LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK
LEHR- UND FORSCHUNGSEINHEIT FÜR PROGRAMMIER- UND MODELLIERUNGSSPRACHEN
Institut für Informatik

Bachelor-Thesis
in Informatik

Association Rule Generation and Evaluation of Interestingness Measures for Artwork Tags

Frederic Sautter

Aufgabensteller:	Prof. Dr. Bry, Clemens Schefels, Arthur Zimek
Betreuer:	Clemens Schefels, Erich Schubert, Arthur Zimek
Abgabedatum:	18.11.2016

Erklärung

Hiermit versichere ich, dass ich diese Bachelor-Thesis selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 18.11.2016

.....
Frederic Sautter

Zusammenfassung

Assoziationen zwischen Gruppen von Werten in einem Datensatz zu finden wird als das Problem der Assoziationsregelanalyse bezeichnet. Dies kann in die zwei Teilprobrobleme Frequent Itemset Mining und Assoziationregel Generierung zerlegt werden. Die Open Source Data Mining Anwendung ELKI, die für Experimente im Zuge dieser Arbeit verwendet wird, unterstützt bis jetzt nur das erste Teilproblem. Deshalb ist die Implementierung von Assoziationsregel Generierung und deren Integration Teil dieser Arbeit.

Das ARTigo-Projekt ist eine Online-Plattform, die dazu gedacht ist eine semantische Kunstwerk-Suchmaschine zu betreiben. Indem Nutzer auf der Webseite des Projekts spezielle Spiele spielen, steuern sie den Kunstwerken beschreibende Tags bei. Diese gesammelten Stichwörter können anhand von verschiedenen Data Mining-Methoden analysiert werden. Diese Arbeit konzentriert sich auf die Assoziationsregelanalyse der ARTigo-Daten unter Verwendung von verschiedenen Interessantheitsmaßen.

Der Hauptteil dieser Arbeit besteht in der Evaluation von geeigneten Interessantheitsmaßen für die ARTigo-Daten. Dafür werden zehn Interessantheitsmaße als Erstes auf Ähnlichkeiten untersucht. Anschließend wird ihre Fähigkeit, uninteressante Regeln zu eliminieren und interessante zu identifizieren, verglichen. Zuletzt werden die tausend am höchsten eingeordneten Regeln anhand von fünf vorgeschlagenen Eigenschaften analysiert.

Abstract

Finding associations between groups of items within a data set is referred to as the problem of association rule mining. This can be decomposed into the two subproblems frequent itemset mining and association rule generation. The open source data mining framework ELKI used throughout this work for experiments so far only supports the first subproblem. Therefore the implementation of association rule generation and its integration in ELKI is part of this work.

The ARTigo project is an online platform with the purpose of powering a semantic artwork search engine. By playing special games on the projects website users contribute descriptive tags to artworks. These collected keywords can be analyzed using various data mining techniques. This work concentrates on applying association rules mining to the ATRigo dataset, using various interestingness measures.

The main part of this thesis is to evaluate appropriate interestingness measures for the given ARTigo data. Therefore, ten interstingness measures are first searched for similarities. Second, their ability of eliminating uninteresting and locating interesting association rules is compared. Last their one thousand highest ranked rules are analyzed by the means of five proposed properties.

Contents

1 Introduction **3**

2 Related Work **5**

3 Theoretical Foundations **7**
3.1 Association Rules 7
3.2 Frequent Itemset Mining 8
3.2.1 Apriori Algorithm 10
3.2.2 FPGrowth Algorithm 12
3.3 Association Rule Generation 14
3.4 Interestingness Measures 16
3.4.1 Certainty factor 18
3.4.2 Conviction 19
3.4.3 Klosgen 20
3.4.4 Leverage 20
3.4.5 Lift 20
3.4.6 Cosine 21
3.4.7 Jaccard 21
3.4.8 Gini index 21
3.4.9 J-Measure 21

4 Experimental Settings **23**
4.1 ARTigo 23
4.2 Data Description 25
4.3 ELKI 26
4.4 Extension 27

5 Experiments and Results **29**
5.1 Data Preparation 29
5.2 Evaluation 31
5.2.1 Similarities between Measures 31
5.2.2 Comparison of Interesting and Uninteresting Rules . . . 32

5.2.3 Comparison of the 1000 Highest Ranked Rules 34
5.3 Results 37

6 Conclusion and Future Work **39**

Appendices **41**

A Manual - Associaition Rule Mining in ELKI **42**

B Experiments **43**
B.1 Top Ten Ranked Rules of all Ten Interestingness Measures . . . 43
B.2 Correlation between all Pairs of Measures 47
B.3 Ten Interesting and Ten uninteresting Rules 48

List of Figures **51**

List of Tables **52**

Bibliography **54**

Chapter 1

Introduction

Nowadays almost every modern digital technologie generates some kind of data [Agg15]. Supermarkets and online retailers collect customer purchase data, airlines gather flight information form their customers, websites track their users IP-addresses. This data is collected for a purpose, whether it is for diagnostic, analysis or with some other intention. Especially the web has made it very easy to gather massive amounts of data. This can be done without users realizing or through active contribution of users. The evaluation of such large data collections can provide very interesting and useful knowledge.

The ARTigo project [WBBL13], developed in close collaboration by the art histroy and the computer science departments of the LMU, is an online platform, with the purpose of powering a semantic artwork search engine. Because of the limited possibilities of computerized image search engines, ARTigo relys on human contribution. Therefore users can play several games on the projects website. Players can enter descriptive or by any other means suitable keywords to the displayed images. If the entered tags have been entered by other users as well, this player scores. These tags make it possible for users to search for images according to specific keywords. Similar to other scenarios like collected customer purchase information these gathered tags can be used for further analysis. Applying various data mining techniques, for instance association rule mining could provide interesting new knowledge for art history as well as computer science. This is done by means of the ELKI framework [SKE+15], being an open source data mining software mainly developed by the database systems group of the LMU computer science department. It includes a variety of different implementations of algorithms, evaluation measures, visualization components and other useful data mining utilities.

This work will concentrate on the usage of association rule mining, which is a data mining method with the purpose of finding associations between groups of artwork tags, when applied to the ARTigo dataset. Until now this mining

method is only partly supported by ELKI, consequently implementations for the missing parts have to be added to the framework before the algorithm can be applied to the dataset. Depending on which parameters are passed and what type of data is fed to the algorithm, the number of detected association rules can be very large and the amount of associations providing new and interesting knowledge may vary significantly. Many rules may be redundant or obvious and therefore contain no useful information. For this reason, various metrics, referred to as interestingness measures, have been defined. By applying different measures, it is possible to increase the amount of valuable rules for the given type of data [STXK07]. Thus, evaluating appropriate interestingness measures for the ARTigo data is a major part of this work. Given this special data type, it is of great interest to search the measures for similarities and analyze their performance. Especially since association rule mining as well as the various interestingness measures have been designed for other data types than artwork tags.

The remainder of this work is structured as follows: Chapter 2 presents related work.
Chapter 3 explains the theoretical foundations of association rule mining and its subproblems. Furthermore the interestingness measures, being used for experimentation will be outlined.
In Chapter 4 ARTigo and ELKI are introduced in further detail.
Chapter 5 covers a characterization of the special type of data collected by ARTigo. Additionally experiments together with their results are illustrated.
Finally, Chapter 6 gives a conclusion and proposes ideas as well as concepts for future projects based on this work.

Chapter 2

Related Work

The association rule mining problem is an especially well-studied topic of data mining. A great amount of research, concerning this problem, concentrated on optimizing the mining process. Since first algorithms were unable to mine rules from very large databases, researchers introduced various new algorithms using pruning methods to improve memory efficiency and reduce computational costs. In Chapter 3, research concerning the process of association rule mining, is introduced and two important algorithms are described in detail. So far, most of the research is done using market basket data, but association rule mining does not have to be limited to this specific application area.

For example Merceron et al. [MY08] applied association rule mining to education data. Their data was collected with the use of the learning management system Moodle, of a course consisting of 84 students. On the one hand, the purpose of this research was to gather information about what students do with these extra resources, like slides and exercises, provided to them by the Moodle platform. On the other hand, researchers wanted to find out, which interestingness measure works best on this specific type of data. Therefore, they applied the four different measures support, confidence, cosine and lift to their data. With the use of adequate thresholds for the right interssting-ness measures they were able to achieve useful knowledge. Their experiments showed that when applying cosine, the greatest amount of interesting rules were achieved.

Moreover, an empirical evaluation of the performance of various interestingness measure on a clinical data set, was part of the research, done by Ohsaki et al. [OKO+04]. Therefore, association rule mining was applied to a dataset on hepatitis. The rules, yielded by the different measures, were compared to those considered interesting by a medical expert. This showed that for instance, the interestingness measure Jaccard has a very high performance on this type of data. Although many measures reveal a complementary trend,

some measures were capable of selecting really interesting rules.

Milk and Kender [MK06] introduced another interesting research work, concerning the topic of evaluating the most suitable interestingness measures for a specific type of data. To cluster web images, they used association rule mining. Therefore, they extracted signal features, such as color and format, from their sample images. Additionally, the websites, containing the respective images, were processed to extract keywords. To this data set, consisting of signal features and textual features, association rule mining was applied. Furthermore, 22 interstingness measures were evaluated on their ability to sort out uninteresting rules. This showed the use of measures, such as Correlation Coefficient and Kappa, result in more precise clustering than standard measures like support and confidence.

Chapter 3

Theoretical Foundations

At first association rules may seem very abstract, therefore this Chapter is intended to provide a basic understanding of association rules and association rule mining. Initially, the origin together with the general concept of association rules is explained. Afterwords, frequent itemset mining, the first subproblem of association rule mining, is defined. Additionally two major frequent itemset mining algorithms Apriori and FPGrowth are introduced. The former, being one of the first algorithms introduced, the latter being one of the most efficient ones. Next, association rule generation, the second subproblem, which is build upon the first, is addressed. This algorithm was added to ELKI. Last, the concept of interestingness measures together with some examples are outlined.

3.1 Association Rules

The problem of association rule mining was originally introduced in 1993 by the three scientists Agrawal, Imielinski and Swami [AIS93] in the context of supermarket data analysis. They presented the first algorithm which was capable of processing the large databases of retailers. For example, the management of a supermarket with a huge amount of different products or items has to make important decisions, including what to put on sale, where to place items in the shelves, which customers should be targeted by ads. Furthermore, the supermarket collects customer purchase data at the checkout counter every day. This data is stored in a database. In particular, the database consists of sets of *items*, where one set is referred to as *transaction* and contains all items purchased by a single customer. The general problem of association rule mining is to find associations between groups of items. This indicates, if certain items contained in an itemset X occur, then specific other items from an itemset Y are likely to occur as well. Therefore, an association rule is an

implication expression of the form $X \Rightarrow Y$. X is called *antecedent* and Y is called *consequent* of a rule, X and Y being disjoint. Assuming there is a rule $\{bread, butter\} \Rightarrow \{milk\}$ which means, someone who buys bread together with butter, the items contained in the *antecedent*, will very likely also buy milk, the elements of the *consequent*. This information for example, can help to determin, where to place items in the shelves and thereby helps to improve the quality of these importent management decisions significantly [Agg15].

Since the problem of association rule mining originated in the context of supermarket data analysis, a great amount of terminology, such as *transaction* and *items*, were borrowed from the supermarket analogy. However, meanwhile the problem is applied to a wide range of different types of data and used in all kindes of different subjects, such as bioinformatics, medical diagnosis, web mining, software bug detection and scientific data analysis [PNSK+06], such as in this thesis.

Generally, the association rule mining problem can be decomposed into two subproblems: first, find all combinations of items, which have a certain statistical significance, also known as frequent itemset mining. Second, given a significant itemset, generate all rules which have a certain strength. The following sections will describe the two subproblems in further detail. To describe these two subproblems, the most common model, which applies raw frequency to quantify the strength of rule, will be used. However, these frequencies do not necessarily corresponde to the most interesting rules. Therefore, a variety of different, so called interestingness measures, have been introduced [Agg15]. These will be explained in greater detail in Section 3.4.

3.2 Frequent Itemset Mining

For detecting association rules in a given dataset [Agg15], it is important to first select only those combinations of items which accomplish sufficient relevance. To formally define the frequent itemset mining problem, a few essential definitions have to be made. Consider a database $\mathcal{T}$ consisting of n transactions, as $T_1, ..., T_n$. Each transaction represented by its *transaction identifier* T_i is a *k-itemset*, being a set of items with cardinality k. Every item is an element of the universe $\mathcal{U}$ of items. Referring to the supermarket analogy the transaction database $\mathcal{T}$ would be all purchases T_i done by customers. The universe of items $\mathcal{U}$ would be all products the supermarket has to offer. Thus $\mathcal{U}$ is very large and can have a cardinality of ten thousand or more, whereas the number of items contained in a transaction is normally relatively small. To quantify the frequency of an itemset, we count the number of transactions in the database $\mathcal{T}$ in which the itemset occurs as a subset. This number is

Transaction	**Itemset**
T_1	$\{apple,\ banana\}$
T_2	$\{banana,\ orange,\ peach\}$
T_3	$\{apple,\ peach\}$
T_4	$\{apple,\ banana,\ orange,\ peach\}$
T_5	$\{apple,\ orange,\ peach\}$

Table 3.1: Customer purchase database with five transactions and the universe of items $\mathcal{U} = \{apple,\ banana,\ orange,\ peach\}$.

known as the *support* of an itemset, whereas the fraction of *support*, divided by the total number of transactions, is referred to as the *relative support*.

