

Vector Quantization of Microarray Gene Expression Data

Dr. T.V. Prasad, Ms. Maitrei Kohli, *Member, IAENG*

Abstract—A series of experiments conducted on various gene expression datasets revealed that the learning vector quantization (LVQ) produced better grouping of genes compared to other known efficient techniques such as self-organizing maps. The LVQ algorithms exhibited consistency and better accuracy compared to other clustering techniques such as SOM, HC, k-means, etc

Index Terms— Data mining, microarray gene expression data, artificial neural networks, vector quantization, clustering, classification.

I. INTRODUCTION

The data mining methods are used to find human-interpretable patterns that describe the data, for example, clustering, associations and classification. Techniques drawn from various other fields such as artificial intelligence, pattern recognition, statistics, database management systems and information visualization together provide efficient methods to mine the volumes. The focus of any clustering or classification technique is to calculate the accuracy of the concerned algorithm and to determine its learn ability. The basic motivation behind the use of LVQ for analysis of gene expression data lies in the fact that LVQ has been used as a tool to minimize classification errors, by means of more appropriate discrimination between decision boundaries of classes [1].

LVQ was applied successfully to areas such as audio compression, data compression, data transmission, facial recognition, radar signal processing, finance and insurance, production control, sale and marketing, and so on. Keeping all these issues in view, LVQ could be applied to such simple structured data, with higher confidence than that of SOM. One of the most amazing features of LVQ algorithm is that it can take very few vectors to obtain excellent classification results. The idea behind LVQ is to take away codebook vectors from the decision surfaces to clearly demarcate the class borders. Let m_c be the codebook vector closest to x in the Euclidean metric, applying training vectors x , and updating the $m_i = m_i(t)$ as follows in table 1:

Table 1: Updation Methods of LVQ Algorithms

Method	Updation method	Remarks
LVQ1	$m_c(t+1) = m_c(t) + \alpha(t) [x(t) - m_c(t)],$ <i>if x is classified correctly</i> $m_c(t+1) = m_c(t) - \alpha(t) [x(t) - m_c(t)],$ <i>if x is classified incorrectly</i> $m_i(t+1) = m_i(t), \text{ if } i \neq c$	The factor $\alpha(t)$ is a scalar gain ($0 < \alpha(t) < 1$), which shrinks monotonically in time
LVQ2	$m_i(t+1) = m_i(t) - \alpha(t) [x(t) - m_i(t)],$ $m_j(t+1) = m_j(t) + \alpha(t) [x(t) - m_j(t)]$ <i>if C_i is the nearest class, but x belongs to $C_j \neq C_i$, where C_j is next to nearest class;</i> <i>furthermore x must fall into the "window"</i> $m_k(t+1) = m_k(t), \text{ in all other cases (2)}$	A further improvement over this is the consideration that one of the two vectors belong to the correct class, hence vector x is defined to lie in the "window" if $\min(d_i/d_j, d_j/d_i) > s$ (3) If w is the relative width of the window in its narrowest point then $s = (1-w)/(1+w)$
LVQ3	$m_i(t+1) = m_i(t) - \alpha(t) [x(t) - m_i(t)],$ $m_j(t+1) = m_j(t) + \alpha(t) [x(t) - m_j(t)]$ where m_i and m_j are the two closest codebook vectors to x , and x and m_j belong to the same class, while x and m_i belong to different classes; x must also fall into the "window"; then $\min(d_i/d_j, d_j/d_i) > (1-w)/(1+w)$ (4)	Finally, the results obtained would be: $m_k(t+1) = m_k(t) + \epsilon \alpha(t) [x(t) - m_k(t)],$ for $k \in \{i, j\}$, if x , m_i , and m_j belong to the same class (5)

Manuscript received March 17, 2010.

Dr. T.V. Prasad is with Lingaya's University as the Dean Industrial Consultancy and Associate Dean Academics. (corresponding author - Phone: 0091 9811211515; e-mail: tvprasad2002@yahoo.com).

Ms. Maitrei Kohli is an Assistant Professor with Dept of Computer Science & Engineering, Lingaya's University, Haryana, India. (e-mail: maitreikohli@gmail.com).

II. APPLICATION OF LEARNING VECTOR QUANTIZATION

It is the user who normally attaches a meaning to each column (or sample); for ANN all columns mean same kind of data. The philosophy behind clustering seems to be similar to internal sorting/ rearrangement on selected

column (fields, attributes, samples, or conditions) [2].

Datasets of breast cancer (Hedenfalk et al, 2001), sugarcane, *Mus musculus*, *A. thaliana* (all of NCBI, 2002) and yeast (Eisen et al, 1998) were used for carrying out experiments on the microarray gene expression data using the three variants of LVQ. For all experiments Euclidean measure was taken as the distance metric. Two algorithms of the self-organizing map (SOM) and the three variants of the LVQ algorithm were used for cluster analysis of microarray gene expression data. Over 650 experiments were conducted on different datasets using the application of the five algorithms covering two variants of SOM and three variants of LVQ. The number of clusters/classes, weights of the ANNs and the number of iterations were kept constant at 9, 0.5 and 1000 respectively. The learning rate (LR) was gradually increased from 0.1 to 1.0 and correspondingly the clustering/classification error was computed. For all the datasets, data log transformed, except in the case of *Mus musculus* dataset in which case data pre-processing was applied for zero filling for genes whose expression value was null. Almost none of the data mining algorithms have been very precise in extensive studies of large-scale datasets and genome-wide expression, except that they have been successful in giving “a fair idea” or “a probable match” about the datasets.

