USING A CLUSTERING ALGORITHM FOR DOMAIN RELATED ONTOLOGY CONSTRUCTION

Hongyan Yi and V. J. Rayward-Smith
School of Computing Sciences, University of East Anglia, Norwich, U.K.

Keywords: Ontology construction, Clustering, Taxonomy.

Abstract: Fisher's clustering algorithm is exploited to build a cluster hierarchy. Then this methodology is used to automatically generate the taxonomies of the nominal attribute values for a real world database. An ontology for a specific analysis task is finally constructed, which reflects some interesting behaviour of real data. Although this semi-automatically constructed ontology may be different from the widely accepted one for the same domain, it may indicate the true character of the data from the statistical point of view and have a semantic interpretation as well as being more suitable for the specific data mining application.

1 INTRODUCTION

An ontology has been defined as "an explicit specification of a conceptualization" (Gruber, 1993). Here the conceptualization means the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them (Genesereth and Nilsson, 1987). An ontology is usually arranged hierarchically, like a taxonomy, where a taxonomy is a classification of things in a hierarchical form that expresses a subsumption relation. However, ontology is definitely not limited to a taxonomic hierarchy, for example it may hold a symmetric or transitive relation between concepts or classes, but the backbone of ontology is often a taxonomy.

Traditionally, domain ontologies are created manually, based on human experts' views of the domain knowledge, but it is a time consuming task. In the past decade, an increasing amount of work has been devoted to automatic or semi-automatic ontology construction. To implement this automation, natural language processing and machine learning techniques are usually used, see e.g. (Khan and Luo, 2002), but many of these efforts have been made to build ontologies are using text-based documents as a knowledge source. In the real world, more and more digital data are collected, processed, managed and stored in relational databases. The patterns, associations, or relationships among all this data can also provide information. With the help of data mining techniques, this hidden knowledge can be discovered, and may then be represented in the form of an ontology for a specific purpose.

In this paper, we will focus on a scenario of building an ontology for a real-life application. A database using in data mining often contains one column/attribute field as a target class for classification or prediction purpose, and the rest of the fields are treated as input classes. For instance, a well-known data set, *Adult* data from UCI data repository (Asuncion and Newman, 2007), contains six numeric and eight nominal attributes representing individual details, such as age, education, and marital-status, and one target class indicating income information. In this case, each field can be considered as a class within an ontology, and the values of each attribute may then form a taxonomy, which we call it an attribute-value taxonomy. If the number of values for a certain attribute is very small, and those values obviously can only form a one or two-level taxonomy, then we may manually add it into the ontology. Otherwise, the use of some automatic way to construct the taxonomy seems more desirable. Such attribute-value taxonomies can help a data mining algorithm to produce more compact, interpretable knowledge for the domain expert or decision maker. To do this, the original values can be replaced by those upper level values of the taxonomy under some strategy, which is called abstraction of attribute values or concept generation. A successful example can be found in (Yi et al., 2005). However, using an ontology developed in a domain different from the application area of the database, poor results can follow. For example, using a geographic based ontology for the Native-country field

Yi H. and J. Rayward-Smith V. (2009).
USING A CLUSTERING ALGORITHM FOR DOMAIN RELATED ONTOLOGY CONSTRUCTION.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 336-341
DOI: 10.5220/0002331803360341

within the *Adult* database will not be appropriate.

Our task is to seek a semi-automatic approach to construct a domain ontology which can represent the specific behaviour of the real data, where, this ontology should have a semantic interpretation and might even be slightly modified to achieve this. In data mining, some clustering algorithms can be used to automatically generate a tree structured hierarchy for the data set. In general, these clustering algorithms can be classified into two categories: (1) hierarchical and (2) partitional. For the given set of data objects, hierarchical algorithms aim to find a series of nested clusters of data so as to form a tree diagram or dendrogram; partitioning algorithms will only split the data into a specified number of disjoint clusters. However, a partitional algorithm can be used iteratively to produce a hierarchy. The output of the traditional hierarchical clustering algorithms is often a binary tree, which is not necessarily an appropriate structure compared with a normal ontology. Thus we will consider the efficacy of exploiting a partitional clustering algorithm, Fisher's algorithm (Fisher, 1958; Hartigan, 1975), in generating relevant semantically interpretable taxonomies, which is then extended to an ontology by manually adding the proper relations and properties to them. Fisher's algorithm is an exact algorithm that can minimised the sum of the distance of points from their cluster means. The number of values in the domain is generally small enough for such an exact algorithm to be applied. Alternative clustering algorithm such as K-means (McQueen, 1967) can be used where the number of values in the domain is large, see (Yi, 2009).