Somehow correlated items, like bread and butter, are more likely to occur together. Clearly, only few people eat plain bread, thus itemsets containing such items, like bread and butter, have a higher support. In other words, a certain minimal support threshold *minsup* has to be defined, whereby all itemsets whose *support* is greater or equal *minsup* are declared *frequent*. The problem of finding all such *frequent* itemsets with $support \geq minsup$ in a database is referred to as frequent itemset mining [Agg15].

Consider the customer purchase database of a supermarket, given in Table 3.1, and let *minsup* be 3. For example, $\{apple, peach\}$ is frequent, because this itemset occurs in transaction three, four and five as a subset and therefore its *support* is 3, which is greater or equal to *minsup*. Instead the itemset $\{apple, banana\}$ is not frequent, since it only occurs in transaction one and four, thus its support of 2 is less than *minsup*. As a matter of fact, the only two *frequent* 2-*itemsets* are $\{apple, peach\}$ and $\{orange, peach\}$. All other itemsets, except the four 1-*itemsets*, $\{apple\}$, $\{banana\}$, $\{orange\}$ and $\{peach\}$, are not *frequent*.

A brute force algorithm for frequent itemset mining would be to first generate all subsets of the universe of items $\mathcal{U}$. Then the support for every subsets X of $\mathcal{U}$ has to be calculated [ZMJ14]. Therefore, for every itemset X it has to be checked, if it is a subset of any transaction T_i in the database $\mathcal{T}$. To compute the *support* of a single itemset X a complete database scan has to be made. More importantly, there are $2^{|\mathcal{U}|}$ subsets of $\mathcal{U}$, resulting in $2^{|\mathcal{U}|}$ database scans. In short, the I/O complexity of this algorithm is $O(2^{|\mathcal{U}|})$. This approach is unsuitable for real world practice, especially, since $\mathcal{U}$ is usually very large.

3.2.1 Apriori Algorithm

The Apriori algorithm was proposed in 1994 by Rakesh Agrawal and Ramakrishnan Srikant [AS+94]. It provides a significant improvement over the brute force approach, by reducing the amount of wasteful computation and the number of complete database scans. Mainly responsible for the improvement is the use of the *downward closure property*, saying every subset X of a frequent itemset Y is frequent as well. On the contrary, this means for every itemset Y being infrequent, any superset of Y is infrequent as well [Agg15].

The Apriori algorithm starts with counting occurrences of single items in the database to determine the support of all 1-*itemsets*. Those, of which are frequent, are retained in the set of frequent 1-*itemsets* $\mathcal{F}_1$, all others are discarded. Next, all 2-*itemsets* are generated out of the itemsets, contained in $\mathcal{F}_1$. These 2-*itemsets* are possible candidates for frequent itemsets. In general, all candidate itemsets of cardinality k are referred to as $\mathcal{C}_k$. $\mathcal{C}_2$ is then pruned, according to the downward closure property. For the remaining itemsets in $\mathcal{C}_2$, the support is calculated, whereas the frequent itemsets are referred to as $\mathcal{F}_2$. The algorithm goes on with this procedure of generating the set of candidate itemsets with next greater cardinality out of the one smaller frequent itemsets, pruning this candidates set and computing the support for the retained ones [Agg15].

To explain one iteration in detail, assume the algorithm has calculated $\mathcal{F}_k$, all frequent Itemsets of length k. Next step is to generate $\mathcal{C}_{k+1}$, all candidates with length $k+1$. Therefore, pairs of frequent itemsets contained in $\mathcal{F}_k$, which have $k-1$ items in common, are joined. For instance if the two itemsets $\{A, B, C\}, \{A, B, D\} \in \mathcal{F}_k$ like in Figure 3.1, having A and B in common, are joined, the result is $\{A, B, C, D\}$. This special generation process ensures, that many itemsets, which will later turn out not to be frequent, will not be generated in the first place due to the downward closure property. Since $\{A, B, C, D\}$ is generated from two frequent itemsets, this guaranties that two of the subsets, namely $\{A, B, C\}, \{A, B, D\}$ as well as their subsets, are frequent. Consider $\{A, B, D\} \notin \mathcal{F}_k$, then $\{A, B, C, D\}$ will not be generated form these two itemsets, as matter of fact $\{A, B, C, D\}$ is surely not frequent. According to the downward closure property, when $\{A, B, D\}$ is infrequent, then its superset $\{A, B, C, D\}$ is infrequent as well. To avoid one $(k+1)$-candidate being generated more than once, a lexicographical ordering is applied to items and only the $k-1$ first common items are considered for the join. As a consequence the only way of generating $\{A, B, C, D\}$ is by joining $\{A, B, C\}$ and $\{A, B, D\}$, for example not by $\{A, B, C\}$ and $\{B, C, D\}$ [Agg15].

The next step is to prune the set of candidates $\mathcal{C}_{k+1}$, according to the downward closure property. In this way, the amout of candidates can be further

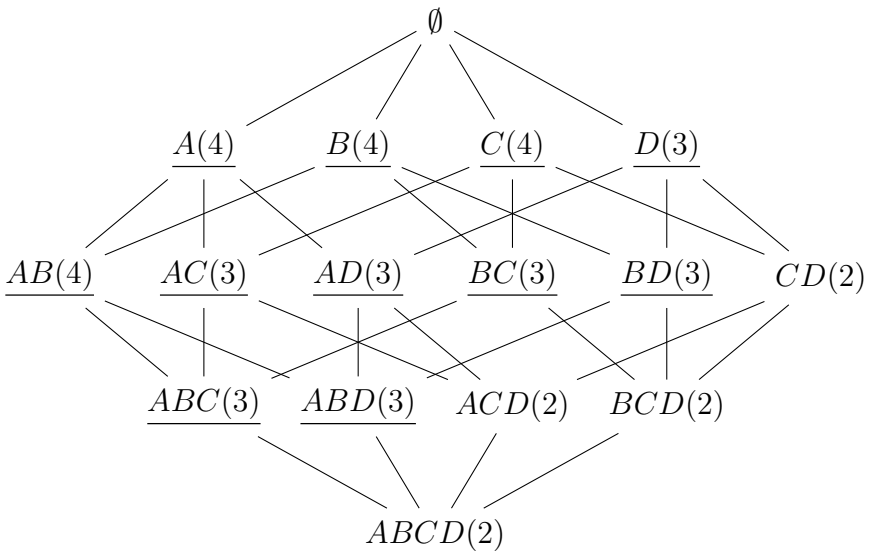

Figure 3.1: All possible itemsets with their support from a transaction database $\mathcal{T}$ = {*ABD, ABCD, ABC, C, ABCD*}, frequent itemsets with *support* ≥ 3 are underlined

reduced. Therefore, the algorithm checks for every itemset $I \in \mathcal{C}_{k+1}$, if all subsets of I with length k exist in $\mathcal{F}_k$. In the previous example $\{A, B, C, D\} \in \mathcal{C}_{k+1}$ was generated from the two frequent itemsets $\{A, B, C\}$ and $\{A, B, D\}$. Assume, the other two subsets of $\{A, B, C, D\}$, $\{A, C, D\}$ and $\{B, C, D\}$, are not in $\mathcal{F}_k$, thus are not frequent. Consequently, $\{A, B, C, D\}$ is removed from $\mathcal{C}_{k+1}$, since not all k-subsets of $\{A, B, C, D\}$ are contained in $\mathcal{F}_k$. This pruning technique, together with the process of candidate generation reduces the amount of itemsets, which are considered for support computation, significantly. As a result, a lot less wasteful computation has to be done [Agg15].

Finally, the support for the remaining itemsets in $\mathcal{C}_{k+1}$ is computed, therefore all subsets with length $k+1$ are calculated for every transaction T_i in the database $\mathcal{T}$. For each calculated subset S it is checked, if S occurs in $\mathcal{C}_{k+1}$, the support of S is incremented when it does. This support computation of all itemsets in $\mathcal{C}_{k+1}$ can be done with one database scan Thus for each iteration only one complete database scan is needed. To further optimize, the support computation process the algorithm makes use of a data structure, referred to as hash tree, to organize candidate itemsets in $\mathcal{C}_{k+1}$. This approach significantly increases the efficiency of support counting. At last, all itemsets, with support less than *minsup* are removed and those with support greater or equal *minsup* are added to $\mathcal{F}_{k+1}$. Now the next iteration starts with $\mathcal{F}_{k+1}$ [ZMJ14].

Compared to the brute force approach, Apriori not only reduces the amount of wasteful computation, but also dramatically decreases the number of costly database scans, resulting in an I/O complexity of $O(|\mathcal{U}|)$ compared to $O(2^{|\mathcal{U}|})$ [ZMJ14].

3.2.2 FPGrowth Algorithm

The FPGrowth algorithm [HPY00], introduced in 2000, is a relatively new approach. The main goal of this algorithm is to avoid the still very costly process of candidate generation, which is used by other approaches such as Apriori. To achieve this, the transaction database $\mathcal{T}$ is indexed by means of an extended prefix tree, referred to as *frequent pattern tree* (FP-tree).

The general concept of the algorithm is first to construct the *frequent pattern tree*, which then is used instead of the original database. Based on this tree, projected FP-trees are built for every frequent item i in $\mathcal{U}$, out of which all frequent itemsets containing i can be generated [ZMJ14].

As described initially, the *frequent pattern tree* has to be constructed out of the transaction database $\mathcal{T}$. Therefore, first of all, the algorithm computes the support of all items i (1-*itemsets*) in $\mathcal{U}$, immediately discarding infrequent items. The frequent items are sorted in decreasing order of support. This lexicographical ordering is applied to every itemset and transaction during the further procedure. Now, the first transaction in $\mathcal{T}$ is sorted by decreasing item support and then inserted into the FP-tree, the root of the tree being the empty set. Every inserted item is labeled with a count, initially being one. For example, the itemset with corresponding item support $\{A(3), B(5), C(4), D(6), E(4)\}$ would be inserted and labeled with one in the following order: first $D(1)$ then $B(1), C(1), E(1)$ and lastly $A(1)$. All other transactions $T_i \in \mathcal{T}$ are inserted into the same tree. Transactions having the first n items with some previously included transaction in common, will follow the exact same path for these first n items, whereas their remaining items will be inserted in new created nodes. Additionally the count of each existing node, which is passed during the insertion of one transaction, is incremented. The *frequent pattern tree* is complete, once all transactions $T_i \in \mathcal{T}$ are inserted as illustrated in Figure 3.2. It is important to note that storing transaction in decreasing order of item support, guaranties items occurring most often will appear at the top of the the tree. As a consequence the tree is as compact as possible [ZMJ14].

The next step of the algorithm is the procedure for building projected FP-trees for every frequent item i in the constructed *frequent pattern tree*. The procedure accepts the prefix P, initially being empty, and the FP-tree R as parameters. The projected FP-trees are built in increasing order of support

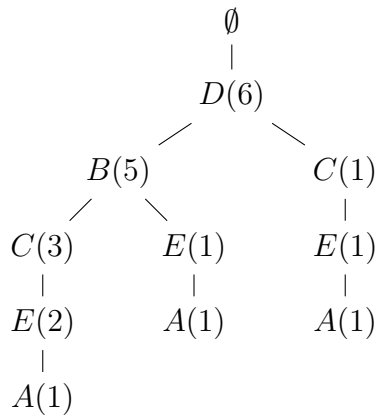

Figure 3.2: Initial *frequent pattern tree* R with count constructed from the transaction database $\mathcal{T} = \{\text{ABDE, BCDE, ABCDE , ACDE, BCD, BD}\}$.

of the corresponding items i in R. The projected FP-tree of one item i and the prefix P is constructed as follows. At first all paths from the root to every occurrence of item i in R are determined. The first path is inserted in the new projected tree $R_{P\cup\{i\}}$, initializing the count of every node in $R_{P\cup\{i\}}$ with the count of the node in R with item i in this particular path. The other paths to i are inserted into $R_{\{i\}\cup P}$ in the same way as the transactions were inserted into the initial *frequent pattern tree*. Except that the count of each node, being passed, is increased with the count of the node in R with item i in this exact path. For example $i = A$, all paths to A in R from Figure 3.2 being $D \to B \to C \to E$, $D \to B \to E$ and $D \to C \to E$ result in the projected FP-tree in Figure 3.3. However, the last item of this path, being i is never inserted, because i is now part of the prefix P. At last all infrequent items are removed from the FP-tree $R_{P\cup\{i\}}$, according to Figure 3.3. The resulting tree $R_{P\cup\{i\}}$ is a projection of the itemset $P\cup\{i\}$. This procedure is now called recursively with the new prefix set $P\cup\{i\}$ and the constructed FP-tree $R_{P\cup\{i\}}$ as parameters [ZMJ14].

