Improved Learning Vector Quantization for Mixed-Type Data

Hung-Yi Tsai
Department of Information Management National Yunlin University of Science and Technology
g9923761@yuntech.edu.tw

Chung-Chian Hsu
Department of Information Management National Yunlin University of Science and Technology
hsucc@yuntech.edu.tw

Jiang-Shan Wang
Department of Information Management National Yunlin University of Science and Technology
g9823703@yuntech.edu.tw

ABSTRACT

With rapid growth of information technology, most of corporations have collected a large amount of digital data, such as data regarding employees, customers and transactions, etc. Thus, mining useful patterns from the data becomes an important issue. Learning Vector Quantization (LVQ) is a prototype-based classification technique and can process a large volume of data within reasonable computation time. However, traditional LVQ process only numeric data due to the use of Euclidean distance but cannot directly handle categorical data which must be converted in advance, by typically using 1-of-*k* method. Nevertheless, after the conversion, categorical data lose their semantic information, leading to reduced classification performance. In this work, we propose an Improved LVQ (ILVQ) to deal with mixed-type data by using distance hierarchy for expressing relationship between categorical data. Experimental results prove ILVQ is better than traditional LVQ in classifying mixed-type data.

Keyword: Learning vector quantization (LVQ), Distance hierarchy, Mixed-type data, Categorical data and Classification.

Improved Learning Vector Quantization for Mixed-Type Data

I. INTRODUCTION

Nowadays, with rapid growth of information technology and e-commerce, business companies and organizations have collected a large amount of data, such as data regarding employees, customers and transactions, etc. How to transform these data into useful information is an issue worthy of exploration. Classification, which is one of commonly used technologies in data mining, can predict unknown data and support enterprise to make decisions based on the result. Classification is particularly useful for enterprises in data analysis and target marketing.

Learning Vector Quantization (LVQ) is a prototype-based artificial neural network (ANN). LVQ is a classifier and can process a large amount of data quickly. However, traditional LVQ cannot process categorical data directly. Some preprocess must convert categorical values to numeric ones prior to training. A typical conversion method is 1-of-*k* coding. After the coding, the semantics inherent in categorical values are lost and the data cannot keep its original structure, leading to increased classification errors.

In this study, we propose an improved LVQ (ILVQ), integrating distance hierarchy which can express the relationship between the categorical values. Moreover, distance hierarchy can be automatically constructed so that the need of human experts can be avoided. We investigate whether ILVQ can improve classification performance.

This paper is organized as follows. In Section 2, we review LVQ and distance hierarchy. In Section 3, we propose an improved LVQ model which can consider distance hierarchy and deal with mixed-type data. In Section 4, we present the experimental results on real datasets to demonstrate that our method is able to improve classified accuracy. Finally, we describe conclusions of this study.

II. LITERATURE REVIEW

To establish background knowledge related to our ILVQ, traditional LVQ and distance hierarchy are reviewed in this section.

2.1 Learning Vector Quantization

Learning Vector Quantization (LVQ) is a prototype-based supervised algorithm proposed by Kohonen for classification. It builds up vector quantization with all data samples at the training step. LVQ is a special case of ANNs and has the training and adjustment process similar to SOM. The structure of LVQ network has three layers: an input layer, a Kohonen classification layer and an output layer (Bezdek et al. 1995; Kohonen 1988; Kohonen et al. 1995). As shown in following Figure 1.

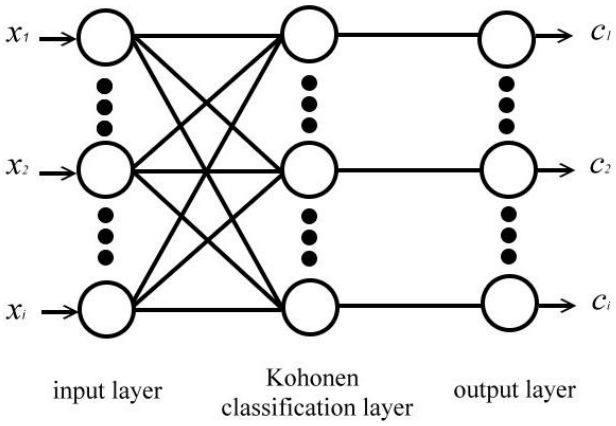


Figure 1. Network architecture of LVQ

The data space of LVQ can be regarded as a Voronoi diagram as shown in Figure 2. In Voronoi diagram, there are several prototypes and each prototype has exclusive non-overlapping decision region with the others. When input data *x* locates in prototype *a*'s decision region, *x*'s class label should be the same as *a*'s class label. Therefore, through the training phase LVQ will construct a favorable Voronoi diagram model so that LVQ classifies unknown input data well in the testing phase.

