Mutual Information based Clustering of Market Basket Data for Profiling Users

Bartholomäus Ende
E-Finance Lab
Johann Wolfgang Goethe-University
60054 Frankfurt, Germany
ende@wiwi.uni-frankfurt.de

Rüdiger Brause
Dept. of Computer Science and Mathematics
Johann Wolfgang Goethe-University
60054 Frankfurt, Germany
r.brause@informatik.uni-frankfurt.de

Abstract

Attraction and commercial success of web sites depend heavily on the additional values visitors may find. Here, individual, automatically obtained and maintained user profiles are the key for user satisfaction. This contribution shows for the example of a cooking information site how user profiles might be obtained using category information provided by cooking recipes. It is shown that metrical distance functions and standard clustering procedures lead to erroneous results. Instead, we propose a new mutual information based clustering approach and outline its implications for the example of user profiling.

1 Introduction

The internet's growth steadily fosters competition among portals whose operators aim at selling all kind of services and products. Therefore, one of their main concerns is high attractivity as commercial sites being uninteresting or non-functional loose their customers rapidly. One promising way for portal operators is the adaptation to changing user needs and interests. This can be achieved by collecting and analyzing information of all users visiting their web site. The aggregated, user related information is called user profile and is difficult to obtain in the anonymous as well as stateless world of the HTTP protocol.
In this contribution we use the example of a cooking site, providing recipes, tools and advice for cooking. In this context, user profiles might be utilized for

- *individualizing feedback emails* to users in order to provide notice only for recipes and goods that correspond to the user's information desires,
- *generating recommendations* for interesting portal contents that have not been visited yet,
- *ameliorating the site's structure* and
- *structuring discussion groups* as well as *identifying their main interests*.

To obtain data for these purposes portal operators might either record *page view information* like time duration, or explicit *user feedback*, e.g. ratings, downloads and printouts of specific texts as well as any other kinds of services.
Beside the multiple applications from above, the modeling of state transitions and sequences is not covered by this contribution. Attempts that compute user profiles by modeling user action sequences via *Markov chains* can be found in [20] and [2]. Other approaches like those of [18] and [21] incorporate frequent item sets for predicting page visits. [14] even advances this idea by introducing *click stream trees* which represent frequent sequences of page views.
One common enhancement of all these approaches is the reduction of dimensionality by clustering portal contents. Thus, for market basket data these applications might benefit from the mutual information based clustering approach described in this paper.

2 Methods for forming user profiles

User profiles are supposed to reflect preferences. For the cooking site the only indications available for these preferences are the visited recipes which in turn are composed of ingredients. Further, our data does not possess any quantitative weighting, except the binary information concerning an ingredient's presence within a recipe. Thus, for this case user profiles can be obtained from user-typical item collections which might be computed by the following three strategies:

1. The classical *bottom-up approach for basket data analysis*, e.g. using the A-Priori Algorithm [17] which aggregates all current items (here ingredients) to a few typical sets. Unfortunately, the $d = 45$ ingredients let this become infeasible as they lead to a combinatorial explosion of $2^{45} > 35 \cdot 10^{12}$ possible set patterns.

2. The *top-down approach* that takes the existing recipes patterns and aims at generalizing them [4, 19]. This yields to more valid rules for user preferences than the first one, but the approach also suffers from the combinatorial explosion although some heuristics described in [19] might alleviate this problem a bit.
3. Sets of user preferences might also be regarded as *corresponding to clusters*. Following this perspective the objective is to identify classes within the data set via cluster analysis methods.

We have chosen the third approach for this contribution. Since no classes are defined a priori, suitable clustering methods are to seek for structures within all users' data. After obtaining this classification basis, assignments of classes for each user or actual visit statistics for different classes can be stored as individual user profiles. For the clustering two different options exist:

1. A *statistical* data base *clustering* that is obtained from all recipes available on the web site's server. Thus, its clusters reflect the internal occurrence probabilities of ingredient combinations within the data base.
2. A *user weighted clustering* that is based on all recipes weighted by their download frequencies. Here, clusters represent the users' interest focuses.

3 Clustering Requirements for Market Basket Data

Traditionally, algorithms for analyzing cluster structures are based on distance measures for pairwise comparisons of data patterns. Further, multiple parameters are used including average distances within clusters, means of cluster attributes and maximal intra as well as minimal inter cluster pattern distances [8]. Unfortunately, these approaches are inherently based on metrical data, i.e. pattern tuples consisting of real numbers.
The cooking site provides information for recipes downloaded by registered users. Further, the recipe patterns are represented by the corresponding ingredients omitting their quantitative weighting. Thus, the objective is to cluster categorical patterns and not metrical ones. The following definition states this task more precisely.

