Towards an Automatic Construction of Contextual Attribute-Value Taxonomies

Dino Ienco, Yoann Pitarch, Pascal Poncelet, Maguelonne Teisseire

Towards an Automatic Construction of Contextual Attribute-Value Taxonomies

Dino Ienco
CEMAGREF, Montpellier, France
LIRMM, Montpellier, France
dino.ienco@teledetection.fr

Yoann Pitarch
LIRMM, Montpellier, France
pitarch@lirmm.fr

Pascal Poncelet
LIRMM, Montpellier, France
poncelet@lirmm.fr

Maguelonne Teisseire
CEMAGREF, Montpellier, France
LIRMM, Montpellier, France
maguelonne.teisseire@teledetection.fr

ABSTRACT

In many domains (e.g., data mining, data management, data warehouse), a hierarchical organization of attribute values can help the data analysis process. Nevertheless, such hierarchical knowledge does not always available or even may be inadequate or useless when exists. Starting from this consideration, in this paper we tackle the problem of the automatic definition of data-driven taxonomies.To do this we combine techniques coming from information theory and clustering to obtain a structured representation of the attribute values: the Contextual Attribute-Value Taxonomy (CAVT). The two main advantages of our method are to be fully unsupervised (i.e., without any knowledge provided by an expert) and parameter-free. We experiments the benefit of use CAVTs in the two following tasks: (i) the multilevel multidimensional sequential pattern mining problem in which hierarchies are involved to exploit abstraction over the data, (ii) the table summarization problem, in which the hierarchies are used to aggregate the data to supply a sketch of the original information to the user. To validate our approach we use real world datasets in which we obtain appreciable results regarding both quantitative and qualitative evaluation.

Categories and Subject Descriptors

H.2 [**Database Management**]: Knowledge Management Applications—*data exploration and discovery*

General Terms

Algorithms, Data Management

Keywords

Contextual Taxonomies, Clustering, Data Summarization

1. INTRODUCTION

The hierarchical organization of values provided by Attribute Value Taxonomies (AVTs) are widely used in many domains such as biological science [2], e-commerce [9], semantic web [14] and they are also used to support different types of data analysis like: anonymization [13], OLAP exploration [11] and summarization [3]. For instance, data summarization can be very helpful to a marketing director to have a quick overview of a large sale table (e.g., for decision making, it could be much more interesting to know sales in Europe than sales at a lower granularity such as cities). Usually this kind of hierarchies need to be specified by a domain expert. Unfortunately this specification phase could be very time consuming and requires huge human resources (which are not always available), especially if the number of attributes of a dataset is very high. These considerations motivate the development of methods to infer these hierarchies in a completely unsupervised way (*i.e.*, without any knowledge supplied by the expert). Consequently, they can fill the information gap when there are no available taxonomies or when existing taxonomies are too general and thus lead to poor-quality data analysis. In some previous works [18, 5] the authors build contextual attribute hierarchies to improve, for instance, the performance of the Naive Bayes classifier and the Bayesian network. These approaches are classification oriented which are thus human resource consuming and they build contextual attribute hierarchies related to the class attribute. Instead in our proposal we focus on a fully unsupervised approach without requiring any class information over the data. To the best of our knowledge, very few approaches have tackled this problem despite the numerous applications that could benefit from an automatic definition of CAVTs. Considering our work, the main contributions are:

- We introduce a method to automatically build contextual attribute hierarchies that are useful in many domain of the data analysis
- We do not require the class information to guide the

process

- We supply a parameter free method that avoids human-expensive activity related to the manual construction

To validate our approach we use two different approaches: the first one is the multi-level multidimensional sequence pattern mining problem[12]. This method uses attribute hierarchies to extract sequence patterns from the data. The second one is based on the privacy-preserving approach presented in [13]. We adapt it to deal with the data summarization problem. The remainder of this paper is organized as follows. We first give a motivating example in Section 2. In Section 3 we describe our approach. The experimentations are presented in Section 4 and the conclusions are drawn in Section 5.

