Growing hierarchical self-organizing map method using category utility

Kazushi Murakoshi [a,*], Satoshi Fujikawa [a]

[a] *Department of Knowledge-based Information Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tenpaku-cho, Toyohashi 441-8580, Japan*

Received 8 June 2011; revised 21 July 2014; accepted 29 March 2015

Abstract

In order to automatically obtain hierarchical knowledge representation from a certain data, an unsupervised learning method has been developed that overcomes two problems of the growing hierarchical self-organizing map (GHSOM) method, which uses the quantization error, the deviation of the input data, as evaluation measure of the growing maps: proper control of the growth process of each map is difficult due to the use of the quantization error and the clusters in the hierarchical structure may be excessively subdivided. This improved GHSOM method uses the category utility (CU), a measure used in conceptual clustering for predicting the preferred level of categorization, instead of the quantization error. The CU is useful for organizing the clustering so that people can effortlessly understand it. The basic principle of this method is that the growth and unification processes are appropriately and autonomously controlled by the CU. Evaluation using computer experiments showed that the proposed method can automatically construct an appropriate hierarchical and topological knowledge representation for high-dimensional input data through unsupervised learning. It also showed that it is easier to use and more effective than the original conventional GHSOM method using the quantization error as an evaluation measure.

Key words:
Growing hierarchical self-organizing map; Category utility; Unification; Cluster goodness

* Corresponding author. phone: +81-532-44-6899; fax: +81-532-44-6873.
Email address: `mura@tut.jp` (Kazushi Murakoshi).

Preprint of an Article published in International Journal of Software Engineering and Knowledge Engineering, 26(2): 217-237, 2016. doi:10.1142/S0218194016500108

1 Introduction

The use of self-organizing maps (SOMs) (Kohonen, 2001) is a prominent unsupervised learning method and an automated adaptive knowledge representation scheme, for clustering and visualizing high-dimensional input data. Although SOMs are widely used in various applications, they suffer at least two limitations: a static network architecture in terms of map size and a limited ability to represent the hierarchical data relationships.

These limitations led to the development of the "growing hierarchical self-organizing map " (GHSOM) method (Dittenbach et al., 2002; Rauber et al., 2002). A GHSOM is an artificial neural network model with a hierarchical architecture comprising independent growing self-organizing maps. GHSOM method was applied to visualization of real data (Palomo et al., 2012; Chattopadhyay et al., 2014). Although it overcomes the two limitations, we indicate the two problems as follows. First, it is possible that the growth process of each map cannot be properly controlled because the quantization error, the deviation of the input data, is not a suitable measure for clustering on each map. Second, the clusters in the hierarchical structure excessively subdivided when sub-maps are constructed due to the excessively detailed data representation. The primary cause of these problems is that the quantization error, which serves as a measure of the quality of maps, cannot be used to appropriately evaluate the quality of input-data allocation. This makes it difficult for the users of the GHSOM to provide appropriate depth and area parameters for cluster formation. Therefore, an alternative measure is necessary for automatically forming appropriate maps.

Such measures have been developed in the field of conceptual clustering (Michalski et al., 1983). They are used to estimate the meanings of clusters. The conventional conceptual methods (Fisher, 1987; Gennari et al., 1989; Li and Biswas, 2002; Scanlan et al., 2006, 2008; Godoy and Amandi, 2006; Chien et al., 2009), however, cannot learn the topology among the sub-clusters in a cluster, whereas the GHSOM method can. We have developed an unsupervised learning method for automatically constructing an appropriate hierarchical and topological knowledge representation for high-dimensional input data. It uses an evaluation measure used in the field of conceptual clustering, the category utility (CU) (Corter and Gluck, 1992; Gennari et al., 1989), instead of the quantization error used in the GHSOM method. The CU is an evaluation function used to predict, on the basis of psychological research, the preferred level of categorization in human hierarchical organizations. It is thus a useful measure for organizing the clustering so that people can effortlessly understand it. The effort required is minimal because a hierarchical knowledge representation can be automatically constructed. This is because the parameter that determines the growth process of a map does not need to be set when using the CU, unlike when using the quantization error.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce the GHSOM method, and describe its problems. Section 3 explains the CU. Section 4 describes our proposed method. Section 5 presents the results of computer experiments used to evaluate our proposed method in comparison with conventional methods. Section 6 concludes the paper with a brief summary of the main points.

2 Growing hierarchical self-organizing map

2.1 Description

A GHSOM (Dittenbach et al., 2002; Rauber et al., 2002) is a growing hierarchical self-organizing map that resolves the two problems with conventional SOM: (1) that users must set an appropriate map size before learning, and (2) there is a limitation in intuitively representing hierarchical data relationships. The sizes of the SOMs and the depth of the hierarchy of the GHSOM are determined during unsupervised training processes in accordance with the data structure. The depth of the hierarchy is defined as the maximum difference between the uppermost layer and any other layer. After the training, the map of the higher layer in the GHSOM provides a rather coarse organization of the main clusters in the input data while the map of the lower layer provides a more detailed organization.

In the GHSOM method, the growth processes are based on quantization errors. The quantization error of unit i is represented by

$$qe_i = \sum_j \parallel \boldsymbol{m}_i - \boldsymbol{x}_j \parallel, \tag{1}$$

where $\boldsymbol{m}_i$ is the reference vector of the i-th unit, and $\boldsymbol{x}_j$ is the j-th input vector that is mapped onto the i-th unit. The mean of the quantization errors (qe_i) for all units on map m is represented by MQE^m. Using these quantization errors, the GHSOM method controls both the growth process for each SOM and the global hierarchical growth process.

The procedure of the GHSOM is as follows.

(1) Create the 0-th unit in layer 0 (the upper most layer), calculate quantization error qe_0, and define this unit as parent unit u.
(2) Create a new sub-map, m, in the next lower layer corresponding to parent unit u; give it an initial size and set expansion number n to 0.

(3) Train map m with a random order of inputs per period for T periods in accordance with the standard SOM training procedure using a linear decreasing learning rate (initial learning rate is α_0) and Gaussian neighborhood function.
(4) After the training, assign each input to the unit, i.e., cluster with the smallest Euclidean distance from the input.
(5) Calculate quantization error qe_i of each unit, and calculate the mean quantization error MQE_n^m.
(6) If $MQE_n^m < \tau_m \cdot qe_u$, go to step 8; qe_u is the quantization error of parent unit u for the i-th unit in the present layer and τ_m $(0 < \tau_m < 1)$ is the parameter that controls the size of the map.
(7) Expand map m on the basis of the error unit with the maximum quantization error $(n := n + 1)$, and go to step 3.
(8) If $MQE_n^m \geq \tau_h \cdot qe_0$, set each i-th unit in map m to parent unit u and go to step 2; τ_h $(0 < \tau_h < 1)$ is the parameter that controls the creation of the lower hierarchical maps.