Definition: Clustering of item sets
Let $\{x_i\}$ *denote the set of all categorical attributes, called* items. *Then, a categorical pattern or* item set pattern *is defined as a set of these items' categorical values* c_j

$$x = \{x_i = c_j | i = 1, ..., d, j = 1, ..., m_i\} = (x_1, x_2, ..., x_d)$$

where each item x_i *can take* m_i *such possible values and* $m_i = 2$ *for binary data. All observed item set patterns form the data base* D*. A clustering of* D *is defined as a partition* $\mathcal{C}_k$ *of the data base into* k *disjoint subsets* Ω_i*, e.g.*

$$\bigcup_{i=1}^{k} \Omega_i = D, \qquad \Omega_i \cap \Omega_j = \emptyset \ \textit{for} \ i \neq j \tag{1}$$

Further, ω_k *denotes that a pattern* x *belongs to subset* Ω_k*.*

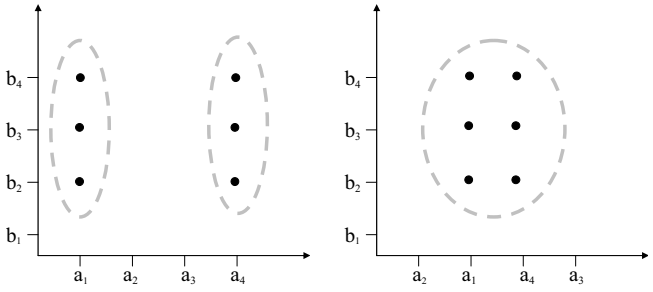

Figure 1: (a) Metrical and (b) categorical clustering

Different from metrical clusterings, categorical data lacks meaningful distance functions among its symbolic entities. To illustrate this challenge, we consider the six 2-dimensional item set patterns defined by the attributes a and b as $(a_1, b_2), (a_1, b_3), (a_1, b_4), (a_4, b_2), (a_4, b_3), (a_4, b_4)$. Fig. 1(a) depicts them within metrical coordinates assuming that $a_i = i$ and $b_j = j$. Within this metrical setup two clusters, highlighted by grey dotted circles, can easily be identified. But with data exhibiting a pure categorical interpretation, relabeling the attributes changes the scene completely. E.g., when the variable a represents *colors* no inherent order exists among the variable's values. Since, intrinsically also no distances are defined among colors, the order of a_1 and a_4 might be changed leading to the situation that is depicted in Fig. 1(b). Here, only one class instead of the two can be deduced. To overcome these kinds of challenges for categorical data literature proposes only few approaches [13]. Their main problem is the definition of an adequate distance measure which meets the requirements that

- members of one cluster are supposed to be as similar as possible and
- members from different clusters shall exhibit no or only little similarities i.e. possess high pairwise distances.

The computations for implementing these demands are NP hard [12] and depend heavily on the distance function [10]. For binary encodings, as those of market basket data or categorical item sets, which contain a one for an attribute's presence and a zero for an attribute's absence, different distance measures have been proposed like the Hamming Distance or the City Block Metric. For binary encodings these measures are equal and compute distance rankings identical

cluster 1		cluster 2	
pattern	present items	pattern	present items
x^1	$\{1,3\}$	x^6	$\{2\}$
x^2	$\{1,3,9\}$	x^7	$\{2,4,5\}$
x^3	$\{3,9\}$	x^8	$\{2,4,5,6,10\}$
x^4	$\{3,7\}$	x^9	$\{2,4,5,6,8,10\}$
x^5	$\{7,9\}$	x^{10}	$\{2,4,5,8,10\}$
		x^{11}	$\{4,5,6,8,10\}$
		x^{12}	$\{4,5,6,8\}$

Table 1: Clustering of item set patterns

to those of the Euclidian Distance. Thus, we will further consider the Hamming Distance only.
Table 1 presents a second example. It depicts an idealized clustering of binary market basket data and shall illustrate the following six typical characteristics of this data type:

1. Item set patterns might possess different numbers of items that are present.
2. Compared to the overall number of possible items only few ones are actually present (e.g. $x^i : i = 1, ..., 5$).
3. Item set patterns might be rather similar although possessing different amounts of present items (e.g. x^6 and x^9).
4. Data might be scattered as it is typically made of a small fraction of the 2^d discriminable item set patterns.
5. Clusters are characterized by individual or joint occurrences of typical items that are present (e.g. the presence of the 2^{nd} or 4^{th} item for the 2^{nd} cluster).
6. Item set patterns belonging to the same cluster might share little or even no present items at all. Thus, an item set patterns's cluster affiliation is established via other item set patterns possessing typical items for this cluster (e.g. item set patterns x^1 and x^5 are interconnected by x^2, x^3 and x^4).
7. The number of item set patterns within clusters might vary. Further, the larger a cluster is the lower the average similarity among its cluster members becomes as for an item set patterns x there are $\sum_{j=0}^{i} \binom{d}{j}$ item set patterns x' with a Hamming distance $\leq i$. Among these the maximal distance of $2i$ appears quite often as there $\binom{d}{i}$ border item set patterns can be found and each of them in turn possesses $\binom{d-i}{2\,i}$ neighbors with a distance of $2i$.

The immediate implications concerning distance measures for binary market basket data are twofold:

1. The superior number of absent items requires an asymmetrical treatment of their matches compared to those for items being present. E.g. within the cooking site, recipes are represented by 45 ingredient categories whereas on average only ingredients from 7 of these categories are used.
2. Measures like the Hamming Distance are misleading for binary market basket data as they bear the risk of assessing object pairs sharing present items to be less similar than those lacking this similarity (e.g. $dist_{Ham}(x^1, x^6) < dist_{Ham}(x^9, x^6)$).

