Chapter 5:

Automatic Metadata Generation and Digital Cultural Heritage

Iliya Mitov, Benoit Depaire, Krassimira Ivanova, Koen Vanhoof, Dimitar Blagoev

1 Automatic Generation of Metadata

The studies presented in [Shreve et al, 2003] and [English et al, 2002] show that the use of metadata can significantly improve the quality of revealing resources. Metadata can help search engines and people to distinguish relevant from non relevant documents in the process of information extraction. Addressing this challenge attracts more research in automatic metadata generation. The proposed approaches can be categorized into two major subcategories: *harvesting* and *mining* (*extraction*) of metadata [Greenberg, 2004].

Harvesting of metadata is a process of automatic extraction of predefined fields. In the collection process relies on metadata produced by humans or semi-automatic processes, with appropriate application software. For example, Web editing software (e.g. Microsoft Publisher) automatically produces metadata when creating or updating a resource of type "format", "creation date", "date updated". Microsoft Word also automatically fills in the administrative metadata for the document type of "Created", "Modified", "Accessed", "Author", "Company", "Word count", "Pages", etc., taking information from the computer, where the document is edited, or from the content of the document. The harvesting process is usually accomplished by creating an analyzer of the source of metadata using pre-defined grammar and transformation results in a certain format (standard) using the comparison rules. Examples of harvesting are the processes assuring interoperability of metadata from various systems and

platforms [Martines and Morale, 2002] and extraction of metadata from non-cooperating digital libraries [Shi et al, 2003].

Extraction of metadata occurs when an algorithm automatically extracts metadata from the content of the resource. Sources for the extraction of metadata can be grouped mainly in: document content analysis, document context analysis, document usage, and composite document structure [Cardinaels et al, 2005].

The most commonly used approaches are based on *Regular Expressions* (RegEx), *Rule-based Parsers* and *Machine Learning Algorithms*.

1.1 Regular Expressions

A large class of problems for the extraction of data can be covered using carefully constructed *regular expressions*. Typical examples of such suitable for automatic extraction elements are e-mail addresses, phone numbers, credit card numbers, social security numbers, etc. In the context of information retrieval, preliminary work has concentrated on teaching the regular expressions using scripts with relatively small and usually their training is done through steps such as parsing (Part-Of-Speech: POS tagging), morphological analysis and set directory (called Gazetteer Matching) [Ciravegna, 2001].

In 2004 a team of Prof. William Cohen from Carnegie Mellon University starts creating collection of classes for storing text, annotating text, and learning to extract entities and categorize text called MinorThird [Cohen, 2004]. It contains a number of methods for learning to extract and label spans from a document, or learning to classify spans (based on their content or context within a document). The creating of such collections is a useful tool not only for the particular investigation support, but also for creating common notion for the area as a whole.

1.2 Rule-based Parsers

A classic example of *rule-based systems*, are expert systems, which use rules to make inferences or choice. In the field of natural language texts processing these systems are widely used in the commission of lexical analysis. Programming, based on the rules, tries to retrieve instructions for execution, using a starting set of data and rules. A typical system based on rules, consists of four main components: (1) List of rules, which is a specific form of knowledge base; (2) Inference rule, which determines what action to take based on the interaction of input and on the basis of rules; (3) Ad-hoc working memory; (4) User interface, or other communication with the outside world.

Rule-based systems [Mao et al, 2004] [Bergmark, 2000] use programmed instructions that specify how to retrieve information from the target documents. With a sufficiently powerful language these techniques are able to extract high-quality metadata. However, heterogeneity leads to the need for complex set of rules for setting and testing a lot of time [Klink et al, 2000]. Analogy with typical software metrics to assess the complexity suggests that a linear increase in the number of rules grow more complex. In such cases, even well-trained team of experts in the subject area and writing makes it difficult to cope with the unequivocal support of a heterogeneous collection of rules. In places, where the documents usually have a fixed structure, such as automatic generation of metadata from electronic documents in an organization, they are favourites.

1.3 Machine Learning Algorithms

Machine learning is the direction in artificial intelligence that deals with algorithms and methods for automatic learning of rules, signs and characteristics by which a computer program can take appropriate decisions. Any learning process is based on the data set (training set), of which the program learns. Machine learning methods are most frequently used methods for extracting metadata because they are reliable and adaptable. Theoretically can be used on any number of documents, but the generation of the training sample may take some time and money. Unlike the machine learning methods, regular expressions and rules based systems are directly applicable. But they are closely tied to the specific application area and require the presence of experts from the field to determine the rules or regular expressions. These are the main reasons these methods are relatively limited use.

2 Data Mining

Data mining is an essential part of whole global process of knowledge discovery [Fayyad et al, 1996]. Data Mining concerns application, under human control, which in turn is defined as algorithms designed to analyze data, or to extract patterns in specific categories from data, while Knowledge Discovery is a process that seeks new knowledge about an application domain [Klosgen and Zytkow, 1996]. This process consists of many steps among with one of them being data mining.

The Knowledge Discovery process was defined by many authors. For instance [Fayyad et al, 1996] define it as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". [Friedman, 1997] considers the process of knowledge discovery as an automatic exploratory data analysis of large datasets.

The process has formed by different phases, which iteratively interact with each other. During the years, several models are proposed (for instance in [Fayyad et al, 1996]). Generally, the process of knowledge discovery can be divided into following phases [Han and Kamber, 2006]:

1. Data cleaning (to remove noise and inconsistent data).
2. Data integration (where multiple data sources may be combined).
3. Data selection (where data relevant to the analysis task are retrieved from the database).
4. Data transformation (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance).
5. Data mining (an essential process where intelligent methods are applied in order to extract data patterns).
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on some interestingness measures).
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present the mined knowledge to the user).

Data Mining is the process of analyzing a large set of raw data in order to extract hidden information which can be predicted. It is a discipline, which is at the confluence of artificial intelligence, data bases, statistics, and machine learning. The questions related to data mining present several aspects, the main being: classification, clustering, association and regularities. Technically, data mining is the process of analyzing data from many different dimensions or sides, and summarizing the relationships identified [Kouamou, 2011].

The data mining methods are divided essentially in two main types [Maimon and Rokach, 2005] (Figure 56):

- Verification-oriented (the system verifies the user's hypothesis);
- Discovery-oriented (the system finds new rules and patterns autonomously) [Fayyad et al, 1996].

Verification methods deal with the evaluation of a hypothesis proposed by an external source. These methods include the most common approaches of traditional statistics, like *goodness-of-fit test*, *t-test of means*, and *analysis of variance*. Such methods are not usually associated with data mining because most data mining problems are concerned with the establishment of a hypotheses rather than testing a known one.

Most of the *discovery-oriented techniques* are based on inductive learning [Mitchell, 1997], where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples.

The underlying assumption of the inductive approach is that the trained model is applicable to future unseen examples.

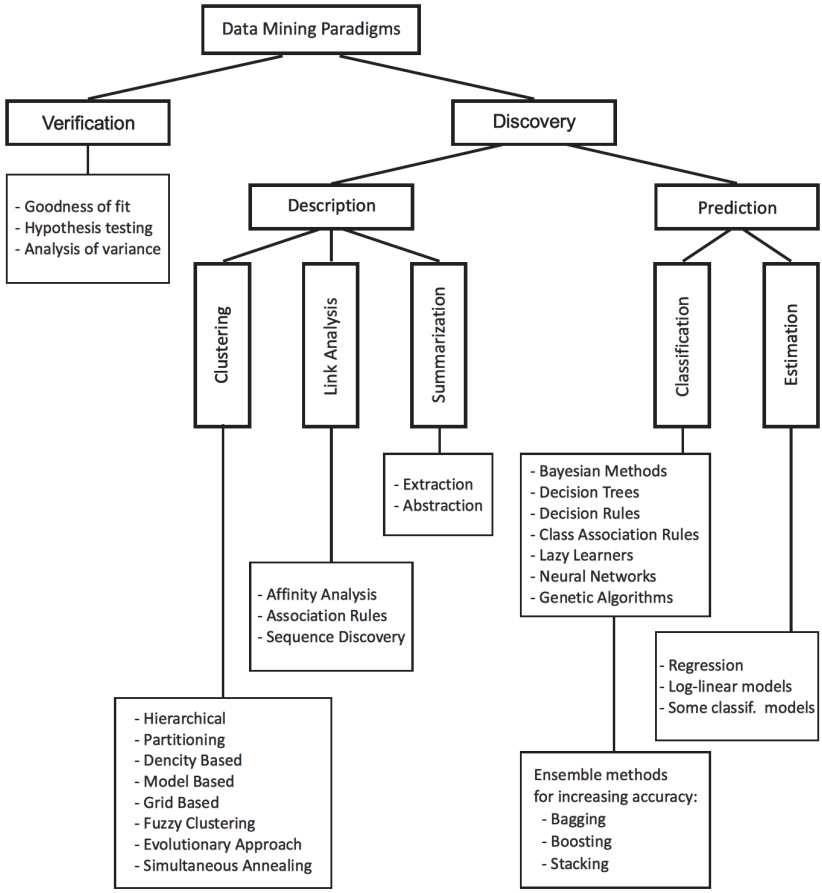

Figure 56. The taxonomy of data mining methods

Discovery methods are methods that automatically identify patterns in the data. The discovery method branch consists of *description methods* versus *prediction methods*.

Description-oriented data mining methods focus on understanding how the underlying data operates. The main orientations of these methods are clustering, summarization and visualization.

The main directions of description-oriented methods are *clustering*, *link analysis* and *summarization*.

Clustering is the process of grouping the data into classes or clusters, so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters. Dissimilarities are assessed based on the attribute values describing the objects, using different kinds of distance measures.

Link Analysis uncovers relationships among data. It is used for 3 primary purposes [Berry et al, 2004]: (1) Find matches in data for known patterns of interest; (2) Find anomalies where known patterns are violated; and (3) Discover new patterns of interest. In this direction falls such disciplines as affinity analysis, association rules mining, and sequence discovery.

Summarization is the process of reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. There are two main methods for this: *extraction* and *abstraction*. *Extractive methods* work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, *abstractive methods* build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original.

Prediction-oriented methods aim to build a behavioural model that can create new and unobserved samples and is able to predict the values of one or more variables related to the sample. Here two main branches are gained: *classification* and *estimation*. These two forms of data analysis are used to extract models describing important data classes or to predict future data trends. The main difference between classification and estimation is that classification map the input space into predefined classes, while estimation models map the input space into a real-valued domain.

Classification models predict categorical (discrete, unordered) labels. The classification is the problem of identifying the sub-population to which new observations belong, where the identity of the sub-population is unknown, on the basis of a training set of data containing observations whose sub-population is known. The new individual items are placed into groups based on quantitative information on one or more measurements, traits or characteristics, etc.), and based on the training set in which previously decided groupings are already established. There are several big groups, in which classifiers belongs: Bayesian Methods, Support Vector Machines, Decision Trees, Decision Rules, Class Association Rules,

Lazy Learners, Neural Networks, and Genetic Algorithms. For increasing the received accuracy, upper technique for ensemble methods, or so-called meta-classifiers as upper stage is used.

Estimation models construct a continuous-valued function, or ordered value, which are used as predictor (estimator). The most common used technique are different kinds of regression models (involving single predictor variable or two or more predictor variables; linear or non-linear regression, etc.), while other models are also used (such as log-linear models that approximate discrete multidimensional probability distributions using logarithmic transformations). Some of the classifier models can also be tuned to be used for estimation (such as Decision Trees, Neural Networks, etc.) [Han and Kamber, 2006].

3 Data Extraction from Web Documents Using Regular Expressions

In recent years multiple machine learning approaches have been proposed for information extraction [Li et al, 2008]. A large class of entity extraction tasks can be accomplished by the use of carefully constructed regular expressions. Examples of entities amenable to such extractions include e-mail addresses, software names (web collections), credit card numbers, social security numbers (e-mail compliance), gene and protein names (bioinformatics), etc. With a few notable exceptions, there has been very little work in reducing this human effort through the use of automatic learning techniques.

In the context of information extraction, prior work has concentrated primarily on learning regular expressions over relatively small alphabet sizes and usually learning of regular expressions is done over tagged tokens produced by other text-processing steps such as POS tagging, morphological analysis, and gazetteer matching [Ciravegna, 2001].

[Rozenfield et al, 2008] propose approach, which use the immense amount of unlabelled text in the web documents in order to create large lists of entities and relations. Based on this approach the system SRES is a self-supervised web relation extraction system that learns powerful extraction patterns from unlabelled text, using short descriptions of the target relations and their attributes.

The proposed in [Li et al, 2008] learning algorithm ReLIE takes as input not just labelled examples but also an initial regular expression, which provides a natural mechanism for a domain expert to provide domain knowledge about the structure of the entity being extracted and meaningfully restriction of the space of output regular expressions.

Within the frame of the project, an approach for information extraction by learning restricted finite state automata from marked web documents created using Bulgarian language was developed and tested. The approach uses heuristics to generalize initial finite-state automata that recognizes only the positive examples and nothing else into automata that recognizes as larger language as possible without extracting any non-positive examples from the training data set. The proposed approach is a good base for building system from the class of Semi-Automated Interactive Learning (SAIL) systems [IBM, 2009].

3.1 Data Extraction by Learning Restricted Finite State Automata

The approach for information extraction by learning restricted finite state automata from marked web documents contains four main steps:

1. Setting up the hierarchical structure of the data to be extracted. Every element and sub-element which is to be identified has to be specified. The data structure is expressed as a tree of elements and their sub-elements.
2. Scanning and manual tagging the initial documents for the required information.
3. Extracting the examples for the different elements and building an initial parsing grammar.
4. Data extracting from new documents. The user can improve the accuracy of the results by manually correcting the annotations for a particular document and add it to the learning set.