The base case of this recursion is reached, when the input FP-tree R is a single path with no branches. In this case all subsets S of this one path are generated and joined with the input prefix of this last call. The support of each of this itemsets S corresponds to the support of the least frequent item in S. All these itemsets are then added to $\mathcal{F}$ [ZMJ14].
Finally, when all recursive calls have reached the base case, the set $\mathcal{F}$ contains all frequent itemsets in $\mathcal{T}$.

FPGrowth has two advantages compared to Apriori and similar approaches: First of all, it avoids the costly candidate generation through successively

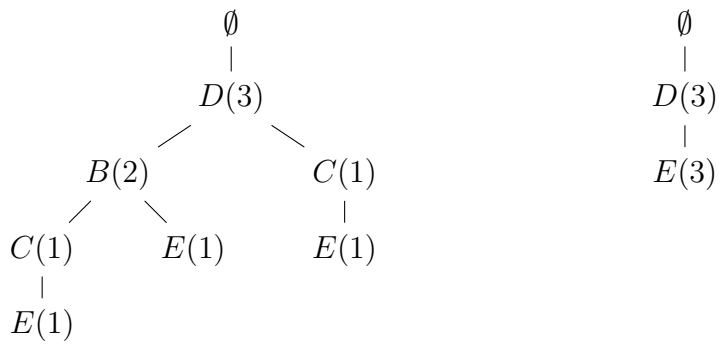

Figure 3.3: Projected FP-tree $R_{\emptyset \cup E}$ constructed from initial FP-tree, before and after pruning infrequent items.

concatenating frequent 1-*itemsets* detected in the FP-tree. Second, the constructed *frequent pattern tree* is very compact, usually being smaller than the original database. Additionally, this saves the cost of database scans. Showing these benefits, FPGrowth has gained a lot of attention and many further improvements to this approach have been introduced [HPY00].

3.3 Association Rule Generation

After finding all itemsets with a certain significance, the second subproblem of association rule mining is to generate rules from these frequent itemsets. Since only relevant rules are of interest, the question arises how to measure this relevance [ZMJ14]. The most common way to quantify this, is by means the measure called confidence. The confidence of a rule, $X \Rightarrow Y$, is known as the conditional probability of a transaction containing Y given that it contains X. This can be calculated according to this formula:

$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{3.1}$$

Since only rules of certain relevance are of interest, similar to *minsup* in frequent itemset mining, a specific minimum confidence threshold is needed to generate only the most relevant rules. The term association rule was already loosely defined in Section 3.1, now that all required formal definitions have been made here the exact formal definition follows.

Definition. *Given two disjoint itemsets X and Y. An implication expression of the form $X \Rightarrow Y$ is referred to as an* ***association rule*** *with a minimum*

support level minsup and a minimum confidence level minconf, if both following criteria are satisfied:

1. $sup(X \cup Y) \geq minsup$
2. $conf(X \Rightarrow Y) \geq minconf$

On the one hand, the first criteria guaranties, that only itemsets supported by a sufficient number of transactions are considered. On the other hand, the second criteria guaranties the conditional probability is high enough, ensuring a sufficient strength of the rule [Agg15]. This makes clear, why the problem of association rule mining can easily be decomposed in two subproblems, the first being *frequent itemset minimg*, the second being *association rule generation*. Whereby, the first part is computational more intensive and most research of the last years was focused on improving frequent itemset mining. The phase of association rule generation is more straightforward.

Given a set of frequent itemsets $\mathcal{F}$ a brute force approach for association rule generation would be to divide every itemset $Z \in \mathcal{F}$ into all combinations of itemsets $X \subset Z$ and $Y = Z \setminus X$, where $X \cup Y = Z$. For every rule $X \Rightarrow Y$ the confidence is calculated. Those rules, satisfying the criteria of minimum confidence are outputted, discarding the others [Agg15].

This brute force approach can be improved, as there is a property, which enables pruning. Therefore the computational expense can easily be reduced. This property, often referred to as *confidence monotonicity* [Agg15], implies for a rule $X \Rightarrow Y$ with $conf(X \Rightarrow Y) < minconf$, that for all subsets $W \subset X$ the confidence $conf(W \Rightarrow Z \setminus W)$ will be less than $minconf$ as well. This ensures the checking of all rules, containing a subsets of X as antecedent, can be avoided, since there confidence is known to be less than *minconf*. The most common algorithm for association rule generation [ZMJ14] makes use of this *confidence monotonicity*. Given the set of frequent itemsets $\mathcal{F}$ it initializes for each itemset $Z \in \mathcal{F}$, with $|Z| \geq 2$, a set of antecedents $\mathcal{A}$, containing all non empty subsets of Z. For each itemset $X \in \mathcal{A}$ the confidence for its corresponding rule $X \Rightarrow Z \setminus X$ is computed. In the case of $conf(X \Rightarrow Z \setminus X)$ being less than *minconf*, all subsets of X are removed from $\mathcal{A}$ according to the *confidence monotonicity*, otherwise the rule is outputted.
This approach for generating association rules is implemented and integrated into ELKI (see Chapter 4. Additionally it is used for experiments in the course of this work (see Chapter 5).

3.4 Interestingness Measures

Association rules obtained by mining procedures, like those described in Section 3.2 and 3.3, have a support of at least *minsup* and a confidence of at least *minconf*. The common use of this frequency based model to quantify the strength of rules is due to its convenience. Using the raw frequencies of itemsets, algorithms can benefit from the *downward closure property* and *confidence monotonicity*, thus very efficient mining algorithms can be designed [Agg15]. However, as mentioned in Section 3.1, it is important to note, that row frequencies do not necessarily correspond to the most interesting rules, especially since in different application areas different rules may be of specific interest. Perhaps these frequencies can even be misleading, this can be illustrated with the following example from Brin, Motwani, and Silverstein [BMS97].

The information shown in Table 3.2 makes it possible to extract the association rule $\{Tea\} \Rightarrow \{Coffee\}$ with a support of 0.15 and a confidence of 0.75. It apparently indicates people drinking tea like to drink coffee as well. Even if this may seem reasonable at first, a closer look reveals that the fraction of people who drink coffee, despite whether they drink tea is 0.80, whereas the fraction of tea drinkers who also drink coffee is just 0.75. As a consequence, knowing that somebody drinks tea actually reduces the probability of the person being a coffee drinker from 0.80 to 0.75.

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

Table 3.2: Contingency table displaying the prefered hot drinks among 1000 people [BMS97].

This Example reveals limitations of the confidence measure, thus it shows the need for other measures. Apart from the limitations of support and confidence a major motivation for defining new measures is the possibility of designing different interestingness measures, that determine specific types of rules for different types of data. Especially since databases grow larger and larger, thus the number of mined rules increases even faster and it becomes impossible for humans to locate interesting rules. For this reason the possibility of computerized elimination of uninteresting rules becomes more and more important. Many interestingness measures have been proposed with the intention of ranking rules interesting for a specific application area highest, facilitating the extraction of interesting rules even form huge databases.

	Y	$\overline{Y}$	
X	$sup(X \cup Y)$	$sup(X \cup \overline{Y})$	$sup(X)$
$\overline{X}$	$sup(\overline{X} \cup Y)$	$sup(\overline{X} \cup \overline{Y})$	$sup(\overline{X})$
	$sup(Y)$	$sup(\overline{Y})$	N

Table 3.3: Contingency table storing frequency counts needed for most interestingness measures with N being the total number of transactions [GH06].

Almost all measures proposed so far are functions of a 2×2 contingency table as shown in Table 3.3 [GH06]. Thereby $sup(A)$ refers to the absolute support. To be able to analyze and categorize all these measures, various properties have been introduced. This work concentrates on the two most widespread sets of properties.

In 1991 Piatetsky-Shapiro [PS91] proposed three principals every interestingness measure should accomplish.

Property 1 The measures value for a rule $X \Rightarrow Y$ should be 0 if X and Y are statistically independent, meaning a rule occurring by chance should be not interesting.

Property 2 The interest value should increase with the support of $X \cup Y$ when the support of X and Y is unchanged. In other words, the greater the correlation between X and Y, the more interesting is the rule.

Property 3 When $sup(X \cup Y)$ and $sup(Y)$ (or $sup(X \cup Y)$ and $sup(X)$) are constant and $sup(X)$ (or $sup(Y)$) decreases, the measures value should increase.

Unlike these three principals which are considered desirable the following set of five properties introduced by Tan et al. in 2002 [TKS02] can help categorize the different measures.

Property 4 The value of the interestingness measure is the same for $X \Rightarrow Y$ and $Y \Rightarrow X$. Measures which accomplish this property are referred to as symmetric measures, whereas measures, where the value changes when interchanging antecedent and consequent, are called asymmetric.

Property 5 When multiplying any row or column of the 2×2 contingency table with a positive factor, the interest value should stay the same.

Property 6 The sign of the value should change when replacing the antecedent or the consequent by its negation, meaning the measure is capable of showing positive and negative correlation.

Property 7 Rules satisfying this property have the same value even if antecedent and consequent are replaced by their negations.

Property 8 Adding transactions, which contain neither antecedent nor consequent do not change the value of the interest measure. In other words the measure only depends on transactions containing X or Y or both.

Tan et al. [TKS02] analyzed 21 different interestingness, showing weather they satisfy these eight measures. Additionally, they were able to categorize these measures, assigning them to three groups. Despite not all measures of a group satisfy the same of these eight properties, they are highly correlated, meaning the correlation value with any other measure in the same group is greater or equal to 0.85. For this reason, at least two measures of every one of the groups identified in Table 3.4 of their paper were chosen for evaluation within this thesis and are described in the remainder of this chapter.

These objective interestingness measures all use specific probabilities to mine association rules. However, rules which were ranked high by these measures do not necessarily have to be interesting. First and foremost, these rules have some very high or low probability, therefore, these highly ranked associations are often obvious rules, which are already known to the user. The actual interestingness of a rule is an important area of research, but so far no consistent formal definition has been found [GH06]. This may occur because of the subjective nature of interestingness. Rules may be declared uninteresting, unless they reveal unexpected information or useful knowledge. Measures based on probability, which guarantee that only these subjective interesting rules will be provided, are hardly possible. It is impotent to note that besides these objective interest measures there are approaches based on humans interactive feedback during the mining process. These methods require a lot of time, since users have on evaluate individual rules presented to them by the system to slowly improve the precision. Therefore, interest measures referred to as subjective interestingness measures, which unlike their objective counter parts are not based on probabilities, are used. Examples for these measures are, for instance, unexpectedness and novelty. We often consider rules, appearing unexpected or new to us, as interesting. This thesis concentrates on evaluating objective measures for the purpose of sorting out as many uninteresting rules as possible from the associations obtained from the ARTigo data set.

3.4.1 Certainty factor

The Certainty Factor was first developed to measure the uncertainty in rules of expert systems [SB75]. Since Berzal et al. [BBV+02] introduced certainty

Measure	P1	P2	P3	P4	P5	P6	P7	P8
Certainty factor	yes	yes	yes	no	no	no	yes	no
Conviction	no	yes	no	no	no	no	yes	no
Klosgen	yes	yes	yes	no	no	no	no	no
Leverage	no	yes	yes	yes	no	no	no	yes
Lift	no	yes	yes	yes	no	no	no	no
Confidence	no	yes	no	no	no	no	no	no
Cosine	no	yes	yes	yes	no	no	no	yes
Jaccard	no	yes	yes	yes	no	no	no	yes
Gini Index	yes	no	no	no	no	no	yes	no
J-Measure	yes	no	no	no	no	no	no	no

Table 3.4: The ten measures used in this work sorted by groups together with the eight properties.

factor as an alternative to confidence in 2002, it was applied in the area of data mining as well. For a rule $X \Rightarrow Y$ it is defined with support being relative support as:

$$CF(X \Rightarrow Y) = \frac{conf(X \Rightarrow Y) - sup(Y)}{sup(\overline{Y})} \tag{3.2}$$

Certainty Factor measures the variation of the probability that a transaction contains Y, taking into account only those transactions containing X [BBV+02]. Certainty Factor satisfies the first three properties and achieves the same value even if the row or columns of the 2×2 contingency table are permuted. This measure can achieve values in the range from -1 to 1, with 0 indicating statistical independence.

3.4.2 Conviction

The Conviction measure was introduced by Brin et al. in 1997 [BMUT97] as alternative to confidence. Conviction of a rule $X \Rightarrow Y$ is defined with support being relative support as:

$$conviction(X \Rightarrow Y) = \frac{1 - sup(Y)}{1 - conf(X \Rightarrow Y)} \tag{3.3}$$

Conviction measures the expected error of a rule [ZMJ14]. In other words, how often X occurs in a transaction which does not contain Y. The conviction considers the antecedent as well as consequent [BMUT97], additionally it is not symmetric, thus it is directional and quantifies actual implication. The

value of conviction ranges from 0 over 1, indicating X and Y are independent, to infinity for perfect implication.

3.4.3 Klosgen

The Klosgen measure was introduced by Klosgen [Klö96]. For a rule $X \Rightarrow Y$ Klosgen is defined with support being relative support as:

$$klosgen(X \Rightarrow Y) = \sqrt{sup(X \cup Y)} \cdot (conf(X \Rightarrow Y) - sup(Y)) \tag{3.4}$$

The Klosgen measure satisfies the first three properties. Therefore its value is 0 when antecedent and consequent are statistically independent. The value of this measure ranges from -1 to 1.