In this map expansion process, a row or column of units is inserted into map m as illustrated in Fig. 1 (a). The insertion location is the place between the *error unit* and the *dissimilar unit*, as shown in Fig. 2. The error unit is the unit with the highest quantization error, and the dissimilar unit is the nearest neighboring unit with the maximum Euclidean distance between its reference vector and the reference vector of the error unit. To obtain a smooth positioning of the newly added units in the input space, their initial reference vectors are set to the average of the reference vectors for their nearest neighbors.

2.2 Problems

We have already indicated the two problems of the GHSOM method as follows.

- Because the growth processes are based on the quantization error, they are difficult to properly control.
- Because the classifications are excessively detailed in the maps, the clusters in the hierarchical structure are unnecessarily subdivided.

Regarding the first problem, the quantization error is not a suitable measure for clustering on each map. It is not suitable because it becomes smaller as the number of units increases. Therefore, it does not provide sufficient information for goodness of clustering on each map. Thus, the GHSOM restricts the expansion of maps on the basis of parameter τ_m, which the user adjusts. It is hard for the user to determine an appropriate value for τ_m. Therefore, growth processes based on the quantization error cannot be adequately controlled.

Regarding the second problem, when the high-dimensional input data is arranged on a low-dimensional map by the SOM, there is not always one cluster per unit. The GHSOM method, however, creates the lower hierarchical map for each unit as one cluster, even if the cluster corresponds to numerous units. Thus, the clusters in the hierarchical structure formed by the GHSOM method are excessively subdivided when new sub-maps are constructed due to the excessively detailed data representation in the map.

3 Category utility

The primary cause of the problems with the GHSOM method described in Section 2 is that the quantization error, which serves as the measure of the quality of maps, is not suitable for evaluating the quality of input-data allocation. Therefore, an alternative measure is necessary for preparing appropriate maps.

Such measures have been developed in the field of conceptual clustering (Michalski et al., 1983), as mentioned in the Introduction. They can be used to estimate the meanings of clusters. The conventional conceptual methods (Fisher, 1987; Gennari et al., 1989; Li and Biswas, 2002; Scanlan et al., 2006, 2008; Godoy and Amandi, 2006; Chien et al., 2009), however, cannot learn the topology among the sub-clusters in a cluster, whereas the GHSOM method can.

Of the various measures for clustering developed in the field of conceptual clustering, we use the category utility (CU) one (Corter and Gluck, 1992; Gennari et al., 1989) for the three reasons.

- It is based on the results of psychological experiments.
- It is effective as a measure for clustering.
- It is widely used for various applications.

The CU is a normative information theoretic measure developed by Corter and Gluck Corter and Gluck (1992) to predict the basic level in human classification hierarchies. The basic level was defined from psychological experiments by Rosch et al. Rosch et al. (1976) as the level at which categories are most distinctive from one another; basic level categories (e.g., *apple*) are generally identified more quickly than either more general (e.g., *fruit*) or more specific (e.g., *golden delicious apple*) categories during object recognition. The theory of the basic level is based on a principle called cognitive economy: people attempt to obtain the maximum information with the least cognitive effort (Ungerer and Schmid, 1996). Corter and Gluck Corter and Gluck (1992) conducted numerical experiments to show that the CU is maximum at the basic level. It is also indicated that the CU is equivalent to the mutual information (Corter and Gluck, 1992). When the value of CU is large, the category can be considered to be a good cluster (Corter and Gluck, 1992). Moreover, the measure is used for various applications such as database design (Ionnidis et al., 1992), semantic web discovery (Clerkin et al., 2001), and sentence classification (Theodorakis et al., 2004). The CU is thus a suitable measure for estimating the quality of clustering.

The value of the CU for a category c_k is given by

$$CU(c_k) = P(c_k) \sum_i \sum_j \{P(a_i = f_{ij}|c_k)^2 - P(a_i = f_{ij})^2\}, \tag{2}$$

where a_i is the i-th attribute, f_{ij} is the j-th feature value of a_i, and $P(\cdot)$ is a probability function. Namely, $P(a_i = f_{ij})$ is the unconditional probability of attribute a_i taking on value f_{ij}, and $P(a_i = f_{ij}|c_k)$ is the conditional probability of $a_i = f_{ij}$ given category c_k. We derive the following equation from Eq. (2) and use it to compare the probability of a category with that of the immediate upper category.

$$CU^{nominal}(c_k) = P(c_k) \sum_i \sum_j \{P(a_i = f_{ij}|c_k)^2 - P(a_i = f_{ij}|c_l)^2\}, \tag{3}$$

where c_l is the immediate upper category of c_k.

The COBWEB system for hierarchical conceptual clustering (Fisher, 1987) uses the CU measure as the criterion function to represent concepts probabilistically in a hierarchical tree structure. COBWEB is incremental and computationally economical and thus can be flexibly applied in a variety of domains. However, the clustering tree structure obtained with COBWEB strongly depends on the instance input order. Thus, the clustering structure cannot excellently preserve the topology of input data.

The CU measure in the original COBWEB (Fisher, 1987) can handle only nominal data. In the CLASSIT (Gennari et al., 1989) method, the extended version of COBWEB, a modified CU that matches the numerical data is used. For a set of numeric feature values, CU is defined as

$$CU^{numeric}(c_k) = \frac{P(c_k)}{2\sqrt{\pi}} \sum_i \left\{ \frac{1}{\sigma_{ik}} - \frac{1}{\sigma_{il}} \right\}, \tag{4}$$

where σ_{ik} and σ_{il} are the standard deviation of attribute i in category k and in immediate upper category l, respectively. To overcome the problem that the standard deviations are zero, a limit parameter, *acuity*, was introduced to specify the minimum value for the standard deviations. It corresponds to the lower limit on our perception ability in psychophysics.

The CU measure was extended in the COBWEB/3 (Li and Biswas, 2002) to handle a mix of nominal and numeric feature values. The overall CU is the sum of the nominal and numeric CU:

$$CU(c_k) = CU^{nominal}(c_k) + CU^{numeric}(c_k). \tag{5}$$

The complete expression for category utility is

$$MCU = \sum_{k=1}^{N} \frac{CU(c_k)}{N}, \tag{6}$$

where N is the number of categories. The division lets one compare different size clusterings. In our method, a cluster (as a unit) obtained by using a SOM is considered to be in category c_k above. That is, the category labels do not need; our proposed method is one of unsupervised learning methods.

4 Proposed method

Our proposed method uses the category utility (CU) measure (Corter and Gluck, 1992) instead of the quantization error used in the GHSOM method (Dittenbach et al., 2002; Rauber et al., 2002). Its basic principle is that the growth and unification processes are appropriately and autonomously controlled by the CU. Note that category c_k described in Sec. 3 is the cluster in the following procedure. The procedure of the proposed method is as follows.

