Identifying Most Predictive Items*

Daniel A. Keim and Markus Wawryniuk

University of Konstanz, Department of Computer and Information Science
78457 Konstanz, Germany
{keim,wawryniu}@informatik.uni-konstanz.de

Abstract. Frequent itemsets and association rules are generally accepted concepts in analyzing item-based databases. The Apriori-framework was developed for analyzing categorical data. However, many data include numerical values. Therefore, most existing techniques transform numerical values to categorical values. The transformation is done such that the rules are optimal with respect to support or confidence.
In this paper we choose a different approach for analyzing data with numerical and categorical data. We present methods to identify items, which have a strong impact on a given numerical attribute. With other words, we want to identify items, whose occurrence in an itemset allows us to make predictions about the distribution function of the numerical attribute.

1 Introduction

Analyzing categorical data is an important field in data mining. Examples include market-basket data as well as census data. The Apriori-framework [1, 2] is widely used in this field. But, most datasets do not only contain categorical attributes, they also contain numerical attributes.

The straightforward approach in order to use the Apriori-framework is to discretize the numerical attribute [13]. There are two problems with the discretization: Finding a good binning of the data is a challenging task, especially if one considers, that a good discretization might depend on the context of the analysis or on the presence or absence of other values. To continue, a distance between two numerical values can be defined, and this information should be preserved during the analysis. Several methods, reported in section 2, were developed in order to achieve an optimal binning and to get the best rules with respect to support and confidence.

Another solution comprises methods, which do not discretize the data [3]. The categorical values are used as a filter, which selects a subset of the database. After the selection the numerical attributes are analyzed. The goal is to find significant statistical differences between the full database and the selected subset. The method proposed in this paper belongs to this category.

* This work was partially funded by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2001-33058 PANDA project (2001-2004).

Instead of finding frequent itemsets and association rules, which consist of categorical and numerical data, we aim at finding items, which have an impact on the distribution function of the numerical attributes. For instance, in our experimental analysis of a census dataset we observe, that for the group of persons with the property *Own Child* the fraction of younger people is higher than when looking at the entire dataset. "Impact on the distribution function" means, that the presence of *Own Child* leads to a left-skewed distribution function of *Age*, i.e., the distribution function has more weight at lower values.

With our method we can find such items. The "impact" is not pre-specified at all - the only measure, which describes the deviation from the average, is a statistical test comparing the two distribution functions.

This paper is organized as follows: Up to now we presented the motivation of our work. Next, in section 2, we describe existing work, which is relevant for this paper. In section 3 we formalize the idea presented so far and give the necessary algorithms. In section 4 we apply our method to real datasets and show the relevance of our findings. We conclude the paper with a summary and an outline of future work in section 5.

2 Related Work

Frequent itemsets and association rules are generally accepted concepts to analyze market-basket data in order to improve the decision making. Starting with a database of transactions, which in turn consist of items sold together, algorithms were developed to find associations between the items. The basic principles were developed in [1], and the well-known *Apriori*-algorithm was proposed in [2]. Many extensions were developed, and the concept of frequent itemsets and association rules is applied in many areas.

The Apriori-framework was developed for analyzing categorical data. However, many data include numerical values. In [13] *quantitative association rules* were proposed. The range of a quantitative attribute is partitioned into intervals and they are handled as categorical values. The question is how to partition a quantitative attribute. Increasing the number of intervals results in low support, but decreasing the number of intervals results in loss of information. Solutions proposed in [13] include joining adjacent intervals until a threshold of maximum support is reached. Further on, a measure for the best number of intervals, which requires equi-depth partitioning, as well as an interestingness measure for the patterns is given.

For rules of the type "Quantitative → Categorical" with only one numerical attribute algorithms were developed to maximize the support or the confidence [5] of the rules. This approach was extended to rules "Quantitative, Quantitative → Categorical" with two numerical attributes [4]. A fully automated system for such a type of association rules was proposed in [7], which does not require the user to choose any of the parameters for generating the rules.