This paper is organized as follows. In section 2, we review the Fisher's algorithm and its implementation, then the strategy of constructing the taxonomy based on the clustering results is proposed. We use the *Adult* database to do the experiment in section 3, the semi-automatically constructed ontology is then checked for semantic interpretation. Section 4 is devoted to discussion and conclusions.

2 FISHER'S ALGORITHM

In partitional clustering, it is often computationally infeasible to try all the possible splits, so greedy heuristics are commonly used in the form of iterative optimization. However, when the number of points is small, an exact algorithm can be considered, e.g. Fisher's algorithm.

Working on an ordered data set, or continuous real values, Fisher's algorithm seeks an optimal partition with respect to a given measure, providing the measure satisfies the constraints that if x, y are both in a cluster and the data z, $x < z < y$, then z is also in the cluster.

2.1 Algorithm Description

Given a set of points $D = \{x_1, x_2, \ldots, x_n\}$, with $x_1 < x_2 \ldots < x_n$ on the real line, we seek a partition of the points into K clusters $\{C_1, C_2, \ldots, C_K\}$, where

C_1 comprises points $x_1 < x_2 \ldots < x_{n_1}$,

C_2 comprises points $x_{n_1+1} < x_{n_1+2} < \ldots < x_{n_2}$,

$\vdots$

C_K comprises points $x_{n_{K-1}+1} < x_{n_{K-1}+2} < \ldots < x_{n_K}$, and $x_{n_K} = x_n$.

Thus, such a clustering is uniquely determined by the values $x_{n_1}, x_{n_2}, \ldots, x_{n_{k-1}}$. Any clustering of the points of this form will be called an *interval-clustering*. For certain quality measures on clusters, an optimal interval clustering will always be an optimal clustering.

For example, consider a within-cluster fitness measure for the K interval clusters $C = \{C_1, C_2, \ldots, C_K\}$ of dataset D

$$Fit(D,K) = \sum_{i=1}^{K} d(C_i), \tag{1}$$

where, $d(C_i)$ is a measure of the value of the cluster C_i and

$$d(C_i) = \sum_{x_j \in C_i} (x_j - \mu_i)^2, \text{ where } \mu_i = \sum_{x_j \in C_i} \frac{x_j}{|C_i|}. \tag{2}$$

The optimal clustering is the partition which minimizes $Fit(D,K)$, and this must necessarily be an interval clustering. The time complexity of this algorithm is O(n^K).

2.2 Algorithm Implementation

Fisher (Fisher, 1958; Hartigan, 1975) pointed that optimal K interval clusters can be deduced from the optimal $K-1$ clusters, which means we can successively compute optimal $2, 3, 4, \ldots, K-1$ partitions, and then the optimal K partition. The steps of this dynamic programming procedure are listed below.

1. Create a matrix $dis(j,k)$ which contains the values of the measure $d(C_{jk})$ for every possible interval cluster, $C_{jk} = \{x_j, \ldots, x_k\}$, i.e.

$$dis(j,k) = \begin{cases} d(C_{jk}) & 1 \le j < k \le n, \\ 0 & 1 \le k \le j \le n. \end{cases}$$