To preserve the geometrical perspective of traditional clustering algorithms we propose to employ *Jaccard's Coefficient* [22]. This measure accounts only for matches of present items as it standardizes the Hamming Distance via a division by the number of items being present at least within one object. Unfortunately, it turns out that several classical methods still yield to erroneous results.
Partitioning clustering approaches that are based on the principle of minimizing variance of cluster members around central objects (*prototypes*) are misleading, because prototypes cannot capture cluster characteristics like multiple attribute co-occurrences. Further, their objective seeks for spherical, equal sized clusters tending to split larger ones as this decreases variance. Characteristic 7 also claims that within larger clusters object dissimilarities increase. This leads to prototype patterns converging to zero vectors because with increasing cluster size more items are present within the clusters but their relative occurrence frequencies decrease. These considerations are supported by validations of multiple partitioning approaches on our data set via the *Silhouette Coefficient* proposed by Kaufman and Rousseuw [10] concluding that no cluster structure prevails or these methods are simply not the right one.
Density-based methods are not applicable here either, as categorical data possesses no such metrical proportions. Instead, objects are placed in the corners of the d-dimensional hypercube.
Although, the *hierarchical clustering approach* seems promising because of its flexibility, traditional distance measures like the *Single-*, *Complete-* and *Average-Link approaches* lack a global perspective that is required to satisfy characteristic 6. To overcome this problem Guha proposes the ROCK (RObust Clustering using linKs) Algorithm [13]. It defines a neighborhood via links among objects whose similarity exceeds a predefined threshold. Further, its objective function aims at maximizing the number of within cluster links via a hill-climbing procedure. Although this approach is reported to work well in the case of pure categorical data, for the case of binary market basket data it seems less promising as its main assumption is a constant number of ones, i.e. a constant number items being present which is not satisfied in our case. Therefore, we have not further investigated this approach.
In conclusion, a geometric perspective employed by traditional clustering algorithms is not suitable for binary market basket data. Furthermore, instead of local, pairwise comparisons it seems that a global perspective accounting for typical cluster characteristics is more required.

4 A mutual information based approach

As profiles represent approximations of the user's information desire, a probabilistic perspective seems suitable. For this purpose a desirable objective is the maximization of the information for the guess of an item's presence within one pattern on the basis of its cluster affiliation. To further motivate this idea, one might consider evaluation methods for cluster quality. Among them one approach used for metrical patterns is the calculation of bits that are required to codify the cluster assignments. This corresponds to the so called minimum description length (MDL) principle [8, 15]. It can be shown that the lower bound for the description length L equals the entropy of the data set's probability distribution. Therefore, the best clustering is the one which minimizes the entropy of the clustering. This can be achieved by minimizing the *conditional entropy* $H(X|k)$ [7] of the patterns within the k clusters where X is a multidimensional random variable, taking values from all possible item sets, and k is a random variable which describes the patterns' cluster affiliation. Further, [16] argue that for binary objects that result from a superposition of multiple Bernoulli distributions a clustering with minimal cluster entropy $H(X|k)$ is an optimal estimation of this model according to the maximum-likelihood principle. Certainly, as the entropy $H(X)$ for all patterns is constant for the data set, the optimal clustering $\mathcal{C}$ maximizes also the **mutual information** $I(X;k)$ between the patterns X within a cluster and the cluster assignment itself.

$$\min_{\mathcal{C}} H(X|\omega_k) \iff \max_{\mathcal{C}}[H(X) - H(X|\omega_k)] = \max_{\mathcal{C}} I(X;\omega_k) \quad (2)$$

Thus, as desired a categorical pattern's cluster assignment indicates its items' presence values and vice versa. The objective to maximize mutual information corresponds to a minimization of an *information based distance measure* between two clusters Ω_i and Ω_j implemented by conditional entropy. This distance is expressed as the difference between the conditional entropy of all fused clusters $\Omega_i \cup \Omega_j$ and the average of the particular clusters:

$$d_I(\Omega_i, \Omega_j) = H(X| \bigvee_{k\in\{i,j\}} \omega_k) - \langle H(X|\omega_k)\rangle_k \quad (3)$$
$$= H(X|\omega_i \vee \omega_j) - H(X|\omega_i)P(\omega_i) - H(X|\omega_j)P(\omega_j)$$

If the items X_i are independent, the equation for computing the entropy can be simplified to [7]

$$H(X) = H(X_1, X_2, ..., X_d) = \sum_{i=1}^{d} H(X_i) \quad (4)$$

which is also valid for the mutual information

$$\max_{\mathcal{C}} I(X;\omega_k) = \max_{\mathcal{C}} \sum_{i=1}^{d} I(X_i;\omega_k) \quad (5)$$

By this, we approximate the probability density function (pdf) of the compound random variable by its marginal densities facilitating our task significantly. Under the independence assumption, we are not supposed to compute the relative frequencies of all $|a_i|^d$ possible patterns any more which is quite a high number in our case: For $a_i \in \{0,1\}$ and $d = 45$ items per recipe pattern one encounters $2^{45} > 35 \cdot 10^{12}$ possible patterns, corresponding to more than a million times of the data available. Therefore, it is impossible to compute the high-dimensional pdf anyway. Nevertheless, the assumption of independent items is neither meaningful nor valid. First, without any dependencies there is no basis for clustering as all combinations are equally probable. Second, a correlation analysis revealed that there are some medium correlations which imply also that dependencies exist. Certainly, if the dependencies are too strong, we will not be able to separate the pattern set into distinct clusters anymore as there will be only one cluster or none. Therefore, we will assume only small or moderate dependencies in the remaining paper and will check this later, see section 5. Thus, the mutual information based clustering algorithm (MIBAC) can be formulated like this with $O(d\,n^2 \log(n))$ time and $O(d\,n^2)$ space requirements:

Algorithm MIBAC: Mutual **I**nformation **BA**sed **C**lustering

Input: An item set pattern database $D = \{x^i | i = 1, ..., n\}$
Objective: Compute a hierarchy of clusterings $\{\mathcal{C}_k\}_{k=n,...,1}$ and return the best one.

