Association Rule Mining in The Wider Context of Text, Images and Graphs

Frans Coenen
Department of Computer Science, The University of Liverpool
Liverpool L68 3BX, UK

ABSTRACT

Association Rule Mining(ARM) is now a well establish data mining tool. In this short paper a review of the wider application of ARM, beyond the processing of simple tabular datasets, is presented concentrating on its application to Text, Image and Graph Mining.

1. Introduction

Association Rule Miming (ARM) is a well understood technology within the field Data Mining. ARM, first popularised in the work of Agrawal et al. (1993) is concerned with the identification of relationships between items (attributes) in binary valued datasets. The relationships are expressed as Association Rules (ARs) of the form $X \rightarrow Y$, "read as if X exits then Y is also likely to exist", where X and Y are disjoint subsets of the global set of items present in the data set. An AR is therefore formed from an itemset I of cardinality k where $I=X \cup Y$ and $k>2$.

So as to broaden the scope of ARM work has been done on variations of the ARM concept such as weighted ARM, incremental ARM, unique pattern mining, ordinal ARM etc. Work has also been conducted to extend ARM know how into the fields of Classification Association Rule Mining (CARM) and distributed ARM. Again much of this work is now well established. Work on producing faster ARM algorithms is also still in progress, however the advantages to be gained by reducing execution times by micro-seconds is debatable. Much greater benefit is likely to accrue from research on the wider refinement and application of ARM. In this short paper a review is presented of some of the wider applications of ARM. In particular the application of ARM techniques to data sets other than "classical" tabular data sets such as document and image collections (text and image mining), and sets of graphs (graph mining).

2. Text Mining

Text mining is concerned with the application of data mining techniques to document collections. The objective of text ARM is to find patterns and relationships linking documents within the collection. Extensions of text mining include its application to WWW content mining (as opposed to WWW usage mining) and email analysis.

The most common (traditional) way of representing documents for text ARM purposes is the Vector Space Model (VSM) where documents are represented as a single, high dimensional, numeric vector d (where d is a subset of some vocabulary V). In effect each vector d is a transaction. The real issue is the composition of the vocabulary. There are a number of mechanism whereby the desired vocabulary used in the vector representation may be generated, examples include: the *bag of words* representation, the *bag of phrases* representation, *N-grams*, etc. The bag or words/phrase approaches are discussed in some more detail here.

In the bag of words approach the entire document based is represented as a collection of words with individual documents represented as a sub-set of this collection. Note that in the bag of words approach the ordering of words within documents as well as the structure of the documents is lost. The problem

with the bag of words approach is how to select a limited number of words from the entire set representing the document base. There are a number of mechanisms where by the number of words in the "bag" may be reduced, for example we might make use of *Stop Lists*, use *Stemming* or *Word Normalisation* technology, or adopt some strategy to collapse *Synonyms*.

Another approach is to allow V to comprise a pre-determined set of key words or named entities (a process sometimes referred to as *document indexing*), the issue is then how best to identify these entities. Various machine learning approaches have been used to identify keywords in text documents (see Sebastiani 2002).

An alternative to the bag of words approach is the bag of terms (phrases) approach. Terms in this case represent ordered combinations of N words (*N-wordgrams*) appearing one after the other (i.e. with no *word gap*) or appearing in sequence with some minimal word gap. In Lewis (1992) and Scott and Matwin (1999) a sequence of experiments were described comparing the bag of keywords approach with the bag of phrases approach in the context of text categorisation. The expectation was that the phrase based approach would work better than the keyword approach, because a phrase carries more information, however the reverse was discovered. In Sebastiani (2002) a number of reasons for this phenomena are presented:

1. Phrases have inferior statistical properties.
2. Phrases have lower frequency of occurrence (than keywords).
3. The bag of phrases includes many redundant and/or noise phrases.