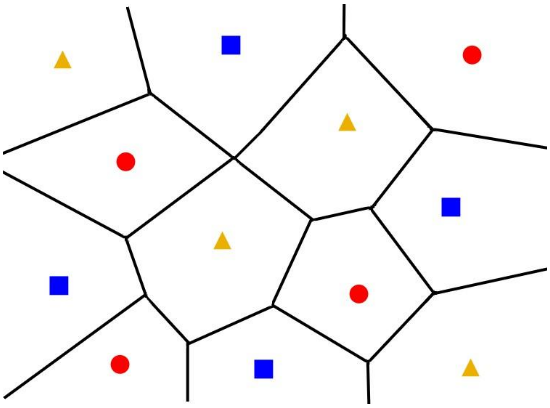


Figure 2. Voronoi diagram

At first of the training, LVQ neural network is defined by a set of prototypes referred to as codebook vectors, which are randomly drawn from the training data instances. Then for each data sample $x(t)$ in discrete time step *t*, the closest codebook vector $m_c(t)$ is selected by identifying the shortest distance between $m_i(t)$ and $x(t)$ according to some distance function, such as Euclidean distance (Bezdek et al. 1995; Kohonen et al. 1995). Formally, the process of the closest vector identification is defined by Eq. (1).

$$C = \arg\min_i \{\|x(t) - m_i(t)\|\} \quad (1)$$

Second, the winner codebook vector gets the update as follows. If $x(t)$ and $m_c(t)$ have the identical class label, then m_c is pushed closer to $x(t)$ by some amount proportional the distance between $x(t)$ and $m_c(t)$; otherwise m_c is pulled away from $x(t)$ by a similar amount. The updating functions are defined as follows.

$$m_c(t+1) = m_c + \alpha(t)[x(t) - m_c(t)]$$

If x and m_c have the same class,

$$m_c(t+1) = m_c - \alpha(t)[x(t) - m_c(t)] \quad (2)$$

If x and m_c have different classes,

$$m_i(t+1) = m_i(t) \text{ for } i \neq c$$

where $\alpha(t)$ is the monotonically decreasing learning rate with t, and the range is $0<\alpha(t)<1$.

2.2 Distance Hierarchy

Distance Hierarchy (DH) can be used to express similarity between values which can be categorical, ordinal or numeric (Hsu et al. 2012; Hsu et al. 2011; Hsu 2006). DH has a tree-like hierarchical structure consisting of nodes, links, and weights. Each node and link represents a concept and the hierarchical relationship between two different concepts, respectively. Each link is associated with a weight representing a distance. The distance between the root and a node expresses similarity extent between the concept and the root. The more distant the more dissimilar. For categorical data, the link weights can be assigned according to some hierarchal clustering algorithm automatically (Das et al. 1998; Palmer et al. 2003) or assigned according to expert's domain knowledge manually. The structures of DH for categorical, ordinal and numeric attributes are depicted in Figure 3.

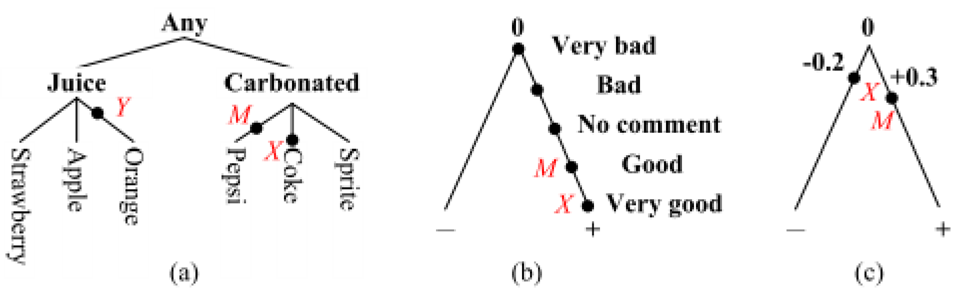


Figure 3. Distance hierarchies for categorical, ordinal and numeric data.

As shown in Figure 4, a point can be placed at any position in a distance hierarchy. Formally, a point X is described by an anchor and a positive offset, denoted as $X = (N_X, d_X)$, representing a leaf node and the distance from the root to X, respectively. X is an ancestor of another point Y if X is in the path from Y to the root. If they are at the same position, X and Y are equivalent, i.e., $X \equiv Y$. The lowest common ancestor of two points represents the most specific common ancestor node of X and Y, denoted as $LCA(X,Y)$. In Figure 4, LCA of X and M is Carbonated, and LCA of Y and M is the root Any. The lowest common point $LCP(P,Q)$ of two points P and Q is defined as following Eq. (3).

$$LCP(P,Q) = \begin{cases} P, \text{if } P \equiv Q \\ P, \text{if } P \text{ is an ancestor of } Q \\ Q, \text{if } Q \text{ is an ancestor of } P \\ LCA(P,Q), \text{otherwise} \end{cases} \quad (3)$$

The distance between two points in a distance hierarchy is defined by the weight between the two points; the distance formula can be expressed as Eq. (4)

$$|P - Q| = d_P + d_Q - 2d_{LCP(P,Q)} \quad (4)$$

where d_P, d_Q and $d_{LCP(P,Q)}$ refer to the distance from P, Q, and $LCP(P,Q)$ to the root, respectively.