(1) Create the 0-th unit in layer 0 (the upper most layer), and define this unit as parent unit u.
(2) Create a new sub-map, m, in the next lower layer corresponding to parent unit u; give it an initial size and set expansion number n to 0.
(3) Train map m with a random order of inputs per period for T periods in accordance with the standard SOM training procedure using a linear decreasing learning rate (initial learning rate is α_0) and Gaussian neighborhood function.
(4) After the training, assign each input to the unit, i.e., cluster with the smallest Euclidean distance from the input.
(5) Calculate the category utility CU_i of each unit, and calculate the mean category utility MCU_n^m.
(6) Unify clusters on the basis of the category utility.
(7) Recalculate the category utility of each cluster, and recalculate the mean category utility.
(8) If $MCU_{n-1}^m \geq MCU_n^m$, go to step 10.
(9) Expand map m on the basis of the error unit with the minimum category

utility ($n := n + 1$), and go to step 3.

(10) If $MCU^m_{n-1} \geq \tau_h$, set each i-th unit in map m to parent unit u and go to step 2, where τ_h is the parameter that controls the creation of the lower hierarchical maps.

This differs from the conventional GHSOM method in several areas.

- The control process used for expanding the map.
- The criterion for selecting the unit insertion location on the expanded map.
- The criterion for stopping the expansion of the hierarchical structure.
- The process of unifying clusters.

Remarkably, each of these differences is based on the CU described in Section 3.

Regarding the first difference, the process controlling the map expansion depends on the use of the CU instead of the quantization error. Figure 1 illustrates the differences in map expansion between the control processes used in the conventional method (a) and in the proposed method (b). With the conventional GHSOM method (Fig. 1 (a)), the control process is executed until the mean quantization error, MQE_n, of the n-th expansion map is less than the product of parameter τ_m and quantization error qe_u, where τ_m is the parameter that controls the size of the map, and qe_u is the quantization error of the corresponding (parent) unit in the upper layer. Since the quantization error measure cannot be used to evaluate the quality of clusters in the present map, the value of the measure for the present map is compared with that of the corresponding unit in the upper layer. Additionally, users of the conventional GHSOM method must provide the proper value of τ_m. With the proposed method (Fig. 1 (b)), on the other hand, the control process is executed as long as the mean CU, MCU_n, of the n-th expansion map is more than the mean CU, MCU_{n+1}, of the next map. Thus, map expansion is stopped at the maximum value of the CU measure. Since the CU measure can be used to evaluate the quality of clusters in the present map, the value of the measure for the present map is compared with that for the next expansion map. Note that users of the proposed method do not face the difficult task of setting up an appropriate parameter, such as parameter τ_m in the conventional GHSOM method.

Regarding the second difference, the CU is used instead of the quantization error to choose the locations for unit insertion during the expansion process. In the conventional GHSOM method, the insertion location is the place between the *error unit* and the *dissimilar unit*, as shown in Fig. 2. The error unit is the unit with the highest quantization error, and the dissimilar unit is the nearest neighboring unit with the maximum Euclidean distance between its reference vector and the reference vector of the error unit. To obtain a smooth

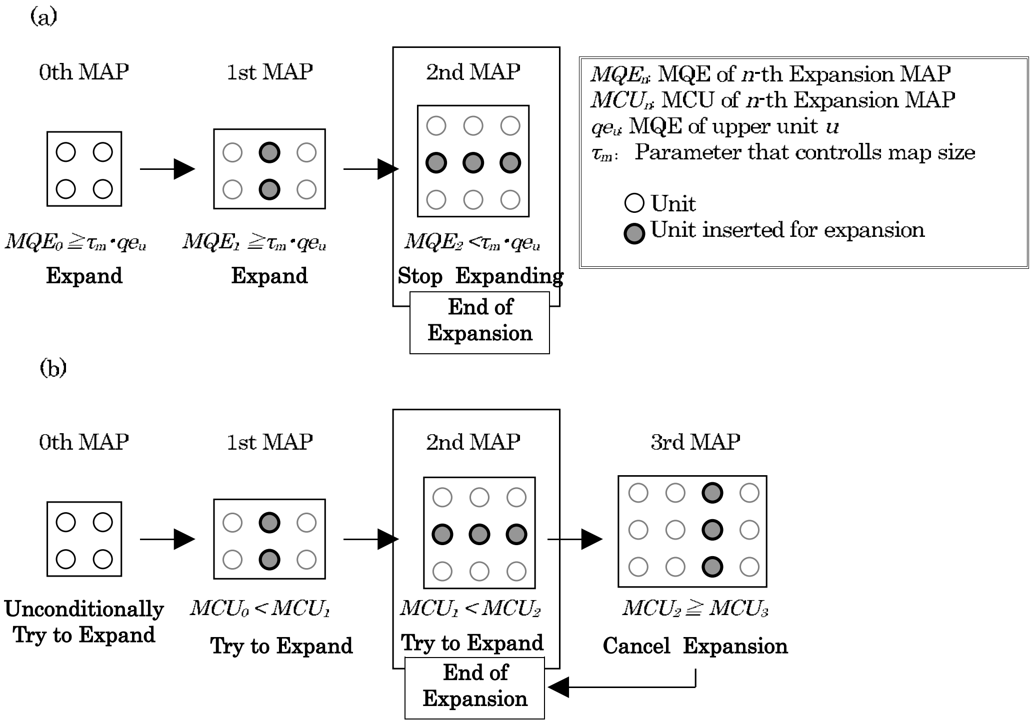

Fig. 1. Difference in map expansion between control processes for (a) GHSOM method (mean quantization error) and (b) proposed method (category utility).

positioning of the newly added units in the input space, their reference vectors are initialized as the average of the reference vectors of their nearest neighbors. In the proposed method, the error unit is selected as the unit with the lowest CU. The other processes are the same as those of the conventional method. The error unit with the lowest CU has the lowest quality clustering while that with the highest quantization error does not entirely has the lowest quality clustering. Thus, using CU for inserting units works better for clustering than using the quantization error.

Regarding the third difference, the CU is used instead of the quantization error to determine when to stop expanding the hierarchical structure. In the conventional GHSOM method, the global hierarchical expansion is executed until the mean quantization error, MQE^m, of map m is less than the product of parameter τ_h and quantization error qe_0 of the 0-th unit in layer 0 (the uppermost layer). In the proposed method, it is executed until the mean CU, MCU^m, of map m is less than τ_h. The users of both methods must provide the value of τ_h. However, that in the proposed method is the threshold value of the category utility while that in the conventional GHSOM method is the ratio used to compare the values of two quantization errors; the τ_h in the proposed method represents the quality of clustering while that in the conventional GHSOM has no explicit meaning. That is, it is easier for users of the proposed method to provide a value for τ_h than it is for users of the