The building of the parsing grammar consists of two sub-steps:

a) combining all positive examples;

b) generalizing the resulting tree into restricted finite state automata.

At the sub-step a) all marked instances of the structured data are flattened in strings containing the text of the main element with special symbols marking the beginnings and ends of the sub-elements and the HTML tags in the case when the text is a web document. Then all the flattened strings from all documents are combined in a single tree. This tree is then used as the initial finite state automata. It recognizes all learned positive examples without misrecognizing negative ones.

The sub-step b) is the generalization of the automata using heuristic methods for combining states and extrapolating the transitions' characters into predefined alphabets. After each generalization the automata is checked for consistency by re-scanning the learning texts and if the extracted data differs from the initial (manually annotated) data the modification is rolled back.

There are many ways in which the finite state automata can be generalized [Baltes, 1992]. To prevent the computational complications that arise from this condition we use restricted finite state automata. The building elements that are used in these automata are (Figure 57):

- Single character transition;
- Matching any character from a given class;
- Shadow transition that unites the output transitions' alphabets.

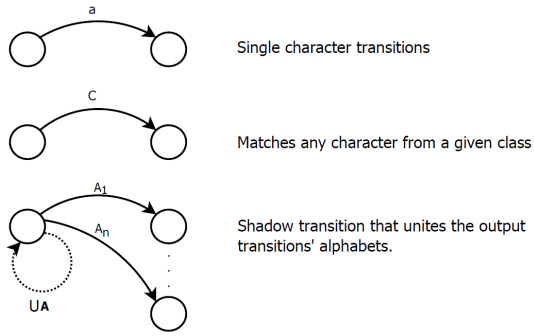

Figure 57. Elements of the restricted finite state automata [Baltes, 1992]

In addition to the automata generalization heuristics described later in the section the generalizator employs the use of a custom character class list. The class list specifies what characters belong to a given class and how many of them have to be present in a state's output transitions before class generalization is attempted. Table 5 shows one sample list which includes classes for both English and Bulgarian letters.

Table 5. Sample character class list

Min	Characters	Class
3	abcdefghijklmnopqrstuvwxyz	English lowercase
3	ABCDEFGHIJKLMNOPQRSTUVWXYZ	English capitals
3	Abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ	English
3	абвгдежзийклмнопрстуфхцчшщъьюя	Bulgarian lowercase
3	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian capitals
3	абвгдежзийклмнопрстуфхцчшщъьюя АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЮЯ	Bulgarian
1	0123456789	Digits
1	\b \t \n \r	White space characters
1	' " " „ " « » ` ' ,	Quotes

The generalization algorithm (Figure 58) in sub-step b) is done in the following way:

1. Class generalization (top-down) which tries to generalize as much as possible output transition characters for a given state into classes;
2. State merging (bottom-up) with character comparison tries to merge a state with one of its next possible states if the two states have identical characters and classes on their output transitions. If it is successful, the two states are merged into a single state and a shadow transition is added over the union of the other output transitions' alphabets. Further testing is made to find the upper repetition limit for the newly formed state;
3. State merging (bottom-up) without character comparison, essentially same as above except it does not require two states to have comparable characters and classes on their output transitions. If it is necessary, character transitions are merged with class transitions. This operation is more prone to making erroneous generalizations or one that block the further generalization of upper states therefore it is performed after the previous generalizations;
4. Character and class merging (bottom-up) tries to merge a character transition in a given state with a class or another character transition in the same state resulting in a transition over a new class which was not predefined in the classes list;
5. State skipping (bottom-up) which tests if all output transitions on a given state can be skipped thereby advancing onto all sub-states without matching any of the transitions. Every output transition to another state is complemented with an epsilon transition (one that matches the zero-length string) to the same sub-state.

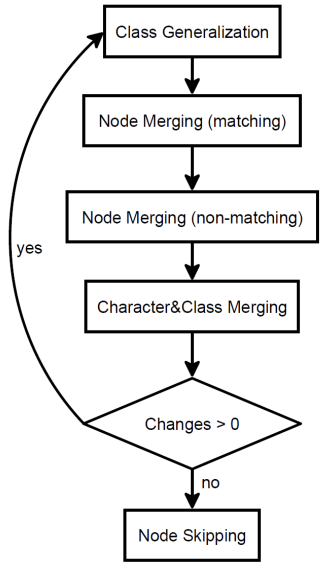

Figure 58. Generalization algorithm

After every change of the automata a test run is performed with the new automata over the learning data set. If the result differs from the initial ones the attempted transformation is rolled back. All generalization

and merging steps (without state skipping) are repeated until there are no more states which can be merged and/or generalized.

Once the module's learning phase is complete a parsing grammar is being generated. This grammar employs regular expressions for data extraction and generates structured XML output containing all elements found in the parsed documents. The sub-elements of the hierarchical data structure are encoded as named groups in the regular expressions. This grammar can then be used for performing background batch processing on a large number of documents or to analyze the produced regular expressions and make inferences for the structure of the elements of interest.

3.2 Program Realization

The presented algorithm has been realized in the experimental system InDES. The system contains two separate modules: a graphical user interface (GUI) and a command-line learning and extracting module. The GUI is developed in C#.NET and employs the embedded Internet Explorer browser component to display the web documents.

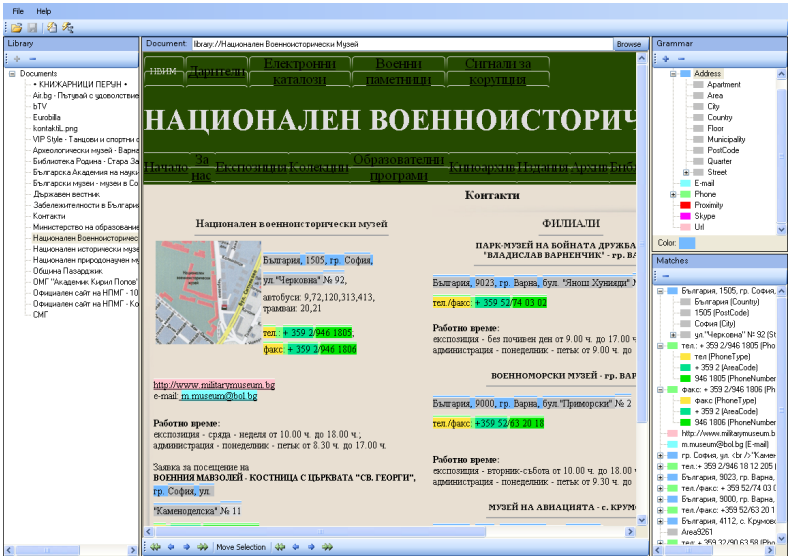

Figure 59. Screenshot of the program system InDES

The learner and extractor are written in C++ for increased performance and smaller memory footprint. Both modules use the same html pre-processing routine for cleansing the given web documents. The cleansing's purpose is to normalize or eliminate characters in the input document without changing the structure of the contained information or the way it appears on the screen. This includes but is not limited to Unicode character normalization where explicit character codes are replaced with their respective characters and JavaScript removal (since the current version of the system does not execute JavaScript prior to learning or extracting).

A screenshot of the GUI with a loaded web document is given on Figure 59. In the left, the sub-screen for selecting the web documents contain some already connected websites and corresponded documents. At the right the generated grammar and founded matches are shown. In the centre of the screen the current document with market texts is presented.

3.3 Experiments

We have made several types of experiments over different web documents in Bulgarian language for extracting elements such as addresses, phones and e-mails from randomly chosen companies' and social institutions' web pages containing contact information. To test the ability of the proposed method to extract new information we used a set of 100 pre-marked web documents in Bulgarian language for three types of elements: addresses, phones and e-mails with corresponded sub-elements. To simplify the experiments, the phone numbers and e-mails are taken as such elements.

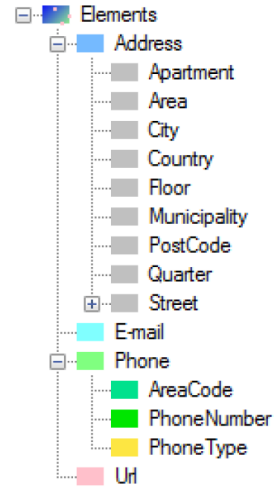

Figure 60. Sample element hierarchy for information extraction

The experiments were provided following the steps of the proposed approach.

At first step the hierarchical structure of the data to be extracted was set up as it is shown on Figure 60. The structure consists of:

- Addresses with sub-elements "Country", "Area", "Municipality", "City", "Post Code", "Quarter", "Street", "Floor" and "Apartment";

- Phones with sub-elements "Area Code", "Phone Number" and "Phone Type";
- E-mails without sub-elements.

At the next step the data set was chosen. For the purposes of the experiments, the web document set was created using web pages for five categories organizations: companies, schools, museums, municipalities and libraries. The documents were picked out in html format using Google possibilities. For each category were selected first twenty websites after searching for combination keywords "address" and one of keywords "company", "school", "museum", "municipality", "library" and with restriction "pages in Bulgarian".

Then, all documents from the data set was scanned and manually tagged in accordance with chosen hierarchical structure. Some of the documents are used later as instances in the learning set and other are used as instances in the testing set. At the first experiment we used ten-fold cross validation. At the second experiment the data set was split into learning set and testing set in random principle.

Since the task is to find and extract all data that represents a given element we tested the system using the following criteria:

- Recall – the percentage of manually annotated elements for which an overlapping element is found in the results of the search;
- Precision – the percentage of found elements that overlap with a manually annotated element [Taylor, 1982];
- Accuracy – the average similarity between the original annotated elements and the correctly extracted elements.

For a similarity measure we propose to set the ratio of the length of the overlapping to the length of the union of the original and extracted texts:

$$similarity = \frac{A \cap B}{A \cup B}$$

where *A* and *B* are the original and extracted line segments respectively.

3.3.1 Ten-fold Cross-validation Test

Because of the wide diversity in which those elements can occur we split the data into 10 parts and performed a 90% learn – 10% test evaluation testing once each part [Kohavi, 1995]. Table 6 shows the results for each of the three element types.

During generalization the number of states in the automata has been reduced on average by 71%, 72% and 90% for addresses, phone

numbers and e-mails respectively. The automata could be further compacted by merging common sub-trees.

Table 6. Results for extracting addresses, phones and e-mails without sub-elements

	Count	Recall	Precision	Accuracy
Address	134	51.54%	74.56%	57.25%
Phone	296	82.71%	87.45%	69.41%
E-mail	102	89.96%	97.29%	95.44%

This experiment shows the ability of the learning method to build generalized automata for parsing web documents. There appears to be a relation between the algorithm's performance and the structural variance of the information to be extracted.

3.3.2 Examination Trend of Reaching Satisfactory Results with Increasing the Cardinality of Learning Set

In other group of experiments the data set was split in two parts in a random principle – 40 instances were used as a learning set and the rest 60 documents were used as a testing set.

The system was learned using respectively 10, 20, 30 and 40 web documents from the learning set (each set contained the documents of the lower learning sub-set). Each time the testing was provided with all documents from the testing set. The test results were analyzed to obtain values for the numbers of fully extracted, partially extracted and elements that should have been but were not extracted. These experiments were provided in order to examine the trend of reaching satisfactory results. For each case multiple randomized runs have been performed to obtain more stable average values. We assume the address is recognized if its sub-elements, given in the text, are recognized. The telephone number is recognized if the system has recognized at least the phone code and the number. In several cases the system has recognized the string as a whole without recognizing its sub-elements. Partial recognizing means that some sub-elements are recognized (in particular one of them), but not the element as a whole. For instance only "town", "street", etc. Table 7 shows the obtained results.

Table 7. Precision for extracting addresses, phones and e-mails when learning sets were 10, 20, 30 and 40 documents respectively

Number of Learning Documents	Address	Phone	E-mail
10	45.33%	85.54%	93.30%
20	50.43%	81.17%	86.72%
30	54.24%	84.23%	88.73%
40	66.19%	85.57%	93.35%

Figure 61, Figure 62 and Figure 63 show the trend of increasing learning accuracy with increasing of the cardinality of learning set for elements and sub-elements of addresses and phones respectively.

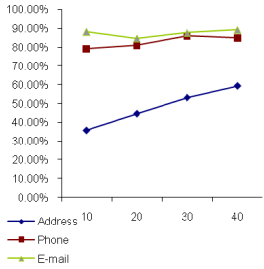

Figure 61. F-measure for addresses, phones and e-mails

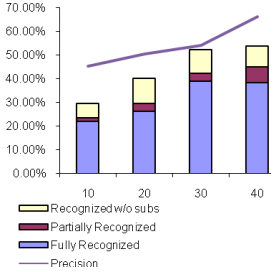

Figure 62. Recall and precision for the addresses

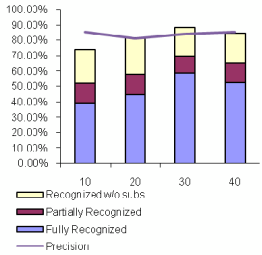

Figure 63. Recall and precision for the phones

In this experiment we found there is a trend for increasing the accuracy of the extractor by increasing the learning data set. As expected, e-mail addresses show the highest recall and precision and achieve high accuracy with a small cardinality of the learning set. The main reason for it is probably the existence of a strict and short structure for an e-mail address which leads to little variety in the different element instances. In that case learning over wider range of documents can actually sometimes prevent the optimal generalization resulting in worse results (Table 7). Bulgarian addresses show the worse results. Given the extremely wide variety of the indirect representation of Bulgarian addresses the results for this element are very promising. Furthermore, by increasing the learning and testing data sets the automata should begin to comprise of the most common cases which will lead to results comparable to the ones for the other two elements.