2. MOTIVATING EXAMPLE

To explain why deriving contextual attribute hierarchies could be useful let us consider the toy dataset in Table 1. In this dataset, we consider information about persons which are represented using three different attributes: *City*, *Main Sport* and *Type of cooking*. We can assume that each attribute ranges over a discrete domain. For the attribute *City* we use the following values {Turin (Italy), Porto (Portugal), Miami (USA), Denver (USA)}. The attribute *Main Sport* describes the principal sport of a person that lives in a particular city, it ranges over values: {ski, surf, snowboard, beach volley}. The last attribute is the *Type of cooking*, it explains which kind of cooking is usual for a particular person. Allowed values are: {meat, fish}. The problem to solve is the summarization of the dataset using Attribute Value Taxonomies [3]. The goal is to produce a good and informative summary of the data for the user using the metadata supplied by the taxonomies. Let us use the classical geographical taxonomy for the *City* attribute (see Figure 1(a)) and a taxonomy for the *Main Sport* attribute (see Figure 1(b)). Looking at the information represented in Table 1, we observe that the geographical information is not so adequate to the context induced by the data. Nevertheless, using the available taxonomies we obtain the summarization represented in Table 2. The summary is composed by the same attributes of the original table (with a generalization step involved) and an extra attribute *Count* that represents the count of how many distinct tuples are involved in the generalization. We represent a generalization of two or more values of an attribute without a specific label but with the set of all the values (*i.e.,* using the taxonomy in Figure 1(a) the generalization of the values *Denver* and *Miami* will be {*Denver*,*Miami*}). It is out of the scope of this work to investigate how to label each node of the hierarchy[1].

An in-depth analysis of Table 1 shows that the habits of people living in *Turin* and *Denver* are very close and also the habits of the people living in *Porto* and *Miami* are similar between them. This behavior is influenced by the fact that *Turin* and *Denver* are two mountain towns while *Porto* and *Miami* are two sea towns. By using this information we build an alternative (contextual) taxonomy, represented in Figure 2. Using this new taxonomy and the taxonomy over *Main Sport*, we summarize the original Table 1 in Table 3. We observe that by taking into account the contextual taxonomy

[1]Interested reader may consider the approaches presented in [4] or [15] as useful references.

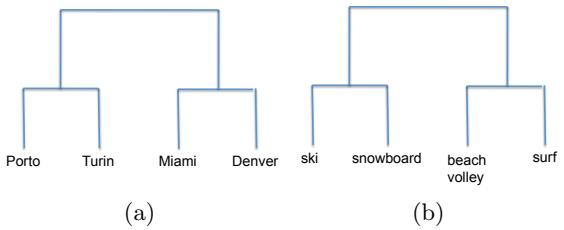

Figure 1: Available Taxonomies over the attributes (a) City and (b) Main Sport

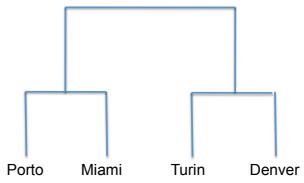

Figure 2: A Contextual Taxonomy over the attribute City

City	*Main Sport*	*Type of cooking*
Denver	ski	meat
Porto	surf	fish
Miami	beach volley	fish
Porto	surf	fish
Porto	surf	meat
Denver	ski	meat
Porto	surf	fish
Turin	snowboard	meat
Miami	surf	fish
Denver	snowboard	meat
Denver	snowboard	meat
Miami	surf	meat
Porto	surf	fish
Turin	ski	meat
Porto	beach volley	fish
Miami	surf	fish
Denver	ski	fish
Turin	ski	meat

Table 1: Toy Example Dataset

City	*Main Sport*	*Type of cooking*	*Count*
{Denver, Miami}	{ski, snowboard}	{meat}	4
{Denver, Miami}	{ski, snowboard}	{fish}	1
{Denver, Miami}	{surf, beach volley}	{meat}	1
{Denver, Miami}	{surf, beach volley}	{fish}	3
{Porto, Turin}	{ski, snowboard}	{meat}	3
{Porto, Turin}	{surf, beach volley}	{meat}	1
{Porto, Turin}	{surf, beach volley}	{fish}	5

Table 2: Summarization using the original Taxonomies

City	*Main Sport*	*Type of cooking*	*Count*
{Denver, Turin}	{ski, snowboard}	{meat}	8
{Denver, Turin}	{ski, snowboard}	{fish}	1
{Porto, Miami}	{surf, beach volley}	{meat}	2
{Porto, Miami}	{ski, beach volley}	{fish}	9

Table 3: Summarization using the contextual Taxonomy for the attribute *City* and the original Taxonomy for the attribute *Main Sport*

we obtain a more compressed and still informative table. In particular we observe that, intuitively, this summarization captures the knowledge behind the data, and the context dependent information coming from them.