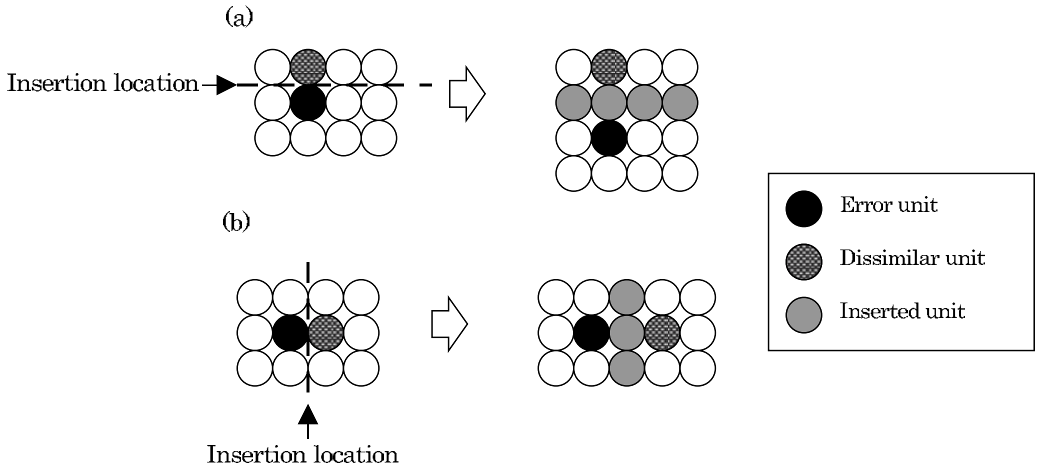

Fig. 2. Insertion of units. (a) a row. (b) a column.

conventional GHSOM method.

Regarding the fourth difference, the unification of clusters is newly added. To overcome the problem that clusters in the hierarchical structure for input data are excessively subdivided, as described in Section 2, the hierarchical growth process is optimized by unifying clusters on the basis of the category utility after training each map. If a single cluster is represented by numerous units, i.e., numerous clusters, it should be one cluster in this unification process. A cluster, however, is occasionally excessively subdivided by learning in SOM. Thus, the unification process is necessary for improving clustering quality. The procedure for unifying clusters is illustrated in Fig. 3. First, the cluster with the lowest CU_i is selected as a merge candidate cluster. Second, an attempt is made to find the best candidate cluster for unification from the four neighboring clusters. The best candidate is the cluster for which the CU of the sum of the cluster and the merge candidate cluster is the maximum among the four neighboring clusters. Third, if the global mean CU, MCU^m, of map m increases after trying to unify the clusters, the unification of clusters is accepted, and the procedure returns to the first step. Otherwise, unification is terminated. Since a merge candidate cluster on a map has a lower CU than those of the other clusters. We simply need to recursively find the cluster with the minimum CU to unify clusters. This unification process prevents excessive cluster subdividing. This unification process is effective for the method with the CU, but not for that with the quantization error, because the quantization error monotonically decreases with an increasing number of units, as discussed in Sec. 2.2. In the conventional GHSOM method with the unification process, in contrast, the quantization error increases due to the decreasing number of units resulting from the unification process.

These differences lead to several positive effects.

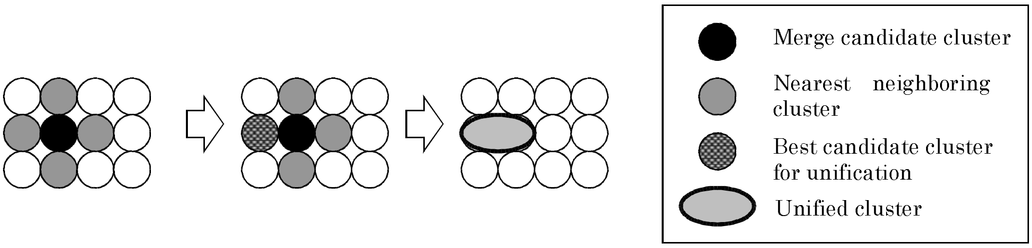

Fig. 3. Unifying clusters.

- The expansion process is stopped appropriately without having to set a parameter.
- Unit insertion using CU works better for clustering than that using the quantization error.
- It is easier to provide τ_h with the proposed method than with the conventional GHSOM method.
- The unification prevents excessive subdividing of clusters.

5 Evaluation

To evaluate the proposed method in comparison with other methods, we conducted computer experiments. The proposed method is denoted as CUU-GHSOM, which corresponds to the GHSOM method using the category utility (CU) measure with cluster unification. CU-GHSOM is the GHSOM method using the category utility (CU) measure without unification. The original conventional GHSOM method (Dittenbach et al., 2002; Rauber et al., 2002) is denoted as QE-GHSOM, namely, the GHSOM method using the quantization error. When we refer to a common characteristic of these methods, it is denoted as simply GHSOM. Comparison of the results with CU-GHSOM and with CUU-GHSOM revealed the effects of cluster unification. We did not use QE-GHSOM with cluster unification, because it is not effective, as described in Section 4. We additionally compared these methods with the COBWEB/3 method (Li and Biswas, 2002) because it uses the CU measure. In summary, the experimentally investigated methods were (1) COBWEB/3, (2) QE-GHSOM, (3) CU-GHSOM, and (4) CUU-GHSOM.

To evaluate the methods, we used two kinds of data: 1) artificially generated data, with nominal and numeric feature values and 2) real data, the zoo data obtained from the UCI machine learning database (Asuncion and Newman, 2007). The artificial data was quantitatively evaluated while the real data was qualitatively evaluated, because there is seldom any real hierarchical clustering problem with its hierarchical answer. Moreover, even if there were such a problem, the number of instances of real data is very small. For example, the number of instances for fruit hierarchical data, as shown by Tversky and Hemenway Tversky and Hemenway (1984), is 6. Therefore, we constructed an artificial hierarchical input data structure to quantitatively evaluate the results for any type of hierarchical cluster structure data. On the other hand, for the real data, the zoo data, we present illustrative results generated by the GHSOM methods in order to evaluate qualitatively. In the following experiments, since the normalized inputs for SOMs are recommended in order to emphasize the differences (Kohonen, 2001), we normalized inputs in each numeric attribute for SOM learning. However, the inputs for CU measure were not normalized so that the evaluations of the clusterings were correctly executed.

5.1 Artificial data

We first describe the procedure used for generating the artificial data. Then, we present the indexes defined for quantitatively evaluating the results. We next explain the experimental method. Finally, we present the results and discuss them.

5.1.1 Data description

We generated two different types of artificial data, as shown in Table 1: 1) nominal and 2) numeric. For the nominal data, a cluster consists of several attributes. For simplicity, we used binary data as the nominal data. When a cluster includes an instance, at least one of the feature values of the attributes is true (1); all other feature values are randomly true (1) or false (0). If a cluster does not include an instance, the feature values are always false (0). For the numeric data, several categories are included in an attribute. The feature values are generated by sampling normal distributions with different means and standard deviations for each cluster. The limit parameter, *acuity*, was set to 1 in each case. The random values in the nominal and numeric data acted as noise.

5.1.2 Evaluation indexes

Accuracy is often used as an evaluation index for non-hierarchical clustering (Huang and Ng, 1999; Devaney and Ram, 1997). Using only accuracy index, however, has a problem for hierarchical clustering as the following consideration.

Then, we defined two indexes for quantitatively evaluating performance: accuracy and cohesion. Accuracy is defined as the average probability of the

Table 1
Characteristics of artificial data.