A generalization of [4, 5] is given in [11]. First, rules are permitted to contain disjunctions over uninstantiated attributes, i.e., the rules have the type $A \in$

$I_1 \vee A \in I_2 \vee \ldots \vee A \in I_n \rightarrow C$. Second, the number of uninstantiated attributes of a rule is not limited. And third, an uninstantiated attribute can be either numerical or categorical. The attribute is called uninstantiated, if the boundaries of the interval are not specified in advance, i.e., they must be found by the algorithm. The optimal solution with respect to support and confidence turns out to be NP-hard. Therefore, the follow up work [10, 12] concentrates on rules with one or two numerical attributes.

The problem of finding good intervals was addressed in [8]. The drawbacks of the equi-depth partitioning in [13] are pointed out. Consequentially, goals for association rules over interval data are described, which yields in the term *distance-based association rules*. The degree of association is expressed by distance measures. For instance, the rule $C_1 \rightarrow C_2$ has the degree of association of D, if the average inter-cluster distance between C_1 and C_2 is smaller then D.

In [3] a new definition of quantitative association rules based on statistical inference theory was proposed. The goal is to identify subsets of the data, which show a different behavior compared to the full dataset. For instance, selecting a subset can be done by specifying the value of a categorical attribute. The subset is interesting, if the mean and/or the variance of a numerical attribute for this subset differs significantly from the average. Two types of association rules are given: "Categorical → Quantitative" and "Quantitative → Quantitative" rules.

The "Categorical → Quantitative" rules in [3] are most relevant to our work. We present categorical values, which have also a significant impact on the quantitative attributes. In contrast to [3] we pay regard to the complete distribution function of the quantitative attribute and do not restrict ourselves to the mean and the variance. Further on, we take into account, that many frequent itemsets can have a very similar distribution function of the quantitative attribute. Therefore, we determine a grouping of the distribution functions in order to summarize the important information.

3 Algorithms

In this paper we present a framework, which makes it possible to use the items of an itemset to predict the distribution of a numerical attribute. In the next subsection we introduce the notations used in this paper. After that we give a short introduction to the Kolmogorov-Smirnov-Test, which is used in order to compare different distribution functions of quantitative attributes. Finally, two methods are presented: The first method ranks single items with respect to their impact on a quantitative attribute. The second method makes it possible to predict the distribution function of a quantitative attribute.

3.1 Definitions

The underlying database consists of transactions, which in turn consist of items. Formally: The set of items is denoted with $\mathcal{I}$. The number of items is $n_{\mathcal{I}}$, and for simplicity the items are given by consecutive numbers, i.e., $\mathcal{I} = \{1, 2, \ldots, n_{\mathcal{I}}\}$.

The database of transactions is given by $\mathcal{D} = \{t_1, \ldots, t_{n_\mathcal{D}}\}$, where $n_\mathcal{D}$ is the number of transactions. For each i with $1 \leq i \leq n_\mathcal{D}$ it is $t_i \subseteq \mathcal{I}$. Each transaction t_i is associated with a numerical value given by $value(t_i)$.

A set of items is called itemset. The support $sup(I)$ of an itemset $I \subseteq \mathcal{I}$ is defined as the number of transactionsh containing I:

$$sup(I) = |\{t | t \in \mathcal{D} \wedge I \subset t\}|$$

A numerical value is "attached" to each transaction. Therefore, it makes sense to define the set of all numerical values, which are attached to the transactions containing a given itemset I. This set is a multi-set, because duplicates are not removed. It is given by:

$$values(I) = \{value(t) | t \in \mathcal{D} \wedge I \subset t\}$$

The goal of this paper is to analyze $values(I)$ in order to find dependencies between items, itemsets and the numerical attribute.

