



## A Novel Texture Classification Procedure by using Association Rules

L. Jaba Sheela<sup>1</sup> & V. Shanthi<sup>2</sup>

<sup>1</sup> Panimalar Engineering College, Chennai.

<sup>2</sup> St. Joseph's Engineering College, Chennai

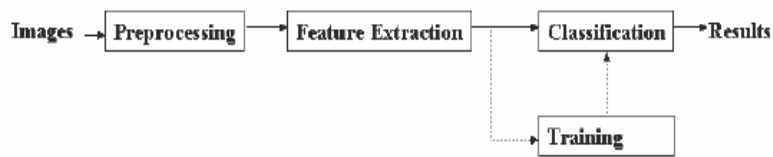
**Abstract.** Texture can be defined as a local statistical pattern of texture primitives in observer's domain of interest. Texture classification aims to assign texture labels to unknown textures, according to training samples and classification rules. Association rules have been used in various applications during the past decades. Association rules capture both structural and statistical information, and automatically identify the structures that occur most frequently and relationships that have significant discriminative power. So, association rules can be adapted to capture frequently occurring local structures in textures. This paper describes the usage of association rules for texture classification problem. The performed experimental studies show the effectiveness of the association rules. The overall success rate is about 98%.

**Keywords:** *association rules; feature extraction; image mining; texture classification.*

### 1 Introduction

According to [1], an image region has constant texture if some local statistics or properties in that region are slowly varying or approximately periodic. Texture is an important characteristic for analysis of many types of images that including natural scenes, remotely sensed data and biomedical modalities [2]. The perception of texture is believed to play an important role in human visual system for classification and presentation.

Texture can be defined as a local statistical pattern of texture primitives in observer's domain of interest. Texture classification aims to assign texture labels to unknown textures, according to training samples and classification rules. Two major issues are critical for texture classification: the texture classification algorithms and texture feature extraction. The main aim of texture classification is to find a best matched category for a given texture among existing textures. Figure 1 gives a overview of the entire process.



**Figure 1** Overview of Texture Classification.

Many previous works are proposed to solve texture classification problem [3-7]. A comparison of textural features from Fourier power spectrum, second-order gray level statistics and first-order statistics of gray level differences were shown in [3]. Other textural features including co-occurrence features, Gabor features, Markov random fields (MRF) based features and fractal features were compared in another work [4]. In [5], a new rotational and scale invariant feature set for textural image classification was presented and a combined invariant feature (CIF) set was proposed. Arivazhagan et al [2] used wavelet statistical features, wavelet co-occurrence features and combination of wavelet statistical features and co-occurrence features of one level wavelet transformed images with different feature databases. A distance classifier is used for measurement of similarity and consequent labeling. Chen et al; gave a comparative study for filtering methods for texture classification according to their discrimination ability [6]. Their main aim was to evaluate the performance of four filtering methods including Fourier transform, spatial filter, Gabor filter and wavelet transform for classification. The superiority of the wavelet transform is shown by their experimental study. Rushing et al; proposed an association rule based feature extraction method for texture images [7]. They embedded the concept of association rules to capture the frequently occurring local intensity variation in textures. The frequency of occurrence of these local patterns within a region is used as texture features.

In this paper, we propose an algorithm that classifies textures based on association rules features. We use Apriori algorithm for mining rules from the texture image database. We applied the proposed algorithm on a set of Brodatz texture images and the results are satisfactory.

The organization of this paper is as follows, in Section 2 we provide an introduction of association rules and Apriori algorithm. Several definitions can be seen at this Section. In Section 3; the methodology and the implementation of the proposed process is given. In Section 4; experimental study is introduced and the classification results are shown. Finally in Section 5, we conclude the study.

