# iARM: Image Association Rule Mining Language

*Hassan H. Malik*

COMS W4115: Programming Languages and Translators
Department of Computer Science
Columbia University

logicators@yahoo.com

February 6, 2005

## 1. Introduction

Image mining [1] deals with extraction of implicit knowledge, image data relationship or other patterns not explicitly stored in images and uses ideas from computer vision, image processing, image retrieval, data mining, machine learning, databases and AI. The fundamental challenge in image mining is to determine how low-level, pixel representation contained in an image or an image sequence can be effectively and efficiently processed to identify high-level spatial objects and relationships. Typical image mining process involves preprocessing, transformations and feature extraction, mining (to discover significant patterns out of extracted features), evaluation and interpretation and obtaining the final knowledge. Various techniques from existing domains are also applied to image mining and include object recognition, learning, clustering and classification, just to name a few. Association rule mining [2] is a well-known data mining technique that aims to find interesting patterns in very large databases. Some preliminary work has been done to apply association rule mining on sets of images to find interesting patterns [3] [4].

Extracting association rules from images remains a complex, tedious process. It typically requires writing several hundred lines of code to read images, extract features and apply a mining algorithm such as APRIORI [2]. iARM is a scripting language that makes it easier to extract association rules from images. It allows defining a list of source image files and customizing association rule parameters. These parameters include number of terms, filters on text feature and configuration of signal features (i.e. color), support and confidence. Using iARM, association rules can be extracted by writing simple, easy to understand code that can be written and maintained by end users (i.e. knowledge workers) with no programming knowledge.

## 2. Background and Related Work

An association rule [2] is represented in LHS => RHS form with both LHS and RHS allowed to contain multiple items. *Support* of an association rule is defined as the percentage of transactions that contains all items (both LHS and RHS) in an association rule and *confidence* of an association rule is defined as the percentage of LHS items that also contains RHS. An association rule holds if its support is greater than *minsup* and confidence is greater than *minconf*, where *minsup* and *minconf* are configurable. The problem of finding association rules is decomposed into sub-problems of finding all itemsets with at-least minimum support (also called large itemsets) and using these large itemsets to generate the desired rules (tested for minimum confidence). Large itemset generation is achieved by generating candidate itemsets and keeping the ones with minimum support.

Ordonez and Omiecinski [4] presents a data mining algorithm that finds association rules from two-dimensional color images, without using domain specific information. An experimental system was built on top of content based image retrieval system (CBIR) a.k.a. Blobworld system from UC Berkeley. The CBIR system supports object-based queries (queries that search for images that contain particular objects) and eliminates the

need of manually indexing images. These queries are performed on descriptors generated from the image content that contain information about color, texture, shape and size. Objects are determined by the similarity of these attributes. Each detected object is assigned an ID and association rules are generated based on the presence / absence of objects in images (using support and confidence as measures). In order to generate association rules, objects extracted by "BlobWorld" are considered analogous to items and images are considered analogous to transactions. Candidate itemsets are generated from the set of objects and "large" itemsets are determined by checking individual images for presence / absence of objects (support calculation). This information is further used to calculate confidence. Unfortunately, this approach is impractical for complex images because it heavily relies on object recognition by another system (BlobWorld).

Haddad and Mulhem [3] presents a more practical approach to generate association rules from images using both manual annotations of regions (symbolic elements or concepts) inside the image and features like prominent colors, directions and texture indicators. Images are segmented to generate regions based on spatial connectivity and visual similarity. These regions are later used to assist in association rule generation. Results of experiments performed on a fairly small dataset (100 images of same type) are included.

## 3. Language Features

### 3.1. Ease of Use

Writing image association rule mining code in a typical programming language (i.e. Java) is a non-trivial task and requires advanced programming and image processing skills. In contrast, iARM provides specialized association rules extraction instructions that can easily be understood by end users.

### 3.2. Supports Multiple Image Features

iARM supports both text feature (associated text provided in a separate file) and a signal feature (color histogram). These features can be combined to form a single association rule, enabling extraction of versatile and powerful rules (i.e. 65% of the images that contains dark green color has the word 'grass' in their label).

iARM also provides a text processing feature that enables stop word elimination and low / high frequency word elimination, based on Zipf's law.

### 3.3. Customizable

iARM is fully customizable. It allows defining input image files, which features to use, configuring text filters, color histogram parameters and various thresholds.

### 3.4. Portable

Written in Java, iARM is portable to all systems with a supported JVM.

### 3.5. Robust

iARM is developed to be a dependable language. It performs compile time checks to detect all syntax and several semantic errors (i.e. color histogram configured while colors feature turned off). It also provides clear and understandable run-time errors (i.e. can't open the image file).

## 4. Implementation Issues

Any language that deals with images needs to understand image formats. For simplicity sake, iARM is limited to an uncompressed standard image format and supports only color images.

Several hundred-research papers (i.e. [5]) were written in last decade or so suggesting improvements to the original association rule-mining algorithm APRIORI, presented by Aggrawal et al [2]. iARM implements the original APRIORI algorithm and ignores the proposed optimizations because of time limitations.

## 5. Summary

iARM is a simple, easy to learn language that facilitates mining association rules from images. It can be customized according to the end-user needs and can extract versatile rules utilizing both textual and signal features. Extracted rules represent implicit knowledge contained in images, and implicit relationship that exists in a set of images. This knowledge could further assist in classification and clustering of images.

## 6. References


***1. Image Mining: Trends and Developments***
*http://www.comp.nus.edu.sg/~whsu/publication/2002/JIIS.pdf*

***2. Fast Algorithms for Mining Association Rules***
*http://citeseer.ist.psu.edu/agrawal94fast.html*

***3. Association Rules for Symbolic Indexing of Still Images***
*http://www-clips.imag.fr/mrim/User/hatem.haddad/PUBLICATIONS/ICIA.pdf*

***4. Discovering Association Rules based on Image Content***
*http://www.comp.nus.edu.sg/~ssung/research/readings/download_by_sp/ordonez99discovering.pdf*

***5. A fast APRIORI implementation [FAI]***
*http://citeseer.ist.psu.edu/cache/papers/cs/31223/http:zSzzSzSunSITE.Informatik.RWTH-Aachen.dezSzPublicationszSzCEUR-WSzSzzSzVol-90zSzbodon.pdf/bodon03fast.pdf*