In this paper we assume that an algorithm to find the frequent itemsets of $\mathcal{I}$ was already applied. The set of the mined frequent itemsets is denoted with $\mathcal{FI} = \{I_1, \ldots, I_{n_{\mathcal{FI}}}\}$. $n_{\mathcal{FI}}$ is the number of frequent itemsets. The value specifying the minimum support parameter is denoted with min_sup.

3.2 The Kolmogorov-Smirnov-Test

The Kolmogorov-Smirnov-Test (KS-Test) [6, 9] tries to determine, if two sets of values differ significantly. The KS-Test has the advantage of making no assumptions about the distribution of the data. That is, it is non-parametric and distribution free. However, this generality comes at some cost: Other tests may be more sensitive to the data. In the context of this paper we can not make any assumptions about the data. For generality we use the KS-Test.

The KS-Test uses the empirical distribution function. For a set of values - in our context $values(I)$ for a given itemset I - the empirical distribution function F_I is defined as follows:

$$F_I(x) = \frac{\text{number of values in}\, values(I)\, \text{that are } \leq x}{\text{number of values in}\, values(I)}$$

For completeness we note, that the distribution function of a random quantity X is given by

$$F(x) = Pr(X < x) = \text{probability that}\, (X < x)$$

The KS-Test requires, that the two sets of values have continuous distribution functions without jumps. In our context these two sets are given by $values(I')$ and $values(I'')$ for corresponding itemsets I' and I''. The test statistic K is the largest absolute deviation between $F_{I'}$ and $F_{I''}$ over the range of the random variable, i.e., over the range of the numerical attribute:

$$K = \max_x |F_{I'}(x) - F_{I''}(x)|$$

This value is compared with a critical value in order to accept or reject the null hypothesis H_0 that the two sets are drawn from the same population. At a significance level of α the critical value D_α is given by

$$D_\alpha = c(\alpha)\sqrt{\frac{sup(I') + sup(I'')}{sup(I') \cdot sup(I'')}}$$

where $c(\alpha)$ depends on the significance level:

α	0.10	0.05	0.025	0.01	0.005	0.001
$c(\alpha)$	1.22	1.36	1.48	1.63	1.73	1.95

If K is equal or larger than D_α the null hypothesis must be rejected. In order to use the value of D_α it must be ensured that both $sup(I')$ and $sup(I'')$ are larger than 40. Otherwise there are equations for D_α, which are more adequate for smaller values of $sup(I')$ and $sup(I'')$. The values K and D_α depend on I' and I''. Therefore, we had to write $K(I', I'')$ and $D_\alpha(I', I'')$. But, if the context is clear, we write simply K and D_α.

The work presented in this paper requires it, that different pairs $(F_{I'}, F_{I''})$ of empirical distribution functions must be compared. Because the interpretation of $K(I', I'')$ depends on $D_\alpha(I', I'')$, which in turn depends on the support of the corresponding itemsets, a normalization of $K(I', I'')$ is needed. This is achieved by defining

$$KS(I', I'') = \frac{K(I', I'')}{D_\alpha(I', I'')}$$

With this definition the similarity of itemsets is given by the similarity of the corresponding empirical distribution functions of $values(I')$ and $values(I'')$. The detailed application of this similarity measure will be clear in the next subsections.

3.3 Analyzing $\mathbf{F}_I$ for single items

In this subsection we will give a method to answer the question: "There are any items such that the set of transactions containing this particular item has an unusual distribution function of the numerical attribute?" We define the term "unusual" as "different from the average", which refers to the similarity of two distribution functions. This can be formalized as follows: The item $\mathsf{j} \in \mathcal{I}$ corresponds to the itemset $\{\mathsf{j}\}$, and the set of corresponding numerical values is given by $values(\{\mathsf{j}\})$. Further on, the empirical distribution function is given by $F_{\{\mathsf{j}\}}$. The average is simply given by all transactions, which can be achieved by using $\emptyset$ as the particular itemset, i.e., we use $values(\emptyset)$ and $F_\emptyset$.