1: Create an initial clustering $\mathcal{C}_k = \{\Omega_i = \{x^i\} | x_i \in D\}$.
2: Build a heap h that stores the information distance $d_I(\Omega_i, \Omega_j)$ for all cluster pairs. This requires $O(d\,n^2)$ time and space.
3: **for** $k = n$ to 2 **do** repeat until only one cluster is left:
4: Retrieve $\{\Omega_i, \Omega_j\}$ from heap h with minimal distance d_I.
5: Fuse them to $\Omega \leftarrow \Omega_i \cup \Omega_j$.
6: Define the $k-1$ clusters for the next hierarchy stage by replacing the fused sets with their fusion:
$\mathcal{C}_{k-1} \leftarrow (\mathcal{C}_k \setminus \{\Omega_i, \Omega_j\}) \cup \Omega$
7: Remove from heap h all distances d_I regarding Ω_i or Ω_j.
8: For all $\Omega_x \in \mathcal{C}_k : x \neq i,j$ add the distance $d_I(\Omega_x, \Omega)$ to to the fused set Ω to heap h. Steps 7 and 8 take each $O(d\,n\,log(n))$ time.
9: Evaluate the quality for each clustering stage k.
10: **return** the clustering that yields best quality.

The above time and space considerations require calculating the distance between two clusters (see eq. (3)) in time $O(d)$. This can be achieved under the independence assumption by storing d probabilities for each cluster concerning the items'

presence. In order to maintain these probabilities after a fusion the cluster weights have to be stored also. For practical applications the item number d is bounded by a constant due to the curse of dimensionality.
For the evaluation of the computed clusterings multiple measures shall be considered. Beside mutual information it turns out that classification success statistics are useful to judge the interconnection between cluster assignments and attribute characteristics. This is further supported by [16] who claim that clusters possessing minimal entropy $H(X|\omega_k)$ maximize Symon's Classification Likelihood. The evaluation via classification success statistics require prior knowledge of the data categories. Therefore, we assume the computed clusters to correspond to class labels and check how well common classifiers can capture the classes' pattern sets. For this evaluation we choose *precision* and *recall* from Information Retrieval [2]. For a set of patterns D, a given query q with its result set Q on D and the subset Ω of all relevant patterns within Q the precision of the query is defined as:

$$Prec(q, D) = \frac{|\Omega|}{|Q|} \tag{6}$$

The relation of Ω to all relevant patterns R within D defines the recall:

$$Rec(q, D) = \frac{|\Omega|}{|R|} \tag{7}$$

In order to reflect both precision and recall we use their harmonic mean that is typically called the F_1-*Measure*:

$$F_1(q, D) = \frac{2 \cdot Prec(q, D) \cdot Rec(q, D)}{Prec(q, D) + Rec(q, D)} \tag{8}$$

5 Results

In this section we will present the results of the mutual information based clustering approach. Therefore, we start with a brief presentation of the data set before the characteristics of our clustering procedure are discussed.

5.1 The data set

The used data basis constitutes of site usage records over a 6 week period in 2005 that subsumes to over 18 millions user actions for computing statistical user states. Our first concern has been the extraction of useful attributes, i.e. items within recipes. Therefore, we focused on the item distribution. Fig. 2 depicts the item (ingredient) occurrence frequency. As expected, only few ingredients are included frequently in recipes (> 100), whereas the majority of the ingredients is referenced seldom. The dotted line represents the average item reference count computed out of four

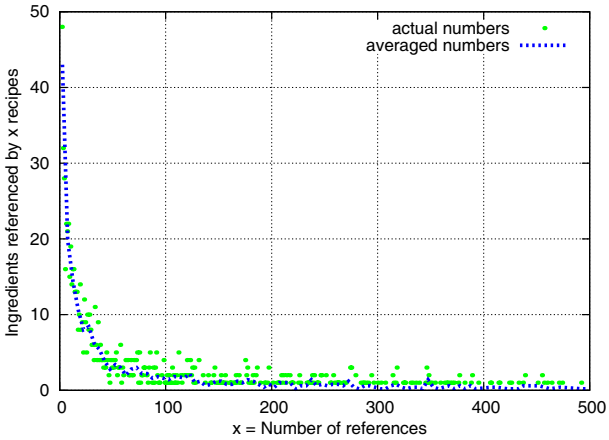

Figure 2: Frequency of item references by all recipes

neighbors corresponding to a power law distribution [2]:

$$P(k) = \frac{1}{k^\alpha} \left[\sum_{i=1}^{\infty} \frac{1}{i^\alpha} \right]^{-1} \tag{9}$$

For a manageable item number one typical feature selection procedure is the removal of infrequent items. On first sight a threshold of 4% seems suitable. This selection yields to 47 items. But these are referenced only within 12% of the recipes which is not much. Therefore, we aimed at broadening our data basis by mapping similar items into the same category using a self-made thesaurus as well as standard text pre-processing stages like stemming. Fig. 3 exhibits a comparison between the number of resulting categories within recipes and the initial items. As one can see, an average recipe contains about 10 items (ingredients), but only about 7 categories which results from a mapping of multiple ingredients within one recipe to the same category.