Our results are compatible with [Cohen, 2004]. The differences are coming from different languages and different grammars' structure in the languages.

The proposed approach, developed system InDES and provided experiments show that this is suitable to cover practical needs for automatic data extraction from web documents created using Bulgarian language.

4 ArmSquare: an Association Rule Miner Based on Multidimensional Numbered Information Spaces

Data mining stands at the crossroad of databases, artificial intelligence, and machine learning. Association rule mining (ARM) is a popular and well researched method for discovering interesting rules from large collections of data. Association rule mining has a wide range of applicability, such as market basket analysis, gene-expression data analysis, building statistical thesaurus from the text databases, finding web access patterns from web log files, discovering associated images from huge sized image databases, etc.

The contemporary databases are very large, reaching giga- and terabytes, and the trend shows further increase. Therefore, for finding association rules one requires efficient scalable algorithms that solve the problem in a reasonable time. The efficiency of frequent itemset mining algorithms is determined mainly by three factors: (1) the way candidates are generated; (2) the data structure that is used; and (3) the implementation details. Most papers focus on the first factor, some describe the underlying data structures, and implementation details are almost always neglected [Bodon, 2003].

The description of the problem of association rule mining is firstly presented in [Agrawal et al, 1993]. Let $\mathbf{D}$ be a set of items, then $X = \{i_1,...,i_k\} \subseteq \mathbf{D}$ denotes an itemset or a k-itemset. A transaction over $\mathbf{D}$ is a couple $T = (tid, I)$ where tid is the transaction identifier and I is an itemset. A transaction $T = (tid, I)$ is said to support an itemset $X \subseteq \mathbf{D}$ if $X \subseteq I$.

A transactional database D over $\mathbf{D}$ is a set of transactions over $\mathbf{D}$. The cover of an itemset X in D consists of the set of transaction identifiers of transactions in D that $support\ X = cover(X,D) = \{tid \mid (tid, I) \in D, X \subseteq I\}$. The support of an itemset X in D is the number of transactions in the cover of X in D, i.e. $support(X,D) = |cover(X,D)|$. Note that $|D| = support(\{\},D)$. An itemset is called frequent if its support is no less than a given absolute minimal support threshold $MinSup$, with $0 \le MinSup \le |D|$.

An association rule is an expression of the form $X \Rightarrow Y$, where X and Y are itemsets, and $X \cap Y = \{\}$. X is called the body or antecedent, and Y is called the head or consequent of the rule. The support of an association rule $X \Rightarrow Y$ in D, is the support of $X \cup Y$ in D and can be

interpreted as a measure of evidence that the rule $X \Rightarrow Y$ is real and not a noise artefact. The confidence or accuracy of an association rule $X \Rightarrow Y$ in D is the conditional probability of having Y contained in a transaction, given that X is contained in that transaction, i.e. $confidence(X \Rightarrow Y, D) = P(Y \mid X) = \frac{support(X \cup Y, D)}{support(X, D)}$. The rule is called confident if $P(Y \mid X)$ exceeds a given minimal confidence threshold $MinConf$.

4.1 A Brief Overview of Previous ARM Algorithms

The main pillar of ARM-algorithms is Apriori [Agrawal and Srikant, 1994]. It is the best-known algorithm to mine association rules, which uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. Over the years, a lot of improvements of Apriori, supported with different types of memory structures, are proposed.

Recent ARM-algorithms, based on graph mining can be roughly classified into two categories. The first category of algorithms employs a breadth-first strategy. Representative algorithms in this category include AGM [Inokuchi et al, 2003] and FSG [Kuramochi and Karypis, 2001]. AGM finds all frequent induced sub-graphs with a vertex-growth strategy. FSG, on the other hand, finds all frequent connected sub-graphs based on an edge-growth strategy. Algorithms in the second category use a depth-first search for finding candidate frequent sub-graphs. A typical algorithm in this category is gSpan [Yan and Han, 2002], which was reported to outperform both AGM and FSG in terms of computation time.

A different approach for association rule searching is used in ECLAT [Zaki et al, 1997]. It is the first algorithm that uses a vertical data (inverted) layout. The frequent itemsets are determined using sets of intersections in a depth-first graph.

In graph ARM approaches the bottleneck is the necessity of performing many graph isomorphism tests. To overcome this problem, alternative approaches use hash-based techniques for candidate generation. The representatives in this direction are DHP [Park et al, 1995] based on direct hashing and pruning, [Özel and Güvenir, 2001] which proposed the use of perfect hashing, and IHP [Holt and Chung, 2002] that uses inverted hashing and pruning.

FP-Tree [Han and Pei, 2000], Frequent Pattern Mining is another milestone in the development of association rule mining, which breaks the main bottlenecks of the Apriori. FP-tree is an extended prefix-tree

structure storing quantitative information about frequent patterns. The tree nodes are arranged in such a way that more frequently occurring nodes will have better chances of sharing nodes than less frequently occurring ones. The efficiency of FP-Tree algorithm has three reasons: (1) FP-Tree is a compressed representation of the original database; (2) it only scans the database twice; (3) it uses a divide and conquer method that considerably reduces the size of the subsequent conditional FP-Tree. The limitation of FP-Tree is its difficultness to be used in an interactive mining system, when a user wants to expand the dataset or change the threshold of support. Such changes lead to repetition of the whole mining process.

The Hmine algorithm [Pei et al, 2001] introduces the concept of hyperlinked data structure "Hyper structure" and uses it to dynamically adjust links in the mining process. Hyper structure is an array-based structure. Each node in a Hyper structure stores three pieces of information: an item, a pointer pointing to the next item in the same transaction and a pointer pointing to the same item in another transaction.

The innovation brought by TreeProjection [Agarwal et al, 2000] is the use of a lexicographical tree which requires substantially less memory than a hash tree. The number of nodes in its lexicographic tree is exactly that of the frequent itemsets. The support of the frequent itemsets is counted by projecting the transactions onto the nodes of this tree. This improves the performance of counting the number of transactions that have frequent itemsets. The lexicographical tree is traversed in a top-down fashion. The efficiency of TreeProjection can be explained by two main factors: (1) the transaction projection limits the support counting in a relatively small space; and (2) the lexicographical tree facilitates the management and counting of candidates and provides the flexibility of picking efficient strategy during the tree generation and transaction projection phrases.

Other data structure that is commonly used is a "trie" (or prefix-tree). Concerning speed, memory need and sensitivity of parameters, tries were proven to outperform hash-trees [Bodon and Ronyai, 2003]. In a trie, every node stores the last item in the itemset it represents its support and its branches. The branches of a node can be implemented using several data structures such as hash table, binary search tree or vector.

Another algorithm for efficiently generating large frequent candidate sets, which use different data structures, is Matrix Algorithm [Yuan and Huang, 2005]. The algorithm generates a matrix with entries 1 or 0 by passing over the cruel database only once, and then the frequent candidate sets are obtained from the resulting matrix. Finally association

rules are mined from the frequent candidate sets. Experiment results confirm that the proposed algorithm is more effective than the Apriori.

The Morishita & Sese framework [Morishita and Sese, 2000] efficiently compute significant association rules according to common statistical measures such as a chi-squared value or correlation coefficient. Because of anti-monotonicity of these statistical metrics, Apriori algorithm is not suitable for associative rule generation. They present a method of estimating a tight upper bound on the statistical metric associated with any superset of an item-set, as well as the novel use of the resulting information of upper bounds to prune unproductive supersets while traversing item-set lattices.

This short overview of available algorithms and used structures shows the variety of decisions in association rule mining. As we can see graph structures, hash tables, different kind of trees, bit matrices, arrays, etc., are used for storing and retrieving the information.

Each kind of data structure brings some benefits and bad features. Such questions are discussed for instance in [Liu et al, 2003] where the comparison between tree-structures and arrays is made. Tree-based structures are capable of reducing traversal cost because duplicated transactions can be merged and different transactions can share the storage of their prefixes. But they incur high construction cost especially when the dataset is sparse and large. Array-based structures incur little construction cost but they need much more traversal cost because the traversal cost of different transactions cannot be shared.

4.2 Association Rule Miner ArmSquare

Here we offer one approach, which is focused on proposing appropriate coding of the items in database in order to use the possibilities of direct access to the information via coordinate vectors into multidimensional numbered information spaces. This structure combines the convenience of the work with array structures with economy and performance of tree structures, which lies in the ground of realized access method. The algorithm of obtaining association rules is very simple, we focus our attention over the possibilities of using such structures for storing information in data mining systems. In future more smart algorithms can be realized using multidimensional numbered information spaces as storage data structures.

The proposed approach is realized in the module ArmSquare. It is aimed to make analysis and monitoring over the produced association rules from frequent datasets. ArmSquare is a part of Data Mining Environment PaGaNe [Mitov et al, 2011]. The main data structures in PaGaNe use the advantages of specific model for organization of the

storage of information, called Multidimensional Numbered Information Spaces, which gives convenient apparatus for operating with the structures [Markov, 2004]. The model is realized in the access method ArM 32. Let's mention the existing confusion of abbreviation ARM used in literature for short denotation of "association rule miner" and ArM, which means "Archive Manager". The name ArM was born in 1991 year (see [Markov et al, 2008]), two years before the defining of the association rule mining in [Agrawal et al, 1993]. The duplicating was used in the name of the realization: ArmSquare.

4.3 Multidimensional Numbered Information Spaces

Following the Multi-Domain Information Model (MDIM), presented in [Markov, 2004] and realized by ArM 32, the elements are organized in a hierarchy of numbered information spaces with variable ranges, called ArM-spaces.

4.3.1 Constructs

There exist two main constructs in MDIM – *basic information elements* and *numbered information spaces*. Basic information element is an arbitrary long sequence of machine codes (bytes). Basic information elements are united in numbered sets, called numbered information spaces of range 1. The numbered information space of range n is a set, which elements are numerically ordered information spaces of range $n-1$. ArM32 allows using of information spaces with different ranges in the same file.

Every element may be accessed by correspond multidimensional space address (ArM-address) given by a coordinate array. The coordinate array is represented as numerical vector $A=(n,p_1,...,p_n)$, in which starting position shows the dimension of the space and next positions contains the coordinates of the points, thorough which the information can be reached. Sometimes, accounting the difference between the meaning of starting coordinate and next coordinates, the vector is written as $(n:p_1,...,p_n)$.

Another constructs, connected with MDIM are *indexes* and *metaindexes*. Every sequence of space addresses $A_1,A_2,...,A_k$, where k is an arbitrary natural number, is said to be an *index*. Every index may be considered as basic information element and may be stored in a point of any information domain. In such case, it will have a space address which may be pointed again. Every index which point only to indexes is said to be a *meta-index*.

Special kind of space index became the *projection*, which is analytical given index. There are two types of projections: (1) *Hierarchical projection* – in which the top part of coordinates is fixed and low part vary for all possible values of coordinates, where non-empty elements exist; and (2) *Arbitrary projection* – in this case is possible to fix coordinates in arbitrary positions and the rest coordinates vary for all possible values of coordinates, where non-empty elements exist.

4.3.2 Operations

It is clear; the operations are closely connected to the defined structures. So, we have operations with:

- *Basic information elements*: Because of the rule for existing of the all structures given above we have need of only two operations: updating and getting the value and two service operations: getting length and positioning in the element;
- *Spaces*: With two spaces we may provide two operations: copying the first space in the second and moving the first space in the second with modifications specifying clearing or remaining the second space before operation;
- *Indexes and meta-indexes*: Using the hierarchical projection we may crawl the defined area and extract next or previous empty or non-empty elements as well as to receive the whole index or its length, of the non-empty elements, which addresses fall into defined projection. The same operations (but only for non-empty elements) can be made for arbitrary projection. The operations between indexes are based on usual logical operations between sets. The difference from usual sets is that the information spaces are built by interconnection between two main sets: set of co-ordinates and set of information elements.

4.4 Algorithm Description of ArmSquare

We propose to use the abilities of multidimensional numbered information spaces for storage the information about interconnections between items and their combinations for facilitating association rule mining. The proposed algorithm makes special analysis for each transaction and stores the frequency information in ArM-space, using the possibility of these spaces for accessing to the data via coordinate arrays. The algorithm consists of three phases: (1) pretreatment; (2) data processing and (3) analysis and monitoring.

4.4.1 Pretreatment

In the pretreatment phase, the following steps are made:

- The transactions of the incoming dataset D may be split into subsets, $D_b, b = 1,...,d \quad \bigcup_{b=1}^{d} D_b = D$ by some condition (periods, regions, etc.). The mapping between the names of these groups and natural numbers $b = 1,...,d$ is made;
- Creating a mapping between incoming items $i_j \in \mathbf{D}$ and natural numbers $c_j \in \mathbf{N}$ by order of first occurrence of the item. This way if $\mathbf{D}$ is a set of items, then $\overline{\mathbf{D}}$ is also a set of items that contains the numbers from 1 to n ($n = |\mathbf{D}|$), which code the items of $\mathbf{D}$. Each incoming transaction $T = (tid, I), \; I = \{i_1,...,i_k\} \subseteq \mathbf{D}$ is transformed into $\overline{T} = (tid, C), \quad C = \{c_1,...,c_k\} \subseteq \overline{\mathbf{D}}$. The transaction database D over $\mathbf{D}$ is transformed to $\overline{D}$ over $\overline{\mathbf{D}}$;
- The items in each transaction $\overline{T}$ are sorted in increasing order. The received transaction is denoted by $\vec{T}$. Ordering the items in the transaction has a great importance for the consequent steps.