2. Compute the fitness of the optimal 2-partition of any t consecutive points set $D_t = \{x_1, x_2, ..., x_t\}$, where $2 \leq t \leq n$, and find the minimum by

$$Fit(D_t, 2) = \min_{2 \leq s \leq t} \{dis(1, s-1) + dis(s, t)\},$$

3. Compute the fitness of the optimal L-interval-partition of any t consecutive points set $D_t = \{x_1, x_2, ..., x_t\}$, where $L \leq t < n$, and $3 \leq L < K$ by using

$$Fit(D_t, L) = \min_{L \leq s \leq t} \{Fit(D_{s-1}, L-1) + dis(s, t)\}.$$

4. Create a new matrix $f(t, L)$ which stores the fitness computed in the above two steps for all optimal L-partitions $(1 \leq L < K)$ on any t points set $D_t = \{x_1, x_2, ..., x_t\}$, where $1 \leq t \leq n$.

$$f(t, L) = \begin{cases} Fit(D_t, L) & 1 < L < K, L < t, \\ dis(1, j) & L = 1, 1 \leq j \leq t, \\ 0 & 1 < L < K, L \geq t. \end{cases}$$

The optimal K-partition can be discovered from the matrix $f(t, L)$ by finding the index l, so that $f(t, K) = f(l, K-1) + dis(l, n)$.
Then the Kth partition is $\{x_l, x_{l+1}, \ldots, x_n\}$, and the $(K-1)$th partition is $\{x_{l^*}, x_{l^*+1}, \ldots, x_{l-1}\}$, where $f(l-1, K-1) = f(l^*-1, K-2) + dis(l^*, l-1)$, and so on.

2.3 Automatic Taxonomy Construction

When the number of cluster is large, each cluster can be replaced by its centroid, where the centroid of a cluster C of reals is the average value and is easily computed; these clusters can then be clustered by applying the algorithm on their centroids. Repeating this procedure, a tree hierarchy of the clusters can be gradually built from bottom to top.

Given a value set, $V = \{V_1, V_2, ..., V_n\}, V_i \in \mathbb{R}$, of a feature/attribute, A, the procedure of partitional clustering based attribute-value taxonomy construction is described as below.

1. Let the number of clusters, k, equal the size of value set, V, then the leaves of the tree are $\{V_i\}$ for each value $V_i \in V$. Call this clustering, C.
2. Determine a suitable k which is less than the current number of clusters, and apply Fisher's algorithm to C to find k clusters.
3. Replace each cluster with its centroid, and reset C to be the new k singleton clusters.
4. Go to step 2 until k reaches 2 or the distance between successive centroids are all sufficiently similar.

3 CASE STUDY

We conducted a case study to demonstrate the automatic construction of taxonomies for a real world database. The *Adult* dataset, extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau, is chosen to carry out the experiment. There are 30,162 records of training data and 15,060 records of test data, once all missing and unknown data are removed. The distribution of records for the target class is shown in table 1.

Table 1: Target Class Distribution.

Data set	Target Class	%	Records
Train	$\leq 50K$	75.11	22,654
Train	$> 50K$	24.89	7,508
Test	$\leq 50K$	75.43	11,360
Test	$> 50K$	24.57	3,700

3.1 Data Preprocessing

As described in section 2, Fisher's algorithm works on a set of ordered or continuous real values. To cluster data with nominal attributes, one common approach is to convert them into numeric attributes, and then apply a clustering algorithm. This is usually done by "exploding" the nominal attribute into a set of new binary numeric attributes, one for each distinct value in the original attribute. For example, the `sex/gender` attribute can be replaced by two attributes, `Male` and `Female`, both with a numeric domain $\{0, 1\}$.

Another way of transformation is using of some distinct numerical (real) values to represent nominal values. If again, using `sex/gender` attribute as an example, a numeric domain $\{1, 0\}$ is a substitute for its nominal domain $\{Male, Female\}$. A more general technique, frequency based analysis, can also be exploited to perform this transformation. For instance, the domain of attribute `race/ethnicity` can be transformed from $\{White, Asian, Black, Indian, other\}$ to $\{0.56, 0.21, 0.12, 0.09, 0.02\}$, according to their occurrence in data.