	Nominal	Numeric
Number of instances	168	189
Number of attributes	114	3
Layer 1: Data type	nominal	numeric
Number of clusters	2	3
Number of attributes per cluster	3	—
Number of clusters per attribute	—	3
distribution		N($\mu = 50, \sigma = 2$)
		N($\mu = 100, \sigma = 2$)
		N($\mu = 150, \sigma = 2$)
Layer 2: Data type	nominal	numeric
Number of clusters	6	3
Number of attributes per cluster	3	—
Number of clusters per attribute	—	3
distribution		N($\mu = 10, \sigma = 2$)
		N($\mu = 40, \sigma = 2$)
		N($\mu = 70, \sigma = 2$)
Layer 3: Data type	nominal	numeric
Number of clusters	2	3
Number of attributes per cluster	3	—
Number of clusters per attribute	—	3
distribution		N($\mu = 10, \sigma = 2$)
		N($\mu = 30, \sigma = 2$)
		N($\mu = 50, \sigma = 2$)

correct classification at each level, whereas the probability of correct classification means the ratio of the number of instances, which exceeds the majority in a unit, of a cluster to the number of instances in the unit. Cohesion is defined as the average of the reciprocals of the numbers of classifications from one cluster of the original input data. High accuracy means that the data were accurately classified, whereas low accuracy means that the classified data include other data from other clusters. High cohesion means that the data in one cluster of the original input data were classified into a single group, whereas low cohesion means that the data in one cluster were classified into numerous groups.

It is desirable for both indexes to be high, but there is usually a trade-off between accuracy and cohesion at the limits of clustering. In extreme examples, if each instance is classified into one group, accuracy is 1, and cohesion approaches 0; if all instances are classified into one group, accuracy is 0, and cohesion is 1. Therefore, we defined a new index, cluster goodness. Cluster goodness is defined as the harmonic average of accuracy and cohesion.

5.1.3 Method

We conducted 100 runs per parameter combination with the input data in random order for each method and averaged the evaluation indexes of the results. Whereas the COBWEB/3 method has no parameters that users assign, the parameters in the GHSOM methods were assigned as follows. The initial size of the map was 2×2. The number of periods for a training run, T, was set to 25. Parameter σ, which determines the neighborhood distance, was set to 1. Parameter τ_h, which controls the hierarchical growth, was set to 0.003 to ensure that the entire hierarchical structure was obtained. These parameters were kept constant for simplicity.

Two additional parameters were variable because they are thought to greatly affect performance. Initial learning rate α_0 was varied from 0.1 to 0.9 in steps of 0.1 Parameter τ_m, which controls the size of the map, was varied from 0.1 to 0.9 in steps of 0.1 (only for QE-GHSOM).

5.1.4 Results and discussion

The results for the nominal data and numeric data are shown in Tables 2 and 3, respectively. For the GHSOM methods, we show the results at the best, worst, and average of cluster goodness, respectively. The results obtained by the proposed CUU-GHSOM method for each type of input data were the highest in terms of quality for the best, worst, and average of the index. They were also the highest for the other input data with the other settings (number of instances, depth of hierarchy, number of clusters, and number of attributes), although they are not shown here.

Note that the performance of CUU-GHSOM and CU-GHSOM did not depend on the variable parameter, α_0, whereas that of QE-GHSOM strongly depended on the map expansion parameter, τ_m. This means that the proposed method is easier to use than the QE-GHSOM method, which requires much effort to assign the appropriate parameters. Moreover, the indexes of CUU-GHSOM were better than those of CU-GHSOM, especially for the numeric data, as shown in Table 3. This indicates the beneficial effect of the unification process. The poor performance of COBWEB is attributed to the fact that COBWEB strongly depends on the instance input order.

Here, we discuss the computational cost of adding the CU and the unification process. Surely, the computational cost of our method increases. However, the cost is not a problem for the user because CUU-GHSOM requires no effort to assign the appropriate parameters. The users of CUU-GHSOM can obtain a fine result by one-shot while the users of QE-GHSOM must execute it repeatedly with tuning of parameters to obtain a fine result. On the other hand, the users of COBWEB do not need tuning of parameters. However, the results of COBWEB are worse than those of our method.

Table 2
Comparison of results of 100 runs for artificial nominal data.

Method		τ_m	α_0	Accuracy	Cohesion	Cluster Goodness
COBWEB/3	—	—	—	0.771	0.752	0.762
QE-GHSOM	best	0.9	0.3	0.673	0.854	0.735
	worst	0.1	0.8	0.938	0.180	0.302
	average	—	—	0.764	0.613	0.616
CU-GHSOM	best	—	0.1	0.860	0.829	0.844
	worst	—	0.8	0.821	0.820	0.820
	average	—	—	0.837	0.829	0.833
CUU-GHSOM	best	—	0.1	0.887	0.894	0.891
	worst	—	0.9	0.874	0.884	0.879
	average	—	—	0.886	0.886	0.886

5.2 Real data

Our evaluation using real data focused on the illustrative final results generated by CUU-GHSOM. Some results for the other GHSOM methods are given to illustrate the differences between the results generated by CUU-GHSOM and those by the other GHSOM methods.

5.2.1 Data description

The zoo data was obtained from the UCI machine learning repository (Asuncion and Newman, 2007) contained a mixture of nominal and numeric at-

Table 3
Comparison of results of 100 runs for artificial numeric data.

Method		τ_m	α_0	Accuracy	Cohesion	Cluster Goodness
COBWEB/3	—	—	—	0.616	0.374	0.466
QE-GHSOM	best	0.2	0.1	0.854	0.544	0.664
	worst	0.1	0.3	0.920	0.311	0.465
	average	—	—	0.771	0.534	0.622
CU-GHSOM	best	—	0.4	0.951	0.471	0.630
	worst	—	0.2	0.950	0.441	0.603
	average	—	—	0.948	0.456	0.616
CUU-GHSOM	best	—	0.1	0.977	0.939	0.958
	worst	—	0.9	0.951	0.840	0.892
	average	—	—	0.949	0.890	0.918

tributes. The cluster labels (types in the zoo data) were not used as an attribute in the clustering process for unsupervised learning. Note that we cannot evaluate the result quantitatively, since this data set has no hierarchical answer. The zoo data consists of 101 instances defined by 16 attributes: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic, and catsize. The attribute legs has only numeric feature values while the other attributes have Boolean (nominal) feature values. The limit parameter, *acuity*, was set to 1 for the legs attribute. The initial size of the map was 2×2. The number of times for a training run, T, was set to 25. Parameter σ, which determines the neighborhood distance, was set to 1.

5.2.2 Results and discussion

Examples (up to the second layer) of final clustering results generated by CUU-GHSOM with $\alpha_0 = 0.1$ and $\tau_h = 0.8$ for the zoo data are shown in Figs. 4-5. The other clustering results with different learning rate parameters obtained by the CUU-GHSOM method were very similar to those presented here. The display style is based on that in the software package for the conventional GHSOM method (Rauber, 2007). The map is displayed as a table, and each cell in the table is a unit, i.e., a cluster. The value of CU, the attribute names, and their average values (plus the standard deviation for the numeric attribute) are shown on the left, and the names of the instances clustered in the cell are displayed on the right. An alphabetic character (a to g) representing the type of zoo is shown at the head of each instance name. This type was not used in learning process. The shading indicates the attributes that had a mean feature value at least 0.2 greater than that in the upper (parent) unit. No instances were clustered in the units marked with '*****'. Hereafter, the unit (cluster) location is indicated as (x, y), where x represents horizontal position and y represents vertical position.