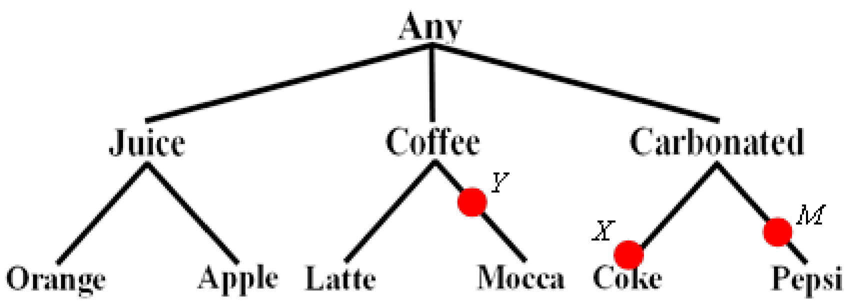


Figure 4. A categorical distance hierarchy for the categorical attribute Drink. *X* and *M* represent the mapping points of the Drink attribute of a training pattern *x* and an ILVQ prototype *m*.

For the example in Figure 4, assume that $X = (Coke, 2.0)$, $Y = (Mocca, 1.2)$ and $M = (Pepsi, 1.6)$. LCA of Y and M is the root Any, i.e., $LCA(Y,M)$ = Any. LCA between

X and M is Carbonated, i.e., $LCA(X, M) = \text{Carbonated}$. LCP of X and M is also Carbonated since neither X or M is an ancestor of the other. The distance between X and M is $|(Coke, 2.0) - (Pepsi, 1.6)| = 2.0 + 1.6 - 2 \times 1.0 = 1.6$ while the distance between X and Y is $|(Coke, 2.0) - (Mocca, 1.2)| = 2.0 + 1.2 - 2 \times 0 = 3.3$. As shown in the example, the semantic similarity inherent in categorical values can be reflected by the scheme of distance hierarchy.

III. IMPROVED LEARNING VECTOR QUANTIZATION

3.1 Improved Learning Vector Quantization

In this section, the improved learning vector quantization (ILVQ) is presented. The main idea of LVQ is to drag closer the prototype of the best matching codebook vector to its training datum if the prototype has the same class label with the datum and to push away the prototype if it has a different class label from the datum. ILVQ is an LVQ extended with distance hierarchy to address the handle of categorical data. Therefore, ILVQ can process mixed-type and categorical datasets well. The procedure of ILVQ training algorithm is as follows.

At the training step, the first step is to find the winner codebook vector. In order to find the winner, ILVQ requires a formula for calculating the distance between the input data and each codebook vector. DH can measure the distance in a unified manner in various type of values including categorical, ordinal and numeric values. The distance between input data $x(t)$ and codebook vector $m_i(t)$ is defined by Eq. (5).

$$\|x(t) - m_i(t)\| = \sqrt{\sum_{j=1}^{k} \left(dh_j(x_j(t)) - dh_j(m_{i_j}(t))\right)^2} \tag{5}$$

where k is the number of attributes, j represents the jth attribute, and $dh_j(\bullet)$ represents the mapping of its argument to its associated distance hierarchy.

At the adjustment step, the concept of distance measure is the same with Eq. (5) and the mapping points of the prototype are moved closer to or away from the mapping points of the input. In fact, for adjusting numeric values, we only need to adjust the value by plus or minus the adjustment amount δ. For adjusting categorical values, we must adjust the value and consider their new position in the hierarchy. In some cases, the anchor of the point needs to be changed after the adjustment. In the following section, we will introduce the adjustment of a point in a categorical distance hierarchy by referring to Figure 5 and 6. Suppose X and M represent the mapping points of two categorical values from a training data sample $x(t)$ and ILVQ winner prototype $m_c(t)$, respectively.

ILVQ is based on LVQ, and thus ILVQ will encounter two situations in adjusting the codebook vector. In the first situation, assume the winner codebook vector *M* has the same class label with data sample *X*, *M* moves towards *X*. In this situation, all possible adjustments are described in Figure 5. As shown in Figure 5(a) and (b), *M* moves towards *X*. The offset of *M* pluses the adjustment amount δ and the anchor does not change. In the case Figure 5(c), the offset of M minus the adjustment amount δ and the anchor does not change. In the case Figure 5(d), the offset of *M* must be calculated again because the anchor of *M* will be changed; the newest offset value is $2d_{LCA(P,Q)} - d_M + \delta$ and the anchor changes from *Q* to *P*.