With the *Adult* data set, prediction is usually interested in identifying what kind of person can earn more than \$50K per year, based on the various personal information, such as education background, marital status, etc. This prediction/classification is very practical for some government agencies, e.g. the taxation bureau, to detect fraudulent tax refund claims. Thus a frequency based transformation seems more appropriate for this task, because each numeric value to be transformed also reveals the statistical information of its original nominal value. Our transformational

scheme replaces each nominal value with its corresponding conditional probability (conditional on the target class membership). In order to benefit from the construction of an attribute-value taxonomy, the nominal attributes with big domains (say, number of values are greater than five) are more interesting.

In this section, three nominal attributes, Education, Marital-status, and Native-country, are chosen for taxonomy construction by using Fisher's algorithm. The "$>$50K" class, denoted by C_h, is chosen as the prediction target, and each value of all selected attributes will be replaced by the conditional probabilities of the person being classified to C_h, given he/she holds this specific value. The training data are used for doing this replacement.

Let $A = \{A_1, A_2, ..., A_n\}$ represent the attributes of *Adult* data set, and $V = \{V_1, V_2, ..., V_n\}$ be the corresponding value set of A. Given $V_{ij} \in V_i$ denotes the jth value of attribute A_i, the conditional probability above can be defined as

$$P(C_h \mid V_{ij}) = \frac{P(C_h, V_{ij})}{P(V_{ij})} = \frac{|C_h\ V_{ij}|}{|V_{ij}|} \qquad (3)$$

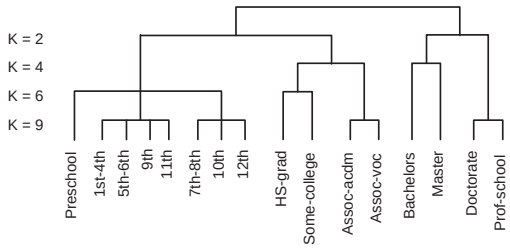
where $|C_h\ V_{ij}|$ is the number of instances classified to C_h, whose value of attribute A_i is V_{ij}, and $|V_{ij}|$ is the total number of instances that hold the attribute value V_{ij}.

For example, suppose Marital-status is the fourth attribute in *Adult* data, and "*Divorced*" is its second value, then $P(C_h \mid V_{42})$ is the probability of the person who can earn more than \$50K per year, given he or she is divorced.

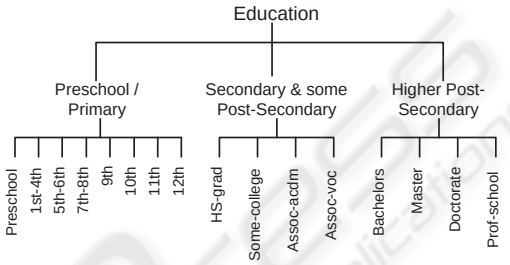
3.2 Experiments and Results

According to the procedure of taxonomy construction described in section 2.3, the number of clusters, k, needs to be preset before running Fisher's algorithm at each iteration. In our inital experiments, k has been chosen manually but, as we develop this research, we expect to use some technique, such as silhouette (Kaufman and Rousseeuw, 1990), for the intelligent selection of k. Figure 1 and figure 2 each show the pair of nominal attribute-value taxonomies for the first two selected fields, respectively. They were built by iteratively exploiting Fisher's algorithm and then modified and labelled to be semantically interpretable. All the taxonomies built by Fisher's algorithm are biased towards partitions that reflect people's yearly income, since all the nominal values are replaced with the conditional probability as described above.

Interestingly, these two automatically generated nominal attribute-value taxonomies have obvious semantic interpretation. For example, after applying the technique to the values of the Education field, three main clusters can be obtained between the first and second top levels in figure 1(a), all of which have obvious semantic interpretation. The modified taxonomy with labelled internal nodes is shown in figure 1(b). Similarly, the taxonomy of Marital-status, shown in figure 2(a), also represents an obvious semantic hierarchy. Its modified version is shown in figure 2(b).