Cluster (0,0) in the first layer in Fig. 4 is distinguished by high average values for hair and toothed, and a low value for eggs, and so on, corresponding to mammalian. Cluster (0,1) is distinguished by low average values for hair and legs, and a high value for eggs, aquatic, and so on, corresponding to nonmammalian which have few legs, for example, reptiles, fish, etc. Cluster (1,1) is distinguished by low average values for hair, milk, toothed, and a high average for eggs, and so on, corresponding to non-toothed nonmammalian, for example, birds, insects, etc. Note that similar clusters are located more closely to each other than distinct ones; the topology of the input data is embedded into the map on the basis of the ability of the SOM. However, the conventional conceptual methods such as the COBWEB (Fisher, 1987), CLASSIT (Gennari et al., 1989), SBAC (Li and Biswas, 2002), DynamicWeb (Scanlan et al., 2006, 2008), and WebDCC (Godoy and Amandi, 2006) cannot excellently preserve the topology of the input data. For this reason, we do not show the final clustering results generated by COBWEB/3. In the second layer, for example, the major difference between cluster (0,0) and cluster (1,1) at the sub-map level, as shown in Fig. 5, is in the average value for predator; the instances in cluster (0,0) are seldom predatory mammalian while those in cluster (1,1) are predatory mammalian.

Finally, we consider the differences between the CUU-GHSOM (proposed GHSOM) results above and some of the CU-GHSOM (GHSOM using CU but without unification process) and QE-GHSOM (original GHSOM using quantization error) results. Figure 6 shows a map of the final results generated by CU-GHSOM with the same α_0 and τ_h as used for CUU-GHSOM. This map (MCU = 0.266739) is more subdivided than the map (MCU = 0.380885) (Fig. 5) generated by CUU-GHSOM, because the unification process is not executed for CU-GHSOM, so the map is excessively subdivided in the early stages of the expansion process. In the case without the unification process, knowledge on predatory mammalian (Fig. 6 compared with Fig. 5) is divided, that is, we hard to understand the knowledge on predatory mammalian. This shows that the unification process is effective. Figure 7 shows a map of the final results generated by QE-GHSOM for $\tau_m = 0.1$ and the same α_0 and τ_h as used for CUU-GHSOM and CU-GHSOM. This map is even more subdivided than the CUU-GHSOM map (Fig. 4), because the expansion process for QE-GHSOM greatly depends on τ_m. In other words, the GHSOM using quantization error (QE-GHSOM) does not appropriately stop the expansion process while the GHSOM using CU (CUU-GHSOM) can appropriately stop it. In the case of QE-GHSOM (original GHSOM), there is the case that knowledge such as mammalian (Fig. 7 compared with Fig. 4) is not obtained. Of course, the results for QE-GHSOM with an appropriate τ_m may be similar to those for CUU-GHSOM, but finding the appropriate value is a time-consuming process. Note that users of the CUU-GHSOM need not to repeat setting the parameter and checking the results. This shows that using CU instead of quantization error is effective. In summary, users of the proposed method CUU-GHSOM do not face these problems because they can automatically construct an appropriate hierarchical knowledge representation.

6 Conclusion

Our proposed growing hierarchical self-organizing map (GHSOM) method using the category utility as an evaluation measure with cluster unification automatically constructs an appropriate hierarchical and topological knowledge representation for high-dimensional input data through unsupervised learning. Evaluation using computer experiments showed that our proposed method is easier to use and more effective than the conventional GHSOM method using the quantization error as an evaluation measure and the GHSOM method using the category utility measure without unification.

More verification through experiments and evaluations for various data is still remained. Especially, we will focus on knowledge acquisition from various real data as a future work.

References

Asuncion, A., Newman, D., 2007. UCI machine learning repository. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Chattopadhyay, M., Dan, P. K., Mazumdar, S., 2014. Comaparison of visualization of optimal clustering using self-organizing map and growing hierarchical self-organizing map in cellular manufacturing system. Applied Soft Computing 22, 528–543.

Chien, B.-C., Hu, C.-H., Ju, M.-Y., 2009. Learning fuzzy concept hierarchy and measurement with node labeling. Information Systems Frontiers 11, 551–559.

Clerkin, P., Cunningham, P., Hayes, C., 2001. Ontology discovery for the semantic web using hierarchical clustering. In: Online Proceedings of Semantic Web Mining Workshop at ECML/PKDD-2001.

Corter, J. E., Gluck, M. A., 1992. Explaining basic categories: Feature predictability and information. Psychological Bulletin 111 (2), 291–303.

Devaney, M., Ram, A., 1997. Efficient feature selection in conceptual clustering. Proceedings of International Conference on Machine Learning, 92–97.

Dittenbach, M., Rauber, A., Merkl, D., 2002. Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. Neurocomputing 48, 199–216.

Fisher, D. H., 1987. Knowledge acquisition via incremental conceptual clustering. Machine Learning 2, 139–172.

Gennari, J. H., Langley, P., Fisher, D., 1989. Models of incremental concept formation. Artificial Intelligence 40 (1-3), 11–61.

Godoy, D., Amandi, A., 2006. Modeling user interests by conceptual clustering. Information Systems 31, 247–265.

Huang, Z., Ng, M. K., 1999. A fuzzy k-modes algorithm for clustering categorical data. IEEE Transaction on Fuzzy Systems 7.

Ionnidis, Y. E., Saulys, T., Whitsitt, A. J., 1992. Conceptual learning in database design. ACM Transactions on Information Systems 10 (3), 265–293.

Kohonen, T., 2001. Self-Organizing Maps, 3rd Edition. Springer-Verlag, Berlin; New York.

Li, C., Biswas, G., 2002. Unsupervised learning with mixed numeric and nominal data. IEEE Transaction on Knowledge and Data Engineering 14 (4), 673–690.

Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (Eds.), 1983. Machine Learning: An Artificial Intelligence Approach. Vol. I. Tioga Publishing Company, Palo Alto, California.

Palomo, E. J., North, J., Elizondo, D., Luque, R. M., Watoson, T., 2012. Application of growing hierarchical SOM for visualisation of network forensics. Neural Networks 32, 275–284.

Rauber, A., 2007. GHSOM - the growing hierarchical self-organizing map - homepage.
URL `http://www.ifs.tuwien.ac.at/~andi/ghsom/`

Rauber, A., Merkl, D., Dittenbach, M., 2002. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transaction on Neural Networks 13, 1331–1341.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., Boyes-Braem, P., 1976. Basic objects in natural categories. Cognitive Psychology 8, 382–439.