## 2 Preliminaries

### 2.1 Association Rules

In order to see how association rules can be used in texture feature extraction, firstly we need to define association rules. Association rules are one of the promising aspects of data mining as a knowledge discovery tool, and have been widely explored to date. They allow capturing all possible rules that explain the presence of some attributes according to the presence of other attributes [8]. Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of literals, called items. Let  $D$  be a database of transactions, where each transaction  $T$  is a set of items such that  $T \subseteq I$ . For a given itemset  $X \subseteq I$  and a given transaction  $T$ , we say that  $T$  contains  $X$  if and only if  $X \subseteq T$ . The support count of an itemset  $X$  is defined to be  $\text{sup}_X =$  the number of transactions in  $D$  that contain  $X$ . We say that an itemset  $X$  is large, with respect to a support threshold of  $s\%$ , if  $\text{sup}_X \geq |D| \times s\%$ , where  $|D|$  is the number of transactions in the database  $D$ . An association rule is an implication of the form “ $X \Rightarrow Y$ ”, where  $X \subseteq I$ ,  $Y \subseteq I$  and  $X \cap Y = \emptyset$ . The association rule “ $X \Rightarrow Y$ ” is said to hold in database  $D$  with confidence  $c\%$  if no less than  $c\%$  of the transactions in  $D$  that contain  $X$  also contain  $Y$ . The rule  $X \Rightarrow Y$  has support  $s\%$  in  $D$  if  $\text{sup}_{X \cup Y} = |D| \times s\%$  [8]. Thus, association rules aim at discovering the patterns of co-occurrence of attributes in a database. For instance, an association rule in a supermarket basket data may be in 10% of transactions, 85% of the people buying milk also buy yoghurt in that transaction. Association rules may be useful in many applications such as supermarket transactions analysis, store layout and promotion on the items, telecommunication alarm correlation, university course enrollment and recently texture and image analysis.

### 2.2 Apriori Algorithm

The Apriori algorithm is a state of the art algorithm that most of the association rule algorithms are somewhat variations of this algorithm [8]. The Apriori algorithm works iteratively. It first finds the set of large 1- itemsets, and then set of 2-itemsets, and so on. The number of scan over the transaction database is as many as the length of the maximal itemset. Apriori is based on the following fact: The simple but powerful observation leads to the generation of a smaller candidate set using the set of large itemsets found in the previous iteration. The Apriori algorithm presented in [9] is given as follows;

```

Apriori ()
 $L_1 = \{l \mid \text{arg } e \geq 1 - \text{itemsets}\}$ 
 $k = 2$ 
while  $L_{k-1} \neq \phi$  do
begin
 $C_k = \text{apriori\_gen}(L_{k-1})$ 
for all transactions  $t$  in  $D$  do
begin
 $C^t = \text{subset}(C_k, t)$ 
for all candidate  $c \in C^t$  do
 $c.\text{count} = c.\text{count} + 1$ 
end
 $L_k = \{c \in C_k \mid c.\text{count} \geq \text{min\_sup}\}$ 
 $k = k + 1$ 
end
end

```

Apriori, first scans the transaction databases  $D$  in order to count the support of each item  $i$  in  $I$ , and determines the set of large 1-itemsets. Then, one iteration is performed for each of the computation of the set of 2-itemsets, 3-itemsets, and so on. The  $k^{\text{th}}$  iteration consists of two steps [9]

1. Generate the candidate set  $C_k$  from the set of large  $(k-1)$ -itemsets,  $L_{k-1}$ .
2. Scan the database in order to compute the support of each candidate itemset in  $C_k$

The candidate generation algorithm is given as follows;

```

Apriori_gen( $L_{k-1}$ )
 $C_k = \phi$ 
for all itemsets  $X \in L_{k-1}$  and  $Y \in L_{k-1}$  do
if  $X_1 = Y_1 \wedge \dots \wedge X_{k-2} = Y_{k-2} \wedge X_{k-1} < Y_{k-1}$  then begin
 $C = X_1 X_2 \dots X_{k-1} Y_{k-1}$ 
add  $C$  to  $C_k$ 
end
end
delete candidate itemsets in  $C_k$  whose any subset is not in  $L_{k-1}$ 

```

	X=0	X=1	X=2	X=3	X=4
Y=0	2	0	2	1	0
Y=1	0	2	0	2	1
Y=2	1	1	2	0	2
Y=3	2	2	0	2	1
Y=4	1	0	2	1	2

(a)

$$(0,0,2) \wedge (1,1,2) \Rightarrow (1,0,0)$$

Support:  $3/9 = 33.33\%$

Confidence  $3/5 = 60\%$

(b)

**Figure 2** (a) Sample Image (b) Sample rule with Support & Confidence.

The candidate generation procedure computes the set of potentially large k-itemsets from the set of large (k-1)-itemsets. A new candidate k-itemset is generated from two large (k-1)-itemsets if their first (k-2) items are the same. The candidate set  $C_k$  is a superset of the large k-itemsets. The candidate set is guaranteed to include all possible large k-itemsets because of the fact that all subsets of a large itemset are also large. Since all large itemsets in  $L_{k-1}$  are checked for contribution to candidate itemset, the candidate set  $C_k$  is certainly a superset of large k-itemsets. After the candidates are generated, their counts must be computed in order to determine which of them are large. This counting step is really important in the efficiency of the algorithm, because the set of the candidate itemsets may be possibly large. Apriori handles this problem by employing a hash tree for storing the candidate. The candidate generation algorithm is used to find the candidate itemsets contained in a transaction using this hash tree structure. For each transaction T in the transaction database D, the candidates contained in T are found using the hash tree, and then their counts are incremented. After examining all transaction in D, the ones that are large are inserted into  $L_k$ .