It is hypothesised here that these drawbacks can be overcome by the use of appropriate CARM algorithms. It is clear that phrases will be found in fewer documents than corresponding key words, but conversely we expect them to have a greater discriminating power. To take advantage of this, we require algorithms that will identify classification rules with relatively low applicability as well as very common ones. To avoid problems of noise, conversely, we require the ability to discard rules that fall below defined thresholds of validity. These requirements point us to the use of CARM algorithms to construct classification rules using the identified words and/or phrases.

A possible phrase identification approach is to proceeds as follows, for each document in the training set:

1. Remove *common* words, i.e. words that are unlikely to contribute to a characterisation of the document.
2. Remove *rare* words, i.e. words that are unlikely to lead to generally applicable classification rules.
3. From the remaining words select those *significant* words that serve to differentiate between classes.
4. Generate significant phrases from combinations of the identified significant words and associated words.

Common and rare words are collectively considered to be *noise* words. These can be identified by their *support* values, i.e. the percentage of documents in the training set in which the words appear. Common words are words with a support value above a user defined Upper Noise Threshold (UNT), which we refer to as Upper Noise Words (UNW). Rare words are those with a support value below a user defined Lower Noise Threshold (LNT), and are thus referred to as Lower Noise Words (LNW). The UNT must of course exceed the LNT value, and the distance between the two values determines the number of identified non-noise words and consequently, if indirectly, the number of identified phrases. A phrase, in the context of CARM represents a possible attribute of a document which may be a component of the antecedent of rules. Experiments indicate that the majority of words occur in less than 1% of documents, so LNT must be set at a low value so as not to miss any potential significant words. Relatively few words are common (appearing in over 50% of documents).

The above word categories can then be combined in a number of different manners to construct phrases. Four different schemes for creating phrases, defined in terms of rules describing the content of phrases and the way in which a phrase is delimited, are suggested here (Table 1). In all cases, a phrase is required to include at least one significant word:

Delimiters	Contents
Stop marks and noise words	Sequence of one or more significant words and ordinary words.
	Sequence of one or more significant words and ordinary words replaced by “wild cards".
Stop marks and ordinary words	Sequence of one or more significant words and noise words.
	Sequence of one or more significant words and noise words replaced by “wild cards".

Table 1: *Phrase generation strategies*

3. Image Mining

Image ARM is concerned with the application of ARM techniques to image sets. There are many reasons why we may wish to analyse collections of images. Common example application areas where data mining techniques have been applied to image sets include medical analysis, meteorology and oceanography. These application areas have been addressed in a number of different manners but all include the recasting of the image set into a structured form that will facilitate data mining using established processes (in many cases the representation includes meta-data).

The challenge of applying ARM to image data is thus the transposing of the image data into a form that; (a) allows it to be used with ARM (i.e. an attribute format), and (b) limits the overall number of attributes to a manageable size. The image analysis and retrieval community have undertaken a significant amount of work in this area and established a body of work which the project team will be able to draw on. In this context it is worth noting that the requirements for image mining and image retrieval are not identical. In the case of image mining where (say) we wish to produce a classifier for a limited number of predefined classes (typically no more than 20) or cluster into a finite set of groups (again typically no more than 20) it is conjectured here that a much coarser image representation will suffice. In addition it is worth remembering here that data mining can and is intended to work with noisy data and thus the representation can be relatively crude.

For general image ARM a framework is required that will allow the representation of images in a transaction format. An obvious start point for the proposed research on image representation is to use a vector representation, such as that popularized in the field of text mining, where by each image is represented by a single numeric vector with the elements of the vector representing some global set of image features. This of course then raises the question of what image features we wish to capture. A simplistic approach is to tessellate the image base down to some predefined resolution with each tile having (say) a colour or a luminance associated with it. Each attribute in the representation then represents a particular pairing (tile identifier and colour/luminance). The tesseral approach, well understood in the field of image analysis and retrieval, has two principal disadvantages: (i) to capture all significant features a high resolution is required which means that large uniform areas of the image are unnecessarily represented by many tessellations and (ii) the representation favours features that run parallel to the X and Y axes. An alternative approach is to use a *quad tree* representation which allows different parts of the image to be tessellated down to different levels of resolution according to the varying degrees of uniformity of different parts of the image. Each quadtree can then be serialized into a

vector format. This addresses the first of the above disadvantages but not the second (it also introduces an additional overhead associated with the decoding of the serialization during the data mining process).