In order to express the difference to the average we use the KS-Test, i.e., for a given item j the value $KS(\{\mathsf{j}\}, \emptyset)$ is used. This value measures the deviation of $values(\{\mathsf{j}\})$ from $values(\emptyset)$. An item j with a high value $KS(\{\mathsf{j}\}, \emptyset)$, i.e., larger than one, is interesting, because the numerical values of transactions containing this item have a distribution function significantly different from the average.

For the practical application all items are sorted with respect to $KS(\{\mathfrak{j}\}, \emptyset)$. Then the user can inspect the most interesting items: Items with a high value of $KS(\{\mathfrak{j}\}, \emptyset)$, as mentioned in the previous paragraph, as well as items with a small value, i.e., close to zero. Those items are interesting, because the corresponding distribution functions are very similar to the average. This sorting criterion for items is simple and useful. Items with a specific impact can be identified and give insight into to data. Experimental results are presented in section 4.

3.4 Identifying items with strong impact

Suppose we have a subset of the transactions. Therefore, we can compute the distribution function of the numerical attribute. Different subsets can result in different shapes of the distribution function. Two things are important to note: First, in this section, such a subset is defined by all transactions containing contain a given (frequent) itemset. Second, we want to identify items, whose occurrence in an itemset is an indicator for a characteristic shape of the distribution function. To give an example we refer to the example given at the end of section 1.

As mentioned, there might be several different shapes of the distribution functions (depending on the subsets). For now we assume, that we identified n_C such groups, each named with C_i for $1 \leq i \leq n_C$.

An item typically belongs to different itemsets. In this paper the items with strong impact are those items, whose probability $P(C_i|\mathfrak{j})$ is very high. This probability describes the following: If an itemset I contains item $\mathfrak{j}$, then the distribution function of $values(I)$ has the shape of C_i with probability $P(C_i|\mathfrak{j})$.

The question arises how to define the groups of shapes: In this paper one group consists of itemsets, such that the distribution functions are similar to each other. Further on, itemsets from different groups have dissimilar distribution functions. This is exactly the definition of a clustering, i.e., a clustering of the itemsets.

The next subsection gives information about the clustering step. The subsection afterwards continues to explain how to identify the items.

Applying the clustering algorithm In order to find a clustering of the itemsets the similarity of two itemsets is expressed by the similarity of the corresponding distribution functions. To specify the similarity the Kolmogorov-Smirnov-Test is used as explained in section 3.2.

As the clustering algorithm we use a hierarchical, agglomerative clustering algorithm. This algorithm merges the two clusters, which have the highest similarity, starting with each itemset as its own cluster. The algorithm merges until the specified number of clusters n_C is reached. Finding the best number of clusters is still an open question in theory and practice. The answer depends on the data, and typically one have to try different settings.

The motivation of using a hierarchical algorithm is twofold: First, the input of such an algorithm is a similarity matrix. Only the similarity of the original, unmerged itemsets is used by the algorithm. Therefore, we do not have to define the centroid or the mean for sets of itemsets. Second, the similarity could be determined with a metric. In this case it would be necessary to approximate the distribution function with binning and histograms in order to get a vector-space. This leads to the problem of finding an appropriate binning and an appropriate metric, and problems with bin boundaries must be handled. Using a statistical test as a similarity function might be an intuitive and simple solution for those issues.

The similarity matrix is denoted with $\mathcal{S} \in \mathbb{R}^{n_{\mathcal{FI}} \times n_{\mathcal{FI}}}$, and the entry at (i, j) is given by

$$\mathcal{S}_{ij} = KS(I_i, I_j)$$

where $1 \leq i, j \leq n_{\mathcal{FI}}$, i.e., I_i and I_j are frequent itemsets. Note that $\mathcal{S}$ is a symmetric matrix. The clustering partitions the set of frequent itemsets into n_C clusters, namely C_i for $i = 1, \ldots, n_C$. It is $\cup_{i=1}^{n_C} C_i = \mathcal{FI}$ and $C_i \cap C_j = \emptyset$ if $i \neq j$ for $1 \leq i, j \leq n_C$.