4.4.2 Data Processing

The intermediate phase is data processing. The data processing is closely depended on the length of the derived association rules. The greater the length, the more resources are needed. Because of this usually a special parameter $MaxK$ is used for limiting the maximum length of examined association rules. The algorithm traverse all combinations from 1 to $MaxK$. Let k be the examined number of items $1 \le k \le MaxK$. For each transaction $\vec{T}$, $n = |\vec{T}| \ge k$ we make all possible combinations $Z^l = \{c_1^l,...,c_k^l\}$, $Z \subseteq \vec{T}$, $l = 1,..., \frac{n!}{k!(n-k)!}$. The element of Arm-spaces with coordinates $(c_1^l,...,c_k^l)$ accumulates the number of occurrence of corresponded itemset $Z^l = \{c_1^l,...,c_k^l\}$ (Figure 64).

In the case when D is split in subsets $D_b, b = 1,...,d$ additional coordinate in the space address is placed for marking the number of the group b where transaction belongs to and the space address became following form $(b, c_1^l,...,c_k^l)$.

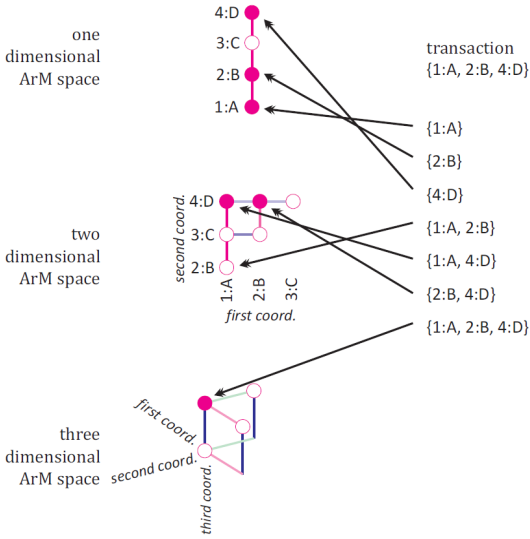

Figure 64. Accumulating in ArM spaces of the number of occurrence of produced itemsets from one transaction

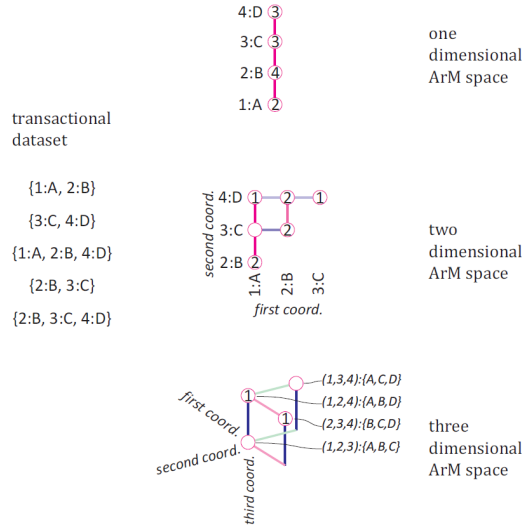

Figure 65. Result of data processing of the database

As far as the processing over combinations with different lengths as well as subsets of database D are independent, operations may be provided in parallel.

Finally, the support of the itemsets, which are driven from the transactions, is accumulated in the corresponded points in ArM-spaces (Figure 65).

Note that because of the ordering, not all coordinates in corresponded space are used. ArM 32 does not waste memory for empty points.

4.4.3 Analysis and Monitoring

The analysis is made over the itemsets with particular length k, $1 \le k \le MaxK$ and using a minimal support $MinSup$, which the itemsets to be included in the resulting list must have and a minimal confidence $MinConf$, which the association rules, created on the basis of the already selected itemsets, must have.

For obtaining all existing k-itemsets, whose support are at least $MinSup$, a traversal of all non-empty elements in a k-dimensional ArM-space is made. The coordinates of each element $(c_1^l,...,c_k^l)$, which contains value, no less than $MinSup$, corresponds to itemset $Z^l = \{c_1^l,...,c_k^l\}$, which is included in the resulting list $F(D, MinSup)$.

In the case where a database is split into groups, the traversal is made for each group taking into account that the first coordinate indicates the number of the group. The support of itemset $Z^l = \{c_1^l,...,c_k^l\}$ for the whole database is received as a sum of values, contained in the points with corresponding coordinates in each group $(b, c_1^l,...,c_k^l), b = 1,...,d$.

One itemset $Z^l = \{c_1^l,...,c_k^l\}$ is a source of producing several association rules. Let $Z = \{c_1,...,c_k\}$ be a k-itemset, $Z \in F(D, MinSup)$. The collection of association rules, which confidence exceeds $MinConf$ is obtained by examining all possible combinations with length from 1 to k-1, which is given as a body of the rule $X^j = \{c_1^j,...,c_p^j\}$, $p = 1,...,k-1, X^j \subset Z$. The rest of the items forms the head of the rule $Y^j = Z \setminus X^j$. For an association rule $X^j \Rightarrow Y^j$ from the point with the space address $(c_1^j,...,c_p^j)$, which correspond to X^j, the $support(X^j)$ is received. Taking into account that the body is part of an already existing itemset, this value is more than zero. The confidence of this association rule is calculated as

$confidence(X^j \Rightarrow Y^j) = \frac{support(Z)}{support(X^j)}$. If $confidence(X^j \Rightarrow Y^j) \geq Minconf$, then $X^j \Rightarrow Y^j$ is included in the list of resulting association rules $R(D, MinSup, MinConf)$.

4.5 Program Realization

The proposed algorithm was realized as analyzing tool in data mining environment PaGaNe.

4.5.1 Input Data

The system allows creation of a new database as well as adding new transactions to the same or different groups in an already existing database. During the input, the system accumulates the information for maximal length and average length of the transactions by each group separately. The repeated elements within the transaction are omitted.

Each transaction in the system is presented as numerical vector, with the length equal to the number of the elements, which participate in the transaction. The elements in the vectors are numbers, which correspond to the position of the element in dynamically expanded nomenclature, which contains the names of the items. Finally, these numbers are sorted. Sorting the elements in the vectors has a great importance for the consequent steps.

4.5.2 Pretreatment in ArmSquare

Before starting the processing, one has to give a maximal number of combinations, which will be interesting – *MaxK.* Usually not all combinations between elements are interesting, but only a limited number of them – 2, 3, or 4, and no more than 10.

The pretreatment performs a special analysis for each transaction in all groups and stores the frequency information in ArM-spaces, using the ability of the ArM-spaces for accessing the data via coordinate arrays. The user is interested in combinations from 1 to *MaxK.*

Let's trace the process for the transaction $\vec{T}$, which belongs to the group with number b. The length $n = |\vec{T}|$ varies depending on the numbers of the elements that were in the transaction. The system loops by *k* from 1 to $\min(n, MaxK)$ in order to traverse all possible combinations. Each combination $\{c_1, ..., c_k\}$ is used to form ArM-address

for "k+1" dimensional space $(b,c_1,...,c_k)$ and at this point a support value is incremented by 1.

4.5.3 Analysis and Monitoring

The user can choose which length of itemsets he wants to observe. This length can vary between 2 and *MaxK*. Other parameters are minimal support and minimal confidence for a given database.

For obtaining all existing k-itemset with a support of at least $MinSup$, crawling over the k-dimensional ArM-space is done using the function *ArmNextProj*, starting with hierarchical projection $(-,-,...,-)$. In case of observing only a concrete group b, the crawling is made in k+1-dimensional ArM space with starting projection $(b,-,...,-)$. Using the function *ArmRead* for the current extracted non-empty element, the value is read and is compared with $MinSup$. If this value is no less than $MinSup$, the corresponded itemset is included into the resulting list of itemsets $F(D,MinSup)$. Optionally the resulting itemsets is sorted by decreasing support.

For receiving all association rules, created on the basis of itemsets from $F(D,MinSup)$, which have a confidence no less than $MinConf$, for each itemset $Z=\{c_1,...,c_k\}$ where $Z\in F(D,MinSup)$, every possible combination with a length from 1 to k-1, where $X^j=\{c_1^j,...,c_p^j\}$, $p=1,...,k-1, X^j\subset Z$ is assumed as a body, while the rest of the items are taken as a head of the rule $Y^j=Z\setminus X^j$, is examined. For association rule $X^j\Rightarrow Y^j$:

- The value of $support(X^j)$ is received using function *ArmRead* from coordinate space address $(c_1^j,...,c_p^j)$, which correspond to the body X^j;
- The $confidence(X^j\Rightarrow Y^j)=\frac{support(Z)}{support(X^j)}$ is calculated;
- The association rule $X^j\Rightarrow Y^j$, whose confidence is no less than $MinConf$ is included in the list of resulting association rules $R(D,MinSup,MinConf)$.

Association rules are sorted by decreasing confidence (optional).

Using the ArM functions, the following additional operations, which can be used for analysis of the database, can be executed:

(1) Observation of all itemsets with given length k. For this purpose a function *ArmProjIndex* with hierarchical projection on the highest level in k-dimensional space is used. In the case of viewing the itemsets, belonging to a given group – the projection is one level lower in k+1 dimensional space with the highest coordinate equal to the number of the group fixed.

(2) Looking for k-itemsets, containing concrete item with given number c. A loop from 1 to k allows crawling the arbitrary projection $(-,...,-,c,-,...,-)$, where position of c varies accordingly to the loop phase. A function *ArmProjIndex* uses these projections and extracts all non-empty elements, which define the corresponding itemset. The union of all these elements is the result of the request.

4.6 Advanced Specifics of ArmSquare

In Apriori algorithm min-support is set globally for combinations with different lengths. In our algorithm, after building the spaces, statistics for min-support for each area can be derived separately (the amount of space is equal to the number of elements in combinations), which allows to give for further analysis different min-support for different numbers of elements in combinations.

In a higher value of min-support Apriori is highly convergent and reaches a relatively short itemsets, where a small amount of min-support is close to total exhaustion of short itemsets.

Structuring the support of the itemsets in ArM-space allows subsequent analysis to be made very quickly by setting a different min-support and profiles of different lengths of itemsets, while other ARM-approaches derive all successive combinations in ascending order and changing the min-support causes a repetition of the whole algorithm.

The information for itemsets with particular length containing a specific element can be directly extracted.

The database can be interactively expanded as well as the processing of the transactions can be made in parallel.

4.7 Implementation

The realized tool allows different types of useful implementations in a wide spectrum of applications. Here we made experiment over a dataset that included several types of colour harmonies and contrast features, extracted by 600 paintings of 19 artists from different movements of West-European fine arts and Eastern Medieval Culture [Ivanova et al,

2010]. The pictures were obtained from different web-museums sources using ArtCyclopedia as a gate to the museum-quality fine art on the Internet (Table 8).

Table 8. List of the artists, which paintings were used in experiments, grouped by movements

Movement	Artist
Icons (60)	Icons (60)
Renaissance (90)	Botticelli (30); Michelangelo (30); Raphael (30)
Baroque (90)	Caravaggio (30); Rembrandt (30); Rubens (30)
Romanticism (90)	Friedrich (30); Goya (30); Turner (30)
Impressionism (90)	Monet (30); Pissarro (30); Sisley (30)
Cubism (90)	Braque (30); Gris (30); Leger (30)
Modern Art (90)	Klimt (30); Miro (30); Mucha (30)

Each row of formed dataset contained the name of the artists, followed with harmonies and contrast features, presented in the manner of transactional dataset: "feature name"="value".

Using the possibility of binning the dataset by class label allowed to use ArmSquare as element in the generation rule phase of CAR-algorithm and extract typical combinations of features for examined artists. For instance, for more than one third of paintings, combinations of four attributes are presented in Table 9.

Table 9. Combinations of four attributes, with more than 33.33% support for examined artists

Artist	4-items combinations	Support (%)
CARAVAGGIO	Sat. Harmony =3-SMOOTH Lum. Harmony =2-SMOOTH Warm-cold contrast =WARM-NEUTRAL Clear-dull contrast =CLEAR-DULL	33.33
GRIS	Hue Harmony =PARTIAL TRIAD Sat. Harmony =4-VARIETY Lum. Harmony =3-SMOOTH Dark-light contrast =MIDDLE	36.67
GRIS	Hue Harmony =PARTIAL TRIAD Lum. Harmony =3-SMOOTH Clear-dull contrast =SPECTRAL-GROUND Dark-light contrast =MIDDLE	33.33

ICON	Hue Harmony =ANALOGOUS Lum. Harmony =3-SMOOTH Warm-cold contrast =WARM Dark-light contrast =MIDDLE	35.00
ICON	Hue Harmony =ANALOGOUS Sat. Harmony =3-SMOOTH Warm-cold contrast =WARM Dark-light contrast =MIDDLE	31.67
MUCHA	Hue Harmony =ANALOGOUS Sat. Harmony =3-SMOOTH Lum. Harmony =3-SMOOTH Warm-cold contrast =WARM	36.67
MUCHA	Hue Harmony =ANALOGOUS Sat. Harmony =3-SMOOTH Lum. Harmony =3-SMOOTH Clear-dull contrast =SPECTRAL-GROUND	33.33
MUCHA	Hue Harmony =ANALOGOUS Lum. Harmony =3-SMOOTH Warm-cold contrast =WARM Clear-dull contrast =SPECTRAL-GROUND	33.33
RUBENS	Hue Harmony =ANALOGOUS Sat. Harmony =3-SMOOTH Lum.Harmony =2-SMOOTH Warm-cold contrast =WARM	33.33
REMBRANDT	Hue Harmony =ANALOGOUS Warm-cold contrast =WARM Clear-dull =CLEAR-DULL Dark-light =DARK	40.00
REMBRANDT	Warm-cold contrast =WARM Hue harmony =ANALOGOUS Clear-dull contrast =CLEAR-DULL Sat. Harmony =1-MONOINTENSE	36.67
REMBRANDT	Warm-cold contrast =WARM Clear-dull contrast =CLEAR-DULL Lum. harmony =1-MONOINTENSE Dark-light =DARK	33.33

Such approach of extracting rules from frequent datasets as well as their extension in the direction of class association algorithms can be used for defining semantic profiles of observed phenomena – movement, artists style or thematic, connected with abstract space of the taxonomy of the art image content, discussed by Tomas Hurtut in [Hurtut, 2010].