3. HOW TO EXTRACT CONTEXTUAL ATTRIBUTE VALUE TAXONOMIES?

Starting from a dataset, our goal is to produce a taxonomical organization for each attribute. To obtain such taxonomical organization (a hierarchy) we employ Hierarchical Clustering. This algorithm uses the distances between each pair of objects as input to produce the tree structure. In our case we need to define the distances between each pair of values of the same attribute. Frequently the involved attributes are categorical (they range over a finite set of values) and defining distances between the values of these attributes is not trivial. For this reason we employ a recent data mining techniques [7] that is able to produce, for each attribute, an intra-attribute distance matrix exploiting the interaction among attributes. Each matrix represents the dissimilarity between each pair of values of an attribute. Finally, our proposed approach is a two steps methodology where first the intra-attribute distance matrices are computed and then a hierarchical clustering algorithm is applied over each matrix to automatically derive a taxonomy for each attribute.

3.1 Intra-Attribute Distance Computation

Here we briefly recall *DILCA* (DIstance Learning for Categorical Attributes), a framework for computing distances between any pair of values of a categorical attribute [7]. The *DILCA* method is based on information theory, in particularly it uses the Symmetrical Uncertainty [17] that it is a normalized version of the mutual information. This measure allows to evaluate the interaction between the attributes. In particular, given a target attribute, this method is based on a two steps strategy: (i) Selection of the context (ii) Using the context to compute the intra-attribute distances. The process is iterated for all the attributes of the dataset. The result is a set of distance matrices (one for each attribute) where each matrix contains the value-value distance for a specific attribute.

3.1.1 Selection of the context

Given a target attribute Y (over which we want to obtain the value-value distance matrix), we select a set of other attributes, called $context(Y)$, related to Y to use as context to compute the distances. In [7] a fully automatic way to compute this context is introduced. In particular the target attribute is used as class attribute for a supervised feature selection task. In this way the context of an attribute is defined as the set of features that are relevant and not redundant for the prediction of Y. The process is performed for all the attributes in the dataset to determine the relative context. Considering the example in Section 2, the context of attribute *City*, $Context(City)$, is equal to: $\{Main\ Sport\}$. In this case only this attribute is selected as a context because the attribute *Type of cooking* was considered redundant. Consequently the intra-attribute distance of *City* will be influenced only by the attributes in the context.

3.1.2 Using the context to compute the intra-attribute distances

Given the context of an attribute $context(Y)$, to compute the distance between any pair of values of the attribute, the following formula is used:

$$d(y_i, y_j) = \sqrt{\frac{\sum_{X \in Context(Y)} \sum_{x_k \in X} (P(y_i|x_k) - P(y_j|x_k))^2}{\sum_{X \in Context(Y)} |X|}}$$

where $P(y_j|x_k)$ is the conditional probability that the value y_j of the attribute Y appears in the same tuple in which the value x_k of the attribute $X \in Context(Y)$ appears. $|X|$ indicates the cardinality of the attribute X. Always considering the toy example in Section 2 we obtain intra-attribute distances in Table 4. We note that these distances represent the contextual taxonomy in Figure 2 in which at the first agglomeration level *Denver* is aggregated with *Turin* and *Porto* is aggregated with *Miami*.

Value	Value	dist.	Value	Value	dist.
Denver	Porto	0.60	**Porto**	**Miami**	**0.13**
Denver	Miami	0.55	Porto	Turin	0.47
Denver	**Turin**	**0.19**	Miami	Turin	0.41

Table 4: Intra-Attribute distances for the *City* attribute

3.2 Taxonomies Building

To extract and generate contextual Attribute-Value taxonomies, we use the results of DILCA, i.e. a distance matrix for each attribute of the database containing the value-value distances. Then, starting from a distance matrix of an attribute, we use the Ward Hierarchical clustering [1] to obtain a hierarchy/taxonomy of it.