A general objection to the methods outlined above, from the perspective of data mining, is that these representations fail to capture higher-level structural properties of images that are likely to be relevant in ARM. An alternative approach is to represent images as "blobs" as advocated by the Blob World project (Carson et al 1997). Blobs in this context are simple groupings of pixels that share some uniform feature. The Blob World project was concerned with content-based image retrieval, however, in the context of data mining the concept of blobs is an interesting one with great potential. It is suggested here that instead of "blobs" image primitives, arranged in a lattice according to some spatial connectivity, are used --- the problem then becomes a graph mining problem.

Some work on image ARM has been undertaken, for example Ordonez and Omiecinski (1999) describe a partition based Apriori algorithm to find ARs based on image content from a simple image set comprising geometric shapes. Segmentation results are used from the Blob-World *region based query system*. The global set of segmented regions represent a set of attributes (each image in the dataset is then a record). Criticism of this approach is that it is not clear how it relates to the real world.

Ding et al. (2002) extract ARs from remote sensed imagery by considering ranges of the spectral bands to be attributes and pixels to be transactions. They also include *auxiliary* information at each pixel location such as crop yield. Criticism of the approach is that analysis at the pixel level is to fine a resolution.

Haiwei Pan, Jianzhong Li and Zhang Wei (2005) describe an ARM application for Computerized Tomography (CT) images. Images are described in terms ROIs (Regions Of Interest), objects, presented in a tabular form for mining. ROIs are identified using a *water immersion segmentation* algorithm. Attributes represent objects and other information. Identical objects may appear in the same image and across a number of images. The ARM is carried out in an Apriori manner, however the authors do not explain what they mean by identical objects, and the whole approach seems a bit simplistic (even though this is quite recent work).

4. Graph Mining

Advocates of graph mining argue that complex data has structure regardless of whether we are talking about traditional data tables, documents, WWW pages, bio informatics, images, etc. This structure can be represented in various forms of graphs/trees/lattices. Graph ARM aims at discovering interesting patterns in tree/graph structured data. Graph mining, and especially mining for frequent patterns in graphs, can be categorised as follows:

Transaction Graph Mining where the dataset comprises a collection (forest) of small graphs (trees) such that each graph/tree is considered to be a "transaction". The goal is then to find frequent patterns that exist across the transactions (see for example Inokuchi et al. 2000 or Zaki 2005).
Single Graph Mining where the data set comprises a single large graph. The objective is then to discover frequently occurring patterns within this single graph (see for example Borglet and Berhold 2002 or Vanetik et al. 2002).

The ARM issue, with respect to graph mining, is again how to represent the data in a tabular format so that it can be mined. A popular representation is to use adjacency matrices. An adjacency metric in this context is a NxN matrix where N represents the number of possible graph nodes (vertices). The available nodes are ordered according to some labeling and ranged along the matrix axies. The intersections in the

matrix, the attributes in the context of ARM, are set to 1 if a pair of nodes are linked by an edge and 0 otherwise. This representation is used in, for example, Inokuchi et al. (2000) who process the resulting matrices in an Apriori like manner.

An alternative is to use some sort of coding to label vertices and edges in graphs. This is done by Zaki (2005) in the TreeMiner algorithm. Zaki represents graphs using a numbering system based on level in the tree and a sibling ordering (mining is founded on the equivalence class concept). A similar approach is used in Borglet and Berhold (2002) who process their graph using an ECLAT style algorithm with candidate graphs generated using information from sibling nodes.

Another idea is to define graphs according to collections of attributes that represent unique atomic sub-graphs. For example Vanetik et al. (2002) describe a single graph mining system where each transaction is a node (vertex) in the graph which may appear on one or more paths. Vanetik et al. process this representation in an Apriori manner with candidate sets generated level by level.