Identifying the items Now we come back to the goal of identifying items with a strong impact, i.e., we want to identify items $\mathsf{j} \in \mathcal{I}$ with high $P(C_i|\mathsf{j})$ for any C_i $(1 \leq i \leq n_C)$. It turns out, that we can compute this probability with Bayes's Law.

The probability that an itemset from a given cluster C_i contains the item j is given by

$$P(\mathsf{j}|C_i) = \frac{|\{I : I \in C_i \wedge \mathsf{j} \in I\}|}{|C_i|}$$

Further on, the probability of cluster i is given by

$$P(C_i) = \frac{|C_i|}{n_{\mathcal{FI}}}$$

The first equation enables us to predict the presence of an item in an itemset – under the condition that this itemset belongs to a particular cluster. The second equation enables us to predict a cluster.

The goal of this paper is to predict the cluster (that is, the distribution function) of an itemset – under the condition that the itemset contains a given item.

The prediction of the cluster corresponds to the probability $P(C_i|\mathsf{j})$. With Bayes's Law this probability is given by

$$P(C_i|\mathsf{j}) = \frac{P(\mathsf{j}|C_i) \cdot P(C_i)}{\sum_{k=1}^{n_C} P(\mathsf{j}|C_k) \cdot P(C_k)}$$

The interpretation of this probability is as follows: Given an itemset I, which contains an item j, the probability, that the itemset I has a distribution function similar to the distribution functions of itemsets in cluster C_i, is given by $P(C_i|\mathsf{j})$.

It is clear, that one is interested in items j with a high value of $P(C_i|\mathsf{j})$ for any i ($1 \leq i \leq n_C$). These are the items, which have a strong impact on the distribution function of the numerical attribute. The predictive power of those items is very high.

4 Evaluation

The methods presented in this paper were evaluated with the *adult* dataset from the *UCI Machine Learning Repository* at `http://www.ics.uci.edu/~mlearn`. The dataset contains continuous and categorical attributes. Each transaction consists of 13 values. Continuous values include age, amount of capital gain and capital loss in dollars, number of working hours per week and number of years of education. Categorical values include the marital status, the education, the occupation, the workclass, the relationship, the race, the sex and the home country. In our analysis we do not use the zip code and the class label. The set of items is constructed as follows: It is the union of the domains of the categorical attributes. For the evaluation the *Age* attribute is chosen as the numerical attribute, because the findings can be easily verified. The findings about single items are reported in the next subsection. Results about predicting the distribution function are reported afterwards.

4.1 Analysis of single items

In this section we show the ordering of the items of the *adult* dataset. The item j is ordered with respect to the difference between the distribution function of $values(\{\mathsf{j}\})$ and the distribution function of $values(\emptyset)$. Figure 1 presents the results. From the figure it can be seen, that persons with the item *Never Married* for marital status are considerably younger than the average. Further on, it can be seen, that persons which are *Husband*, are older than the average. Conversely, the attribute *United States* does not have an impact on the distribution function of *Age*. The census dataset comes from the US, and therefore, nearly all people, i.e., the majority which forms the average, have this attribute value.

4.2 Items with strong impact

In this subsection we present the items of the *adult* dataset, which are capable of predicting the distribution function of the numerical attribute.

As pointed out in section 3.4 the first step is to find the frequent itemsets. There are 4135 frequent itemsets when choosing $min\_sup = 300$ (absolute value) corresponding approximately to 10% of the data. The set of numerical values for each frequent itemset is computed, which is required for computing the similarity matrix $\mathcal{S}$, whose entries are the normalized test statistic of the KS-Test.