3.4.4 Leverage

The Leverage measure, also referred to as Piatetsky-Shapiro Measure was introduced in 1991 by Piatetsky-Shapiro [PS91]. Leverage is defined for a rule $X \Rightarrow Y$ with support being relative support as:

$$leverage(X \Rightarrow Y) = sup(X \cup Y) - sup(X) \cdot sup(Y) \tag{3.5}$$

This symmetric measure expresses the difference between the actual probability of $X \cup Y$ occurring in a transaction and the probability when X and Y are statistically independent [ZMJ14]. The smallest possible value is -1 and the highest is 1, whereby a value of 0 indicates statistically independence of X and Y.

3.4.5 Lift

The Lift measure also known as interest introduced by Brin et al. in 1997 [BMS97] is one of the most common alternatives to confidence. The Lift is defined for a rule $X \Rightarrow Y$ with support being relative support as:

$$lift(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X) \cdot sup(Y)} = \frac{conf(X \Rightarrow Y)}{sup(Y)} \tag{3.6}$$

This measure can only take on non-negative values. The value 1 indicates antecedent X and consequent Y are statistically independent. The preferred values are much larger or smaller than 1, whereby values less than 1 mean X and Y are negatively correlated and greater values indicate positive correlation. Lift is vulnerable to noise in small datasets, because infrequent itemset can have very high lift values.

3.4.6 Cosine

The Cosine aslo known as IS-Measure introduced by Tan and Kumar in 2000 [TK00] is similar to the definition of lift. The cosine for a rule $X \Rightarrow Y$ is defined as follows, with support being relative support:

$$cosine(X \Rightarrow Y) = \frac{sup(X \cup Y)}{\sqrt{sup(X) \cdot sup(Y)}} \tag{3.7}$$

The possible values of cosine range from 0 to 1. For statistical independence cosine achieves the value of $\sqrt{sup(X) \cdot sup(Y)}$, therefore cosine, not achieving a value of 0 for statistical independence, does not satisfy the first property.

3.4.7 Jaccard

The Jaccard coefficient was introduced by Rijsbergen [vR79] for a rule $X \Rightarrow Y$ is defined according to the following formula:

$$jaccard(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X) + sup(Y) - sup(X \cup Y)} \tag{3.8}$$

This measure quantifies the similarity of antecedent X and consequent Y [ZMJ14], with values between 0 and 1. Higher values indicate greater similarity between X and Y. As a symmetric measure, it satisfies property number four. Additionally the Jaccard coefficient accomplishes property eight, thus it depends only on transaction containing X and Y or both.

3.4.8 Gini index

The Gini index [BFSO84] is used to measure impurity and is defined with support being relative support as follows:

$$\begin{aligned} gini(X \Rightarrow Y) = sup(X) \cdot (conf(X \Rightarrow Y)^2 + conf(X \Rightarrow \overline{Y})^2) + sup(\overline{X}) \\ \cdot (conf(\overline{X} \Rightarrow Y)^2 + conf(\overline{X} \Rightarrow \overline{Y})^2) - sup(Y)^2 - sup(\overline{Y})^2 \end{aligned} \tag{3.9}$$

This measure is not symmetric, thus quantifies real implication [SMAM13]. Its possible values range from 0 to 0.5, where a value of 0 indicates consequent and antecedent are statistically independent.

3.4.9 J-Measure

The J-measure introduced by Smyth and Goodman [GS91] originated in information theory but is now also used as interest measures [GH06]. It is defined

according to following formula with support being relative support:

$$\begin{aligned} j(X \Rightarrow Y) = sup(X \cup Y) \cdot \log\left(\frac{conf(X \Rightarrow Y)}{sup(Y)}\right) \\ + sup(X \cup \overline{Y}) \cdot \log\left(\frac{conf(X \Rightarrow \overline{Y})}{sup(\overline{Y})}\right) \end{aligned} \tag{3.10}$$

This measure is not symmetric and satisfies the first property. The value 0 indicating antecedent and consequent are statistically independent. Additionally is 0 the smallest value this measure can accomplish, with 1 being the greatest value.

Chapter 4

Experimental Settings

This chapter describes the platform, which was used to collect the data. Additionally, the framework used for data mining is introduced. Therefore, Section 4.1 takes a closer look at the ARTigo platform. Afterwards the data set containing the tags collected by means of ARTigo is characterized. In Section 4.3, an overview of the data mining framework ELKI used for evaluation is given. Finally after outlining ELKI the implementations made in the course of this thesis are roughly illustrated.

4.1 ARTigo

ARTigo [WBBL13], as shortly described in the introduction, is a web platform with the purpose of collecting tags, which are used to power a semantic artwork search engine. Thanks to the more than 58,000 images of paintings, fashion artifacts, art photographs, etc. provided by ARTigo, users of this search engine can access a large database of artworks. The metadata needed to ensures a reliable searching, is collected from the "footprints", left by users participating in the games, which are playable on the platform. Such games are referred to as Games With A Purpose (GWAP), meaning the purpose of these games is not only to entertain the player but also, for instance, to gather image tags or other useful tasks. This approach is taken, because without human contributions it would not be possible for software to achieve the quality of search results as provided by ARtigo.

On the ARTigo web platform (`artigo.org`) [WBBL13] users can chose between six different GWAPs, which can be classified in four different categories, being description, dissemination, diversification and integration games. Whereby games, belonging to different categories provide different types of tags. In description games, such as ARTigos main game, players describe a displayed artwork superficially by entering tags. These tags can be related to

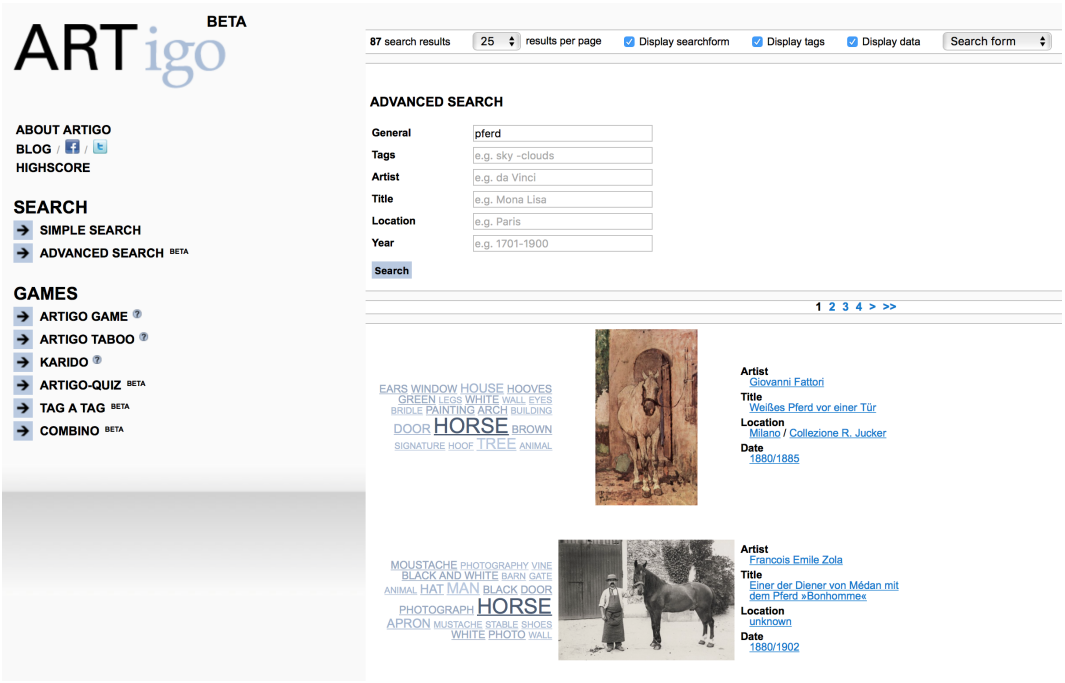

Figure 4.1: Image of the ARTigo search engine

anything visible on the image. The keywords collected by this type of games are mostly obvious descriptions or surface tags. Dissemination games are intended to discriminate artworks, which are similarly tagged. Therefore, users have to add missing tags to these images. Additionally, they can be used to translate tags from one language to another. Games, sorted in the diversification category, generate more precise tags. To achieve this, users have to distinguish between artworks, which have been tagged alike, with new tags. At last integration games cluster tags to accomplish more precise tags. These games are more challenging for participants, since in some cases even detailed knowledge is needed for clustering tags. In most of these games the entered tags are validated if one or more other players have entered the same tag as well. Needed to produce correct data, this approach ensures the gaming platform is not misused.

The combination of the tags provided by these various games, produces a very precise and comprehensive description of artworks, which can only be achieved by means of human intervention. These keywords are primarily used to enable accurate semantic searching on the large database of artworks [WBBL13]. In this work the data provided by ARTigo will be analyzed using association rule mining. In the next section the structure of this data will be characterized in detail.

4.2 Data Description

The data provided by ARTigo consists of natural language, words which the players contribute during the game. In the case of this work the data is obtained from the German version of ARTigo, due to the fact that the German data set is the most mature one. In ARTigo every word a user enters while playing is registered. Since ARTigo is designed to power a artwork search engine a deep data set is preferred. In other words, every tag is stored unchanged in the database, resulting in 4177618 tags for 58901 different artworks. These four million tags consist of 240338 unique ones, which are mostly simple words we use every day, such as FRAU, WEISS or OFFEN, nevertheless more technical vocabulary such as, REALISMUS or artist names such as CLAUDE MONET are contained in the data as well. It is important to note that this data does not consist of well distinguished objects as it is the case in supermarket data, where every item has a unique name. Since the 240338 tags in this data set are syntactically unique but not by any means semantically. The data set contains synonyms as well as different spellings of the same word, such as FOTO and PHOTO. Furthermore misspelled words are contained as well. Due to the presence of synonyms, different spellings of words and misspelled ones finding interesting association rules becomes more difficult as shown in Chapter 5.

In the data set, obtained from ARTigo, each one of the 58901 images is represented by a identifier, being a unique number. Assigned to this identifier are all the tags related to this image. In the context of association rule mining the transaction database $\mathcal{T}$ is this entire data set with all identifiers and tags. Whereby each image together with its related tags is one transaction T_i, the image identifier corresponding to the transaction identifier and the related tags being the items contained in the transaction. Therefore, the transaction database consists of 58901 transactions and in average each transaction consists of about 71 tags (items). Furthermore, the universe of items $\mathcal{U}$ corresponds to the set of all 240338 unique tags found in the database.

To show an other property of the data consider each itemset being represented as a bit vector of the form $(i_1, i_2, i_3, \ldots, i_n)$ with $n = 240338$ being the cardinality of $\mathcal{U}$ and $i_{j\in\{1,2,3,\ldots,n\}} \in \mathcal{U}$ representing one unique item. Each value i_j can either be 1 or 0, indicating the corresponding item is element of this itemset if $i_j = 1$ otherwise the item is not contained in this itemset. Since in average every transaction contains only 71 items (tags) and the universe of items with its over 200 thousand tags is quite large, the itemsets are sparse, meaning the great majority of the values i_j are 0 and only few are 1. This makes sense, since, for instance a picture displaying a ship will have tags assigned to it, concerning ships, which is a relatively small scope compared to

all the things, which could be shown by pictures.

4.3 ELKI

The ELKI framework (**E**nvironment for Deve**l**oping **K**DD-Applications Supported by **I**ndex Structures) [AKZ08] addresses the problem of precisely evaluating the efficiency of newly proposed algorithms, due to the fact that they are often only compared to a small number of related algorithms. Even if the authors provide an implementation to the algorithm, a comparison of these algorithms would still be difficult, since many factors for instance the performance of different programming languages, frameworks and implementation details are not always sufficiently considered. For this reason, and due to the fact, that other frameworks focus on specific areas of data mining, ELKI supports a variety of different algorithms for various different data mining techniques in one framework. The open source java software framework ELKI separates data types and data management from the algorithms, thus ensures the algorithms can be compared on equal terms. Additionally, ELKI ensures an intuitive structure to ease third-party contributions.

The architecture of ELKI [AKZ08] separates the different tasks and allows an independent implementation of different parts of the system. This modular structure, together with various different parsers provided by ELKI, make it possible to read data of arbitrary data types, including CSV, ARFF and libSVM format. For example the simple transaction parser allows to read text files outputting transactions. This parser will be used during experimentation. In particular the parsed data can be processed by any algorithm suitable for this data type. Furthermore, ELKI supports a large number of index structures for instance R*-tree variants, VA-files and LSH allowing to efficiently process large and high dimensional data sets. The user can chose between many different algorithms from many different categories such as outlier detection, classification, frequent itemset mining and clustering. In particular, major support for clustering uncertain data has been added in version 0.7.0 [SKE+15]. Considering these features ELKI is well suited for evaluation and benchmarking of algorithms, as well as for data mining tasks. ELKI is actively developed and refined by a contentiously growing community.

So far, association rule mining and respective interestingness measures were not available, but they are added throughout this work. Since frequent itemset mining and a transaction parser were already included, implementations for association rule generation as described in Section 3.3, and for interestingness measures were still missing. Additionally, to be able to present a human readable version of the association rule mining output, visualization modules have

to be extended. In the following section, additional implementations, made to enable full association rule mining capabilities, are explained.