5 PGN: Classification with High Confidence Rules

Within the data mining community, research on classification techniques has a long and fruitful history. Classification techniques based on association rules, called class-association rule (CAR) algorithms, are relatively new. CAR-classifiers generate a set of association rules from a given training set and use these rules to classify new instances. As it is mentioned in [Zaïane and Antonie, 2005], the advantages of associative classifiers can be highlighted in five major ones:

- The training is very efficient regardless of the size of the training set;
- Training sets with high dimensionality can be handled with ease and no assumptions are made on dependence or independence of attributes;
- The classification is very fast;
- Classification based on association methods present higher accuracy than traditional classification methods [Liu et al, 1998] [Li et al, 2001] [Thabtah et al, 2005] [Yin and Han, 2003];
- The classification model is a set of rules easily interpreted by human beings and can be edited [Sarwar et al, 2001].

In 1998, CBA is introduced in [Liu et al, 1998], often considered to be the first associative classifier. During the last decade, various other associative classifiers were introduced, such as CMAR [Li et al, 2001], ARC-AC and ARC-BC [Zaïane and Antonie, 2002], CPAR [Yin and Han, 2003], CorClass [Zimmermann and De Raedt, 2004], ACRI [Rak et al, 2005], TFPC [Coenen and Leng, 2005], HARMONY [Wang and Karypis, 2005], MCAR [Thabtah et al, 2005], 2SARC1 and 2SARC2 [Antonie et al, 2006], CACA [Tang and Liao, 2007], ARUBAS [Depaire et al, 2008], etc.

Typically, the generation of association rules from a training set is guided by the support and confidence metrics. Many associative classifiers set a minimum support level and use the confidence metric to rank the remaining association rules. This approach, with a primary focus on support and confidence as the second criterion, will reject 100% confidence rules if the support is too low.

We question this common approach which prioritizes support over confidence. We study a new associative classifier algorithm, called PGN, which turns the priorities around and focuses on confidence first by retaining only 100% confidence rules. The main goal of this research is to verify the quality of the confidence-first concept.

5.1 The Structure of CAR-algorithms

Generally the structure of CAR-algorithms consists of three major data mining steps:

1. Association rule mining.
2. Pruning (optional).
3. Classification.

The mining of association rules is a typical data mining task that works in an unsupervised manner. A major advantage of association rules is that they are theoretically capable of revealing all interesting relationships in a database. But for practical applications the number of mined rules is usually too large to be exploited entirely. This is why the pruning phase is stringent in order to build accurate and compact classifiers. The smaller the number of rules a classifier needs to approximate the target concept satisfactorily, the more human-interpretable is the result.

5.1.1 Association Rule Mining

All associative classifiers start by generating association rules from a given training set. Different implementations of associative classifiers use different association rule mining techniques. For example, the Apriori algorithm is used by CBA, ARC-AC, ARC-BC, ACRI and ARUBAS, while CMAR uses the FP-tree algorithm, CPAR uses the FOIL algorithm and CorClass uses the Morishita&Sese Framework.

Class association rules can be generated from a single data set containing all training transactions, which is e.g. the case for ARC-AC, CMAR or CBA, or can be generated from a set of data sets, where training cases are grouped per class label. The latter is the case for ARC-BC and makes it more probable for small classes to have representative class association rules. Furthermore, all association rule mining algorithms produce the same set of class association rules, but differ in terms of computational complexity. One exception is the FOIL algorithm used in the CMAR implementation, which is a heuristic rather than an exact solution and only gives an approximation of the exhaustive set of class association rules meeting specific support and confidence criteria.

5.1.2 Pruning

Once the CARs are generated from the training set, most associative classifiers apply some pruning strategy to reduce the size of the set. Even if there is no separate post-pruning step, all algorithms apply some sort of pre-pruning during the rule generation step by setting a support and/or confidence threshold. This pre-pruning technique is an isolated pruning technique as the CARs are evaluated individually, in isolation from the

other CARs. Other isolated pruning techniques are Pessimistic Error Pruning and Correlation Pruning. Pessimistic Error Pruning, applied in CBA, uses the pessimistic error rate from C4.5 [Quinlan, 1993] to prune, while Correlation Pruning, which is applied in CMAR, uses the correlation between the rule's body and the rule's head.

Some associative classifiers use non-isolated pruning techniques which take multiple rules into account when deciding whether or not to prune a specific rule. A well-known non-isolated pruning technique is the Data Coverage Pruning technique (DCP), which is applied in CBA, ARC-AC, ARC-BC and CMAR. DCP consists of two steps. First, the rule set is ordered according to confidence, support and rule size. Rules with the highest confidence go first. In case of a tie, rules with the highest support take precedence. In case of a tie in terms of confidence and support, the smaller the rule, i.e. the more general a rule is, the higher the ranking. Once the rule set is ordered, the rules are taken one by one from the ordered rule set and are added to the final rule set until every record in the training set is matched at least a times. For CBA, ARC-AC and ARC-BC this parameter a is fixed to 1, while in CMAR a is a parameter which needs to be set by the user. Confidence Pruning (ConfP) is another non-isolated pruning technique which is used by CMAR, ARC-AC, ARC-BC. ConfP prunes all rules which are generalized by another rule with a higher confidence level.

5.1.3 Classification

Once the CARs are generated and pruned, the associative classifier uses all these pieces of local knowledge to classify new instances. While some associative classifiers apply order-based classification, others use non-order-based classification. With order-based classification, the association rules are ordered according to a specific criterion, while non-order-based classifiers do not rely on the order of the rules.

Among the order-based classification schemes, the Single Rule Classification approach has to be distinguished from the Multiple Rule Classification approach. The former approach orders the rules and uses the first rule which covers the new instance to make a prediction. The predicted class is the selected rule's head. This classification scheme is used by CBA, CorClass and ACRI. CBA and CorClass order the rules according to confidence, support and rule size in the same way the data coverage pruning technique does. ACRI on the other hand, allows the users to select from four different ordering criteria, i.e. a cosine measure, the support, the confidence or the coverage.

Multiple Rule Classification, which is used by CPAR, selects those rules which cover the new instance, groups them per class and orders them

according to a specific criterion. Finally, a combined measure is calculated for the best *Z* rules, where *Z* is a user-defined parameter. With CPAR, the rank of each rule is determined by the expected accuracy of the rule.

Furthermore, some associative classifiers, such as CMAR, ARC-AC, ARC-BC and CorClass, use a non-order-based classification scheme. These classification schemes select the rules, which cover the new instance; groups them per class and calculates a combined measure per class value. This approach is almost identical to the order-based multiple rule classification scheme, except for the ordering step.

5.2 Algorithm Description of PGN Classifier

One of the main specifics of PGN is that it is a parameter free classifier. Let mention that in classical CAR algorithms user must give the support and confidence level.

The association rule mining goes from longest rules (instances) to the shorter ones until no intersections between patterns in the classes are possible. In the pruning phase the contradictions and inconsistencies of more general rules are cleared, after that the pattern set is compacted throwing all more concrete rules within the classes.

To illustrate the algorithm, a simple data set shown in Table 10 is used as example.

Table 10. Example Data Set. The first attribute (separated with "|" from others) is class label

```
R1:  (1| 1, 2, 4, 1)
R2:  (1| 1, 2, 3, 1)
R3:  (1| 3, 1, 3, 2)
R4:  (1| 3, 1, 4, 2)
R5:  (1| 1, 2, 4, 1)  Equal to R1
R6:  (1| 3, 1, 4, 2)  Equal to R4
R7:  (2| 3, 1, 1, 2)
R8:  (2| 2, 1, 1, 2)
R9:  (2| 3, 1, 2, 2)
```

5.2.1 Training Process

The training process consists of:

- Generalization – the process of associative rule mining;
- Pruning – the process of clearing exceptions between classes and lightening the pattern set;
- Searching patterns with unique attributes. This step is optional as well as it not typical CAR strategy and from other side it created very powerful patterns, which is good for some data set, but not for the others.

Step 1: Generalization

The step of creating the pattern set consists of two phases:

1. Adding instances to the pattern set.
2. Creating all possible intersection patterns between patterns within the class.

PGN starts by adding all records to the appropriate set of association rules. For each class, a separate set of association rule is generated (Figure 66).

Figure 66. Adding instances in the pattern set

Later, for each class every combination of two patterns is intersected.

The intersection between P^i and P^j is the result of matching of these patterns.

$$P^i \cap P^j = (c^l \mid a_1^l,\ldots,a_n^l):\ c^l = \begin{cases} c^i : c^i = c^j \\ \text{"}-\text{"} : c^i \neq c^j \end{cases} \text{ and } a_k^l = \begin{cases} a_k^i : a_k^i = a_k^j \\ \text{"}-\text{"} : a_k^i \neq a_k^j \end{cases}.$$

If a new pattern exists, it is added to the pattern set. If the patterns-candidates to be written into the pattern set (instances as well as patterns) are already in it, then they are not duplicated, only the set of instances that are possible creators of the pattern is expanded. The process goes iteratively until no intersections are possible. Figure 67 shows the process of creating the pattern set on the example data set.

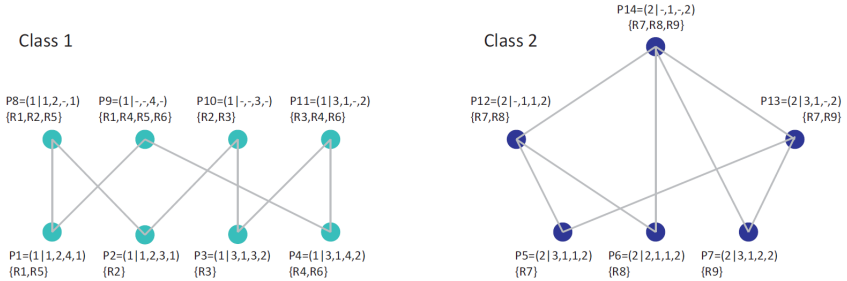

Figure 67. Adding intersections in the pattern set

Step 2: Pruning

In this step some patterns are removed from the pattern set using two phases:

1. Deleting contradictory patterns as well as general patterns that have exception patterns in some other class.
2. Removing more concrete patterns within the classes. This step ensures compactness of the pattern set that can be used in the recognition stage.

In the first phase the patterns, belonging to different classes are paired. If one pattern matches another pattern (but they have a different class value), then the more general is removed. If two patterns match each other then both of them are removed.

$$P^i, P^j \in PS,\ c^i \neq c^j : \begin{cases} \left|P^i \cap P^j\right| = \left|P^i\right| < \left|P^j\right| : \text{remove } P^i \\ \left|P^i \cap P^j\right| = \left|P^j\right| < \left|P^i\right| : \text{remove } P^j \\ \left|P^i \cap P^j\right| = \left|P^i\right| = \left|P^j\right| : \text{remove } P^i, P^j \end{cases}$$

If a dataset does not contain missing values, then all instances have equal $|R| = n$ and all other patterns will have smaller sizes. This means that checking up for data consistency can be done with comparison of patterns only with the instances but not with all patterns from other classes.

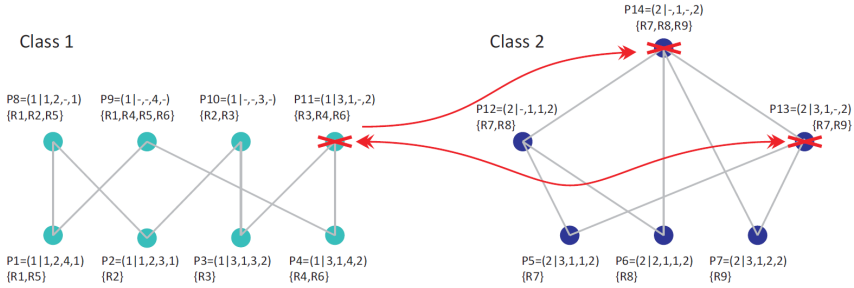

Figure 68. Supplying maximum confidence of the rules

Figure 68 shows the first pruning phase for example data set. The process pass in two steps: first labelling, than removing.

This step tries to supply the maximum confidence of the resulting rules. This operation removes the patterns that do not formulate a representative for a given class combination, because in another class

there exists pattern with an equal or more concrete combination of the same values of attributes, which can pretend to recognize the request. Furthermore, by removing incorrect patterns (records with equal attributes, which belongs to different classes) this operation ignores the possible inconsistencies in the learning set. Of course, the tendency of supplying a maximum confidence for final patterns suffers from the impossibility to create patterns in noisy datasets.

The second phase, which retains most general rules, is provided again within the classes. Patterns from equal classes are compared and, conversely to the previous phase, more concrete patterns are deleted, i.e. the larger pattern is removed.

$$P^i, P^j \in PS,\ c^i = c^j : \begin{cases} \left|P^i \cap P^j\right| = \left|P^i\right| < \left|P^j\right| : \text{remove } P^j \\ \left|P^i \cap P^j\right| = \left|P^j\right| < \left|P^i\right| : \text{remove } P^i \end{cases}$$

The simple idea is that after first phase in the pattern set remains only patterns that are not exceptions to the other class. Because of this, we can make lighter the pattern set by removing patterns for which other patterns are subsets.