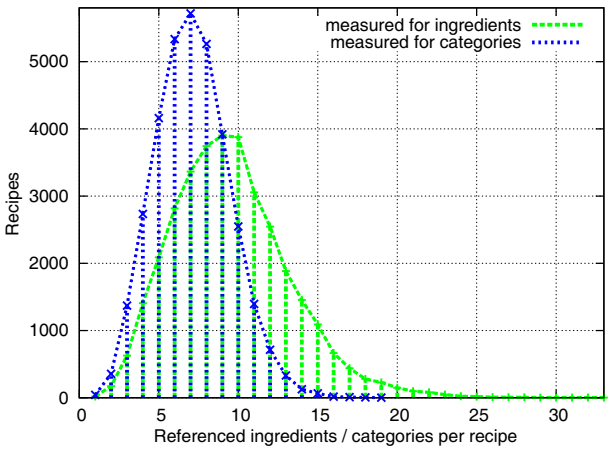

Figure 3: Number of recipes exhibiting x items/categories entries

Further, recipes cannot be characterized uniquely any more by their items as about 28% of them correspond to duplicates which reduces the initial data basis of $34{,}109$ recipes to $24{,}481$ unique pattern representations after the introduction of 45 ingredient categories.
In order to test the independence assumption for the mutual information based approach we perform a correlation analysis. It reveals relatively week correlation coefficients whose absolute values are on average 0.095. The only exception is a moderate correlation of 0.515 among the presence of flour and eggs. Later this leads to a baking cluster. Altogether, these results support the independence assumption.

5.2 Determining the number of categorical clusters

For categorical data, the number of clusters is difficult to obtain. As stated in section 3 we tried several traditional approaches like partitioning, density-based as well as hierarchical clustering methods, but all have failed. It seems that it is not possible to find a suitable method for our categorical data from the classical toolboxes. Therefore, we used the mutual information based approach described in section 4 where clustering is directed not by a geometrical distance measure. Instead, it is guided by the information content that patterns possess for each cluster. Thus, the characteristics of this measure might be a useful indicator.

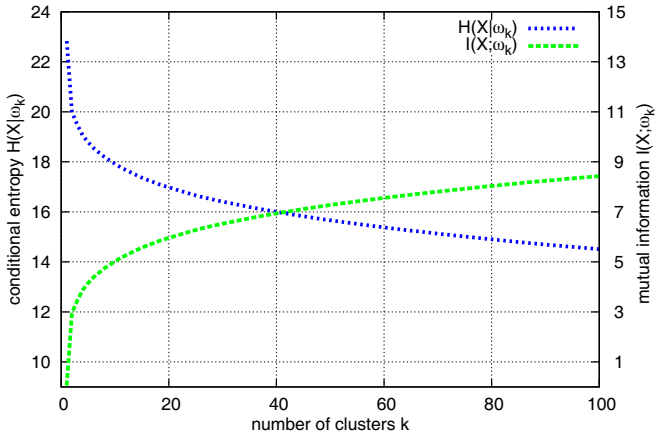

Figure 4: Conditional entropy and the mutual information of the user patterns as function of the cluster number

The conditional entropy $H(X|\omega_k)$ is the entropy of a random variable X, describing all possible attribute instances within cluster k. It decreases monotonically with increasing cluster numbers as is exhibited in Fig. 4 by the dotted line. The mutual information $I(X;\omega_k)$ between the cluster subset Ω_k and the attributes is also highlighted there. By definition it is inverse to the conditional entropy. Unfortunately,

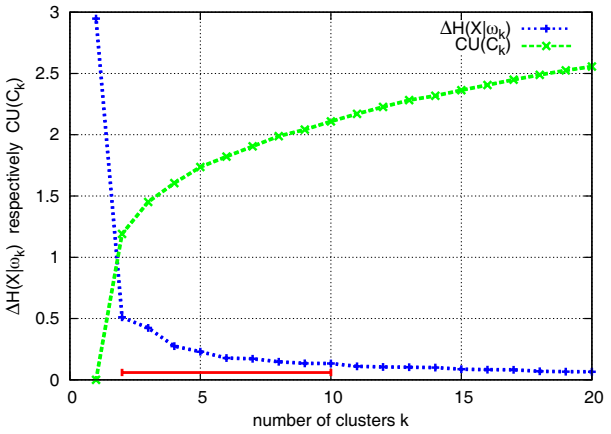

Figure 5: The conditional entropy gain and Category Utility as functions of the cluster number

no significant steps or irregularities offering the possibility to decide a proper cluster number can be found. Also the purity measure called Category Utility Function $CU(\mathcal{C}_k)$ [1] in Fig. 5. provides no deeper insights. Hence, we magnify the decline of conditional entropy defined as the decrease $\Delta H(X|\omega_k) = H(X|\omega_k) - H(X|\omega_{k+1})$. Fig. 5 shows this for the interesting part of the graph, the first 20 clusters. Although, one can see that the decrease ΔH is monotonous it becomes evident that ΔH is not constant for all merging steps. This fact is further underlined by plotting the change in the decrease, defined by $\Delta^2 H(X|\omega_k) = \Delta H(X|\omega_k) - H(X|\omega_{k+1})$ which is depicted in Fig. 6. Here, striking changes for the conditional entropy decrease can be seen for the transition from two clusters to one. The decrease is also remarkable for the transition from four clusters to three and to a smaller extend also for the merger leading from six to four clusters. There are no further significant