4.4 Extension

As described earlier, the ELKI framework already includes various algorithms for frequent itemset mining. To provide full association rule mining functionality, the second subproblem of the mining process as described in Section 3.3 had to be added.

Association rules generation builds on frequent itemset mining, as mostly association rules are generated from frequent itemsets. This was considered for the new implementation and thus first a frequent itemset mining algorithms has to be invoked. Therefore, it is possible to pass the name of the preferred itemset mining algorithm, to specify which of the available ones should be used. If this parameter is omitted when launching the program, the default algorithm FPGrowth will be applied. However it may be the case, someone wants to mine association rules directly from the row data. In this case, simply the minimum support parameter for the itemset mining algorithms has to be set to 0. Since the new association rule generation algorithm simply invokes an existing itemset mining algorithm all parameters allowed for them can now be called as well.

After invoking the specified itemset mining algorithm, the set of frequent itemsets $\mathcal{F}$ is received. If confidence is specified as desired interestingness measure, the obtained frequent itemsets are processed according to the algorithm described in Section 3.3. Otherwise, for every itemsets $Z \in \mathcal{F}$ the value of the specified interestingness measure is calculated for all combinations of itemsets $X \subset Z$ and $Y = Z \setminus X$ according to the brute force approach explained in Section 3.3. Hence, the advanced algorithm, which harnesses the potential of pruning, can only be applied when confidence is selected, since the *confidence monotonicity* property only holds for confidence and not for other interestingness measures. However to actually generate rules additionally to the parameter, specifying the interestingness measure, thresholds have to be passed, since only, if the measures value $measure(X \Rightarrow Y)$ is within the specified range, the rule $X \Rightarrow Y$ can be outputted. For example, Lift is passed as desired measure, together with a minimum threshold of 2.0 and maximum threshold of 10.0, as shown in Figure 4.2, which produces association rules, satisfying $2.0 \leq lift(X \Rightarrow Y) \leq 10.0$. If the measure parameter is omitted confidence is used as default.

The now computed association rules are visualized as simple text, using the resultwirter functionality of ELKI. Therefore they are written one rule

```
dhcp-10-180-56-47:target fredi$ java -jar elki-0.7.2-SNAPSHOT.jar KDDCLIApplication -dbc.in art
igo_tags.ascii -dbc.parser SimpleTransactionParser -time -algorithm associationrulemining.Assoc
iationRuleGeneration -associationrulemining.frequentItemAlgo FPGrowth -resulthandler ResultWrit
er -out output -itemsetmining.minsupp 0.035 -associationrulemining.interestMeasure Lift -associ
ationrulemining.minmeasure 2.0 -associationrulemining.maxmeasure 10.0
de.lmu.ifi.dbs.elki.datasource.FileBasedDatabaseConnection.load: 4905 ms
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.raw-items: 151845
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.raw-transactions: 58901
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.minsupp-relative: 0.0350078946028081
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.minsupp-absolute: 2062
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.items: 327
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.nodes: 1321437
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.transactions: 58901
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.fp-tree.construction.time: 379 ms
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.fp-growth.extraction.time: 2949 ms
de.lmu.ifi.dbs.elki.algorithm.itemsetmining.FPGrowth.frequent-itemsets: 46815
de.lmu.ifi.dbs.elki.algorithm.associationrulemining.AssociationRuleGeneration.runtime: 3818 ms
dhcp-10-180-56-47:target fredi$
```

Figure 4.2: Invoking EKLI association rule generation algorithm with various parameters

per line to a text file named `arule-generation.txt` in the specified output directory. Within the file, the rules are presented as antecedent as well as consequent together with their support values and the value of the specified interestingness measure. For Example `BILDNIS, NASE: 4417 -> PORTRAET, MUND: 5048 : 0.774676`.

Expandability was a very important factor for the design of the extension due to the fact that a great many different interstingness measures have been proposed. Within the scope of this work eleven interestingness measure have been implemented containing the ten measures described in Section 3.4 and the Added-Value measure. To ensure easy extension an abstract class (`AbstractInterestingnessMeasure`) was created. In order to add a new interestingness metric to ELKI, simply a new class extending, the previous abstract, one has to be inserted into the `interestingnessmeasure` package. Furthermore, new and different association rule generation algorithms can be easily added as well. Therefore, just a new class extending, the abstract class in the newly created package for association rule mining, has to be implemented.

For detailed information on how to use association rule mining in the ELKI framework, in Section A of the appendix, a manual is provided. There, all mandatory and optional parameters are explained.

Chapter 5

Experiments and Results

In this chapter the evaluation of the ten interestingness measures, described in Section 3.4, on the ARTigo data set is outlined. First of all, the data set, obtained from ARTigo, has to be prepared. This is very important for achieving useful rules. Afterwords, the actual evaluation process is described. To evaluate between the ten interest measures, several steps were taken. At first, they are examined to find possible similarities between them. The next step is to find out which measures achieve the best results in ranking subjective interesting rules high and subjective uninteresting rules low. Afterwards, a closer look is taken on the top ranked rules, according to the various measures.

5.1 Data Preparation

The step of data preparation has a major impact on the quality of the yielded rules. As described in Section 4.2, the data obtained from ARTigo, being natural language, contains many synonyms, different spellings of the same word and even misspelled tags. This highly affects the results of the mining process. Especially these different spellings and misspelled tags are a major factor, being responsible for the high redundancy of the yielded rules. For Example the different spellings FOTO, FOTOGRAFIE and PHOTOGRAPHIE, with FOTO being short for FOTOGRAFIE, are all different versions of the same word with exactly the same meaning. Every one of these three words is, according to the German orthography correctly spelled. Whereas PHOTO which is short for PHOTOGRAPHIE is actually misspelled, but yet is very common. Since all these syntactically different tags have the exact same meaning, unlike synonyms which often have a slightly different meaning, they do not contribute any additional value to the data and the resulting rules. On the contrary, the presence of these tags in the data set ensure highly redundant rules are being mined. Furthermore, various interestingness measures rank these redundant

	Rules
1	{WEIß, FOTOGRAFIE, PHOTO, PHOTOGRAPHIE} ⇒ {FOTO}
2	{WEISS, FOTOGRAFIE, PHOTO, PHOTOGRAPHIE} ⇒ {FOTO}
3	{WEISS, WEIß, FOTOGRAFIE, PHOTO, PHOTOGRAPHIE} ⇒ {FOTO}
4	{WEIß, FOTOGRAFIE, PHOTO, PHOTOGRAPHIE, SCHWARZ} ⇒ {FOTO}

Table 5.1: Four top ranked rules by conviction before data preparation

	Rules
1	{WEISS, GRAU, KRAGEN, AUGEN, BILDNIS} ⇒ {PORTRAET}
2	{SCHWARZ, KRAGEN, HAARE, BILDNIS, MUND} ⇒ {PORTRAET}
3	{GRAU, SCHWARZ, KRAGEN, AUGEN, BILDNIS} ⇒ {PORTRAET}
4	{SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND, NASE} ⇒ {PORTRAET}

Table 5.2: Four top ranked rules by conviction after data preparation

rules very high. For instance, if ARTigo players are showed an image of a photograph, many of them will tag it with FOTO, because it is short. Some might use FOTOGRAFIE and others might tag PHOTO or PHOTOGRAPHIE. Because all tags have the same meaning it does not make any difference for the mined rules. Therefore, it is very likely that images which have one of these tags assigned to them, are also tagged with the others. To illustrate this, Table 5.1 shows the four top ranked rules by conviction before data preparation. All four rules contain the four spellings of FOTO, thus they are highly redundant and not of great interest.

In order to achieve more appealing results the data was prepared. Therefore, first the raw data obtained from the ARTigo database was searched for misspelled words and these were corrected. Additionally all occurrences of the letter ß were replaced with SS, for example WEIß became WEISS. In a second iteration the now corrected words were conflated with the ones, which already have been correct such that only one occurrence of the same tag exists in one transaction. At last, tags of the same stem were conflated, which removed a few of the synonyms. Reducing the amount of syntactically unique tags from 240338 to 151845, these steps of data preparation dramatically improved the quality of the resulting rules, now showing much less redundancy. Most of the highly redundant rules, especially those seen in Table 5.1, now can not be generated any more. To illustrate this, Table 5.2 shows the four top ranked rules after data preparation according to the conviction measure.

5.2 Evaluation

After data preparation, which was able to remove many of the highly redundant rules, it was possible to proceed with the actual evaluation. At first all rules had to be mined. To achieve this, for each one of the ten measures the newly implemented algorithm was invoked, with the FP-Growth algorithm for frequent itemset mining, with a minimum support threshold of 0.035, and the respective measure with no threshold. The support threshold of 0.035 was chosen due to the fact that it was the lowest threshold yielding an appropriate amount of rules and not pruning to many items. Any lower threshold resulted in an inappropriate number of rules, making further analysis very difficult. For the measures no threshold was used, because this ensures no rules are eliminated, yielding all 1387436 possible rules. Thus the output of each measure contains exactly the same rules. This allows to compare the ranks of the rules between the various measures. On the contrary, in everyday use you normally set thresholds to eliminate certain rules. After generating the association rules, each one of the ten outputted sets of rules is sorted by the values of the respective measure. Next, they were ranked, by assigning 1 to the most interesting rule, 2 to the second most interesting one and so on. The ten highest ranked rules, according to each one of the ten measures, can be examined in Section B.1 of the appendix.

The three steps taken to find out, which interestingness measure or measures perform best on this type of data, respectively yield the most interesting rules are outlined in the following sections.

5.2.1 Similarities between Measures

In the first step of evaluation, the yielded rules are searched for similarities. Checking them for existing similarities before evaluating the quality of the rules, obtained by the various measures makes sense. The discovery of groups of similar measures may ease the remainder of the evaluation process, due to the fact that they provide further insight and help to understand the results of the following analyses.

To find similarities, first the correlation coefficient between all ten measures was computed. Therefore, the ranks of the rules are used to calculate a vector of the form $(r_1, r_2, r_3, \ldots, r_{10})$ for every rule, with $r_i \in \{1, 2, 3, \ldots, 1387436\}$ being the rank of the rule, according to the corresponding measure. In other words, this vector contains the ranks of its respective rule, according to each one of the ten measures. These vectors were then used to calculate the correlation between every pair of measures.

Table 5.3 shows the resulting correlations in yellow, indicating the two

	1	2	3	4	5	6	7	8	9	10
1 Confidence										
2 Certainty factor										
3 Conviction										
4 Klosgen										
5 Gini-Index										
6 J-Measure										
7 Cosine										
8 Jaccard										
9 Lift										
10 Leverage										

Table 5.3: Correlation between every pair of measures, yellow for $\rho(A, B) \geq 0.85$ and red for $\rho(A, B) \geq 0.95$. Table B.11 shows the values of the correlation coefficient for all pairs of measures.

measures have a correlation coefficient of greater or equal 0.85 and red of greater or equal 0.95. The results reveal two major clusters. One contains Certainty Factor, Confidence, Conviction as well as Klosgen and the other consists of J-Measure, Cosine, Jaccard and Lift. In the first group all pairwise correlations are actually greater than 0.92. Inside of this group the three measures Certainty Factor, Confidence and Conviction form an even stronger cluster, with their pairwise correlations being greater than 0.98. By looking at their top ranked rules, their strong similarity is indicated, as they share the same 33 first rules, the tables of Section B.1 showing the first ten. Foremost, Certainty Factor and Conviction even share the same 134401 top ranked rules. The similarity of the measures, contained in the second major group is not quite as high with their pairwise correlations being not less than 0.89.

This step of searching for similarities has revealed two major clusters with strongly related measures, which almost divides the measures in two parties. Only Gini-Index and Leverage do not really fit in one of the groups, whereby Gini-Index is still highly correlated with Klosgen, Leverage is only relatively week related to Cosine, with $\rho(A, B)$ being 0.85.

5.2.2 Comparison of Interesting and Uninteresting Rules

This part of the evaluation is intended to analyze the various measures ability of finding interesting rules and eliminating uninteresting ones. In other words, interesting associations should be ranked high whereas uninteresting ones should be ranked low. Therefore, ten subjectively interesting and ten

subjectively uninteresting rules were selected, shown in Section B.3 of the appendix. Then the ranks of the ten association rules according to each one of the ten measures were compared. This was done by means of comparing the average of the ranks. However, since the ranks average of the interesting rules had to be as low as possible, the rank of the uninteresting ones had to be as high as possible, making it difficult to combine them. Hence, the average of the uninteresting rules is calculated using the difference between the average rank and the total number of rules. Thus if r_i with $i \in \{1, 2, 3, \ldots, 10\}$ is the rank of each rule according to one measure, the average is calculated as follows: $1387436 - 0.1 \cdot (r_1 + r_2 + r_3 + \ldots + r_{10})$. As a consequence a low average rank indicating the measure classified the rules as uninteresting. Thus a low average rank is desirable. Now for both, the average for interesting rules and the average for uninteresting ones, low values are desirable. This makes it possible to combine them. Thereby, an overall average can be computed, showing the ability of the measure to select interesting rules and eliminate uninteresting ones.