### 3 Illustration

For example, consider the image as shown in Figure 2a. Assume that the X and Y offsets of the rule are restricted to 0 and 1, as is the case for the rule specified in Figure 1b. Rules with offsets restricted to 0 and 1 can be applied to this

image at nine of the 25 pixel locations. Of these nine locations, the antecedent matches at five locations, and both the antecedent and consequent match at three locations. Therefore, the support for the rule is  $3/9$  or 33.33 percent and the confidence of the rule is  $3/5$  or 60 percent. Note that the pixels at which the rule cannot be applied are not considered in the computation of support or confidence values.

### Step by Step Working:

	X=0	X=1	X=2	X=3	X=4
Y=0	2	0	2	1	0
Y=1	0	2	0	2	1
Y=2	1	1	2	0	2
Y=3	2	2	0	2	1
Y=4	1	0	2	1	2

n\*n matrix in (1, 0)

(0, 0)

2	0	2	1	0
0	2	0	2	1
1	1	2	0	2
2	2	0	2	1
1	0	2	1	2

Min support = 3

Itemset	Count	Frequent
(0, 0, 0)	3	Yes
(0, 0, 1)	1	
(0, 0, 2)	5	Yes

(1, 0)

2	0	2	1	0
0	2	0	2	1
1	1	2	0	2
2	2	0	2	1
1	0	2	1	2

Itemset	Count	Frequent
(1, 0, 0)	3	Yes
(1, 0, 1)	2	
(1, 0, 2)	4	Yes

Itemset	Count	Frequent
(0, 0, 0)	3	Yes
(0, 0, 1)	1	
(0, 0, 2)	5	Yes
(1, 0, 0)	3	Yes
(1, 0, 1)	2	
(1, 0, 2)	4	Yes
(1, 1, 0)	2	
(1, 1, 1)	2	
(1, 1, 2)	5	Yes

$$\{(0, 0, 0), (1, 0, 2)\} \Rightarrow \text{Sup}(\{(0, 0, 0), (1, 0, 2)\}) = 3$$

	X=0	X=1	X=2	X=3	X=4
Y=0	2	0	2	1	0
Y=1	0	2	0	2	1
Y=2	1	1	2	0	2
Y=3	2	2	0	2	1
Y=4	1	0	2	1	2

$$\{(0, 0, 2), (1, 1, 2)\} \Rightarrow \text{Sup}(\{(0, 0, 2), (1, 1, 2)\}) = 5$$

	X=0	X=1	X=2	X=3	X=4
Y=0	2	0	2	1	0
Y=1	0	2	0	2	1
Y=2	1	1	2	0	2
Y=3	2	2	0	2	1
Y=4	1	0	2	1	2

$$\text{Sup}(\{(0, 0, 0)\}) = 3,$$

$$\text{Sup}(\{(1, 0, 2)\}) = 4,$$

$$\text{Sup}(\{(0, 0, 0), (1, 0, 2)\}) = 3$$

$$(0,0,0) \Rightarrow (1,0,2) = \frac{\text{sup}(\{(0,0,0) \cap (1,0,2)\})}{\text{sup}(\{(0,0,0)\})} = 3/3 = 1$$

$$(1,0,2) \Rightarrow (0,0,0) = \frac{\text{sup}(\{(0,0,0) \cap (1,0,2)\})}{\text{sup}(\{(1,0,2)\})} = 3/4$$

#### 4 Texture Classification

In this section, the methodology of using association rules for texture image classification is presented. Texture images appear similar and there exist small variations in the structure and intensity values of the corresponding pixels. Thus, the gray-level textures pose some challenging problems for the use of association rule approach to capture local image attributes. Visually similar textures produce of the itemsets appearing in transaction get discarded because they do not satisfy the support criterion. This means that common attributes presented in images can not be recognized by the mining algorithm because the transactions of a specific pixels that belong to a given texture region do not have many items in common. The association rules based texture image classification procedure which is proposed in this study is composed of 2 steps.