Figure 69 shows lightening the pattern set for example data set.

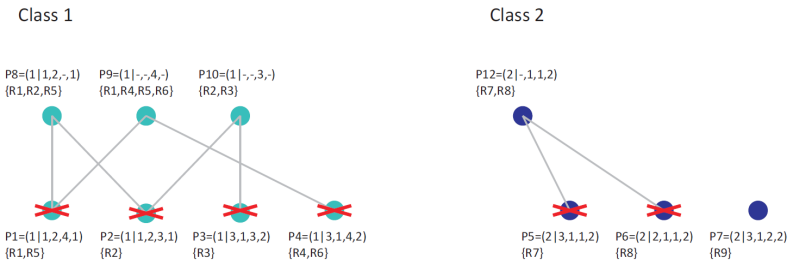

Figure 69. Retain most general rules

As a result of this step in the pattern set remain only patters that are general for the class that they belong and their bodies are not subsets of the bodies of patterns in other classes.

Table 11 shows the pattern set with corresponded support of each remained pattern for the example dataset.

Table 11. Association rules after pruning

	Pattern set	Support	Support set
	Class 1		
P8	(1| 1, 2, -, 1)	3	{R1,R2,R5}
P9	(1| -, -, 4, -)	4	{R1,R4,R5,R6}
P10	(1| -, -, 3, -)	2	{R2,R3}
	Class 2		
P7	(2| 3, 1, 2, 2)	1	{R9}
P12	(2| -, 1, 1, 2)	2	{R7,R8}

The analysis of the presence of attributes in different classes can show that the values of some attribute are contained only in the instances (one or a few) of a given class. It is possible to define an essence threshold, over which this value is assumed representative for the class. In such case a new pattern with corresponding values – class number and value of given attribute (all other attribute values are "-") is created. The patterns created in this phase have big power during the recognition stage. Because of this, after analysis of the application area, the expert has to decide to allow or prohibit the execution of this procedure.

5.2.2 Classification

The record to be recognized is given by the values of its attributes $Q = (?\,|\, a_1, a_2, ..., a_n)$. Some of the features may be omitted.

To classify new instances with the pruned rule set, the definition for the size of an association rule must be introduced first. The association rule size corresponds to the number of non-class attributes which have a non-missing value: $|P| = |\{a_i \,|\, 1 \le i \le n-1, a_i \ne "-"\}|$. The intersection percentage between pattern P and query Q is defined as $IP(P,Q) = \frac{|P \cap Q|}{|P|}$.

To classify a new instance, the intersection percentage between the test case and every rule is calculated. This allows for two different scenarios:

- When the maximum intersection percentage occurs only in one class (for only one single rule or for different rules but in the same class), this class becomes the predicted class for the new instance;
- When the maximum intersection percentage occurs multiple times for rules from different classes, the supports of these rules are summed per class. The class with the highest aggregated support becomes the predicted class for the new instance.

Note that this classification scheme also uses association rules which do not cover the test case perfectly for classification purposes.

Let's illustrate this classification method with the pruned rule sets in Table 11. Assume a new instance $Q = (?|1,2,1,2)$ which needs to be classified.

Firstly, the intersection percentage between Q and every rule is calculated and shown in Table 12.

Table 12. Classification of $Q = (?|1,2,1,2)$

	Pattern set	$P \cap Q$	$IntSize(P,Q)$	Support	Support set
	Class 1				
P8	(1| 1, 2, -, 1)	(?| 1, 2, -, -)	0.667	3	{R1,R2,R5}
P9	(1| -, -, 4, -)	(?| -, -, -, -)	0	4	{R1,R4,R5,R6}
P10	(1| -, -, 3, -)	(?| -, -, -, -)	0	2	{R2,R3}
	Class 2				
P7	(2| 3, 1, 2, 2)	(?| -, -, -, 2)	0.250	1	{R9}
P12	(2| -, 1, 1, 2)	(?| -, -, 1, 2)	0.667	2	{R7,R8}

The maximum intersection percentage is 0.667 and occurs for rules P^8 and P^{12}, which belong to different classes. Considering only the rules with an intersection percentage of 0.667, the summed support for class 1 is 3 and the summed support for class 2 is 2. Consequently, the new instance is predicted to belong to class 1.

5.2.3 Empirical Analysis of PGN

Several experiments were produced to compare PGN against other classifiers, realized in Weka [Witten and Frank, 2005], representatives of most similar recognition models to CAR algorithms as Decision Trees and Decision Rules, which have a similar model representation language.

The experiments were performed on various data sets from the UCI Machine Learning Repository [Frank and Asuncion, 2010]. Data sets containing continuous attributes were discretized first by means of the Chi-merge method [Kerber, 1992]. This discretization method is based on the χ^2 statistic and uses a Chi-square threshold as stopping rule with Chi-square threshold 95%. This method and threshold was chosen because achieves lower classification error than those trained on data pre-processed by the other discretization methods or thresholds [Mitov et al, 2009/iTech].

The comparison of overall accuracy between PGN and other classifiers was made using methodology suggested by Demsar [Demsar, 2006]. The

Friedman test showed statistical difference between tested classifiers. The pos-hoc Nemenyi test showed that our PGN has best overall performance between examined classifiers [Mitov, 2011].

The experimental results are very positive and show that PGN is competitive with classification methods that build similar classification models. At the same time, it has the advantage over the other classifiers that it is parameter free. In general, the results provide evidence that the confidence-first approach yields interesting opportunities.

5.3 PGN and Predictive Analysis in Art Collections

The experiments were made over the same art collection, described in 4.7. The datasets were formed using proposed visual and higher-level attributes. The movements or the artists' names were used as class labels.

5.3.1 Predictive Analysis of the Visual Features

The visual features can be used to classify/retrieve movements and artists styles. We made three-fold cross validation using the datasets that contains hue values, saturation values, luminance values separately and all three together. We analyzed the results of OneR, JRip, J48, and PGN, comparing average accuracies and confusion matrices.

Table 13 and Table 14 and Figure 70 show the accuracies by different classifiers by distribution of hue, saturation, luminance separately and all three together.

Table 13. Accuracy – movements

Database	OneR	JRip	J48	PGN
Hue	27.83	34.00	39.00	42.83
saturation	34.83	33.00	35.33	36.50
luminance	30.67	35.00	38.50	45.83
HSL	33.50	49.00	47.00	63.17

Table 14. Accuracy – artists' names

Database	OneR	JRip	J48	PGN
hue	15.33	24.67	24.17	29.83
saturation	18.83	17.17	22.5	24.67
luminance	17.83	26.33	26.17	32.00
HSL	18.83	36.67	37.17	49.33

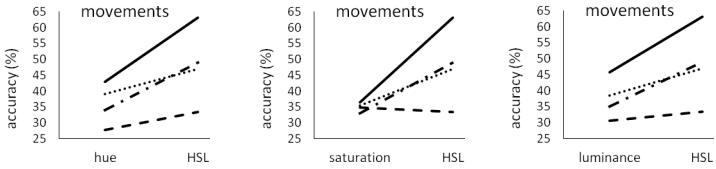

Figure 70. The accuracies of different classifiers by hue, saturation, luminance separately and all three together

As expected the accuracies obtained by the classifiers based on one colour component are similar and we have an increase in accuracy by combining the components. The table shows however a curiosity. As we can see, examining all attributes together does not increase the accuracy of the OneR classifier for movements. In the three fold-cases for "HSL" dataset OneR choose "v0" attribute as most appropriate, but not "s7" or "s8" as in the case of "Saturation" dataset, it leads to decreasing of overall accuracy in HSL dataset than in simpler one "Saturation" dataset.

As we can see PGN shows the best accuracies from examined models for all datasets. Additionally PGN shows the best possibilities to explore specific combinations of attribute values, it achieves the biggest increase of accuracy by examining all three characteristics together.

Figure 71 and Figure 72 show the confusion matrices for movements and for artists' names respectively. In the visualization of confusion matrices, the darker is the square, the bigger is the percentage of images following into corresponded square.

Analyzing the movements results three patterns immediately get attention. Firs the movement Baroque is the most easy to predict, OneR fails to predict Modern Art, PGN is the only classifier with a nice consistent black/grey downwards diagonal. The first pattern repeats patterns seen in the descriptive analysis. It seems that Modern Art pictures can not be characterized with one visual attribute. The characteristic PGN rules can better discriminate than J48 rules especially between the movements Romanticism, Impressionism, Cubism and Modern Art. Let's mention again the specifics of the PGN against other classifiers. All other classifiers take into account in one on other manner the support, controversially to PGN, which focuses primarily on the confidence of the association rules and only in a later stage on the support of the rules.

Analyzing the artist results the three mentioned patterns are confirmed and two new ones are seen: the presence of vertical lines (dark or light) and the presence of "movement" squares.

It is clear that based on visual characteristics OneR is not able to classify the different artist paintings. JRip predicts almost 25% of the paintings as Icon (the vertical line in the JRip confusion matrix). The datasets that we use here are specifics that all artists are represented with equal number of paintings, and all selected movements contain also fixed number of artists, i.e. the distributions are equal. The exception is Icons, which are twice more than each artist and two-thirds than the movements. Because of this, we can see for the precision of Icons the tendencies of losing percentages for movements and enforcing ones for artists for OneR, JRip and J48 – here and in consequent analyses.

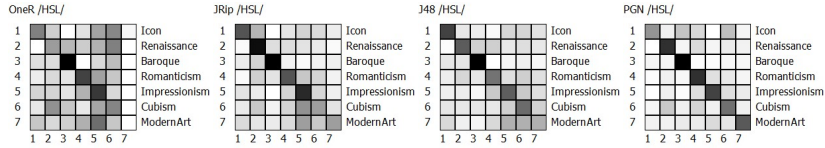

Figure 71. Confusion matrices for HSL features, movements as class labels

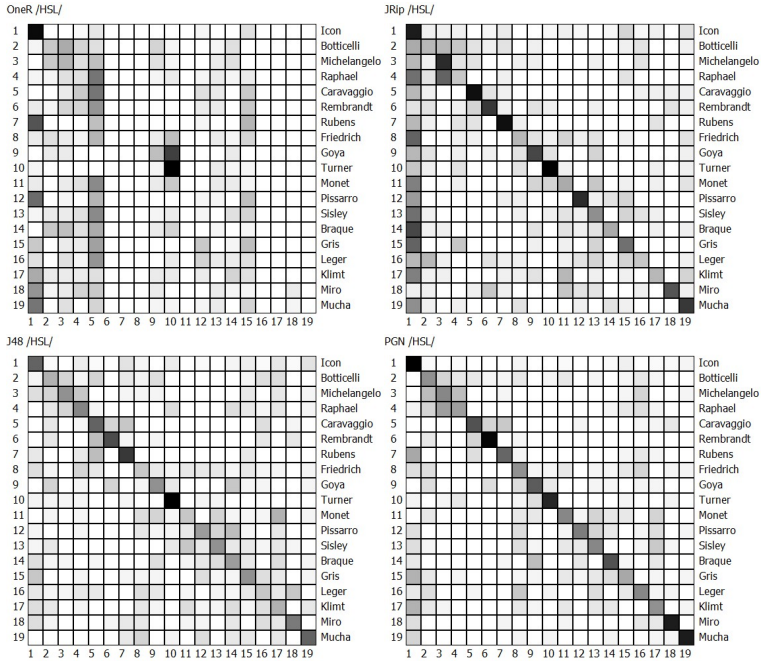

Figure 72. Confusion matrices for HSL features, artists' names as class labels

The grey squares show some common tendencies of recognizing or misplacing the class labels. For instance, it is interesting that the Renaissance painters Botticelli, Michelangelo and Raphael are not recognized correctly but are misclassified mainly within the group. Icons, Michelangelo, Caravaggio, Rembrandt, Rubens, Turner, Pissarro, Miro and Mucha can be labelled as more easy to classify.

5.3.2 Predictive Analysis of the Harmonies/Contrast Features

The harmonies/contrast features try to extract very global colour combinations constructs. As we already mention, there are a lot of reasons that influence of choosing one or another colour combination – the thematic of the painting, fashion style, philosophical visions of the painter, its current emotional condition, etc. Because of this we do not expect that such features can be used for exact classification of movements or artists. We put these descriptors into classification task in order to see whether there are some tendencies.

Table 15 and Figure 73 show the accuracies of different classifiers, based on harmonies/contrast descriptors with movements and artists' names as class labels.

Table 15. Accuracy of different classifiers – based of harmonies/contrast descriptors

Database	**OneR**	**JRip**	**J48**	**PGN**
movements	27.83	36.67	45.00	41.67
artists' names	15.50	21.33	30.17	29.83

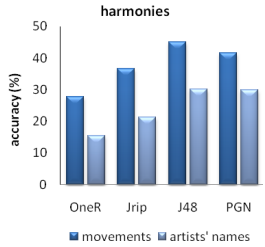

Figure 73. Accuracy of different classifiers – based on harmonies/contrast descriptors

Taking into account the facts that the features are too global and the numbers of class labels are great, we receive not bad results. Here the best classification model stands J48, following by PGN.

Considering the movements results we see that OneR now fails to predict Icons but is able to predict Modern Art. Impressionism is classified well due to the frequent use of partial triads in natural paintings.