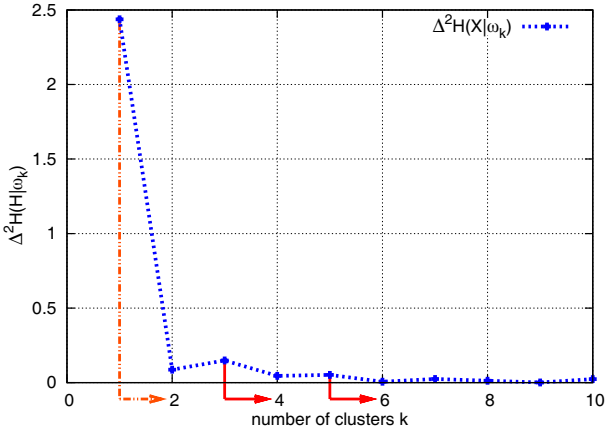

Figure 6: The change in the conditional entropy decrease

changes to be observed. The first transition from two clusters to one shall be treated with special care as it is expected to induce a high change for all data that exhibit a suitable cluster structure. In conclusion, we shall split the data into either four or six clusters, but no more. To verify these results we perform two hypothesis tests resulting in significance. The first test rules out the chance that the reported measures are obtained randomly for our data set whereas the second test's null hypothesis is the assumption that the data possesses no clustering structure at all.

5.3 Separability of the categorical clusters

Another important criterion for a good clustering is the separability of its clusters. Therefore we propose a three-stage process: First, the data shall be clustered by the algorithm to be evaluated. Second, we switch to a classification perspective where cluster assignments are treated as class labels. Therefore, a data subset is taken to train a classifier. Finally, the remaining data is used as a validation set. The classification results retrieved from all validation patterns are then compared with their cluster origins in order to assess the error probability, our indicator for separability. Further, we incorporate 10-fold cross-validation to increase significance.

	Ω_1	Ω_2	Ω_3	Ω_4	Ω_5	Ω_6	F_1
Ω_1	**4582**	14	426	1251	336	66	0.717
Ω_2	1	**5674**	72	85	14	359	0.918
Ω_3	391	80	**2993**	690	174	28	0.723
Ω_4	754	65	318	**5421**	577	121	0.681
Ω_5	339	10	86	890	**2801**	74	0.686
Ω_6	35	314	32	334	69	**4633**	0.866

Table 2: The data base statistic results for 6 clusters

According to this procedure suitable cluster numbers are supposed to lead to superior classification results for multiple classifiers. If no satisfying results can be obtained at all, this is evidence that the data exhibits no cluster structure. In order to compare our mutual information based clusterings consisting of four and six clusters, we choose three classifiers: the naïve Bayes learner [17], the decision tree algorithms C4.5 [17] and the rule learner RIPPER [6]. Altogether, these algorithms obtain all comparable classification errors although the RIPPER algorithm exhibits best performance with an error probability of about only 16%. Further, as only 120 rules are required for these results the cluster patterns of the $24{,}481$ distinguishable objects seem to possess a compact representation. The obtained kappa coefficients [11] range from 0.7 to 0.8 corresponding to a good compliance between the cluster structure and the classification labels. Table 2 depicts the confusion matrix that is computed from the RIPPER learner's classification on the partitioning into six clusters. Within each row the classifier's pattern assignments are listed. The bold numbers

	Ω_1	Ω_2	Ω_3	Ω_4	F_1
Ω_1	**8847**	104	1968	112	0.814
Ω_2	86	**5558**	135	426	0.905
Ω_3	1718	73	**9386**	279	0.805
Ω_4	58	339	377	**4643**	0.854

Table 3: The data base statistic results for 4 clusters

in the diagonal correspond to correctly classified patterns whereas all off-diagonal entries correspond to misclassifications. Although, most patterns are classified correctly one can notice several error entries beside the main diagonal exceeding the 10% level of activity. The last column within the table 1 depicts the values of the resulting F_1-Measure averaging the classifications precision and recall as defined in eq. (8). An overview of the results for the partitioning into four clusters is depicted in table 3. Although, they seem quite similar to those seen before, the F_1*-Measure* indicates higher average precision and recall values for four clusters. The results of the corresponding user weighted clusterings give a similar impression.