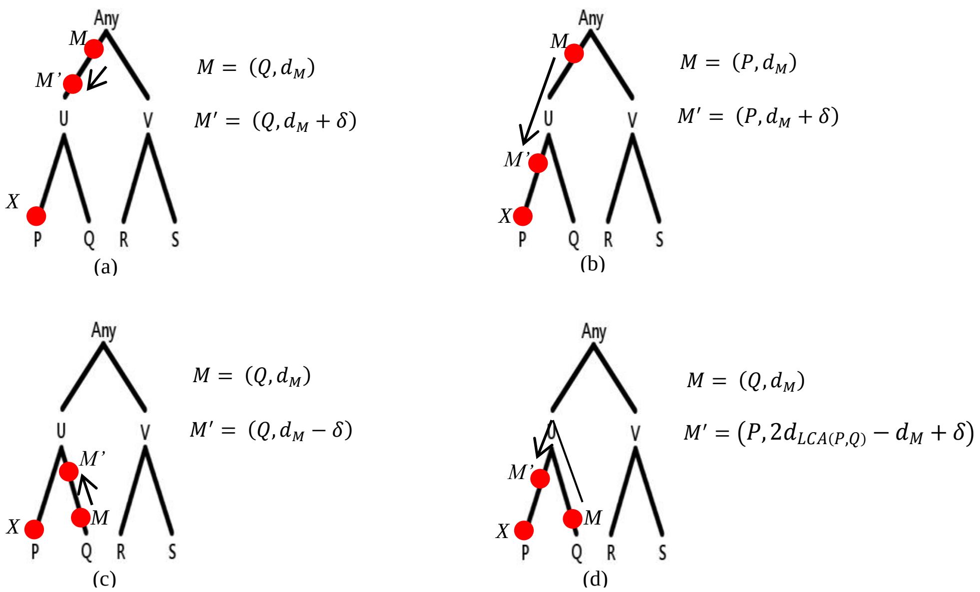


Figure 5. All possible adjustment situations for prototype *M* moving towards data sample *X*.

In the second situation, assume the winner codebook vector *M* has the different class label with data sample *X*, and *M* moves away *X*. In this situation, all possible adjustments are described in Figure 6. In the case of Figure 6(a), *M* moves away from *X* and the offset of *M* pluses the adjustment amount becomes the new position of *M*'. In Figure 6(b), *M* moves away in different direction with *M* in Figure 6(a) but the anchor does not change. In Figure 6(c), *M* crosses the root node. Not only does the offset need to be calculated again but also the anchor of *M* will be changed; the new offset value is $-(d_M - \delta)$ and the anchor changes from *Q* to non-*Q*.

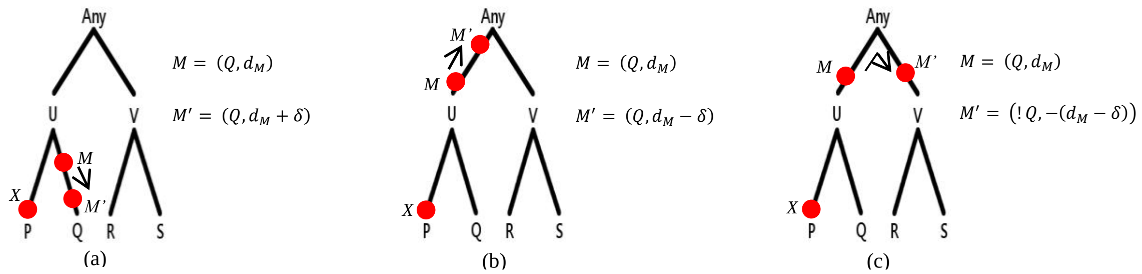


Figure 6. All possible adjustment situations for prototype *M* moving away data sample *X*.

For example, Figure 7 shows the adjustment of the winner codebook vector $m(t)$ with respect to data sample $x(t)$ in a distance hierarchy. Assume that $X = (Coke, 2.0)$, $M = (Pepsi, 1.2)$ and $\delta = 0.25$. If *X* and *M* have the same class label, *M* must move towards to *X*. Therefore, the new value is $M = (Coke, 1.05)$ that calculated by referred Figure 4(d), and the process of adjustment can refer to Figure 7(a). On the other hand, if *X* and *M* have the different class label, *M* must move away from *X*, and the process of adjustment can be referred in Figure 7(b). The new value is $M = (Pepsi, 1.45)$.