The ten interesting rules used for this experiment can be examined in Table B.12. These rules appear interesting and therefore have been selected because they reveal something new, a not obvious relationship between two tags, for example {HINTERGRUND} $\Rightarrow$ {GEMÄLDE}. They all are rather short, whereby antecedent and consequent combined, do not contain more than four items. In contrast the selected uninteresting rules shown in Table B.13 are either quite long and therefore obvious or redundant as for instance {HELL, SCHWARZ} $\Rightarrow$ {DUNKEL, WEISS}. The synonyms (bright, white) and (dark, black) in this example are interchanged, thus antecedent and consequent have a similar meaning. Hence the rule is of no interest.

The results of this experiment shown in Table 5.4 reveal leverage with an average rank of 199261, seems to have the best ability of locating interesting rules. Gini-Index and Klosgen following Leverage show quite promising ranks as well. Whereby the other measures achieve relative similar values, Lift displays by far the worst average rank of 816197. Except for Lift, all the measures accomplished to rank the interesting rules in the upper half. In comparison none of the interesting measures was able to rank the uninteresting rules lower than half of the rules. On the contrary some of them such as Lift actually ranked these rules higher than the interesting ones. Especially the very redundant rule {HELL, SCHWARZ} $\Rightarrow$ {DUNKEL, WEISS} was ranked very high by many measures. In particular Gini Index and Leverage ranked this rule under their two hundred most interesting rules. As synonyms often occur together the measures have great problems with eliminating redundant rules which contain synonyms in antecedent and consequent. Confidence ranked the selected uninteresting rules in average lowest. Closely followed by Certainty

Measure	avg. interesting	avg. uninteresting	total average
Certainty factor	482969	889888	686428
Conviction	482969	889888	686428
Klosgen	360494	964763	662629
Leverage	199261	907697	553479
Lift	816197	937866	877031
Confidence	428709	873489	651099
Cosine	486335	1025747	756041
Jaccard	514158	978470	746314
Gini Index	332956	998541	665749
J-Measure	441030	1024069	732549

Table 5.4: Average ranks for ten interesting and ten uninteresting measures, as well as total average according to the ten measures. For all averages low values are better. Section B.3 shows the ranks of the individual rules.

Factor and Conviction, whereas Cosine and J-Measure achieve the worst results.

Overall Leverage still outperforms the other measures with total average of 553479 due to its by far best ability of locating interesting ones and its medium performance in eliminating uninteresting ones. The other measures all yield quite similar values, whereas Lift accomplishes the by far lowest value with 877031.

5.2.3 Comparison of the 1000 Highest Ranked Rules

As association rule mining produces a huge amount of rules, it is the purpose of the interestingness measures to rank the most interesting rules in the top. Thus only the highest ranked association rules are actually examined. Therefore, in this part of the evaluation, the one thousand top ranked rules, according to each measure, are compared and analyzed in greater detail.

To accomplish reasonably objective results, not individual subjective interesting rules are selected. Instead, the measures and their top ranked rules are searched for properties which increase the probability, that interesting rules are ranked high. Therefore, five assumptions for properties are made. They concern the measures and their one thousand highest ranked rules. These properties, if achieved, make the mined rules more likely to be interesting.

Property 1 Mined rules should contain as few as possible synonyms. If antecedent and consequent contain synonymous words like the previously

used example rule {HELL, SCHWARZ} ⇒ {DUNKEL, WEISS}, antecedent and consequent effectively mean the same. Thus this association does not provide any useful knowledge, consequently it is not interesting. The first column after the measures of Table 5.5 shows the susceptibility of each measure for synonyms.

Property 2 Unsymmetrical measures are preferred. For instance the rule {MÄNNER} ⇒ {MENSCHEN} is obvious, since every man is a human. Thus it is not surprising a picture tagged with men (ger. MÄNNER) may be tagged with humans (ger. MENSCHEN)as well. On the contrary, the rule {MENSCHEN} ⇒ {MÄNNER} is not obvious. It actually reveals that most of the humans displayed on artworks are men, which may be very interesting. Symmetrical measures rank these two rules equally, thus many uninteresting rules will be ranked highly, as the measure does not make any difference between the directions of the rule. Column three of Table 5.5 shows this property.

Property 3 The length of the rules should be rather short. Especially during the experiment of Section 5.2.2 it turned out, long rules are in some way overwhelming. Consequently the human observer considers them rather uninteresting. Hence in column four of Table 5.5 the average rule length of the one thousand top ranked associations of each measure is displayed.

Property 4 The ratio of consequent length and total length of the rules should be rather high. Long rules, especially ones with many items contained in the antecedent tend to anticipate the items of the consequent. For instance, in the rule {WEISS, BLICK, AUGEN, BILDNIS, GESICHT, MUND, NASE} ⇒ {PORTRAET}, the items of the antecedent almost entirely describes a person's head and it is said it is an effigy. Thus the consequent containing the item portrait (ger. PORTRAET) is quite obvious. For this reason, antecedent and consequent, having a similar length, makes a rule more likely to be a matter to the observer. Table 5.5 shows in column five the ration of the average consequent length to the average total rule length. It is important to note that symmetrical measures will always achieve the value 0.5 as they rank the rules $X \Rightarrow Y$ and $Y \Rightarrow X$ equally.

Property 5 As many as possible different consequent itemsets. If the number of unique consequent itemsets is high it is more likely interesting rules are contained, since there is more variety within the rules. Therefore, column six of Table 5.5 displays the number of different consequent itemsets within the top one thousand rules of each measure.

Measure	syn.	sym.	avg. length	$\frac{\text{avg. consequent l.}}{\text{avg. total l.}}$	# consequents
Certainty factor	I	no	5.834	0.171	7
Conviction	I	no	5.834	0.171	7
Klosgen	S	no	4.271	0.299	63
Leverage	VS	yes	3.440	0.500	193
Lift	VS	yes	6.338	0.500	404
Confidence	I	no	5.866	0.170	8
Cosine	S	yes	5.086	0.500	246
Jaccard	S	yes	5.172	0.500	251
Gini Index	N	no	3.833	0.379	120
J-Measure	S	no	4.302	0.413	137

Table 5.5: Five properties for measures and their resulting rules (VI: very immune, I: immune, N: neutral, S: susceptible, VS: very susceptible)

The results of this experiment show that certainty Factor, confidence and conviction accomplish very similar values. On the one hand, they are quite immune against synonyms, but on the other hand, they yield very long rules, with short consequent itemsets, such as {KRAGEN, HAARE, BILDNIS, MUND, NASE} ⇒ {PORTRAET}. Thus their ratio of consequent length to total length is very low. Additionally the mined associations contain no more than eight different consequent itemsets, which is by far the lowest value. Almost all consequent itemsets of these measures contain either PORTRAET or HIMMEL.
The four measures Leverage, Lift, Cosine and Jaccard achieve the best values for the ratio of consequent length to total length, being 0.5. Furthermore, their resulting rules contain the most different consequent itemsets. However, these two values have to be seen in the view of the fact that these measures are symmetrical, thus the value of 0.5 in column five and the high value in the last column of Table 5.5 accrue from this property. As for instance, a rule {NASE} ⇒ {AUGEN, MUND} (rule 15 according to Jaccard) is always followed by a rule, where antecedent and consequent of the previous one are interchanged, like {AUGEN, MUND} ⇒ {NASE} (rule 16 according to Jaccard).
Moreover Leverage, Cosine and Jaccard are quite vulnerable to synonyms, whereas Lift actually reveals to be very susceptible. The most interesting rule according to Lift namely {WEISS, STICH, TEXT} ⇒ {SCHWARZ, SCHRIFT, DRUCK} (Table B.5) is quite redundant, as STICH is a synonym for DRUCK and TEXT is a synonym for SCHRIFT.
Klosgens and Jaccards results can be considered similar. Since association

rules yielded from Klosgen are shorter than those of Jaccard. However, Klosgens average consequent itemsets length is quite small, compared to its overall rule length. Additionally, only very few different itemsets appear on the right side of the implication of rules, mined with Klosgen.
For both J-Measure accomplishes better results. For this experiment Gini Index achieves the most promising results. No strong vulnerability to synonyms was detected and except for Leverage its average rule length is the shortest. Further its results for property four and five are average.

5.3 Results

During the previous three steps of evaluation the ten measures were precisely analyzed. The experiment, described in Section 5.2.1, reveals great similarities between the various measures, pointing out two major cluster of strongly related measures. Evaluation step two in Section 5.2.2 demonstrates the ability of the individual measures to locate interesting rules and to eliminate uninteresting ones. As this experiment is meant to provide an overall view of the quality of the obtained rules, the impact of its results on the evaluation process is smaller than of evaluation step three, described in Section 5.2.3. As the comparison of this third step provides a detailed analysis of the one thousand top ranked rules, being the most important ones.

Summarizing the results of these three steps of evaluation, it can be said, that Lift performs worst on this type of data obtained from ARTigo. In particular as Lift yields the far highest average rank in the second experiment (Section 5.2.2). Additionally Lift accomplishes exceedingly few of the properties presented in Section 5.2.3, especially since the values achieved for property four and five are relativized by the fact that lift is a symmetrical measure.
The measures Cosine and Jaccard achieve pure results as well. Their total average ranks in experiment two are the second and third highest. Further they are both symmetrical, vulnerable to synonyms and yield very long rules.
Though Leverage achieves the overall lowest average rank in the experiment of Section 5.2.2, the values achieved in the last experiment are rather poor. According to the one thousand highest ranked rules Leverage seems to be very susceptible for synonyms and in consideration of the fact that Leverage is symmetrical its yielded rules contain quite few different consequent itemsets.
The interestingness measures contained in the Cluster of Certainty Factor, Confidence, Conviction and Klosgen show quite similar results, especially the first three achieve almost equal values throughout the entire evaluation. The second experiment showed their relatively strong ability to eliminate uninteresting association rules.

Overall J-Measure and Gini Index seem the be best suited for this tag data obtained from ARTigo. Their results in the experiments are consistently average or above, with Gini Index having a small advantage over J-Measure, as it achieved the better results in evaluation step three of Section 5.2.3.

Chapter 6

Conclusion and Future Work

The purpose of this work was to add association rule mining functionality to the ELKI framework and evaluate appropriate interestingness measures for the ARTigo data. Therefore, first an association rule algorithm with the ability of applying various interestingness measures was implemented and integrated into the framework. In the second part of this work the previously added features of ELKI were used to evaluate appropriate measures for the ARTigo data. In order to achieve acceptable results data preparation had to be done. The evaluation was split into three parts. First, the measures were searched for similarities. Second, their ability to locate interesting and eliminate uninteresting association rules was studied and third, they together with their yielded rules were examined by the means of five proposed properties.

As interestingness is subjective, there is no simple way of calculating which interstingness measure generates the most interesting rules of one type of data. Thus selecting the appropriate measure is a difficult task. The evaluation in this work reveals that the Lift measure performs worst, whereas J-Measure and Gini Index yield the most promising results. It is important to note even though J-Measure and Gini Index outperform the rest of the ten examined measures, this does not mean, all of their mined association rules provide new and useful knowledge. This type of data involves many difficulties such as synonyms, which ensure redundant and obvious rules are produced.

Evaluating more than ten measures would have gone beyond the scope of this work, but in the past years many more measures have been proposed. This holds possibilities for further analysis. As other measures, not examined throughout this work, may achieve more promising results than the ten previously analyzed measures. Furthermore, in this work data preparation concentrated on removing different spellings of the same word and misspelled tags, but the data still contains many synonyms. These synonyms tend to be a major problem for mining surprising and interesting association rules. Thus

expanding data preparation to further eliminate synonyms from the data set can improve the yielded results. It is important to note that pruning synonyms reduces data depth, which also affects the resulting rules.

In this work subjective selected rules (Section 5.2.2) and the proposed properties, concerning the rules and their measures (Section 5.2.3), were used to evaluate the various measures. An other approach is to perform an extensive poll to evaluate appropriate interstingness measures. Additionally, a poll may provide new knowledge on which rules are considered interesting and to refine the properties such rules satisfy. This approach has the great advantage that the subjective opinions of the many participants are involved, thus the results may be more significant. An other evaluation method was chosen for this work due to the fact, that creating an informative survey take, enough effort for a thesis of its own.

Additionally, the process of finding interesting association rules can be integrated in ARTigo. Therefore, a new game could be added to the ARTigo platform, for instance, allowing users to propose their own interesting association rules. This can be combined with machine learning techniques, which use the rules contributed be the players to further refine and generate new rules.

In conclusion, this field of study opens up many opportunities some of them have been outlined previously, but this topic may hold many more. This last section was intended to show, there is still a lot which could be done and this topic is by no means completed.