(a) Using Fisher's Algorithm

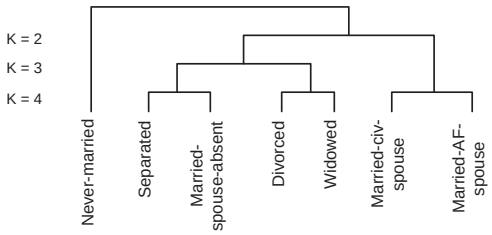


(b) Modified with a semantic interpretation

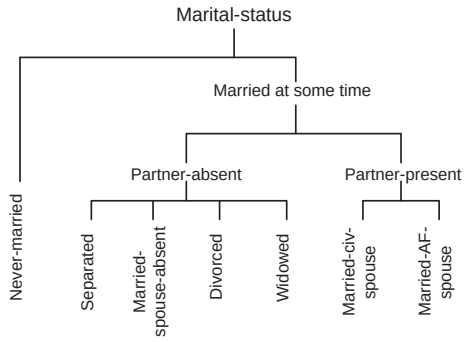
Figure 1: Taxonomies of Education.

However, as we mentioned before, for the Native-country field, neither the geographical location nor the classification based on the economical situation of the country leads to a taxonomy that is suitable for the *Adult* database. This field arises from the census and describes the original countries of people who live in the US. In the *Adult* dataset, about 90% people are native US citizens, and only 25% have more than \$50K yearly income. The use of any widely accepted taxonomy of country is very dangerous and not appropriate. Thus generating a specific taxonomy for this case becomes necessary.

Before applying Fisher's algorithm to the values of Native-country, we selected all the countries with a very small number of samples, i.e. less than 50 records, to be clustered together as a minority class. With US citizens dominating, they are placed in a single cluster. Thus there are three top level nodes in the taxonomy of Native-country. Then we attempt to cluster the remaining 19 countries. Table 2 shows the detailed clusters at each level of the attribute-value



(a) Using Fisher's Algorithm



(b) Modified with a semantic interpretation

Figure 2: Taxonomies of Marital-status.

taxonomy of Native-country in a top-down order, in which the second cluster is the minority class, so we use the "Minorities" to represent it at the bottom levels. For these 19 countries, we noticed that nearly all the Asian countries, except *Vietnam*, and all the European countries are clustered together, and most America countries, except *Canada* and *Cuba*, are grouped in another cluster. All these clusters reflect the income level of US citizens originally from various countries. It is difficult to give a simple semantic interpretation but they could be described as US citizens, minority groups, high earning immigrants, and low earning immigrants. Alternatively, some modification by hand to these clusters could be made to enable a clearer semantic interpretation.

A simple *Adult* ontology can be constructed for some selected fields as shown in figure 3. In this figure, only the relations among the higher level classes, i.e. attributes, are presented, assuming all the attribute-value taxonomies are holding the *hasValue* property. Here we believe there are some relations between the attribute Workclass and Occupation. For instance, the *Transport-moving* may be a self employed job (denoted as *Self-emp-inc* in Workclass field), or provided by a *Private* company. However, this extra detail will not be exploited by our data mining algorithms, such as decision tree or rule induction algorithms (Tan et al., 2006).

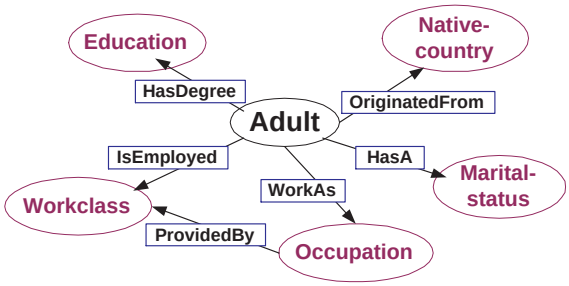


Figure 3: Adult Ontology.

Table 2: The clusters at each level of the attribute-value taxonomy of Native-country.