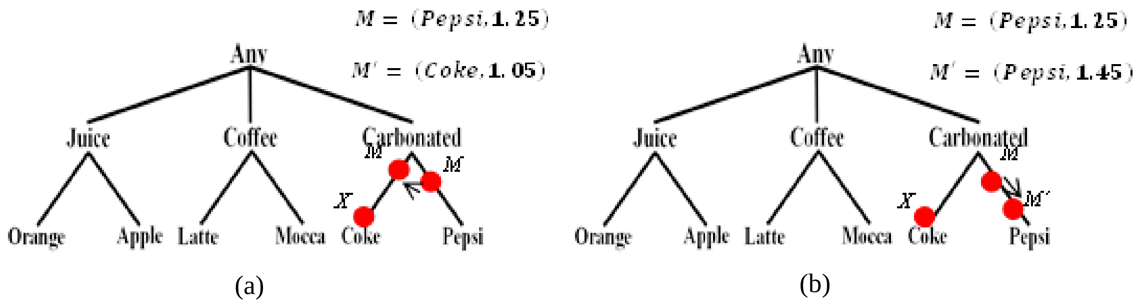


Figure 7. Two distance hierarchies present the example for adjustment calculation.

3.2 Construct Distance Hierarchy

Distance hierarchy can be constructed by two approaches. The first is to use domain knowledge and to construct manually by experts. However, in some applications, categorical values neither have existing domain knowledge nor have experts. It is also possible that the values of categorical attributes are encrypted due to privacy consideration. Thus, the second approach is proposed by using existing information in the data to construct distance hierarchy automatically.

For automatically constructing distance hierarchy, we use a dissimilarity index between a pair of categorical values which measures the relationship between the pair and an external probe. If two categorical values have the same extent of co-occurrence with the external probe, the values are deemed similar(Das et al. 1998). Accordingly, we define the dissimilarity between two categorical values, A and B, in a feature attribute with respect to the set of labels in class attribute P as Eq. (10).

$$d(A,B) = \sum_{D \in P} |conf(A \Rightarrow D) - conf(B \Rightarrow D)| \quad (10)$$

where conf(A⇒D) denotes the ratio of co-occurrence of A and D.

According to the above dissimilarity definition, we can measure the similarity degree between any pair of categorical values in the categorical attribute. The similarity degree of all pairs can form a proximity matrix. Then, the distance hierarchy can be constructed by using hierarchical clustering to cluster the distance matrix. The procedure of constructing a distance hierarchy is depicted in Figure 8.

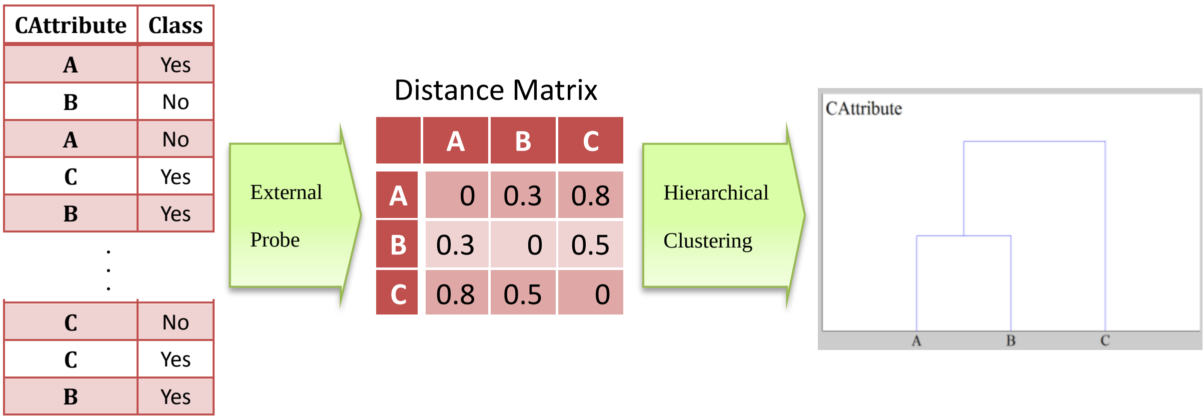


Figure 8. The procedure of constructing distance hierarchy from the dataset.

In this study, we use the popular agglomerative hierarchical clustering algorithm with single-linkage, compete-linkage and average-linkage, respectively, to construct various type of distance hierarchies. Experimental results will show three different performances for ILVQ with different types of distance hierarchies. The results will be described in next section.

IV. EXPERIMENTS

We compare traditional LVQ and ILVQ on three real datasets (Adult, Nursery and Car evaluation) from UCI repository. In our experiments, we use two testing methods, 10-fold cross validation and 70%-30% validation. The 10-fold cross validation which splits the dataset to ten parts with nine parts as training data and one part as testing data in turn, and executes ten rounds. That is, in each round, it will change content of training data and testing data. The 70%-30% validation uses 70% of the dataset as training data and 30% as testing data. We used 10-fold cross validation for the small dataset Car evaluation, and 70%-30% validation for the large datasets Adult and Nursery. Information about the datasets is summarized in Table 2.

Table 1. The experimental UCI datasets.