5.4 Stability of the categorical clusters

If the computed clusters do not depend on the utilized clustering method and parameter set it is quite probable that their cluster structure is stable, i.e. it is independent of user fluctuations. Further, the larger the data basis becomes the more importance is gained by clusterings obtained from randomly selected samples. Of course, these partitionings shall be stable for different subsampled pattern sets. Hence, stability is an interesting as well as important topic.
If it is possible to detect the same cluster number obtained from all data already by subsampling, we call this clustering *"stable"*.
For the statistical as well as the user weighted data base we choose a subsampling set size of 5000 item patterns. Then,

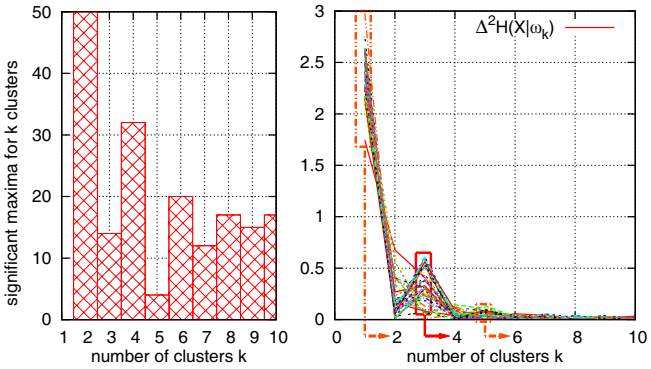

Figure 7: (a) Significant maxima for k clusters within the samples
(b) The change in decrease of conditional entropy for the 50 subsamples

we repeated our clustering procedure 50 times for both data bases. For the user weighted data the 50 resulting graphs for the change in conditional entropy's decrease (see Fig. 6) are depicted in Fig. 7(b). The statistical data results are omitted as they reveal nearly identical proportions.
An automated prediction of a suitable cluster number k depends on the detection of significant changes in entropy decrease. Therefore, we introduce the concept of a *significant maximum* which requires a function value to exceed α times its neighbors' values

$$\textit{significant maximum} \quad \Leftrightarrow \quad f(k) \geq \alpha f(k-1) \ \wedge \ f(k) \geq \alpha f(k+1) \qquad (10)$$

For our data, we obtained good results for $\alpha = 1.08$ although the prediction of each cluster number k for each subsample might include multiple guesses. An overview of the prediction frequencies for each cluster number $k \in \{2, ..., 10\}$ is shown in Fig. 7(a). We notice that as expected all 50 sets have a significant maximum for the change in entropy decrease at $k = 2$ clusters. Predictions for larger cluster numbers are less frequent. The only further peaks can be found for four and six clusters. Consequently, the predictions of two, four as well as six clusters are the most stable and most probable ones. Additionally, Fig. 7(b) depicts these results also by a superposition of the changes in entropy decrease for all 50 subsamples.
In a second step we further investigated how the object assignments for the subsample clusters comply with the clustering obtained from the complete data base. Here, the four cluster structure exhibits the best results although one subsample cluster's composition seems rather ambiguous. Thus, the data seems to possess a significant cluster structure without being too arbitrary.

5.5 Characteristics of the categorical clusters

A characterization of the individual recipe pattern clusters can be achieved via statistics for frequent occurrences of items (ingredients) or item combinations. E.g. within the four cluster structure one partition could be identified as a baking cluster as the combination of flour, eggs, butter and milk is usual for its recipes. However, the individual items' information content is very different.
One way to show this is to rank each item by its contribution to the mutual information of a pattern's cluster assignment and its presence within the pattern. Thus, we can compute for each ingredient the importance of its affiliation to a recipe for the purpose of assigning this recipe's pattern to a cluster. This kind of ranking is only valid for one item. But if the first one is selected, a ranking for the remaining items can be computed under the condition that we add a second item to a list possessing highest mutual information where

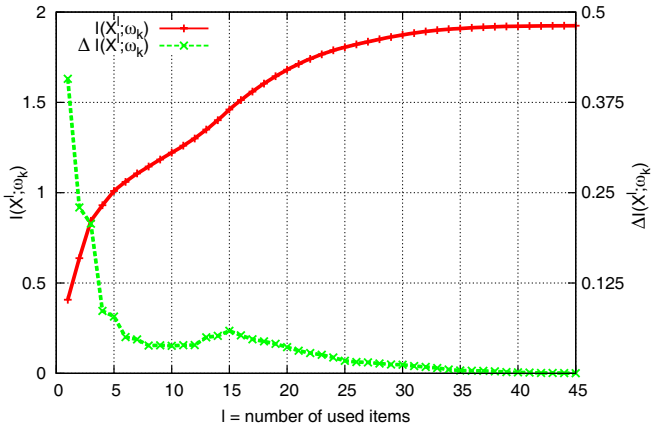

Figure 8: The solid line shows the increase in mutual information when adding a new item. The dotted line shows the decreases of the contribution for each additional item

the first is already fixed. After choosing the second item, we might select, under the condition of two fixed items, the third one and so on. This greedy procedure, known as *forward feature selection*, computes mutual information for a fixed clustering as a function of the item (attribute) number l used for clustering $I(X^l; \omega_k)$. Fig. 8 depicts this function for the case of four clusters. Certainly, the contribution of each additional item decreases with the number of already selected items. This is highlighted in Fig. 8 by the dotted line; the corresponding units can be found at the figure's right hand side. In contrast to the sum of all mutual information contributions, we notice that the first five items have most influence on mutual information.