Algorithm 1 $WARD(distM)$

```
1: for all i = 1 to |distM| do
2:     C_1i = {i}
3: end for
4: C1 = {C_11, ..., C_1n}
5: i = 1
6: while |C_i > 1| do
7:     for j = 1 to |C_i| do
8:         for k = j + 1 to |C_i| do
9:             d[j, k] = distM[C_ij, C_ik]
10:        end for
11:    end for
12:    (s,r) = argmin(d[j, k])
13:    for j = 1 to |C_i| do
14:        if j ≠ r and j ≠ s then
15:            C_{i+1,j} = C_{i,j}
16:        else if j = r then
17:            C_{i+1,j} = C_ir ∪ C_is
18:        end if
19:    end for
20: end while
```

The Ward method, presented in Algorithm 1 is a greedy, agglomerative hierarchical method, that determines a diagram - the dendrogram - that records the sequence of fusions of clusters into larger clusters. This is an iterative approach. It takes in input a matrix containing the distances (`distM`) between any pair of elements. At the beginning it creates as many initial clusters as the number of objects in the distance matrix (line 1 to 3). In our case the objects are the different values of the same attribute. The process is guided by an objective function that measures the cohesion inside the generated groups (line 12). The objective function is based on the computation of an overall measure of goodness

of each candidate clustering solution. At each step i, the two clusters that increase the overall global measure are merged (from line 13 to line 19). In the process, all the candidate solutions are computed but the best one is determined by the minimum increase in the objective function (line 12). At the end of the process a hierarchy of the original values is then obtained. Using this procedure we are able to obtain, in a fully unsupervised way, a hierarchical organization of the values of an attribute. We perform the hierarchical clustering for each attribute of the dataset.

4. EXPERIMENTAL STUDIES

In this section we validate our approach using two distinct and complementary ways. Firstly, we exploit the CAVTs into two different tasks: the sequential pattern mining problem, for which we employ the M3SP method [12] and the data summarization task in which we adapt the algorithm presented in [13] to perform this analysis. In both tasks we compare the results obtained using the original hierarchies with the ones obtained using CAVTs. Secondly, in collaboration with data experts, we analyzed the quality of the extracted CAVTs. Beforehand, we start this section with a description of the real datasets used for the experimentations.

4.1 Description of the Datasets

The *Mali* dataset represents 980 farmers in the Mali state. For each farm, the dataset stores both time series information and static information. Time series information represents typical measurement for agricultural surveillance over cultivated area. In this particular dataset five different indicators are considered. Each time series is built over 11 different timestamps obtained by monitoring the different zones for one year. For the static information associated to each farm, 5 attributes are used (soil type, distance to the village, distance to the river, rainfall, ethnic group). The CAVTs are derived over the 5 static attributes.

The *Adult* dataset comes from a publicly available repository[2]. It is based on census data and has been widely used to evaluate classification and k-anonymization algorithms. Here, the same settings as in [8] are used. We obtain a dataset of 8 attributes. The attribute *age* is discretized in 8 equal-size bins. Starting from this dataset we sample 10% of the original dataset obtaining 4 522 tuples. As original hierarchies we use the taxonomies supplied in [8]. In this dataset the CAVTs are derived over all the 8 attributes.

4.2 CAVT for the Sequential Pattern Mining

Hierarchies are more and more considered in frequent pattern mining techniques since they allow to capture the general trends over a given dataset at different levels of granularities. In particular, recent works (e.g., [12],[10]) focused on exploiting hierarchies to enhance the multidimensional sequential pattern mining problem. Here, it should be noticed that we restrict the definition of sequential patterns since we considering sequences of (multidimensional) items only as input data. More precisely, manipulated sequences are on the form $<(e_1)...(e_l)>$ where e_i represents a multidimensional item at time i. The pattern mining algorithms extract multidimensional sequences that each represents a subset of the dataset. The minimum size of this subset is given by a numerical user-defined threshold: the *minimum support* value, *minSupp*. The benefit of taking attribute hierarchies into account is to make possible the extraction of longer and more descriptive sequences. For instance, given the hierarchies defined in Figure 1, the item $(Porto,\ Surf)$[3] may not be considered as frequent and will be thus disregarded whereas the item (**Europe**, $Surf$) could be frequent since it represents all the european cities where surf is played. Here, we consider the benefits of using CAVTs in order to extract such kinds of sequences. To extract multidimensional multilevel frequent sequences we use the M3SP algorithm [12]. This algorithm is able to take into account the hierarchical knowledge. In our experiments we compare two different ways to obtain such knowledge. The first one is supplied by our method and is denoted M3SP$_{CAVT}$. The second one incorporates hierarchies supplied by an expert and is denoted M3SP$_{EXP}$. In the experiment we ran M3SP with a support threshold varying between 50% to 90% by step 10%. To evaluate the obtained results we used two indicators. The first employed measure is the number of extracted patterns. This is equivalent to the size of the result. The results about this experiment are reported in Table 5. This is a good point because we can observe that M3SP$_{CAVT}$ has a distribution, regarding the size of the results, very similar to the one obtained by M3SP$_{EXP}$. This underlines the fact that our method supplies hierarchies that, for this experiment, are a good replacement w.r.t. hierarchies given by an expert.