Data Set	Samples	Dimension	Numeric attr.	Categorical attr.	Classes
Adult	48,842	22	19	3	2
Nursery	12,960	9	0	9	5
Car evaluation	1,728	7	0	7	4

The experimental setting is noted in Table 3. LVQ has three parameters, the amount of codebook vector, training times and the learning rate. According to the suggested setting in (Kohonen et al. 1995), we set the learning rate to 0.3 and training to 50 times of the amount of codebook vectors. The amount of codebook vectors were not suggested in Kohonen's paper. Due to consideration of computational time, we decide the maximum amount of codebook vectors should not exceed one-third of the training samples.

Table 2. The experimental setting for the UCI datasets.

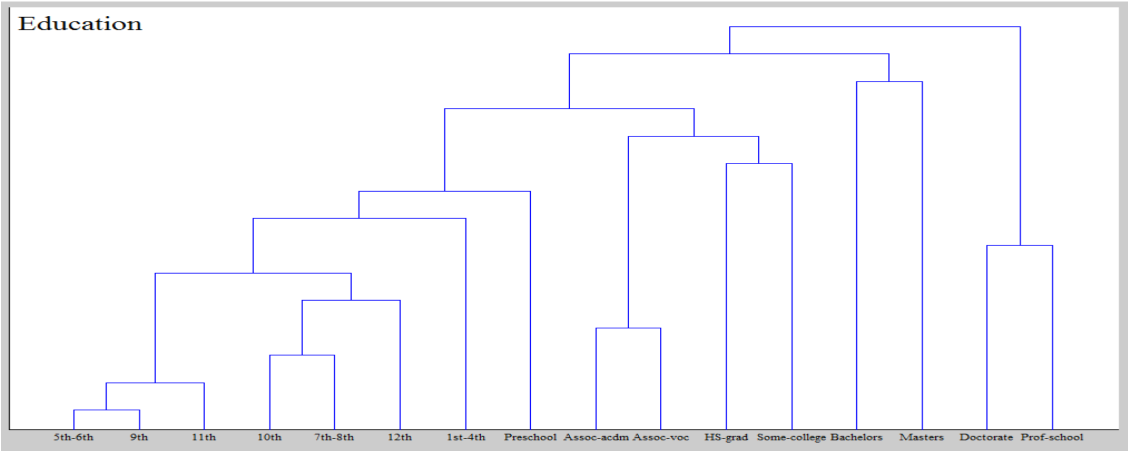
Data Set	Test model	Learning rate	Training samples
Adult	70%-30%	0.3	34,189
Nursery	70%-30%	0.3	9,072
Car evaluation	10-fold	0.3	1,555

For distance hierarchies, we utilize automatically-constructed ones to process categorical data. Due to space limitation, we show just the sample of using single-linkage hierarchical clustering for the three categorical attributes (Education, Marital_status and Relationship) of the Adult dataset. The clustering dendrograms are presented as shown in Figure 8. Distance hierarchies for ordinal and numeric attributes can be easily constructed manually as mentioned in section 2.2.

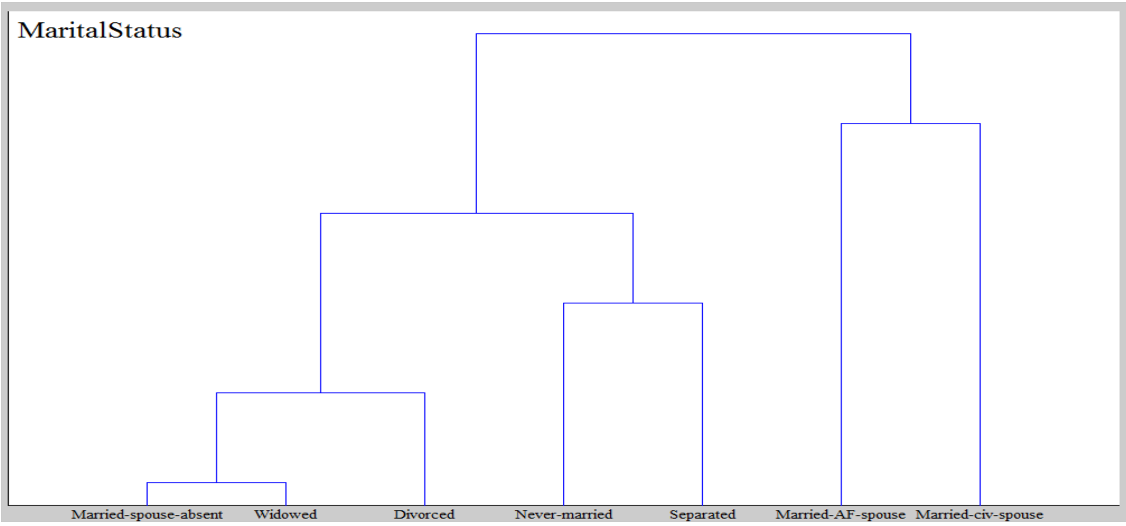
The experimental results for the three datasets listed in Table 4, 5 and 6 demonstrate that ILVQ with distance hierarchy outperforms traditional LVQ with 1-of-*k* coding. The performance of ILVQ with distance hierarchy is enhanced obviously.