Appendices

Appendix A

Manual - Associaition Rule Mining in ELKI

This is a short manual to explain the usage of association rule mining in ELKI. For information on how to invoke ELKI itself consult the projects wiki [web], as this manual only covers the parameters needed for association rule mining. Options:

algorithm	Specifies which mining algorithm should be used, has to be set to associationrulemining.x with x being the class name of the desired association rule algorithm (currently only AssociationRuleGeneration).
[associationrulemining.frequentItemAlgo]	Specifies the frequent itemset mining algorithm to be used. Needs to be passed the class name of the desired algorithm. Optional, if omitted default value is FPGrowth.
itemsetmining.minsupp	Specifies the minimum support threshold. Needs to be passed a number either a double between 0 and 1 for relative support or an integer for absolute support.
[itemsetmining.minlength]	Specifies the minimum length of the mined itemsets. Needs to be passed an integer. Optional, if omitted default value is 0.
[itemsetmining.maxlength]	Specifies the maximum length of the mined itemsets. Needs to be passed an integer. Optional. if omitted default value is maximum interger value.
[associationrulemining.interestMeasure]	Specifies the interestingness measure to be used. Needs to be passed the class name of the desired measure. Optional, if omitted default value is Confidence.
associationrulemining.minmeasure	Specifies the minimum threshold for the measures value. Needs to be passed a double value.
[associationrulemining.maxmeasure]	Specifies the maximum threshold for the measures value. Needs to be passed a double value. Optional, if omitted default value is maximum double value.

Appendix B

Experiments

B.1 Top Ten Ranked Rules of all Ten Interestingness Measures

	Rules
1	WEISS, GRAU, KRAGEN, AUGEN, BILDNIS: 2124 -> PORTRAET: 9508 : 0.9983156823269521
2	SCHWARZ, KRAGEN, HAARE, BILDNIS, MUND: 2097 -> PORTRAET: 9508 : 0.9982939958333078
3	GRAU, SCHWARZ, KRAGEN, AUGEN, BILDNIS: 2078 -> PORTRAET: 9508 : 0.9982783971426594
4	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND, NASE: 2070 -> PORTRAET: 9508 : 0.9982717436050463
5	KRAGEN, BLICK, BILDNIS, MUND, NASE: 2220 -> PORTRAET: 9508 : 0.9978513569143822
6	SCHWARZ, KRAGEN, BLICK, AUGEN, BILDNIS, NASE: 2202 -> PORTRAET: 9508 : 0.9978337930744452
7	KRAGEN, HAARE, BILDNIS, MUND, NASE: 2185 -> PORTRAET: 9508 : 0.997816939290585
8	GRAU, KRAGEN, AUGEN, BILDNIS: 2169 -> PORTRAET: 9508 : 0.9978008355693538
9	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND: 2148 -> PORTRAET: 9508 : 0.9977793353584394
10	WEISS, KRAGEN, HAARE, BILDNIS, MUND: 2144 -> PORTRAET: 9508 : 0.997775192327392

Table B.1: Ten top ranked rules according to Certainty Factor

	Rules
1	WEISS, GRAU, KRAGEN, AUGEN, BILDNIS: 2124 -> PORTRAET: 9508 : 593.7122289944143
2	SCHWARZ, KRAGEN, HAARE, BILDNIS, MUND: 2097 -> PORTRAET: 9508 : 586.1650396427905
3	GRAU, SCHWARZ, KRAGEN, AUGEN, BILDNIS: 2078 -> PORTRAET: 9508 : 580.8540545434995
4	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND, NASE: 2070 -> PORTRAET: 9508 : 578.6178502911665
5	KRAGEN, BLICK, BILDNIS, MUND, NASE: 2220 -> PORTRAET: 9508 : 465.41001001680786
6	SCHWARZ, KRAGEN, BLICK, AUGEN, BILDNIS, NASE: 2202 -> PORTRAET: 9508 : 461.6364153409959
7	KRAGEN, HAARE, BILDNIS, MUND, NASE: 2185 -> PORTRAET: 9508 : 458.07246481384016
8	GRAU, KRAGEN, AUGEN, BILDNIS: 2169 -> PORTRAET: 9508 : 454.7181584353407
9	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND: 2148 -> PORTRAET: 9508 : 450.31563131356006
10	WEISS, KRAGEN, HAARE, BILDNIS, MUND: 2144 -> PORTRAET: 9508 : 449.47705471893516

Table B.2: Ten top ranked rules according to Conviction

	Rules
1	WOLKEN: 10311 -> HIMMEL: 16380 : 0.2576513715394224
2	SCHATTEN: 16713 -> LICHT: 18702 : 0.254701506383997
3	SCHATTEN, WEISS: 14748 -> LICHT: 18702 : 0.25271611579756104
4	TEXT: 7648 -> SCHRIFT: 11715 : 0.2510496301362711
5	BILDNIS: 6397 -> PORTRAET: 9508 : 0.2440364131251936
6	WEISS, WOLKEN: 8272 -> HIMMEL: 16380 : 0.24384615644044547
7	LICHT, WEISS: 16037 -> SCHATTEN: 16713 : 0.23623446590465305
8	SCHATTEN, SCHWARZ: 12808 -> LICHT, WEISS: 16037 : 0.23465710782809898
9	SCHATTEN, SCHWARZ: 12808 -> LICHT: 18702 : 0.2344210073624016
10	WEISS, BILDNIS: 5626 -> PORTRAET: 9508 : 0.23318567567389

Table B.3: Ten top ranked rules according to Klosgen

	Rules
1	WEISS: 33964 -> SCHWARZ: 28995 : 0.15569607168406052
2	SCHWARZ: 28995 -> WEISS: 33964 : 0.15569607168406052
3	SCHATTEN: 16713 -> LICHT: 18702 : 0.1480850563766027
4	LICHT: 18702 -> SCHATTEN: 16713 : 0.1480850563766027
5	SCHATTEN: 16713 -> LICHT, WEISS: 16037 : 0.13847887935568415
6	LICHT, WEISS: 16037 -> SCHATTEN: 16713 : 0.13847887935568415
7	SCHATTEN, WEISS: 14748 -> LICHT: 18702 : 0.1362332815203472
8	LICHT: 18702 -> SCHATTEN, WEISS: 14748 : 0.1362332815203472
9	WEISS, SCHWARZ: 25890 -> GRAU: 21476 : 0.12521349953578512
10	GRAU: 21476 -> WEISS, SCHWARZ: 25890 : 0.12521349953578512

Table B.4: Ten top ranked rules according to Leverage

	Rules
1	WEISS, STICH, TEXT: 2442 -> SCHWARZ, SCHRIFT, DRUCK: 3729 : 14.223610614197906
2	SCHWARZ, SCHRIFT, DRUCK: 3729 -> WEISS, STICH, TEXT: 2442 : 14.223610614197904
3	SCHWARZ, DRUCK, TEXT: 2939 -> WEISS, SCHRIFT, STICH: 3108 : 14.179708447571667
4	WEISS, SCHRIFT, STICH: 3108 -> SCHWARZ, DRUCK, TEXT: 2939 : 14.179708447571665
5	LICHT, WEISS, AUGEN, NASE: 3045 -> PORTRAET, SCHATTEN, MUND: 2915 : 14.167547958439306
6	PORTRAET, SCHATTEN, MUND: 2915 -> LICHT, WEISS, AUGEN, NASE: 3045 : 14.167547958439304
7	BERG, LANDSCHAFT, WEISS, BERGE: 3084 -> HIMMEL, GEBIRGE: 2839 : 14.16103533377283
8	HIMMEL, GEBIRGE: 2839 -> BERG, LANDSCHAFT, WEISS, BERGE: 3084 : 14.161035333772828
9	WEISS, DRUCK, TEXT: 2905 -> SCHWARZ, SCHRIFT, STICH: 3150 : 14.154391607245309
10	SCHWARZ, SCHRIFT, STICH: 3150 -> WEISS, DRUCK, TEXT: 2905 : 14.154391607245309

Table B.5: Ten top ranked rules according to Lift

	Rules
1	WEISS, GRAU, KRAGEN, AUGEN, BILDNIS: 2124 -> PORTRAET: 9508 : 0.998587570621469
2	SCHWARZ, KRAGEN, HAARE, BILDNIS, MUND: 2097 -> PORTRAET: 9508 : 0.9985693848354793
3	GRAU, SCHWARZ, KRAGEN, AUGEN, BILDNIS: 2078 -> PORTRAET: 9508 : 0.9985563041385948
4	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND, NASE: 2070 -> PORTRAET: 9508 : 0.9985507246376811
5	KRAGEN, BLICK, BILDNIS, MUND, NASE: 2220 -> PORTRAET: 9508 : 0.9981981981981982
6	SCHWARZ, KRAGEN, BLICK, AUGEN, BILDNIS, NASE: 2202 -> PORTRAET: 9508 : 0.9981834695731153
7	KRAGEN, HAARE, BILDNIS, MUND, NASE: 2185 -> PORTRAET: 9508 : 0.9981693363844394
8	GRAU, KRAGEN, AUGEN, BILDNIS: 2169 -> PORTRAET: 9508 : 0.9981558321807285
9	SCHWARZ, KRAGEN, BLICK, BILDNIS, MUND: 2148 -> PORTRAET: 9508 : 0.9981378026070763
10	WEISS, KRAGEN, HAARE, BILDNIS, MUND: 2144 -> PORTRAET: 9508 : 0.9981343283582089

Table B.6: Ten top ranked rules according to Confidence

	Rules
1	WEISS: 33964 -> SCHWARZ: 28995 : 0.8250135430456259
2	SCHWARZ: 28995 -> WEISS: 33964 : 0.8250135430456259
3	MUND: 6802 -> AUGEN, NASE: 6258 : 0.8046800054794293
4	AUGEN, NASE: 6258 -> MUND: 6802 : 0.8046800054794293
5	NASE: 7739 -> MUND: 6802 : 0.8028514435268379
6	MUND: 6802 -> NASE: 7739 : 0.8028514435268379
7	SCHATTEN: 16713 -> LICHT: 18702 : 0.7935156240833475
8	LICHT: 18702 -> SCHATTEN: 16713 : 0.7935156240833475
9	NASE: 7739 -> AUGEN: 8183 : 0.7863879677624043
10	AUGEN: 8183 -> NASE: 7739 : 0.7863879677624043

Table B.7: Ten top ranked rules according to Cosine

	Rules
1	WEISS: 33964 -> SCHWARZ: 28995 : 0.69842725727697
2	SCHWARZ: 28995 -> WEISS: 33964 : 0.69842725727697
3	MUND: 6802 -> AUGEN, NASE: 6258 : 0.6722151088348272
4	AUGEN, NASE: 6258 -> MUND: 6802 : 0.6722151088348272
5	NASE: 7739 -> MUND: 6802 : 0.6683111519045434
6	MUND: 6802 -> NASE: 7739 : 0.6683111519045434
7	SCHATTEN: 16713 -> LICHT: 18702 : 0.6559898999345366
8	LICHT: 18702 -> SCHATTEN: 16713 : 0.6559898999345366
9	NASE: 7739 -> AUGEN: 8183 : 0.6475579470198676
10	AUGEN: 8183 -> NASE: 7739 : 0.6475579470198676

Table B.8: Ten top ranked rules according to Jaccard

	Rules
1	SCHATTEN: 16713 -> LICHT: 18702 : 0.21580149622860573
2	LICHT: 18702 -> SCHATTEN: 16713 : 0.20239253951259506
3	WEISS: 33964 -> SCHWARZ: 28995 : 0.1985946774661997
4	SCHATTEN, WEISS: 14748 -> LICHT: 18702 : 0.19776453905205466
5	SCHWARZ: 28995 -> WEISS: 33964 : 0.19397653632141887
6	LICHT, WEISS: 16037 -> SCHATTEN: 16713 : 0.1935648909265485
7	SCHATTEN: 16713 -> LICHT, WEISS: 16037 : 0.18871180217848393
8	WOLKEN: 10311 -> HIMMEL: 16380 : 0.1749212484494438
9	LICHT: 18702 -> SCHATTEN, WEISS: 14748 : 0.1712925463130619
10	SCHATTEN, SCHWARZ: 12808 -> LICHT, WEISS: 16037 : 0.1703812567932914

Table B.9: Ten top ranked rules according to Gini Index

	Rules
1	SCHATTEN: 16713 -> LICHT: 18702 : 0.16561953703296425
2	WOLKEN: 10311 -> HIMMEL: 16380 : 0.16192766617220872
3	TEXT: 7648 -> SCHRIFT: 11715 : 0.16091977324525913
4	SCHATTEN, WEISS: 14748 -> LICHT: 18702 : 0.16007082709516157
5	BILDNIS: 6397 -> PORTRAET: 9508 : 0.15776378969165347
6	MUND: 6802 -> NASE: 7739 : 0.1555303317192933
7	LICHT, WEISS: 16037 -> SCHATTEN: 16713 : 0.15154115007368169
8	AUGEN, MUND: 5821 -> NASE: 7739 : 0.1505569588897519
9	MUND: 6802 -> AUGEN: 8183 : 0.14991295978208424
10	NASE: 7739 -> AUGEN: 8183 : 0.14932465166173636