Scanlan, J., Hartnett, J., Williams, R., 2006. DynamicWEB: Profile correlation using COBWEB. Lecture Notes in Computer Science 4304, 1059–1063.

Scanlan, J., Hartnett, J., Williams, R., 2008. DynamicWEB: Adapting to concept drift and object drift in COBWEB. Lecture Notes in Computer Science 5360, 454–460.

Theodorakis, M., Vlachos, A., Kalamboukis, T. Z., 2004. Using hierarchical clustering to enhance classification accuracy. In: Proceedings of 3rd Hellenic Conference on Artificial Intelligence.

Tversky, B., Hemenway, K., 1984. Objects, parts, and categories. Journal of Experimental Psychology: General 113 (2), 169–197.

Ungerer, F., Schmid, H.-J., 1996. An Introduction to Cognitive Linguistics. Addison Wesley Longman Limited, New York.

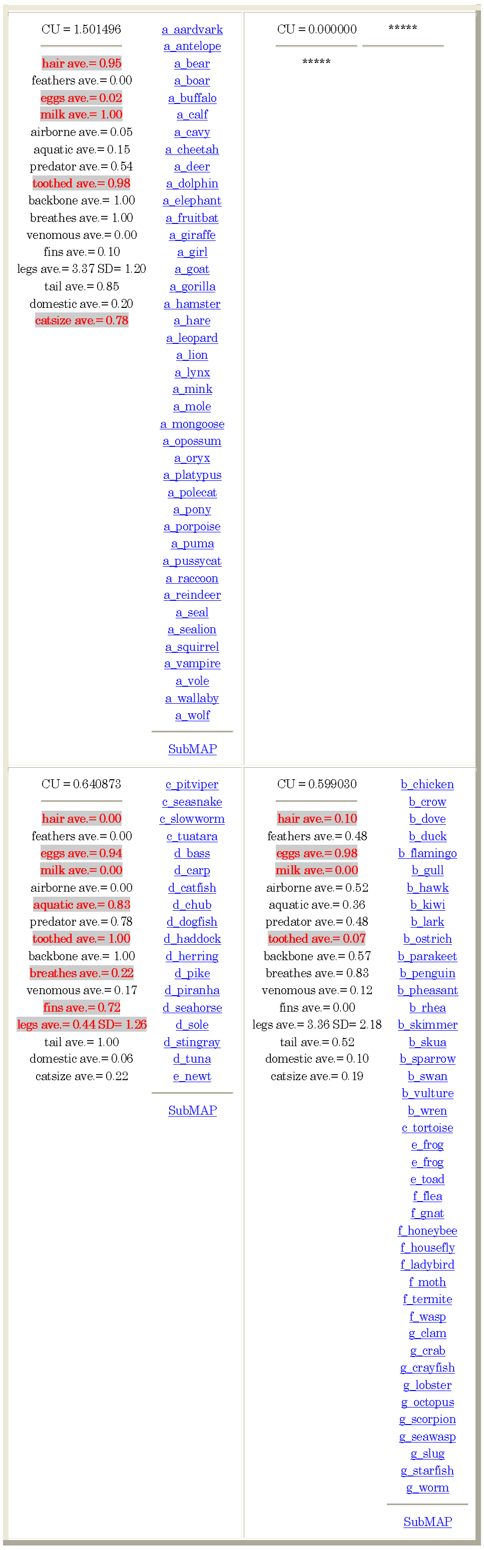

Fig. 4. Example of final clustering results generated by CUU-GHSOM: map in first layer.

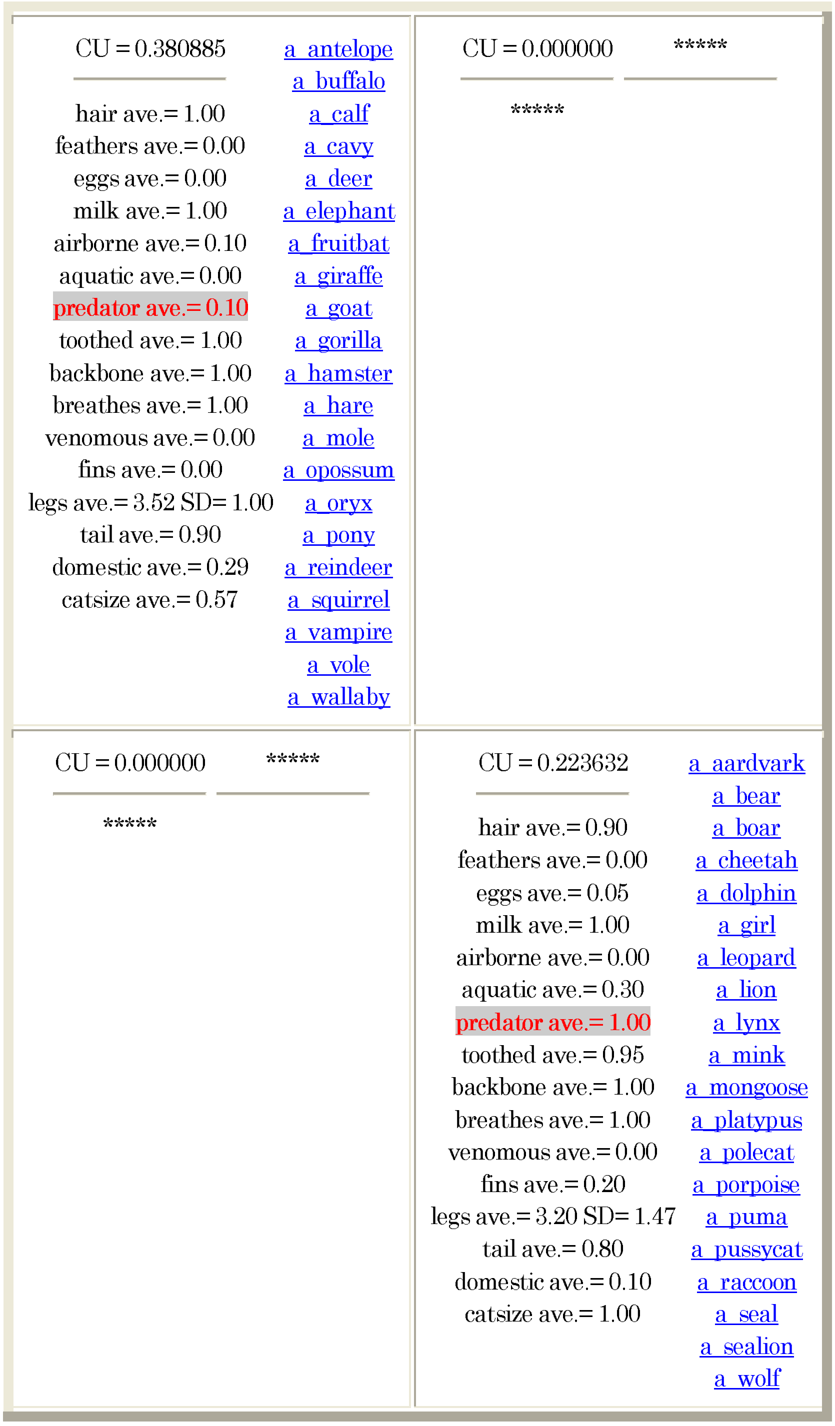

Fig. 5. Example of final clustering results generated by CUU-GHSOM: map in second layer for parent unit (0,0) in first layer (Fig. 4).