As performance of using distance hierarchies constructed by the clustering algorithm with various types of linkages, the ones by average linkage outperformed the ones by single and complete linkage in average.

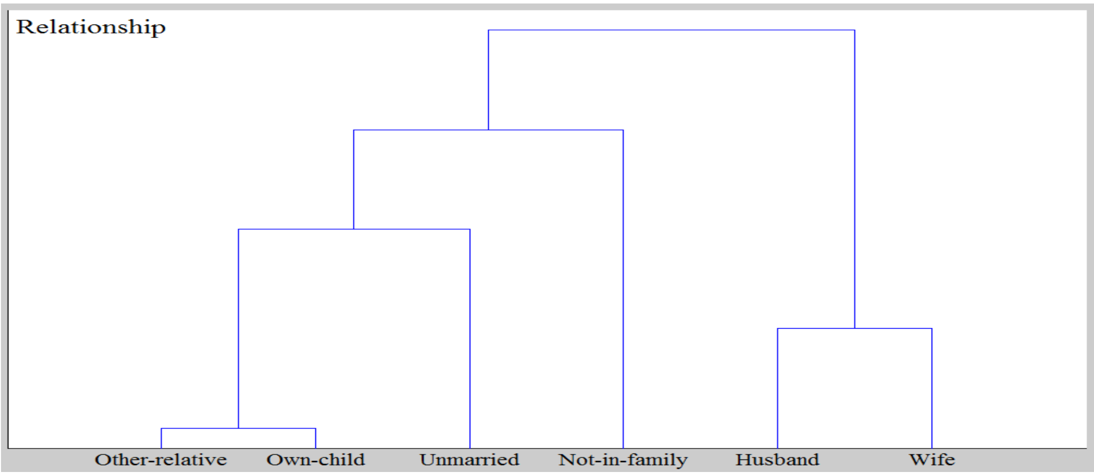
In Nursery and Car Evaluation datasets, the more number of codebook vectors, the better performance. However, in the Adult dataset, classification accuracy did not improve along with the increasing number of codebook vectors. Furthermore, 10-fold cross-validation was used for the small dataset Car Evaluation, and the result shows that the more number of codebook vectors the smaller the standard deviation of the predication accuracy. That is, the classifier was more stable when more codebook vectors were used. Note that the result of using 1-of-*k* coding was obtained by running the LVQ program in Weka which does not give the standard deviation of 10-fold cross validation.



(a)



(b)



(c)

Figure 9. Distance hierarchies constructed by single-linkage hierarchical clustering for (a) Education (b) Relationship and (c) MaritalStatus attributes of dataset Adult.

Table 3. The experimental result on the Adult Dataset.

Model	**# codebook vectors**										**Avg**
	2k	**3k**	**4k**	**5k**	**6k**	**7k**	**8k**	**9k**	**10k**	**11k**	
1-of-*k*	82.54	81.77	81.98	81.44	81.92	81.66	81.16	80.98	80.97	80.98	81.54
single DH	**82.74**	82.37	82.29	**82.46**	81.87	**81.67**	81.68	81.73	**81.75**	81.74	82.03
complete DH	82.27	**82.65**	82.56	81.98	81.60	81.49	81.73	81.29	81.09	81.14	81.78
average DH	82.54	82.46	**82.62**	82.02	**82.52**	81.60	**82.14**	**81.94**	81.68	**82.00**	**82.15**

Table 4. The experimental result on the Nursery Dataset.

Model	**# codebook vectors**										**Avg**
	100	**200**	**300**	**400**	**500**	**1k**	**1.5k**	**2k**	**2.5k**	**3k**	
1-of-*k*	88.04	**89.12**	**89.66**	88.63	87.24	88.30	88.14	88.14	86.75	86.27	88.03
single DH	**88.25**	88.48	88.81	88.30	87.81	88.91	90.35	90.61	90.79	91.49	89.38
complete DH	86.63	87.83	89.53	**89.35**	89.17	91.59	92.88	93.26	**94.19**	**95.04**	90.95
average DH	87.27	88.37	87.94	89.15	**89.81**	**91.82**	**93.42**	**93.93**	94.16	94.26	**91.01**

Table 5. The experimental result on the Car Evaluation Dataset.