	# Sequences	
MinSupp	**M3SP$_{EXP}$**	**M3SP$_{CAVT}$**
50%	105	183
60%	46	75
70%	28	27
80%	12	18
90%	12	10

Table 5: Number of patterns extracted from M3SP with different hierarchical knowledge

The aptitude, of a data mining algorithm, to only extract meaningful and concise results from the data is always an important issue [16]. To do this, the second measure is the percentage of the sequential patterns with more than one item over the whole extracted pattern. We consider this quantity because it represents the true sequential information in the data. At the same time, with this analysis, we measure the ability of the used hierarchies to capture the hidden sequential behavior inside the data. The findings about this experimentation are reported in Table 6. Comparing M3SP$_{CAVT}$ with M3SP$_{EXP}$, we can note that our proposal facilitates the process to mine sequences with more than one element. This is an appreciable result for this problem because the goal is to extract hiding information regarding the sequential nature of the data. This phenomena is particularly evident for high values of the support threshold. These results underline that the extracted CAVTs help the mining process performed by M3SP at least as well as the original taxonomies.

4.3 CAVT for Data Summarization

In this subsection we analyze the results obtained by $CAVTs$ for the data summarization task. The goal of this task, as reported in [3], is to find a compressed representation of

[2]http://archive.ics.uci.edu/ml/

[3]This item designates the fact that surf is played at Porto.

MinSupp	M3SP$_{EXP}$	M3SP$_{CAVT}$
50%	3.8%	5.46%
60%	2.17%	5.33%
70%	0%	0%
80%	0%	27.77%
90%	0%	20.0%

Table 6: Percentage of sequences extracted from M3SP with different hierarchical knowledge

	$MinGen_{CAVT}$		$MinGen_{EXP}$	
K	**Inf. Loss** (x10^3)	**# tuples**	**Inf. Loss** (x10^3)	**# tuples**
10	**28.78**	67	30.44	55
20	**35.21**	33	718.37	37
30	**35.21**	33	35.36	22
40	37.798	18	**35.36**	22
50	**34.66**	20	35.36	22
60	**34.87**	14	35.36	22
70	**34.87**	14	723.29	16
80	**34.87**	14	723.29	16
90	**37.25**	10	723.29	16
100	**37.25**	10	723.29	16

Table 7: Information Loss and Number of Tuples in the compressed table for $MinGen_{CAVT}$ and $MinGen_{EXP}$

the original table using generalization supplied by the attribute value taxonomies. The only constraint is that each tuple of the compressed table needs to represent at least k tuples of the original table. Actually this task shares similarities to the k-anonymization problem, in which each tuple of the anonymized table needs to represent at least k tuples of the original table. For this reason we used the algorithm $MinGen$ presented in [13] and originally developed for the k-anonymity problem. We couple this algorithm with both original available taxonomies ($MinGen_{EXP}$) and with our CAVTs automatically extracted ($MinGen_{CAVT}$). The algorithm needs two parameters. The first parameter k is the minimum number of tuples indistinguishable over the set of sensitive attributes (quasi-identifiers) [13]. The quasi-identifiers are the attributes for which the generalization process is performed (and the attribute taxonomies is demanded). In our context we consider all the attributes of the input table as sensible attributes. We range the k parameter from 10 to 100 at step of 10. The second parameter is the suppression coefficient sc standing for the maximum number of tuples (in percentage w.r.t. the size of the table) that the algorithm suppresses to obtain a k-anonymized table. This parameter is not common for the table summarization task, but it is a reasonable assumption that the final summary represents the majority of the data and not the whole table. For all this reason we set the suppression coefficient equals to 1%. To evaluate the quality of the compressed table we adopt the non-uniform entropy measure [6] to quantify the Information Loss. This measure assumes that the different values of an attribute do not have a uniform distribution. As suggested by the author, this is a more careful way to measure the Information Loss. The Information Loss ranges from 0 to infinity. The idea is that a good k-anonymization (for this reason also a good summary of an original table) is the one that minimizes the Information Loss respecting the given constraints k and sc.