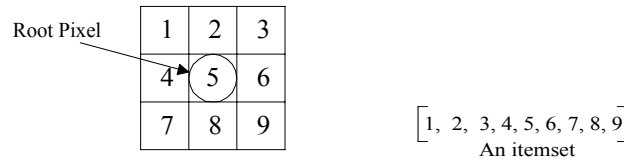
**Step 1:** In this step, all of the texture images that are used in this study are firstly mapped from pixel intensity value to 0, 1 and 2 according to the Equation 1 [7].

$$new\ value = \begin{cases} 0 & root \leq \mu - c * \sigma \\ 1 & \mu - c * \sigma < root < \mu + c * \sigma \\ 2 & root \geq \mu + c * \sigma \end{cases} \quad (1)$$

Where, root is the pixel at the centre of the neighborhood. At each root pixel, the mean  $\mu$  and standard deviation  $\sigma$  of intensity value in a local region of size centered about the root are computed. The new value is assigned to the root.  $c$  is a constant and we select 0.3 for this paper.

**Step 2:** This step contains the Apriori algorithm where the associations of the pixels in a local region are extracted. Apriori algorithm is explained at section 2.2 in detailed so here, we only describe the application of the algorithm on texture images. Apriori algorithm uses only 0 and 1 values as input thus, we code our new pixel values (0), (1) and (2) as (1 0 0), (0 1 0) and (0 0 1) respectively. A 3x3 window is used for all pixels in the texture image for forming the transactions. A transaction has a vector form of size 1x9. The related illustrations are shown in Figure 3.





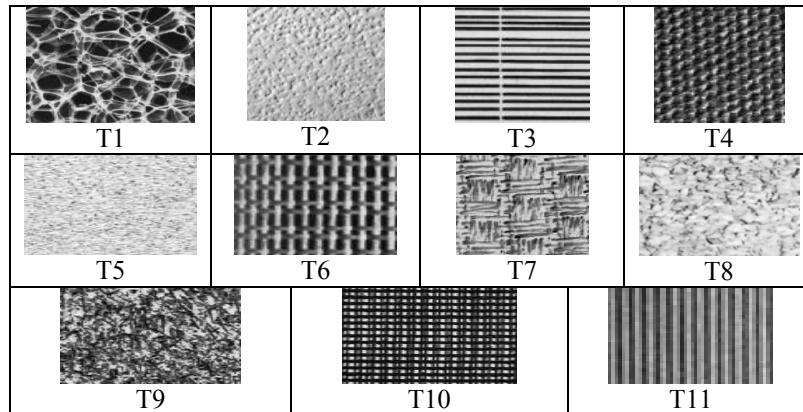
**Figure 3** A transaction form.

### 5 Experimental Results

Experiments are conducted with 11 Brodatz texture images [10] (obtained from [11]) are shown at Figure 4. Instead of using whole texture images for feature extraction with association rules, we divide the texture images into small regions of size 40x40. Because texture images have repetitive regions so using whole image cause more computation and time consuming. These texture image pieces are then transformed into a new form as described at step 1 of Section 3. A 3x3 sliding window is used for constructing the transactions according to the root pixels. This process is also described in step 2 of Section 3. These transactions are the input to the Apriori algorithm. Apriori algorithm generates several itemsets and these itemsets characterize the input texture images. The minimum support value for the Apriori algorithm is chosen as 7%. In our proposal, we do not generate rules; we only use the large itemsets and their minimum support threshold value that the Apriori algorithm generates. This can be viewed as finding the co occurrences of the pixels in the sample texture images.

Table 1 shows the experimental results. This table is obtained as follows; the first and the second column of the Table 1 show the texture images and their support threshold values respectively. Before testing, the large itemsets that characterize each texture image is extracted and saved. Several non-overlapping sub images of size 40x40 are obtained from the each texture images for constructing the test. Each column shows the related texture and it is similarity to the texture which is at the rows of the table. This similarity is calculated as follows,  $Similarity = \arg \min_{i,j} |Support_{T_i} - Support_{T_j}|$

Here i and j are the rows and the columns of the table. The accuracy of the classification can be shown in a percentage form. For example, the support value for the  $T_1$  is 7.8. The Apriori algorithm produced 0.4 for the  $T_1$ . Thus,  $(7.8-0.4)*100/7=95\%$  is obtained.