Table B.10: Ten top ranked rules according to J-Measure

B.2 Correlation between all Pairs of Measures

Measure A	Measure B	$\rho(A, B)$
Certaint Factor	Confidence	0.987
Certaint Factor	Conviction	1.0
Certaint Factor	Cosine	0.393
Certaint Factor	Gini Index	0.847
Certaint Factor	J-Measure	0.617
Certaint Factor	Jaccard	0.336
Certaint Factor	Klsogen	0.960
Certaint Factor	Leverage	0.312
Certaint Factor	Lift	0.383
Confidence	Conviction	0.987
Confidence	Cosine	0.295
Confidence	Gini Index	0.791
Confidence	J-Measure	0.525
Confidence	Jaccard	0.249
Confidence	Klsogen	0.928
Confidence	Leverage	0.242
Confidence	Lift	0.274
Conviction	Cosine	0.393
Conviction	Gini Index	0.847
Conviction	J-Measure	0.617
Conviction	Jaccard	0.336
Conviction	Klsogen	0.960
Conviction	Leverage	0.312
Conviction	Lift	0.383
Cosine	Gini Index	0.747
Cosine	J-Measure	0.953
Cosine	Jaccard	0.964
Cosine	Klsogen	0.566
Cosine	Leverage	0.851
Cosine	Lift	0.940
Gini Index	J-Measure	0.897
Gini Index	Jaccard	0.695
Gini Index	Klsogen	0.957
Gini Index	Leverage	0.699
Gini Index	Lift	0.669
J-Measure	Klsogen	0.768
J-Measure	Leverage	0.813
J-Measure	Lift	0.899
Jaccard	J-Measure	0.894
Jaccard	Klsogen	0.513
Jaccard	Leverage	0.811
Jaccard	Lift	0.912
Klsogen	Leverage	0.506
Klsogen	Lift	0.519
Leverage	Lift	0.656

Table B.11: Correlation between all pairs of measures

B.3 Ten Interesting and Ten uninteresting Rules

	Interesting Rules
I_1	MENSCHEN: 15243 -> MÄNNER: 12359
I_2	BERG: 5745 -> HIMMEL: 16380
I_3	BÄUME: 10428 -> BLAU: 17939
I_4	GRAPHIK: 5727 -> WEISS, GRAU, DRUCK: 4341
I_5	SKIZZE, ENTWURF: 3007 -> ZEICHNUNG: 14750
I_6	BILDNIS, STIRN: 2453 -> PORTRAET: 9508
I_7	HINTERGRUND: 7334 -> GEMÄLDE: 14815
I_8	WALD: 5651 -> FLUSS: 5093
I_9	HIMMEL, ROT: 7311 -> MALEREI: 8622
I_{10}	HORIZONT: 4335 -> HIMMEL, WASSER: 6222

Table B.12: Selected interesting Rules

	Uninteresting Rules
U_1	HELL, SCHWARZ: 9390 -> DUNKEL, WEISS: 10401
U_2	GEBIRGE: 3125 -> BERGE: 4878
U_3	WEISS, SCHWARZ, TEXT: 4087 -> SCHRIFT, SCHWARZWEISS
U_4	HIMMEL, LICHT: 8924 -> BLAU, BRAUN, GEMÄLDE, LANDSCHAFT, SCHATTEN, WEISS, WOLKEN: 2245
U_5	MÄNNER: 12359 -> HIMMEL, MANN, WEISS, GRAU, SCHWARZ, MENSCHEN: 2696
U_6	HIMMEL, WOLKEN: 9487 -> BLAU, WEISS: 13297
U_7	AUGEN, MUND, NASE, LOCKEN: 2275 -> HAARE: 10699
U_8	BLAU, BRAUN, HIMMEL, LICHT, WEISS, BÄUME: 2759 -> GRÜN, WOLKEN: 5149
U_9	MANN, SCHATTEN, WEISS, MENSCHEN: 4092 -> LICHT, SCHWARZ, MÄNNER: 3765
U_{10}	GRÜN, WEISS, BÄUME: 4749 -> BLAU, BRAUN, GEMÄLDE, ROT: 5682

Table B.13: Selected uninteresting rules

Measure	Ranks									
Certainty Factor	550498	108560	851647	553324	10656	689	687876	819234	890546	356661
Conviction	550498	108560	851647	553324	10656	689	687876	819234	890546	356661
Klosgen	33803	7119	704604	363699	16749	4427	571096	820954	845390	237103
Leverage	58	5904	75635	51202	209080	284104	130755	715293	398534	122049
Lift	1118941	1047312	1301771	229568	830160	328065	1228343	609428	1112841	355539
Confidence	463115	95932	559602	603745	17412	1092	516025	829592	808329	392246
Cosine	12156	316796	775339	80476	646336	314016	862229	684103	926344	245551
Jaccard	10806	472456	685757	70658	967684	627978	792977	546329	752684	214249
Gini Index	272	4985	728892	179294	43995	17987	592660	785115	825920	150444
J-Measure	5923	60461	1125183	80802	184456	36895	989913	742628	1000103	183934
	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}

Table B.14: Ranks of the interesting rules according to the various interestingness measures

Measure	Ranks										
Certainty Factor	258875	35265	408951	1101057	1229490	592422	26638	146811	425629	750346	
Conviction	258875	35265	408951	1101057	1229490	592422	26638	146811	425629	750346	
Klosgen	1196	1645	313097	1109346	1241568	277140	54701	101276	334373	792384	
Leverage	147	46165	216507	769487	1111489	4927	905514	509612	239693	993851	
Lift	768081	8568	81927	322987	862329	1167099	509012	72068	65023	638610	
Confidence	256010	56587	471189	1154261	1256112	477640	41082	180470	490531	755586	
Cosine	4097	601	77987	447971	1021831	365617	678950	133754	69233	816845	
Jaccard	3411	1073	60495	765976	1103771	315219	966900	158662	54231	659925	
Gini Index	121	2894	221132	1041463	1226450	91010	139201	112776	243869	810037	
J-Measure	346	943	107728	640360	1083416	420905	323468	110100	108910	837496	
	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	

Table B.15: Ranks of the uninteresting rules according to the various interestingness measures

List of Figures

3.1 All possible itemsets with their support from a transaction database $\mathcal{T} = \{ABD, ABCD, ABC, C, ABCD\}$, frequent itemsets with *support* ≥ 3 are underlined 11
3.2 Initial *frequent pattern tree* R with count constructed from the transaction database $\mathcal{T} = \{$ABDE, BCDE, ABCDE , ACDE, BCD, BD$\}$. 13
3.3 Projected FP-tree $R_{\emptyset \cup E}$ constructed from initial FP-tree, before and after pruning infrequent items. 14

4.1 Image of the ARTigo search engine 24
4.2 Invoking EKLI association rule generation algorithm with various parameters 28

List of Tables

3.1 Customer purchase database with five transactions and the universe of items $\mathcal{U} = \{apple,\ banana,\ orange,\ peach\}$. 9
3.2 Contingency table displaying the prefered hot drinks among 1000 people [BMS97]. 16
3.3 Contingency table storing frequency counts needed for most interestingness measures with N being the total number of transactions [GH06]. 17
3.4 The ten measures used in this work sorted by groups together with the eight properties. 19

5.1 Four top ranked rules by conviction before data preparation . . 30
5.2 Four top ranked rules by conviction after data preparation . . . 30
5.3 Correlation between every pair of measures, yellow for $\rho(A, B) \geq 0.85$ and red for $\rho(A, B) \geq 0.95$. Table B.11 shows the values of the correlation coefficient for all pairs of measures. 32
5.4 Average ranks for ten interesting and ten uninteresting measures, as well as total average according to the ten measures. For all averages low values are better. Section B.3 shows the ranks of the individual rules. 34
5.5 Five properties for measures and their resulting rules (VI: very immune, I: immune, N: neutral, S: susceptible, VS: very susceptible) 36

B.1 Ten top ranked rules according to Certainty Factor 43
B.2 Ten top ranked rules according to Conviction 44
B.3 Ten top ranked rules according to Klosgen 44
B.4 Ten top ranked rules according to Leverage 44
B.5 Ten top ranked rules according to Lift 45
B.6 Ten top ranked rules according to Confidence 45
B.7 Ten top ranked rules according to Cosine 45
B.8 Ten top ranked rules according to Jaccard 46
B.9 Ten top ranked rules according to Gini Index 46

B.10 Ten top ranked rules according to J-Measure 46
B.11 Correlation between all pairs of measures 47
B.12 Selected interesting Rules 48
B.13 Selected uninteresting rules 48
B.14 Ranks of the interesting rules according to the various interestingness measures 49
B.15 Ranks of the uninteresting rules according to the various interestingness measures 50

Bibliography

[Agg15] Charu C Aggarwal. *Data mining: the textbook*. Springer, 2015.

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. *Mining association rules between sets of items in large databases*. In Acm sigmod record, volume 22, pages 207–216. ACM, 1993.

[AKZ08] Elke Achtert, Hans-Peter Kriegel, and Arthur Zimek. *ELKI: A Software System for Evaluation of Subspace Clustering Algorithms*. In Bertram Ludäscher and Nikos Mamoulis, editors, Scientific and Statistical Database Management, 20th International Conference, SSDBM 2008, Hong Kong, China, July 9-11, 2008, Proceedings, volume 5069 of *Lecture Notes in Computer Science*, pages 580–585. Springer, 2008.

[AS+94] Rakesh Agrawal, Ramakrishnan Srikant, et al. *Fast algorithms for mining association rules*. In Proc. 20th int. conf. very large data bases, VLDB, volume 1215, pages 487–499, 1994.

[BBV+02] Fernando Berzal, Ignacio Blanco, Mar Vila, et al. *Measuring the accuracy and interest of association rules: A new framework*. Intelligent Data Analysis, 6(3):221–235, 2002.

[BFSO84] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[BMS97] Sergey Brin, Rajeev Motwani, and Craig Silverstein. *Beyond market baskets: Generalizing association rules to correlations*. In ACM SIGMOD Record, volume 26, pages 265–276. ACM, 1997.

[BMUT97] Sergey Brin, Rajeev Motwani, Jeffrey D Ullman, and Shalom Tsur. *Dynamic itemset counting and implication rules for market basket data*. In ACM SIGMOD Record, volume 26, pages 255–264. ACM, 1997.

[GH06] Liqiang Geng and Howard J Hamilton. *Interestingness measures for data mining: A survey.* ACM Computing Surveys (CSUR), 38(3):9, 2006.

[GS91] Rodney M Goodman and P Smyth. *Rule induction using information theory.* G. Piatetsky, 1991.

[HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. *Mining frequent patterns without candidate generation.* In ACM Sigmod Record, volume 29, pages 1–12. ACM, 2000.

[Klö96] Willi Klösgen. *Explora: A multipattern and multistrategy discovery assistant.* In Advances in knowledge discovery and data mining, pages 249–271. American Association for Artificial Intelligence, 1996.

[MK06] Hassan H Malik and John R Kender. *Clustering web images using association rules, interestingness measures, and hypergraph partitions.* In Proceedings of the 6th international conference on Web engineering, pages 48–55. ACM, 2006.

[MY08] Agathe Merceron and Kalina Yacef. *Interestingness measures for association rules in educational data.* In Educational Data Mining 2008, 2008.

[OKO+04] Miho Ohsaki, Shinya Kitaguchi, Kazuya Okamoto, Hideto Yokoi, and Takahira Yamaguchi. *Evaluation of rule interestingness measures with a clinical dataset on hepatitis.* In European Conference on Principles of Data Mining and Knowledge Discovery, pages 362–373. Springer, 2004.

[PNSK+06] Tan Pang-Ning, Michael Steinbach, Vipin Kumar, et al. *Introduction to data mining.* In Library of congress, volume 74, 2006.

[PS91] Gregory Piatetsky-Shapiro. *Discovery, analysis, and presentation of strong rules.* Knowledge discovery in databases, pages 229–238, 1991.

[SB75] Edward H Shortliffe and Bruce G Buchanan. *A model of inexact reasoning in medicine.* Mathematical biosciences, 23(3):351–379, 1975.

[SKE+15] Erich Schubert, Alexander Koos, Tobias Emrich, Andreas Züfle, Klaus Arthur Schmid, and Arthur Zimek. *A Framework for Clustering Uncertain Data.* PVLDB, 8(12):1976–1979, 2015.

[SMAM13] Mateen Shaikh, Paul D McNicholas, M Luiza Antonie, and T Brendan Murphy. *Standardizing Interestingness Measures for Association Rules*. arXiv preprint arXiv:1308.3740, 2013.

[STXK07] Michael Steinbach, Pang-Ning Tan, Hui Xiong, and Vipin Kumar. *Objective measures for association pattern analysis*. Contemporary Mathematics, 443:205, 2007.

[TK00] Pang-Ning Tan and Vipin Kumar. *Interestingness measures for association patterns: A perspective*. In Proc. of Workshop on Postprocessing in Machine Learning and Data Mining, 2000.

[TKS02] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. *Selecting the right interestingness measure for association patterns*. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 32–41. ACM, 2002.

[vR79] C.J. van Rijsbergen. *Information Retrieval, 2nd Edition*. Butterworths, London, 1979.

[WBBL13] Christoph Wieser, François Bry, Alexandre Bérard, and Richard Lagrange. *ARTigo: Building an Artwork Search Engine With Games and Higher-Order Latent Semantic Analysis*. In In Proc. of Disco 2013, Workshop on Human Computation and Machine Learning in Games at HComp 2013, 2013.

[web] *ELKI Data Mining Framework*. `http://elki.dbs.ifi.lmu.de/wiki`. Accessed: 2016-11-14.

[ZMJ14] Mohammed J Zaki and Wagner Meira Jr. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.