There is a lot of potential in graph mining given its applicability to structured data of all kinds. An extension to graph mining is the mining of data contained in a class hierarchy. This was first addressed in Srikant and Agrawal (1995) who described an algorithm to produce what the authors describe as a *generalised association rule*s, by which they mean ARs generated from a set of attributes arranged in one or more class hierarchies. Each node in such a hierarchy represents some class of object, e.g. shoes are a type of footware which is a type of clothing, etc. This structure can be utilized in the AR generation process given the following observations:

1. The support for any attribute (class) also contributes to the support for all its parent classes. However care must be taken that the support for a parent node is not counted twice. A record may contain (say) two attributes that are both child nodes of a single ancestor node --- in which case the support of the ancestor node should not be incremented twice!
2. Interesting high confidence rules may be discovered amongst rules near the root of the hierarchy that do not exist at the leaf nodes.
3. If a rule $X \rightarrow Y$ is generated with above minimum confidence then all rules $X \rightarrow Z$ will also be valid (where Z is any combination of the parent classes of Y).
4. It is also possible to prune rules where an alternative node involving ancestors of the rule's attributes exist. Generally speaking a rule is interesting if: (i) it has no ancestors, or (ii) it is "R-interesting with respect to its close ancestors among its ancestor nodes" (the terms "R-interesting" and "close ancestor" are also defined in Srikant and Agrawal.

Note that the approach is only appropriate where attributes can be arranged in a hierarchy such as in the case of shopping basket analysis but merits further exploration.

5. Conclusions

In this short paper three application areas have been presented which may prove fruitful in the application of current ARM technology. In particular the application domains of text, image and graph mining have been considered in some detail. In all cases the fundamental challenge is how best to translate the input data into a tabular format that will permit the application of ARM technology.

References

1. Agrawal, R, Imielinski, T. and Swami, A. (1993). *Mining association rules between sets of items in large databases*. Proc. ACM SIGMOD Conference on Management of Data, pp 207-216.
2. Agrawal, R. and Srikant, A. (1994). *Fast algorithms for mining association rules.* Proc. VLDB'94, pp 487-499.
3. Borglet, C. and Berhold, M.R. (2002). *Mining Molecular Fragments: Finding Relevant Substructures of Molecules.* Proc ICDM'02, IEEE, pp 51-58.
4. Ding, Q., Ding, Q. and Perrizo, W. (2002). *Association Rule Mining on remotely sensed images using P-trees.* Proc PAKDD. pp 66-79.
5. Han, J., Pei, J. and Yiwen, Y. (2000). *Mining Frequent Patterns Without Candidate Generation.* Proceedings ACM-SIGMOD International Conference on Management of Data, ACM Press, pp 1-12.
6. Inokuchi, A., Washio, T., and Motoda, H. (2000). *An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data.* Proc. PKDD'2000, LNCS 1910, pp13-23.
7. Lewis, D.D. (1992). *An evaluation of phrasal and clustered representations on a text categorization task.* Proc ACM Int. Conf. on Research and Development in Information retrieval (SIGIR-92), pp 37-50.
8. Ordonez, C. and and Omiecinski, E. (1999). *Discovering Association Rules Based on Image Content.* Proc. IEEE Advances in Digital Libraries Conference, pp38-49.
9. Pan, H., Li, J. and Wei, Z. (2005). *Mining Interesting Association Rulers in Medical Images.* Proc. ADMA'05, LNAI 3584, Springer-Verlag, pp598-609
10. Scott, S. and Matwin, S. (1999). *Feature Engineering for Text Classification.* Proceedings of ICML-99, 16th International Conference on Machine Learning, pp 379-388.
11. Srikant, R. and Agrawal, R. (1995). *Mining Generalised Association Rules*. Proc. of the 21st International Conference on Very Large Data Bases, VLDB'95. pp 407-419.
12. Vanetik, N. Gudes, E. and Shimony, S. (2002). Computing Frequent Graph Patterns from Semistructured Data. Proc. ICDM'2002, pp458-465.
13. Zaki, M.J. (2005). *Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications.* IEEE Transactions on knowledge and Data Engineering, Vol. 17, No. 8, pp 1021-1035.