIBSVM library [9] with default parameter values.

Table 3: Classification accuracy of the SIMPLIcity dataset with MPEG-7 features

method	minimum support	accuracy (%) *flower/ food*	*flower/ elephant*	*flower/ mountain*	*food/ elephant*	*food/ mountain*	*elephant/ mountain*
JES	0.250	92.26	93.68	96.37	30.50	83.50	58.00
	0.200	94.79	95.26	96.89	41.00	89.50	66.00
	0.150	96.37	97.89	96.89	63.50	93.00	74.50
	0.100	97.94	98.95	96.89	85.00	94.00	89.00
	0.050	**98.47**	98.95	96.89	**93.00**	95.50	92.00
	0.025	98.47	98.95	96.89	93.00	**96.00**	**93.50**
	0.005	98.47	98.95	96.89	93.00	95.50	93.50
occJEP [5]	-	97.92	**98.96**	**97.92**	88.00	91.00	88.50
JEP [5]	-	95.83	91.67	96.35	88.50	93.50	83.50
C4.5	-	93.23	89.58	85.94	87.50	92.50	82.00
SVM	-	90.63	91.15	93.75	87.50	84.50	84.50

Secondly, we have included results of classification of a dataset used in our earlier experiments in [5], namely the image database created by the authors of the SIMPLIcity CBIR system [10] (see Fig. 4). This set consists of 10 categories of photographs, 100 images in each class. As reported in Table 3, our current approach is in each case giving better results than any of the others. It may be noted that lowering the minimum support value when mining JESs improves the classification accuracy only to certain point, above which there is no additional gain of discovering greater number of patterns.

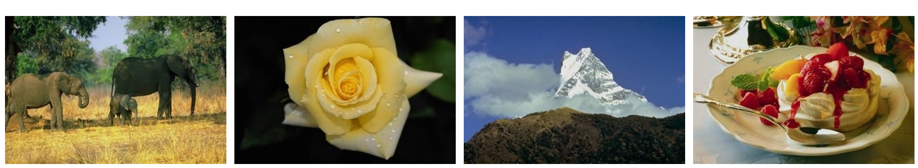

Fig. 4: Example images from the SIMPLIcity test database

7 Conclusions and Future Work

In this paper we have proposed an approach to image classification that combines the methods used for sequence and text mining with image analysis and showed that such methodology may give promising results, surpassing the performance of other data mining methods. Using jumping emerging substrings to distinguish images of different classes in a database has a clear advantage over other pattern-based methods, thanks to its ability to capture spatial relationships between visual features. It is important to note that optimal (linear-time) algorithms exist to mine JESs between sets of sequential data. Furthermore, the proposed approach may be used in conjunction with different feature descriptors, as long as the images are expressed by a matrix of a finite number of symbols.

The following aspects of the described method could be enhanced in future work: invariance to scale by providing multiple layers of symbolic representation of an image, each calculated using a descriptor of a different scale; using a dense grid of points for SIFT and multiple orientations to achieve better results than the MPEG-7 approach.

References

1. Chan, S., Kao, B., Yip, C.L., Tang, M.: Mining emerging substrings. In: Proceedings of the Eighth International Conference on Database Systems for Advanced Applications. (2003) 119–126
2. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (1999) 43–52
3. Fischer, J., Heun, V., Kramer, S.: Optimal string mining under frequency constraints. In: Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases. Volume 4213 of Lecture Notes in Artificial Intelligence. (2006) 139–150
4. Fischer, J., Mäkinen, V., Välimäki, N.: Space efficient string mining under frequency constraints. In: Proceedings of the 8th IEEE International Conference on Data Mining, Los Alamitos, CA, USA, IEEE Computer Society (2008) 193–202
5. Kobyliński, Ł., Walczak, K.: Efficient mining of jumping emerging patterns with occurrence counts for classification. In Chan, C.C., Grzymała-Busse, J.W., Ziarko, W.P., eds.: International Conference on Rough Sets and Current Trends in Computing. Volume 5306 of LNAI., Springer-Verlag (2008) 419–428
6. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
7. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)
8. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd edn. Morgan Kaufmann, San Francisco (2005)
9. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
10. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. IEEE Trans. on Patt. Anal. and Machine Intell. **23** (2001) 947–963