**Figure 4** Texture images.

**Table 1** The classification results.

	Support Values	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
T1	7,80	0,4	6,8	6,2	7,3	7,1	7,2	7,0	6,9	6,93	6,8	6,9
T2	5,19	17,3	0,95	3,7	4,7	4,2	5,1	2,1	3,6	2,5	1,3	3,9
T3	6,28	7,2	4,8	0,96	59	6,0	6,0	5,0	5,6	5,4	4,4	5,6
T4	5,47	5,5	1,9	4,6	0,4	3,1	5,8	1,6	2,4	3,6	3,1	2,4
T5	5,00	5,2	1,4	4,1	2,6	0,9	6,5	1,2	2,6	2,9	3,4	2,4
T6	7,00	7,0	6,9	6,95	6,96	6,96	0,8	6,4	6,9	6,98	6,97	6,8
T7	5,24	5,2	4,0	5,0	4,0	4,7	17,4	0,9	3,9	4,9	5,0	3,6
T8	6,02	7,9	1,9	3,5	1,8	1,7	6,64	1,98	0,58	1,18	2,7	1,2
T9	5,29	27,9	1,29	2,84	3,7	2,9	5,1	1,38	2,0	0,4	2,7	2,3
T10	5,91	11,1	2,7	2,8	5,4	5,2	5,8	3,7	4,6	3,8	0,4	4,7
T11	5,09	5,1	2,1	4,13	2,8	2,37	19,07	1,73	1,40	2,63	2,4	0,9

On the other hand, we conducted another test for the proposed process; we make 100 overlapping sub images of size 64x64 for each texture images. So totally we obtained 1100 texture images for testing the algorithm. The correct classification rates are shown at Table 2. According to the Table 2, the overall success rate is about 98%.

**Table 2** The correct classification rates.

Texture Images	Number of correct classified textures	Number of incorrect classified textures	Total number of textures
T1	99	1	100
T2	95	5	100
T3	100	0	100
T4	99	1	100
T5	100	0	100
T6	98	2	100
T7	100	0	100
T8	100	0	100
T9	96	4	100
T10	99	1	100
T11	100	0	100

## 6 Conclusion

This work indicates the use of association rules for texture image classification. Association rules have been used in business applications such as market basket analysis. They are shown to be capable of capturing frequently occurring local image structure, which are crucial for texture characterization and classification. The frequencies of occurrences of these local patterns within a region are used as texture features. The utility of the association rule features for supervised texture classification is shown using several experimental studies. Experimental studies indicate that the association rules achieve texture segmentation with a high success rate. The overall success is about 98%.

## References

- [1] Sklansky J., *Image segmentation and feature extraction*, IEEE Trans. System Man Cybernat., 8,237-247, 1978.
- [2] Arivazhagan S. and Ganesan L., *Texture classification using wavelet transform*, Pattern Recog. Lett. 24, 1513-1521, 2003.
- [3] Weszka J. S., Dyer C.R., Rosenfeld A., *A comparative study of texture measures for terrain classification*, IEEE Trans. Syst. Man. Cybernet. 6, 269-285, 1976.
- [4] Jain A.K., Farrokhnia F., *Unsupervised texture segmentation using Gabor filters*, Patt. Recog., 24, 1167-1186, 1991.
- [5] Muneeswaran K., Ganesan L., Arumugam S. and Ruba Soundar K., *Texture classification with combined rotation and scale invariant wavelet features*, Pattern Recognition, 38, 10, 1495-1506, 2005

- [6] Chen C. C., Chen C. C. *Filtering methods for texture discrimination*, Patt. Rec. Lett. 20, 783-790, 1999.
- [7] Rushing, J.A.; Ranganath, H.S.; Hinke, T.H.; Graves, S.J., *Using association rules as texture features*, Pattern Analysis and Machine Intelligence, IEEE Transactions on Vol. 23, 845 – 858, Aug. 2001
- [8] Agrawal R., Imielinski T., and Swami A., *Mining association rules between sets of items in large databases*, In proc. ACM SIGMOD international conf. on Management of data, Washington DC, May 1993.
- [9] Agrawal R., Srikant R., *Fast algorithms for mining association rules in large databases*, In proc. 20<sup>th</sup> Int. Conf. on very large databases, pp. 487-499. Santiago, Chile, 1994.
- [10] <http://sipi.usc.edu/database/database.cgi?volume=textures&image=1#top>