Model	**# codebook vectors**										**Avg**
	10	**30**	**50**	**100**	**150**	**200**	**250**	**300**	**400**	**500**	
1-of-*k*	73.96	82.58	85.01	88.60	88.95	89.53	88.77	89.35	88.95	86.69	86.24
single DH	76.60 (0.79)	86.70 (0.91)	88.43 (0.70)	92.09 (0.44)	93.48 (0.42)	93.79 (0.66)	94.42 (0.39)	94.47 (0.27)	94.71 (0.49)	95.06 (0.56)	90.98 (0.56)
complete DH	78.95 (2.23)	88.32 (0.31)	**90.12** (0.74)	**93.52** (0.59)	93.99 (0.64)	94.52 (0.28)	94.80 (0.38)	95.40 (0.30)	95.75 (0.41)	96.15 (0.33)	92.15 (0.62)
average DH	**83.11** (2.21)	**88.41** (1.01)	88.56 (0.53)	91.58 (0.59)	**94.21** (0.44)	**95.36** (0.27)	**95.59** (0.41)	**96.06** (0.71)	**96.36** (0.36)	**96.36** (0.51)	**92.56** (0.71)

V. CONCLUSION

In this study, we have proposed an improved LVQ with distance hierarchy to deal with mixed-type dataset. The improved LVQ shows better performance than traditional LVQ, which uses 1-of-*k* coding for transforming categorical values. Since distance hierarchy considers semantics inherent in categorical values, the similarity relationship between categorical values can be preserved better than by 1-of-*k* coding. In other words, distance hierarchy results in less information loss than 1-of-*k* coding, and thus our model yields better outcome than traditional LVQ.

In addition, we have utilized automatically-constructed distance hierarchy to replace the manually constructed method. Since constructing distance hierarchy manually is time-consuming and infeasible for datasets with encrypted attributes, an approach to automatically constructing distance hierarchy is necessary. Thus, in our work, we proposed an approach to automatically constructing distance hierarchies for categorical attributes. Experimental results prove the improved LVQ can handle the mix-type data better than traditional LVQ.

In the future, we will construct and compare with the other versions of LVQ, such as

LVQ2.1 and LVQ3.

Acknowledgements

This work is supported by the National Science Council, Taiwan, under grant NSC 100-2410-H-224-003-MY2.

REFERENCES

1. Bezdek, J. C., and Pal, N. R. "Two soft relatives of learning vector quantization," *Neural Netw.* (8:5) 1995, pp 729-743.
2. Das, G., and Ronkainen, H. M. a. P. "Similarity of Attributes by External Probes," in: *American Association for Artificial Intelligence*, 1998.
3. Hsu, C.-C., and Lin, S.-H. "Visualized Analysis of Mixed Numeric and Categorical Data Via Extended Self-Organizing Map," *Neural Networks and Learning Systems, IEEE Transactions on* (23:1) 2012, pp 72-86.
4. Hsu, C.-C., Lin, S.-H., and Tai, W.-S. "Apply extended self-organizing map to cluster and classify mixed-type data," *Neurocomputing*) 2011, p 11.
5. Hsu, C. C. "Generalizing self-organizing map for categorical data," *IEEE transactions on Neural Networks* (17:2) 2006, pp 294-304.
6. Kohonen, T. *Self-organization and associative memory*, (2nd ed.) Springer-Verlag, Berlin ; New York, 1988, pp. xv, 312 p.
7. Kohonen, T., Kangas, J., Laaksonen, J., and Torkkola, K. "LVQPAK: A software package for the correct application of Learning Vector Quantization algorithms," Int'l. Joint Conf. on Neural Networks, 1995, pp. 725-730.
8. Palmer, C. R., and Faloutsos, C. "Electricity based external similarity of categorical attributes," *Lecture notes in computer science*) 2003, pp 486-500.

改良式向量量化分析混合型資料

蔡宏益
國立雲林科技大學資訊管理學系
g9923761@yuntech.edu.tw

許中川
國立雲林科技大學資訊管理學系
hsucc@yuntech.edu.tw

王江山
國立雲林科技大學資訊管理學系
g9823703@yuntech.edu.tw

摘要

隨著資訊科技的成長，大部份的公司擁有大量的數位資料，例如：員工、客戶和交易資料…等等。因此，從這些資料探勘出有用的模型變成一個很重要的議題。學習向量量化(LVQ)是一個以代表點為基礎的分類技術並且可以在合理的時間內處理大量的資料。然而，由於傳統的學習向量量化是利用歐基理德距離所以僅能處理數值型資料而無法直接處理種類型資料，種類型資料必須進階使用 1-of-k 的方法做轉換才能處理。不過經由 1-of-k 轉換後，種類型資料會遺失原有的語義資訊並且導致分類效率下降。本篇研究，我們設計一個改良式的學習向量量化藉由利用可以表達種類型資料之間關係的距離階層去處理混合型的資料。實驗的結果證明改良式的學習向量量化在分類混合型資料可以優於傳統型的學習向量量化。

關鍵詞：學習向量量化、距離階層、混合型資料、種類型資料、分類。