6 Discussion

For all kinds of customer-oriented business intelligence approaches user profiles are an important topic to be considered. This contribution outlines how web based user profiling might be established upon clustering of categorical user data. It turns out that classical, metrical clustering algorithms are inappropriate because of the special nature of categorical item sets. Therefore, we suggest a new clustering approach based on probabilities, i.e. the mutual information between item patterns and the cluster candidates they are affiliated to.
The utilization of information as clustering criterion is not new. E.g. the COOLCAT algorithm [3] greedily fuses clusters based only on conditional entropy. This yields to results comparable to those of Hamming Distance based clustering which is inappropriate for categorical item sets. Since the deviations at the beginning of the sequential clustering determine the final results, we did not use this approach. Another algorithm based on an information motivated criterion

is LIMBO (scaLable InforMation Bottleneck) [1]. It incorporates mutual information to calculate information loss for each cluster fusion. For all patterns, the probability is computed separately using the number of present items and assuming that they are equally probable. For clusters possessing only one pattern this procedure results in the inconsistent case of non-zero entropies. Thus, we did not continue this approach either. Finally, after performing this work, we became aware of another approach quite similar to ours. The HierEntro algorithm for classical category information [5] uses the entropy of each item x_i and normalizes it via a division by $\log_2(m_i)$ where m_i describes the number of all possible values for x_i. Thus, the influence of items possessing many possible values is reduced compared to those with only few ones. As clustering criterion this approach considers the increase of average conditional entropy claiming that small entropies indicate a homogeneous clustering.

In conclusion, there are many approaches for building user profiles. But, for categorical user patterns the choice of useful methods diminishes heavily. For the special case of patterns consisting of binary item sets, metrical methods based on Euclidean or Hamming distance shall not be used. Instead, the mutual information guided approach obtains meaningful clusters. Further, it allows not only to identify user defined similarities and preferences, but also to differentiate among several mainstreams by indicate a favorable cluster number. Now, it is essential to verify the value of these clusterings for real-life profiling applications. The first results are promising. We already succeeded to predict future user actions based on assignments to the cluster whose contents have been visited most frequently quite well. But these results constitute only a starting point and require further sophistication. Here, one possibility might be the incorporation of a model for cluster transitions. Finally, a wider data basis is required to obtain significant statement concerning the clusters value for user profiles.

Acknowledgment:
We thankfully acknowledge the support of the *E-Finance Lab*, Frankfurt for this work.

References

[1] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevcik. *LIMBO: Scalable Clustering of Categorical Data*. 9th Int. Conf. on Extending Database Technology, 2004.

[2] P. F. Baldi, P. Frasconi, and P. Smyth. *Modeling the Internet and the Web Probabilistic Methods and Algorithms*. Wiley, New York, USA, 2003.

[3] D. Barbar, J. Couto, and yi Li. COOLCAT: An entropy-based algorithm for categorical clustering. In *IKM'02*, pages 582–589, USA, 2002. IEEE Press.

[4] R. Brause, T. Langsdorf, and M. Hepp. Neural Data Mining for Credit Card Fraud Detection. In *ICTAI-99*, pages 103–106. IEEE Press, 1999.

[5] K. Chen and L. Liu. Towards Finding Optimal Partitions of Categorical Datasets. In *Technical Report*, Georgia Institute of Technology, College of Computing, 2003.

[6] W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *JAIR'99*, volume 16, pages 335–342, USA, 1999.

[7] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley, New York, USA, 1991.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second Edition*. Wiley, New York, USA, 2000.

[9] B. Ende. Adaptive Profilbildung auf Internetportalen. Diploma thesis, J.W.Goethe-University Frankfurt, Department of Computer Science and Mathematics, Germany, 2005.

[10] M. Ester and J. Sander. *Knowledge Discovery in Databases*. Springer, Heidelberg, Germany, 2000.

[11] B. D. Eugenio and M. Glass. The Kappa statistic: a second look. In *Computational Linguistics*, volume 30(1), 2004.

[12] M. R. Garey and D. S. Johnson. Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman, 1979.

[13] S. Guha, R. Rastogi, and K. Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Information Systems*, volume 25(5), pages 345–366, USA, 2000.

[14] S. Gunduz and M. T. Ozsu. A Web Page Prediction Model Based on ClickStream Tree. In *Proc. of the ACM SIGKDD*, volume 9, pages 535–540, USA, 2003.

[15] M. Hansen and B. Yu. Model selection and the principle of Minimum Description Length. In *Journal of the American Statistical Association*, volume 96 (454), pages 746–774, 2001.

[16] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In *ACM Int. Conf. Proc. Series*, 2004.

[17] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[18] B. Mobasher, R. Cooley, and J. Srivastava. Automatic Personalization Based on Web Usage Mining. In *CACM*, volume 43(8), pages 142–151, 2000.

[19] J. Paetz. Intersection Based Generalization Rules for the Analysis of Symbolic Septic Shock Patient Data. In *ICDM'02*, pages 673–676, Japan, 2002.

[20] R. Sarukkai. Link prediction and path analysis using Markov Chains. In *Computer Networks*, volume 33, pages 377–386, USA, 2000.

[21] B. van Eimeren, H. Gerhard, and B. Frees. Mining Navigation History for Recommendation. In *IUI*, pages 106–112, USA, 2000.

[22] B. Zhang and S. Srihari. Discovery of the Tri-Edge Inequality with Binary Vector Dissimilarity Measures. In *Proc. of the ICPR*, volume 4(23-26), pages 669–672, 2004.