The similarity matrix is the input to the clustering algorithm. We specified the number of cluster with $n_C = 3$. The motivation for $n_C = 3$ will be explained

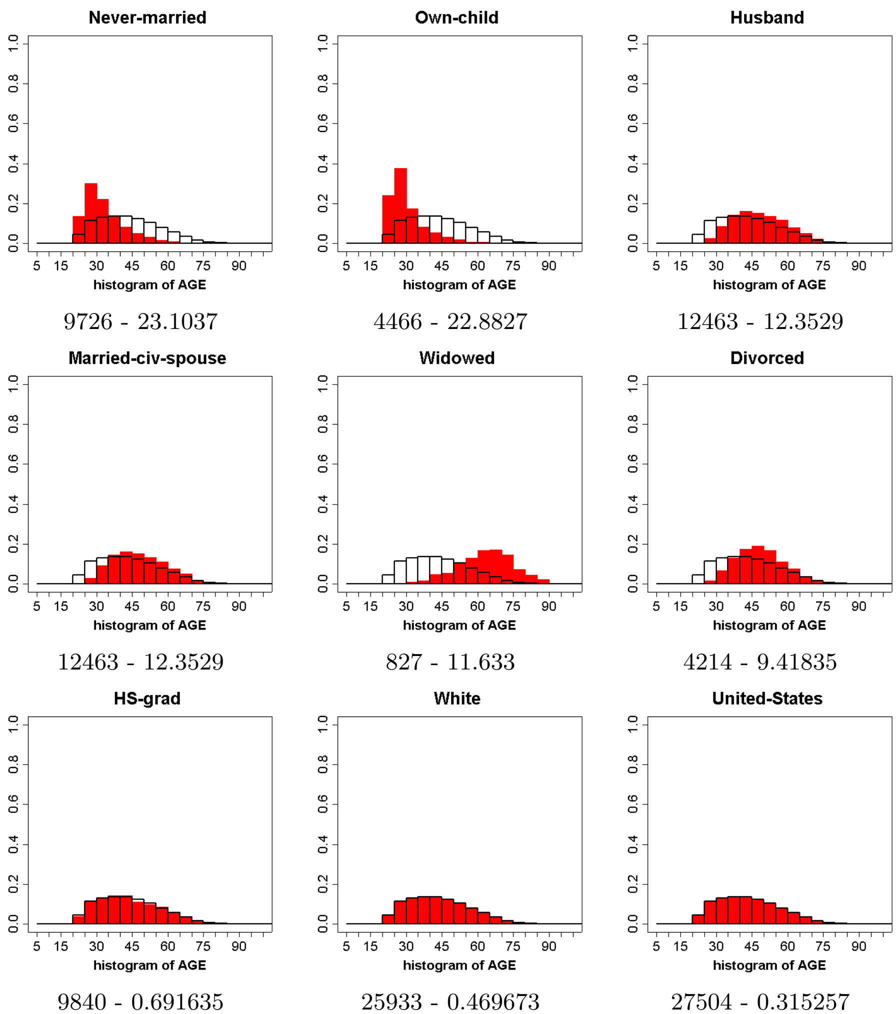
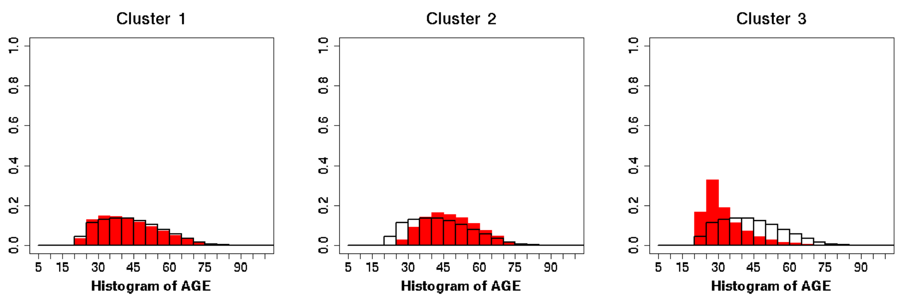