Cluster No.	Clusters
K = 3	US = {United-States}, Minorities = {Ecuador, Honduras, Nicaragua, Peru, Portugal, Ireland, France, Greece, Hungary, Scotland, Holland-Netherlands, Yugoslavia, Iran, Haiti, Trinadad&Tobago, Cambodia, Thailand, Laos, Hong-Kong, Taiwan,Outlying-US(Guam-USVI-etc)}, {Mexico, Canada, El-Salvador, Puerto-Rico, Columbia, Guatemala, Italy, Germany, England, Poland, Cuba, China, India, Japan, Philippines, South-Korea, Vietnam, Jamaica, Dominican-Republic}
K = 4	US, Minorities, {China, Philippines, Japan, India, Canada, Germany, England, Italy, Poland, Cuba, South-Korea}, {Vietnam, Mexico, Guatemala, Jamaica, El-Salvador, Puerto-Rico}
K = 5	US, Minorities, {China, Philippines, Japan, India, Canada, Germany, England, Italy}, {Poland, Cuba, South-Korea}, {Vietnam, Mexico, Guatemala, Jamaica, El-Salvador, Puerto-Rico}

4 DISCUSSION AND CONCLUSIONS

There are some problems arising when building the taxonomies automatically. Firstly, the choice of k is specified in advance for each run, which may result in various taxonomies. Finding an appropriate number of clusters becomes very crucial for unsupervised automatic taxonomy construction. One way of more objectively choosing k is inspired by the use of **Silhouette width** (Kaufman and Rousseeuw, 1990) in general partitional clustering algorithms. Before ap-

plying Fisher's algorithm to the attribute value set at each run, the optimal number of k is selected according to the maximum of overall average silhouette width, which means the corresponding clustering at each level of the taxonomy is an optimal clustering. But this approach is only suitable where the dataset to be clustered has a large number of points, since the **Silhouette** method often suggests the optimal number of clusters should be 2 for a small number of points.

Secondly, nominal values are clustered based on conditional probability, which means the taxonomies reflect the statistical features of the data and, although this may correspond to semantic similarity, it is not guaranteed so to do. Furthermore, to complete the taxonomy, we also need to use some concepts to represent the internal nodes of the taxonomies, but this can be difficult.

It is usually claimed that an ontology should be reusable and easily used across domains, so it must include all the terms and possible relationships. This results in big and complex ontologies so as to make themselves comprehensive. As mentioned by the Native-country study, naive use of even complex ontologies that aim to reflect many parallel semantic interpretations can still be unwise in a data mining exercise. The taxonomy produced by clustering algorithms can be an useful assistance for users, analysts or specialists to avoid a user's subjectivity.

In conclusion, in this paper one partitional algorithm, Fisher's algorithms, has been introduced and exploited to perform the automatic generation of a concept taxonomy (under some supervision) for some selected nominal attributes of *Adult* data set. Here supervision means not only setting the number of clusters before each clustering iteration but also allowing postprocessing to add semantic interpretation. Two generated taxonomies are modified to be semantically interpretable. The taxonomy of Native-country is also constructed after some preprocessing, from which we revealed some statistical characteristic of the data. All these taxonomies provide a good guide on constructing appropriate concept taxonomies. Such modification is likely to be required if a general taxonomy is to be used for a specific database. Experiments have been undertaken to compare the effectiveness of the Fisher clustering based approach with a heuristic K-means based approach (Yi, 2009). As it happens, on the case study considered here, the results show that Fisher's algorithm can produce more interpretable attribute-value taxonomies than K-means algorithm.

REFERENCES

Asuncion, A. and Newman, D. J. (2007). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences.

Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53:789–798.

Genesereth, M. R. and Nilsson, N. J. (1987). *Logical Foundation of Artificial Intelligence*. Kauffman, Los Altos, California.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. In *Knowledge Acquisition*, volume 5, pages 199–220.

Hartigan, J. A. (1975). *Clustering Algorithms*. New York: John Wiley & Sons, Inc. Pages 130-142.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons, Inc.

Khan, L. and Luo, F. (2002). Ontology construction for information selection. In *Proc. of 14th IEEE International Conference on Tools with Artificial Intelligence*.

McQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley.

Tan, P. N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Pearson Education, Boston.

Yi, H. (2009). *The Construction and Exploitation of Attribute-Value Taxonomies in Data Mining*. PhD thesis, University of East Anglia, to be submitted.

Yi, H. Y., Iglesia, B. d. l., and Rayward-Smith, V. J. (2005). Using concept taxonomies for effective tree induction. In *Computational Intelligence and Security International Conference (CIS 2005)*, volume LNAI 3802, pages 1011–1016.