In Table 7 we show the findings of our analysis. For each type of taxonomical knowledge we report the Information Loss as well as the number of distinct tuples in the k-anonymized (compressed) table. We can observe that $MinGen_{CAVT}$ obtains interesting results w.r.t. the original taxonomies. For instance we notice that for values of k equal to 20 and greater than 70 the Information Loss of $MinGen_{CAVT}$ is 20 times lesser than the Information Loss of $MinGen_{EXP}$. Even when our method obtains lower Information Loss compared to the original taxonomies (e.g., k equals to 40) the difference remains very small. About the number of different tuples obtained in the compressed table, we note that the number of tuples in $MinGen_{CAVT}$ is even quite lower than $MinGen_{EXP}$. In particular $MinGen_{CAVT}$ obtains less tuples for values of k greater or equal to 40. Interestingly we note that the Information Loss is not directly proportional to the number of tuples in the compressed table. For instance for $k = 30$, $MinGen_{CAVT}$ produces a compressed table of 33 tuples (22 tuples for $MinGen_{EXP}$), but it obtains a better value of Information Loss. As a conclusion of our analysis, over this dataset, we can state that with the produced $CAVTs$ we obtain results at least as well as to the original taxonomies. This means that our method is able to extract the taxonomical information of an attribute using the whole dataset. This could be useful when: (i) the original taxonomies are too general (*i.e.*, the dataset represents specific information that the original taxonomies do not take into account), (ii) there are no available taxonomies associated to the data.

4.4 Qualitative evaluation

In this subsection we describe some of the results about the extracted $CAVTs$ from the MALI dataset, from a qualitative point of view. To obtain hierarchies generated by the expert we asked to scientist in the field of Agricultural and Remote Sensing Science. The collaboration with scientist in the domain area allows to compare our $CAVTs$ with real hierarchies. We analyzed the hierarchies to understand if they can help the analyst to extract some new knowledge from the data or if the analyst is able to recognize some interesting informations within the $CAVTs$. An example of portion of the extracted hierarchy is reported in Figure 3. For visualization purpose, we retain only the first two levels of the hierarchy. When we supply this representation to the analyst, she has clearly recognized information about the correlation of culture type. For instance, she has recognized that the group (a) is correct (*i.e.,* *sorgho* and *mil* are very similar plants). This means that they live in similar environment. *cotton* and *corn* co-occur in the same group (b) because they are usually cultivated together. The analyst also recognizes that in group (c) all the cultivation are related to the human nutrition. Another interesting result is obtained over the attribute describing the type of land. The whole contextual taxonomy is depicted in Figure 4. This attribute can assume six different values *EC* (Drainage land), *GR* (Gravelly land), *GR_su* (Superficial Gravelly land), *SU* (Superficial Land), *LIAR* (Silty clay) and *LISA* (Silty sands). When we supplying this taxonomy, the expert was initially surprised because in spite the taxonomy is different from the classical taxonomy presented in the literature, it is always very informative regarding the analyzed datset. This means that when we contextualize this hierarchy for the MALI dataset (rural landscapes and agricultural study) this taxonomy captures the intrinsic relationship within the data. For instance, from their structure point of view, the *EC* and *GR* types are very different, but they were automatically grouped together be-

cause, in this context, we cannot find cultivation but only grass over there. Another automatically detected group is the one composed by *GR_su* and *SU*. Both type are characteristic of superficial lands and are well adapted to peanuts and rice cultivation. This example emphasizes the usefulness of the contextual attribute-value taxonomies because they are able to structure the values of an attribute taking into account the context in which this information is used. We obtained similar findings for all the other hierarchies that we extract with our approach. It confirms that we are able to extract interesting and meaningful hierarchies in a fully automatic and unsupervised way.