Fig. 1. This figure shows the ordering of the items of the *adult* dataset. Each diagram corresponds to one item, and the name is given on top of the diagram. The red bars show the distribution function of $values(\{j\})$, i.e., the age pattern for the given item is shown. For comparison, the transparent (white) bars show the distribution function of $values(\emptyset)$, i.e., the age pattern for the full dataset is shown. The support of the item is given below the histogram (left value), and the difference to the average is given by the KS value (right value). The figure shows items, which are either very similar or very dissimilar to the average.

later in this paragraph. In order to understand the clustering of the itemsets, i.e., the clustering of the distribution functions, for a given cluster the distribution function of *Age* is computed as follows: The sets $values(I)$ for all itemsets of a given cluster are merged, and the histogram over this set is computed. Note that multi-sets are used, i.e., duplicate values are not removed. In the case of the *adult* dataset *Age* ranges from 0 to 99. This range is divided into equi-width bins of width 5. The resulting histograms and explanations are given in Figure 2. One can see, that the histograms are pairwise different. When choosing a higher value for n_C this desired property will disappear. That is, when choosing a higher value for n_C two (or more) clusters will have a similar distribution function, or multiple clusters have a distribution function similar to the average. That is, the uniqueness of the clusters disappears.



986 itemsets
(1)

2163 itemsets
(2)

986 itemsets
(3)

Fig. 2. The characteristic distribution function for each cluster is given. The red bars show the distribution function, i.e., the age pattern for the cluster is shown. For comparison, the transparent (white) bars show the distribution function, i.e., the age pattern, of the full dataset is shown. The number of itemsets in a cluster is given below the histogram. From the diagrams the following can be seen: The first cluster contains itemsets I whose distribution function of $values(I)$ is similar to the average. The second cluster contains itemsets I whose distribution of $values(I)$ shows, that young people are missing. The third cluster contains itemsets I whose distribution of $values(I)$ shows, that the fraction of young people is above average.

In order to use an item to predict the distribution function the probabilities $P(C_i|\text{j})$ are computed. Figure 3 gives probabilities for the three clusters and all items which are represented in frequent itemsets. The probabilities, i.e., the prediction, have to be applied as follows: The item *Never married* has a high probability, almost 1, for Cluster 3. That means, that for nearly all itemsets with item *Never married* the distribution function of the numerical attribute has a shape similar to the one shown in Figure 2(3). A similar statement can be made for item *7th-8th*. In this case all itemsets with this item have a distribution function of the numerical attributes similar to 2(2).