The more detailed analysis on confusion matrices, presented in Figure 74 and Figure 75, shows also that in this case rule-based classifiers OneR and JRip do not produce good classification models, creating rules that

predominate Renaissance in the case of movements, as well as Icons in the case of artists' names. In general we can say that the harmonies/contrast features are too global and not the best choice for classifying artist's paintings. The paintings of Rembrandt and Sisley are exceptions of this general rule.

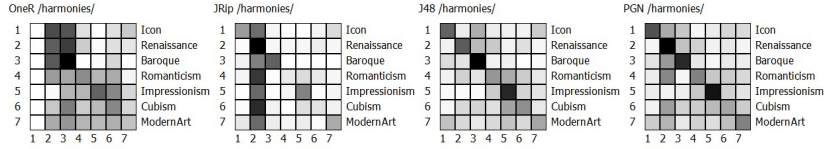

Figure 74. Confusion matrices for harmonies/contrast features, movements as class labels

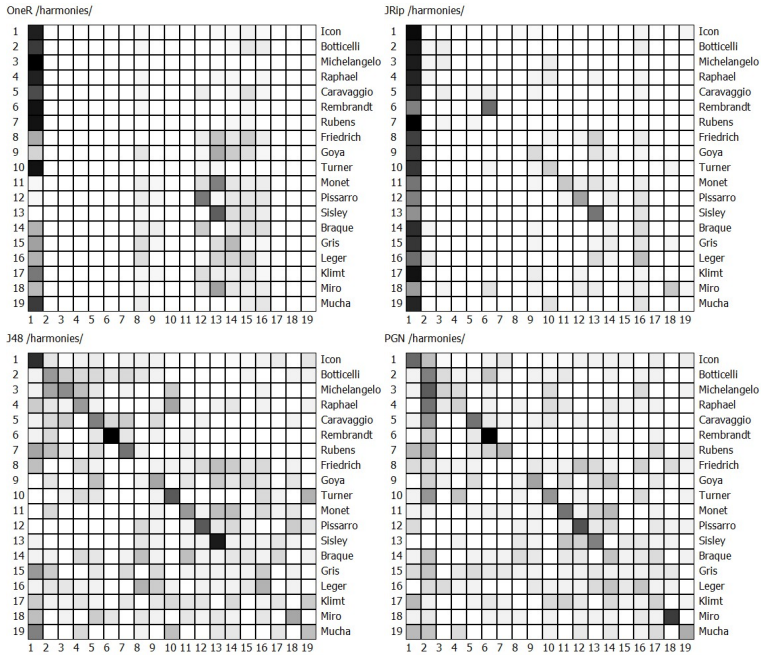

Figure 75. Confusion matrices for harmonies/contrast features, artists' names as class labels

6 Metric Categorization Relations Based on Support System Analysis

The problem of resolving the gaps computer analysis and human understanding has a deep philosophical ground even in the problem of understanding between humans. Lewis Carroll in his book "Through the Looking-Glass" gives us an example of absurd using of the semantic and pragmatics when Humpty Dumpty talks with Alice: "When I use a word it means just what I choose it to mean – neither more nor less".

6.1 The Semantic Complexity

In the ground of the semantic lays the definition of the concepts as name and corresponded content. Our life is fulfilled with learning of the concepts (their names and contents) in order to understand each other. In recent years, when the computer systems stand as part of our interaction, we are faced with the necessity of build the intelligent interface, which includes using of the concepts in the computer systems with the same manner that we accustomed to use. There exist many well suited theories in the area of the concept formation based on the attribute models. But when the computer finds a typical combination of attributes, which defines something, it does not know how to name it.

Our approach extends the learning set with additional metadata, which will be used as a base for finding appropriate names of defined concepts. Of course the problems of the semantic and semiotic cannot be reduced only to the interface – it concerns the expressing the understanding of the world in the opposite side (in this instance – the intelligent system). One very good article, which discusses these questions, is [Scherp and Jain, 2008]. In other words we give the system some description of the real world and we expect that the system will follow our mental model and will generate appropriate names of concepts that arise. In the same time we expect that the system will explain the decision it has made will show these parts of the mental model, which it has used to generate the names of the concepts. This interaction with the intelligent system helps us to improve our mental model and to develop it by extending and/or changing any of its parts. Analysis the answers of the system may show that our mental model does not correspond to the real world. As a result of this process of information interaction with the intelligent system one will receive the possibility to improve the attribute space of some observed area.

In the application areas often is seen that one combination of given feature values defines some concept and in the same time other combination of values of these features are not essential. From other side, we often strive to choose class-section in such way that all values to be subordinated to a common rule, which comes from our sense of order. For instance, if we want to divide the space of art paintings, we usually use such class as "movement" or "artist", etc. But the analysis of colour distribution for different artists [Ivanova et al, 2008] shows that there are some artists, for instance Gauguin and Giotto, which use similar colours in all his life and the others, like Velazquez and Rubens, have great disperse from theirs meaning vectors. And this can be a signal that the work of such authors has to be divided in more short periods. Typical instance of this direction is the variety of styles of Picasso.

6.2 Meta-PGN: Algorithm Description

The basic ideas of the proposed approach were presented in [Ivanova et al, 2009]. Shortly the application of such algorithm in the field art-painting image analysis can be presented as follows:

- The observed set of objects is described by a set of primary measurable features and is classified by a number of viewpoints – complementary set of classifiers. All this information is given in a form of a table which consists of two column parts respectively – descriptive or regular part, and metadata part;
- The primary features, which participate in descriptive part in our case, are automatically extracted for given object, derived from the low and intermediate semantic analysis of the paintings. We use different types of attributes; some of them represent meaning characteristics of some low level visual data; other are given as a result of clustering of MPEG-7 descriptors of tiles of paintings; another derived from intermediate semantic analysis such as colour harmonies and contrasts, which are described in [Ivanova and Stanchev, 2009].

Metadata part contain classification values of different viewpoints – for example: the subject of the painting (landscape, portrait, scene, still life, etc.), the artist (Leonardo, Rubens, Picasso, etc.), the movement (Gothic, Renaissance, Impressionism, etc.) and so on. This metadata can be:

- Automatically extracted from the context – for instance in some cases the artist name presents in the file name, which contains painting;
- Manually annotated – for example the subject of the paintings can be inputted by an expert for the vectors in training set;
- As a result of secondary processing using already given features and corresponded thesauri – for instance the movement, which the

painting belongs to, is determined on the base of the artist (sometimes using the year of the paintings) in one hand, and the defined in advance ontology of the movements, schools and artists in second hand.

The set of records is partitioned into disjoint (sometime intersecting) classes. A *class* is a subset of the records set consisting of all objects that satisfy the *class condition*. The *classification problem* is to classify new objects, i.e. to construct decision rules that describe objects of each class. The *decision rule* is an operator making a decision about the classification of objects. Classification description might be derived to a *simplified* form which is search effectiveness. The metadata part of description consists of tuples of values that correspond to the concepts in the chosen classifier relevant to the column.

We address the problem of understanding and modeling the realistic interconnections between objective categorization given by a set of classifiers and initial object descriptions given by sets of features and native relations of these descriptions.

The overall model is based on several hypotheses which appear in application area analysis. We differentiate three cases – one or several *metrics* (similarity measures) are given as the *application area properties*, and they have to be treated as de facto descriptors of the categorization; a large parameterized set of distances is described over the features sets and several *formal optimization* functional are used to estimate the *correspondence of distances to the categorization*; and finally when several *hypotheses* were found *in terms of* application area *categorization* structures, which helps to find the best distance in the parameterized family of distances.

6.3 Program Realization

For implementing of presented idea we propose an extension of classical classification methods with the purpose of using of metadata for automatic concept identifying of the founded regularities by the system. We have used as a ground a part of already realized classification algorithm PGN in the experimental data mining system PaGaNe [Mitov et al, 2009a].

The enhanced algorithm meta-PGN uses feature vectors with the different structure – the values of preliminary pointed class are equal (all vectors belong to the one general class, which represents the examined area as a whole). Beside of this the vectors contain a second part of values of metadata domains.

At the first stage the learning set is processed by the standard PGN classifier. As a result we receive a set of patterns that define specific

frequently occurred combinations of attributes. Each pattern is connected with the instances from the learning set, which were participated in its creation. The next step consists of traversing of all metadata positions and finding for each value of these positions one or several patterns that correspond to this value. The patterns, connected with examined metadata value, are additionally processed in order to throw out some patterns that are comprised in other ones in the group.

For every metadata value (that define some concept) can be found zero, one or more corresponded patterns. The reason that corresponded patterns not exist usually lays in the fact that chosen primary attributes are not enough to correctly define this concept. If metadata value is connected only with one pattern – we can assume that this is the exact name of this pattern. The content of this concept is represented as a conjunction of significant values of attributes, contained in corresponded pattern. Of course, here also exists risk that primary attributes not represent correctly the examined area (but this is the problem of classification in general). If the value is connected with more patterns – it is represented as a disjunction of conjunctions of significant values of attributes, contained in connected patterns.

6.4 The Next Step: Application in the Field

The process of searching typical combinations, which have to be associated with some concepts, has two sides. From one side this is easy tool for preliminary analysis of the objects features and theirs combinations and class representatives (from different point of view), which is interesting for examining attribute space. This not abolishes the factor analysis algorithms, but vastly relieves the work by throwing off a part of variants and proposing the combinations to be further examined. From the other side the processes of knowledge formation can be extended with supplying of additional metadata, which are used as a base for finding appropriate names of defined concepts.

Some of the concepts can be described in common hierarchical structure and then more abstract concepts inherit properties of their successors. Other concepts may not enter in this hierarchical description as they are connected with different (non-hierarchical) connections with other concepts.

7 Conclusion

In the last decade the core activities of cultural heritage institutions have focused on digitizing and providing access through the World Wide Web to digital resources. The intellectualized access to these resources use data, extracted from the digital object or metadata added to this

object in order to separate "essential" and "unessential" parts of them and to find the most similar objects to the query given by the user using decision algorithms implemented in the cortex of the intellectualized algorithms. In this chapter we explained and demonstrated some automatic metadata generation methods.

The experiments of a harvesting method showed the ability of to build generalized automata for parsing web documents. There appears to be a trade-off relation between the algorithm's performance and the structural variance of the information to be extracted. As expected, e-mail addresses show the highest recall and precision and achieve high accuracy with a small cardinality of the learning set. The main reason for it is probably the existence of a strict and short structure for an e-mail address which leads to little variety in the different element instances. Bulgarian addresses show the worse results. Given the extremely wide variety of the indirect representation of Bulgarian addresses the results for this element are still very promising. Our results are compatible with [Cohen, 2004]. The differences are coming from different languages and different grammars' structure in the languages.

As a metadata mining approach we developed a new association rule miner, which is very specific in it's coding of the items and is special in its preference on confidence. This coding uses the possibilities of direct access to the information via coordinate vectors into multidimensional numbered information spaces. So, the structure combines the convenience of the work with array structures with economy and performance of tree structures, which lies in the ground of realized access method.

Due to this specific coding statistics for min-support for each area can be derived separately (the amount of space is equal to the number of elements in combinations), which allows to give for further analysis different min-support for different numbers of elements in combinations. This new functionality has a high application value. Setting one min-support threshold has always in the past limited the application value of association rules. The extracting rules from frequent datasets as well as their extension in the direction of class association algorithms can be used for defining semantic profiles of observed phenomena – movement, artists style or thematic. With these structures a classifier called PGN has been implemented. The experimental results are very positive and show that PGN is competitive with classification methods that build similar classification models. At the same time, it has the advantage over the other classifiers that it is parameter free. In general, the results provide evidence that the confidence-first approach yields interesting opportunities.

Bibliography

[Agarwal et al, 2000] Agarwal, R., Aggarwal, C., Prasad V.: A tree projection algorithm for generation of frequent item-sets. In Journal of Parallel and Distributed Computing, 61/3, 2000, pp. 350-371.

[Agrawal and Srikant, 1994] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules, In Proc. of the 20th Int. Conf. on VLDB, Santiago, Chile, 1994, pp. 487-499.

[Agrawal et al, 1993] Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD ICMD, Washington, DC, USA, 1993, pp. 207-216.

[Antonie et al, 2006] Antonie, M.-L., Zaiane, O., Holte, R.: Learning to use a learned model: a two-stage approach to classification. In Proc. of the 6th Int. Conf. on Data Mining, 2006, IEEE, Washington, DC, USA, 2006, pp. 33-42.

[Baltes, 1992] Baltes, J.: Symmetric Version Space Algorithm for Learning Disjunctive String Concepts. Technical Report 92/469/06, University of Calgary, Calgary, Alta, March 1992.

[Bergmark, 2000] Bergmark, D.: Automatic Extraction of Reference Linking Information from Online Documents. CSTR 2000-1821, November 2000.

[Berry et al, 2004] Berry, P., Harrison, I., Lowrance, J., Rofriguez, A., Ruspini, E., Thomere, J., Wolverton, M.: Link Analysis Workbench. Technical Report, Air Force Research Laboratory Information Directorate, Rome Research Site, Rome, New York, September 2004.

[Bodon and Ronyai, 2003] Bodon, F., Ronyai, L.: Trie: an alternative data structure for data mining algorithms. In Mathematical and Computer Modelling, 38/7, 2003, pp. 739-751.

[Bodon, 2003] Bodon, F.: A fast Apriori implementation. IEEE ICDM Workshop on FIMI, Melbourne, Florida, USA, 2003.

[Cardinaels et al, 2005] Cardinaels, K., Meire, M., Duval, E.: Automating Metadata Generation: the Simple Indexing Interface. In Proc. 14th Int. Conf. on World Wide Web (Chiba, Japan, May 10--14, 2005). WWW '05. ACM, New York, NY, 2005, pp. 548-556.