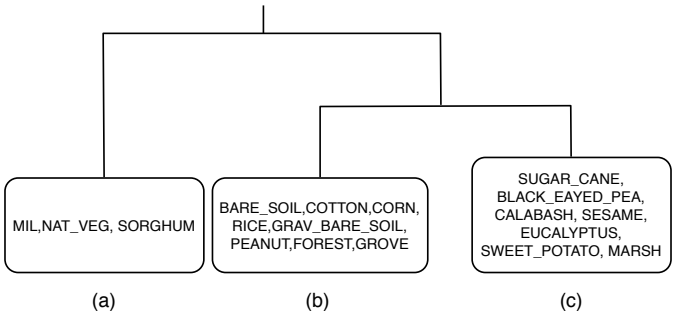

Figure 3: Two levels of the $CAVT$ over the type of cultivations

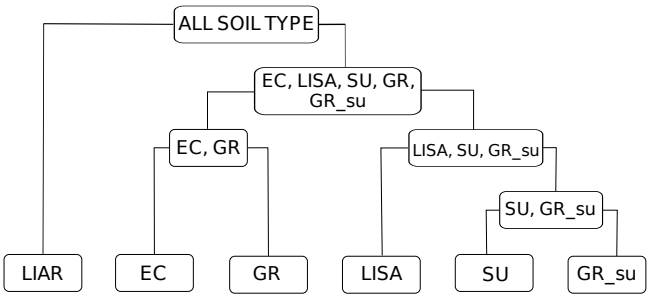

Figure 4: CAVT for the type of lands

5. CONCLUSION

This paper presents a new technique to automatically derive Contextual Attribute-Value Taxonomy (CAVT) from data. Through our approach we are able to first extract existing relationships from categorical data and then automatically build taxonomies for each attribute. The approach was evaluated using a multilevel, multidimensional sequential pattern mining method and a k-anonymity algorithm to summarize a relational table. The experimentations compared with expert taxonomies underline the good quality of the obtained CAVTs in both tasks. In this way we underlined that our approach is not biased by the involved algorithm. We also performed an in-depth analysis over the obtained taxonomies showing that our approach is able to extract taxonomies that are related to the hidden knowledge contained in the data. As future work, we plan to investigate the usefulness of CAVTs with other approaches and we investigate different way to build hierarchies developing new hierarchical clustering techniques driven by the characteristics of our method.

6. REFERENCES

[1] M. R. Anderberg. Cluster analysis for applications, 1973.

[2] M. Ashburner and al. Gene ontology: tool for the unification of biology. *The Gene Ontology Consortium. Nat Genet.*, 25(1):25–29, 2000.

[3] K. S. Candan, M. Cataldi, and M. L. Sapino. Reducing metadata complexity for faster table summarization. In *EDBT*, pages 240–251, 2010.

[4] D. Carmel, H. Roitman, and N. Zwerdling. Enhancing cluster labeling using wikipedia. In *SIGIR*, pages 139–146, 2009.

[5] M. desJardins, P. Rathod, and L. Getoor. Bayesian network learning with abstraction hierarchies and context-specific independence. In *ECML*, pages 485–496, 2005.

[6] A. Gionis and T. Tassa. K-anonymization with minimal loss of information. *IEEE Trans. Knowl. Data Eng.*, 21(2):206–219, 2009.

[7] D. Ienco, R. G. Pensa, and R. Meo. From context to distance: Learning dissimilarity for categorical data clustering. *TKDD*, to appear, 2012.

[8] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, pages 279–288, 2002.

[9] R. Kohavi and F. Provost. Applications of data mining to electronic commerce. *Data Min. Knowl. Discov.*, 5(1/2):5–10, 2001.

[10] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *CIKM*, pages 81–88, 2001.

[11] Y. Pitarch, C. Favre, A. Laurent, and P. Poncelet. Context-aware generalization for cube measures. In *DOLAP*, pages 99–104, 2010.

[12] M. Plantevit, A. Laurent, D. Laurent, M. Teisseire, and Y. W. Choong. Mining multidimensional and multilevel sequential patterns. *TKDD*, 4(1), 2010.

[13] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[14] J. H. T. Berners-Lee and O. Lassila. The semantic web. *Scientific American*, 2001.

[15] P. Treeratpituk and J. P. Callan. An experimental study on automatically labeling hierarchical clusters using statistical features. In *SIGIR*, pages 707–708, 2006.

[16] J. Vreeken, M. van Leeuwen, and A. Siebes. Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.*, 23(1):169–214, 2011.

[17] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, pages 856–863, 2003.

[18] J. Zhang and V. Honavar. Avt-nbl: An algorithm for learning compact and accurate naïve bayes classifiers from attribute value taxonomies and data. In *ICDM*, pages 289–296, 2004.