j	$P(C_1\|j)$	$P(C_2\|j)$	$P(C_3\|j)$	j	$P(C_1\|j)$	$P(C_2\|j)$	$P(C_3\|j)$
10th	0.889	0.111	0.000	Married-spouse-absent	0.000	1.000	0.000
11th	0.000	0.050	0.950	Masters	0.038	0.961	0.000
12th	0.000	0.000	1.000	Mexico	1.000	0.000	0.000
7th-8th	0.000	1.000	0.000	Never-married	0.058	0.000	0.941
9th	0.800	0.200	0.000	Not-in-family	0.540	0.185	0.274
Asian-Pac-Islander	0.333	0.583	0.083	Other-relative	0.076	0.000	0.923
Assoc-acdm	0.468	0.510	0.021	Other-service	0.264	0.190	0.545
Assoc-voc	0.333	0.640	0.027	Own-child	0.003	0.000	0.996
Bachelors	0.283	0.519	0.197	Private	0.292	0.417	0.290
Black	0.456	0.327	0.215	Prof-school	0.000	1.000	0.000
Craft-repair	0.232	0.614	0.153	Prof-specialty	0.270	0.615	0.113
Divorced	0.004	0.996	0.000	Protective-serv	0.272	0.727	0.000
Doctorate	0.000	1.000	0.000	Sales	0.282	0.434	0.282
Exec-managerial	0.166	0.770	0.063	Self-emp-inc	0.000	1.000	0.000
Farming-fishing	0.107	0.892	0.000	Self-emp-not-inc	0.031	0.968	0.000
Federal-gov	0.000	1.000	0.000	Separated	0.653	0.346	0.000
Female	0.328	0.293	0.377	Some-college	0.227	0.426	0.346
HS-grad	0.255	0.546	0.198	Tech-support	0.409	0.568	0.022
Handlers-cleaners	0.364	0.000	0.636	Transport-moving	0.190	0.793	0.015
Husband	0.029	0.970	0.000	United-States	0.233	0.528	0.237
Local-gov	0.175	0.824	0.000	Unmarried	0.208	0.717	0.073
Machine-op-inspct	0.309	0.619	0.071	White	0.231	0.533	0.235
Male	0.214	0.606	0.178	Widowed	0.000	1.000	0.000
Married-civ-spouse	0.041	0.958	0.000	Wife	0.235	0.764	0.000

Fig. 3. The probabilities $P(C_i|j)$ for the three clusters C_1, C_2 and C_3 and the items j are given. Note that only items which appear in at least one of the frequent itemsets are listed. The interpretation of the table is given in the text.

5 Conclusions

In this paper we presented methods to analyze the dependencies between the items and the distribution function of a numerical attribute. In a first step the single items are ordered with respect to their impact on the numerical attribute. This enables to find subsets with a behavior different from the average. In the second step groups of similar distribution functions were identified. A method was proposed in order to find relations between items and the groups of distribution functions. The relations allow predicting a group of distribution functions from the item.

Our future work is directed as follows: One goal is to understand the influence of the parameters *min_sup* and n_C, and to develop criteria for good values. Further on, we want to better understand the term “impact on the numerical attribute”. Until now we can say, that, e.g., *Never married* has a strong impact on *Age*. However, there might be exceptions to this rule, which should be covered by our framework. Additionally we want to define the “impact” of combinations of more than one item.

References

1. Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD*

International Conference on Management of Data, pages 207–216. ACM Press, 1993.
2. Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann, 1994.
3. Yonatan Aumann and Yehuda Lindell. A statistical theory for quantitative association rules. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 261–270. ACM Press, 1999.
4. Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining using two-dimensional optimized accociation rules: Scheme, algorithms, and visualization. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 13–23. ACM Press, 1996.
5. Takeshi Fukuda, Yasuhiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Mining optimized association rules for numeric attributes. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 182–191. ACM Press, 1996.
6. Donald Erwin Knuth. *The Art of Computer Programming*, volume 2 / Seminumerical Algorithms. Addison-Wesley Publishing Company, 3rd edition, 1997.
7. Brian Lent, Arun N. Swami, and Jennifer Widom. Clustering association rules. In *Proceedings of the 13th International Conference on Data Engineering*, pages 220–231. IEEE Computer Society, 1997.
8. Renée J. Miller and Yuping Yang. Association rules over interval data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 452–461. ACM Press, 1997.
9. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 2002.
10. R. Rastogi and K. Shim. Mining optimized gain rules for numeric attributes. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 135–144. ACM Press, 1999.
11. Rajeev Rastogi and Kyuseok Shim. Mining optimized association rules with categorical and numeric attributes. In *Proceedings of the 14th International Conference on Data Engineering*, pages 503–512. IEEE Computer Society, 1998.
12. Rajeev Rastogi and Kyuseok Shim. Mining optimized support rules for numeric attributes. In *Proceedings of the 15th International Conference on Data Engineering*, pages 126–135. IEEE Computer Society, 1999.
13. Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12. ACM Press, 1996.