[Ciravegna, 2001] Ciravegna, F.: Adaptive information extraction from text by rule induction and generalisation. IJCAI 2001, pp. 1251-1256.

[Coenen and Leng, 2005] Coenen, F., Leng, P.: Obtaining best parameter values for accurate classification. In Proc. ICDM'2005, IEEE, pp. 597-600.

[Cohen, 2004] Cohen, W.: Minorthird: Methods for Identifying Names and Ontological Relations in Text Using Heuristics for Inducing Regularities from Data. 2004. http://minorthird.sourceforge.net

[Demsar, 2006] Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res., 7, 2006, pp.1-30.

[Depaire et al, 2008] Depaire, B., Vanhoof, K., Wets, G.: ARUBAS: an association rule based similarity framework for associative classifiers. IEEE Int. Conf. on Data Mining Workshops, 2008, pp.692-699.

[English et al, 2002] English, J., Hearst, M., Sinha, R., Swearingen, K., Yee, K.: Flexible search and navigation using faceted metadata. Jannuary 2002.

[Fayyad et al, 1996] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: an overview. In Advances in Knowledge Discovery and Data Mining. American Association for AI, Menlo Park, CA, USA, 1996, pp.1-34.

[Frank and Asuncion, 2010] Frank, A. Asuncion, A.: UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science, 2010.

[Friedman, 1997] Friedman, J.: Data mining and statistics: what is the connection? Keynote Address, 29th Symposium on the Interface: Computing Science and Statistics, 1997.

[Greenberg, 2004] Greenberg, J.: Metadata extraction and harvesting: a comparison of two automatic metadata generation applications. Journal of Internet Cataloging, 6/4, 2004, pp. 59-82.

[Han and Kamber, 2006] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufman Publ., Elsevier, 2006.

[Han and Pei, 2000] Han, J., Pei, J.: Mining frequent patterns by pattern-growth: methodology and implications. ACM SIGKDD Explorations Newsletter 2/2, 2000, pp. 14-20.

[Holt and Chung, 2002] Holt, J., Chung, S.: Mining association rules using inverted hashing and pruning. Information Processing Letters Archive, 83/4, 2002, pp. 211-220.

[Hurtut, 2010] Hurtut, T.: 2D artistic images analysis, a content-based survey, 2010, http://hal.archives-ouvertes.fr/hal-00459401_v1/, 10.01.2011.

[IBM, 2009] Trainable Information Extraction Systems. http://researchweb.watson.ibm.com/IE/

[Inokuchi et al, 2003] Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: mining graph data. In Machine Learning, Vol.50, 2003, pp. 321-354.

[Ivanova and Stanchev, 2009] Ivanova, K., Stanchev, P.: Color harmonies and contrasts search in art image collections. First Int. Conf. on Advances in Multimedia MMEDIA 2009, Colmar, France, pp. 180-187.

[Ivanova et al, 2008] Ivanova, K., Stanchev, P., Dimitrov, B.: Analysis of the distributions of color characteristics in art painting images. Serdica Journal of Computing, 2/2, 2008, Sofia, pp. 111-136.

[Ivanova et al, 2009] Ivanova, K., Mitov, I., Markov, K., Stanchev, P., Vanhoof, K., Aslanyan, L., Sahakyan, H.: Metric categorization relations based on support system analysis. In Proc. of the VIIth Int. Conf. "Computer Science and Information Technologies", Yerevan, Armenia, 2009, pp.85-88.

[Ivanova et al, 2010] Ivanova K., Stanchev P., Vanhoof K. Automatic tagging of art images with color harmonies and contrasts characteristics in art image collections. Int. J. on Advances in Software, 3/3&4, 2010, pp. 474-484.

[Kerber, 1992] Kerber R.: Discretization of numeric attributes. Proc. of the 10th National Conf. on Artificial Intelligence, MIT Press, Cambridge, MA, 1992, pp.123-128.

[Klink et al, 2000] Klink, S., Dengel, A., Kieninger, T.: Document structure analysis based on layout and textual features. In: Proc. of Fourth IAPR International Workshop on Document Analysis Systems, 2000, pp. 99–111.

[Klosgen and Zytkow, 1996] Klosgen, W, Zytkow, J.: Knowledge discovery in databases terminology. In Advances in Knowledge Discovery and Data Mining. AAAI Press, 1996, pp.573-592.

[Kohavi, 1995] Kohavi, R.: A study of cross validation and bootstrap for accuracy estimation and model selection. Int. Joint Conf. on Artificial Intelligence IJCAI, 1995.

[Kouamou, 2011] Kouamou, G.: A software architecture for data mining environment. Ch. 13 in New Fundamental Technologies in Data Mining, InTech Publ., 2011, pp.241-258.

[Kuramochi and Karypis, 2001] Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In Proc. of the 1st IEEE Int. Conf. on DM, 2001, pp. 313-320.

[Li et al, 2001] Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multiple class-association rules. In: Proc. of the IEEE Int. Conf. on Data Mining ICDM, 2001, pp.369-376.

[Li et al, 2008] Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., Jagadish, H.: Regular expression learning for information extraction. Proc. of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, 2008, pp. 21–30.

[Liu et al, 1998] Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In Knowledge Discovery and Data Mining, 1998, pp.80-86.

[Liu et al, 2003] Liu, G., Lu, H., Yu, J., Wang, W., Xiao, X.: AFOPT: An efficient implementation of pattern growth approach. In Workshop on Frequent Itemset Mining Impl. (FIMI' 03), 2003.

[Maimon and Rokach, 2005] Maimon, O., Rokach, L.: Decomposition Methodology for Knowledge Discovery and Data Mining. Vol. 61 of Series in Machine Perception and Artificial Intelligence. World Scientific Press, 2005.

[Mao et al, 2004] Mao, S., Kim, J., Thoma, G.: A dynamic feature generation system for automated metadata extraction in preservation of digital materials. In: Proc. of the 1st Int. Workshop on Document Image Analysis for Libraries, vol. 225, IEEE Computer Society, Los Alamitos, 2004.

[Markov et al, 2008] Markov, K., Ivanova, K., Mitov, I., Karastanev, S.: Advance of the access methods. Int. J. Information Technologies and Knowledge, 2/2, 2008, pp. 123-135.

[Markov, 2004] Markov, K.: Multi-Domain information model. In Int. J. Information Theories and Applications, 11/4, 2004, pp. 303-308.

[Martines and Morale, 2002] Martines, F., Morale, F.: Investigation of metadata applications at Palermo astronomical observatory. Library and Information Services in Astronomy IV, 2002.

[Mitchell, 1997] Mitchell, T.: Machine Learning, McGraw-Hill, 1997.

[Mitov et al, 2009a] Mitov, I., Ivanova, K., Markov, K., Velychko, V., Vanhoof, K., Stanchev, P.: PaGaNe – a classification machine learning system based on the multidimensional numbered information spaces. Proc. of 4th Int. Conf. ISKE 2009. Publ. in World Scientific Proc. Series on CEIS, N.2, 2009, pp. 279-286.

[Mitov et al, 2009b] Mitov, I., Ivanova, K., Markov, K., Velychko, V., Stanchev, P., Vanhoof, K.: Comparison of discretization methods for preprocessing data for pyramidal growing network classification method. In Int. Book Series Inf. Science & Computing –No: 14. New Trends in Intelligent Technologies, 2009, pp. 31-39.

[Mitov et al, 2011] Mitov, I., Ivanova, K., Depaire, B., Vanhoof, K.: ArmSquare: an association rule miner based on multidimensional numbered information spaces. Proc. of First Int. Conf. IMMM, Barcelona, Spain, 2011, pp.143-148.

[Mitov, 2011] Mitov, I.: Class Association Rule Mining Using Multi-Dimensional Numbered Information Spaces. PhD Thesis, Hasselt University, Belgium, 2011.

[Morishita and Sese, 2000] Morishita, S., Sese, J.: Transversing itemset lattices with statistical metric pruning. In Proc. of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2000, pp. 226-236.

[Özel and Güvenir, 2001] Özel, S., Güvenir, H.: An algorithm for mining association rules using perfect hashing and database pruning. In Proc. of the TAINN, 2001, pp. 257-264.

[Park et al, 1995] Park, J., Chen, M., Yu, P.: An effective hash based algorithm for mining association rules. In ACM SIGMOD Int. Conf. on Management of Data, 24/2, 1995, pp. 175-186.

[Pei et al, 2001] Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., Yang, D.: Hmine: Hyper-structure mining of frequent patterns in large databases. In Proc. of IEEE ICDM, 2001, pp. 441-448.

[Quinlan, 1993] Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Rak et al, 2005] Rak, R., Stach, W., Zaiane, O., Antonie M.-L.: Considering re-occurring features in associative classifiers. In Advances in Knowledge Discovery and Data Mining, LNCS, Vol. 3518, 2005, pp.240-248.

[Rozenfield et al, 2008] B. Rozenfield, R. Feldman. Self-supervised relation extraction from the Web. Knowledge and Information Systems, 17/1, Springer-Verlag New York, Inc. USA, 2008, pp. 17-33.

[Scherp and Jain, 2008] Scherp, A., Jain, R.: Towards an ecosystem for semantics. Proc. of 1st ACM Workshop on the Many Faces of Multimedia Semantics – MS'07, 2007, pp. 3-11.

[Shi et al, 2003] Shi, R., Maly, F., Zubair, M.: Automatic metadata discovery from non-cooperative digital libraries. In Proc. of IADIS Int. Conf. on e-Society, Lisbon, Portugal, 2003.

[Shreve et al, 2003] Shreve, Gregory M., Zeng, M.: Integrating resource metadata and domain markup in an NSDL Collection. Institute for Applied Linguistics, School of Library & Information Science; Kent State University, 2003.

[Tang and Liao, 2007] Tang, Z., Liao, Q.: A new class based associative classification algorithm. IAENG Int. Journal of Applied Mathematics, 36/2, 2007, pp.15-19.

[Taylor, 1982] Taylor J.: An Introduction to Error Analysis. University Science Books, Mill Valley, California, 1982.

[Thabtah et al, 2005] Thabtah, F., Cowling, P., Peng, Y.: MCAR: multi-class classification based on association rule. In Proc. of the ACS/IEEE 2005 Int. Conf. on Computer Systems and Applications, 2005, p.33.

[Wang and Karypis, 2005] Wang, J., Karypis, G.: HARMONY: eficiently mining the best rules for classification. In Proc. of SDM, 2005, pp.205-216.

[Witten and Frank, 2005] I. Witten, E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[Yan and Han, 2002] Yan, X., Han, J.: gSpan: Graph-based structure pattern mining. In Proc. of the 2nd IEEE Int. Conf. on DM, 2002, pp. 721-724.

[Yin and Han, 2003] Yin, X., Han, J.: CPAR: classification based on predictive association rules. In SIAM Int. Conf. on Data Mining (SDM'03), 2003, pp.331-335.

[Yuan and Huang, 2005] Yuan, Y., Huang, T.: A Matrix algorithm for mining association rules. LNCS, Vol.3644, 2005, pp. 370-379.

[Zaiane and Antonie, 2002] Zaiane, O., Antonie, M.-L.: Classifying text documents by associating terms with text categories. J. Australian Computer Science Communications, 24/2, 2002, pp.215-222.

[Zaiane and Antonie, 2005] Zaiane, O., Antonie, M.-L.: On pruning and tuning rules for associative classifiers. In Proc. of Int. Conf. on Knowledge-Based Intelligence Information & Engineering Systems, LNCS, Vol.3683, 2005, pp. 966-973.

[Zaki et al, 1997] Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of assoctiation rules. In 3rd Int. Conf. on KD and DM, 1997, pp. 283-286.

[Zimmermann and De Raedt, 2004] Zimmermann, A., De Raedt, L.: CorClass: Correlated association rule mining for classification. In Discovery Science, LNCS, Vol.3245, 2004, pp.60-72.

Krassimira Ivanova, Milena Dobreva, Peter Stanchev, George Totkov
(editors)

Access to Digital Cultural Heritage:

Innovative Applications of Automated Metadata Generation

Plovdiv University Publishing House "Paisii Hilendarski"
2012, Plovdiv, Bulgaria

Access to Digital Cultural Heritage:
Innovative Applications of Automated Metadata Generation

Edited by:
Krassimira Ivanova, Milena Dobreva, Peter Stanchev, George Totkov

Authors (in order of appearance):
Krassimira Ivanova, Peter Stanchev, George Totkov, Kalina Sotirova, Juliana Peneva, Stanislav Ivanov, Rositza Doneva, Emil Hadjikolev, George Vragov, Elena Somova, Evgenia Velikova, Iliya Mitov, Koen Vanhoof, Benoit Depaire, Dimitar Blagoev

Reviewer: Prof., Dr. Avram Eskenazi

Published by: Plovdiv University Publishing House "Paisii Hilendarski"
2012, Plovdiv, Bulgaria
First Edition

The main purpose of this book is to provide an overview of the current trends in the field of digitization of cultural heritage as well as to present recent research done within the framework of the project D002-308 funded by Bulgarian National Science Fund. The main contributions of the work presented are in organizing digital content, metadata generation, and methods for enhancing resource discovery.

Printed in Bulgaria by Plovdiv University
24, Tsar Assen, Str., Plovdiv-4000, Bulgaria

ISBN: 978-954-423-722-6
Plovdiv, 2012

Edited by K. Ivanova, M. Dobreva, P. Stanchev, G. Totkov

ACCESS TO DIGITAL CULTURAL HERITAGE

Innovative Applications of Automated Metadata Generation

Plovdiv, 2012