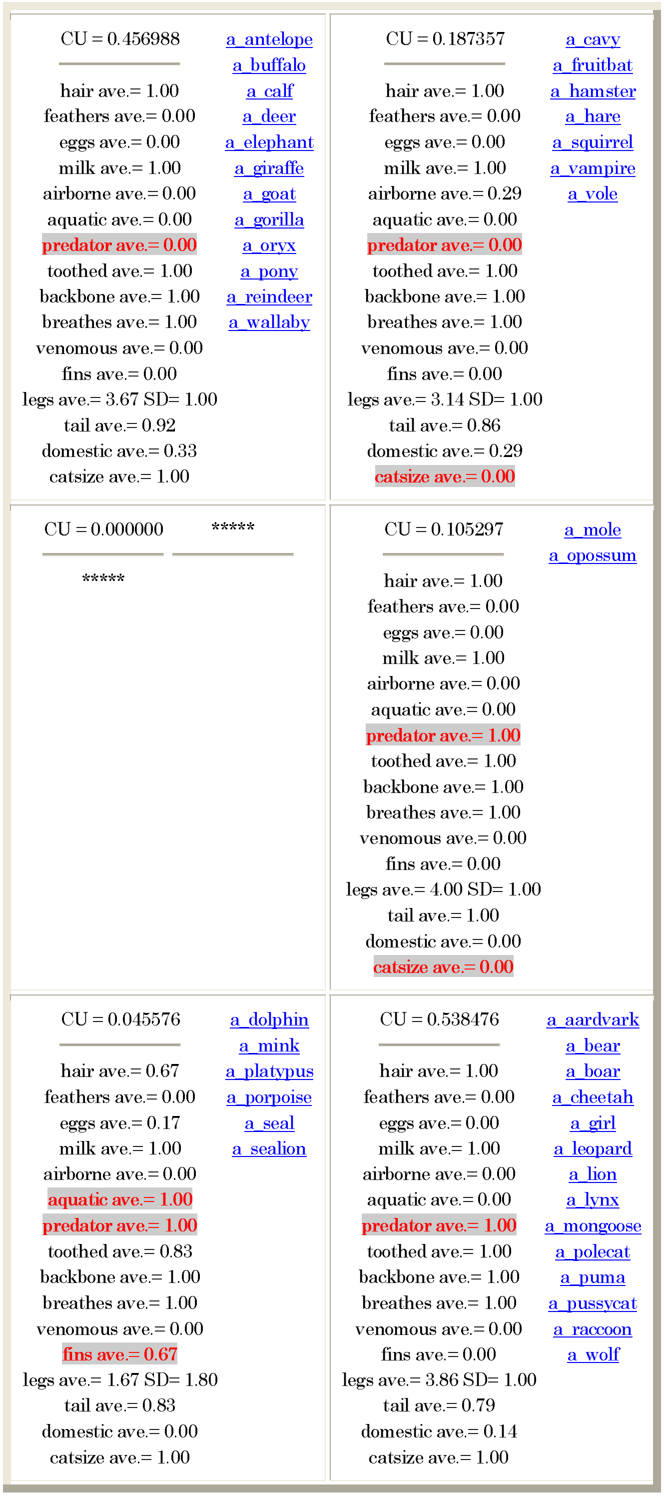

Fig. 6. Example of final result generated by CU-GHSOM (using CU without unification process): map in second layer for parent unit (0,0) in first layer (not shown but similar to that in Fig. 4).

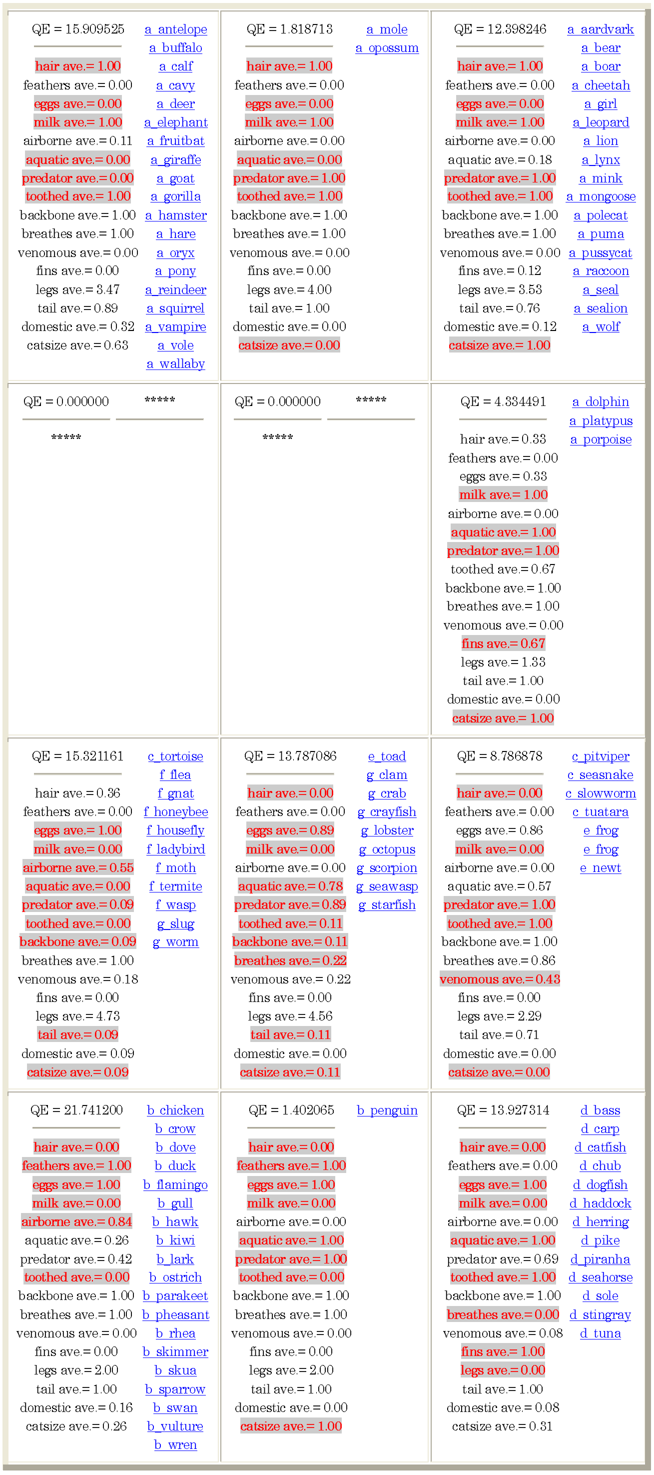

Fig. 7. Example of final result generated by QE-GHSOM (using quantization error): the map in first layer.