Determination of interaction of those genes that exhibit normal or average expression, and which have been omitted in almost all research findings globally, shall remain the largest mystery. Most importantly, these chunks of average expressing genes contribute largely to the overall chain of translation and transcription. Just as a dull or average student in school can exhibit high level of intellectuality at a later stage of life, so also the average expressing genes may participate in the gene regulation process at a later time. The growth and interaction of these genes give an impression that genes too have “individuality” like human [3].

Details listed in table 2, appendix 1. It represents the comparative performance of various LVQ algorithms applied to various datasets.

III. COMPARISON OF LVQ1, LVQ2 AND LVQ3

In order to establish that the performance of LVQ was better than that of other ANNs, it was essential to compare their accuracies. The SOM algorithm was selected for the comparison due to its simplicity, similarity to LVQ and accuracy as known through various well known reports. There are a number of papers highlighting the importance and accuracy of SOM.

Over 650 experiments were conducted on different datasets as described earlier by application of five algorithms covering two variants of SOM and three of LVQ. A number of other experiments with varying parameters were also conducted but were not included in this thesis since change in weight and

number of clusters/classes had no noticeable effect on the output. A marginal change was observed when the number of iterations was increased.

IV. COMPARISON OF ALGORITHMS WITH RESPECT TO THE PARAMETRIC CHANGES

The number of clusters/classes, weights of the ANNs and the number of iterations were kept constant at 9, 0.5 and 1000 respectively. The learning rate (LR) was gradually increased from 0.1 to 1.0 and correspondingly the clustering/classification error was computed. For simplicity, the error value was incorporated on the proximity map visualization form since computation of distances and formation of distance matrix is common to both proximity map as well as clustering quality, see also Prasad and Ahson (2005b). The intention of these experiments was to bring out the accuracy or overall behaviour of the five ANN algorithms, which were plotted in the form of line graphs.

Breast cancer dataset (Data preprocessing – data log transformed): When log transformed dataset of breast cancer was processed, the LVQ1 algorithm produced highly consistent and accurate results compared to the other variants as well as that of SOM. The results were in the range of [91, 92] percent. The LVQ2 and LVQ3 algorithms produced identical outputs and the accuracy kept reducing with increase in LR till it reached 0.7, but the accuracy still remained in the range [86, 91] percent.

Sugarcane dataset (Data preprocessing – data log transformed): As reported, for the sugarcane gene expression dataset, the SOM2D produced the least accurate clustering with accuracy falling in the range [81-91] percent. All other algorithms yielded results beyond 94 percent, which is an excellent outcome than as reported in other works using LVQ. In some cases, it was also found that the LVQ2 and LVQ3 were marginally better than their counterparts. It was for the first time that an accuracy of this order was seen.

Homo sapiens (Data preprocessing – data log transformed): The above results indicate that the SOM1D version resulted in lesser clustering error than the SOM2D version. The LVQ1 algorithm was slightly lesser accurate than the SOM1D algorithm whereas the LVQ2 produced much better results. Of all the algorithms, the LVQ1 produced the best results though slight improvement, by providing proper parameters.

Mus musculus dataset: case I: Data preprocessing precondition – filtered genes on the basis of 90 percent okay genes; data log transformed

Of the five algorithms viz., the SOM1D, SOM2D, LVQ1, LVQ2 and LVQ3, on the *Mus musculus* data that was preprocessed by removing all genes containing most values as null and log transforming, it was observed that the SOM2D resulted in the lowest accuracy, whereas all other algorithms produced much higher accuracies. The LVQ2 and LVQ3 yielded accuracy beyond 94 percent when the LR was kept at 0.9. The LVQ1 algorithm produced highest accuracy for rest of the experiments. It also exhibited consistency throughout

the exercise.

Case II: Data preprocessing precondition – zero filling for genes whose expression value was null; data log transformed

The *Mus musculus* dataset was preprocessed again by filling null values by zeroes and then log transforming the entire dataset. Just as the earlier processing, the LVQ1 again produced the best accuracy of all the five algorithms, except when the LR was in the range [0.2, 0.4]. In this range the accuracy of output fell down as low as 65 percent and in all other cases the accuracy was beyond 85 percent.

Case III: Data preprocessing precondition – zero filling for genes whose expression value was null and duplicate genes merged with row mean; no log transform

In the third experiment, all duplicate genes were merged together in the dataset after filling the null values with zeroes, however, this time the data was not log transformed. The LVQ1 algorithm continued dominating all other techniques producing the least classification error in the range]88, 94[percent, except when the LR was 0.3. At this value of learning rate, the SOM1D performed the best grouping. The SOM2D also resulted in a consistent growth in accuracy.

V. RESULTS

The accuracy was found to be dependent on parameters such as number of clusters and number of rows (genes), but not on number of columns (observations). Dataset (expression values) though not related straightaway to the change in accuracy, influences the overall classification, due to the computation of distances.

Based on the comparison of results produced by SOM and LVQ algorithms on the microarray gene expression datasets, LVQ produced better results than SOM, and out of the three LVQ algorithms LVQ1 was the best. The classification accuracy of LVQ was found to be in the range of 91.8 percent \pm 1.5 percent.

VI. CONCLUSIONS

Application of LVQ to model microarray gene expression

datasets of different organisms through its three variants has been elaborated. Wherever possible, the actual outcome of the GEDAS has been presented through the use of proximity map visualization. The application of LVQ on various datasets proved that it could be preferred over other ANN based techniques for establishing/modeling logical groups in any given dataset. Amongst the three variants, the LVQ1 proved to be the best of all.

Comparison of the accuracy produced by these variants, fine tuning of SOM map using LVQ as well as reasons thereof was also brought out. Wherever the range of gene expression values lie in a narrower range, fine tuning does not bring much enhancement in the accuracy. While fine tuning using the three variants of LVQ, it was found that the LVQ1 improved the accuracy than the other two variants.

REFERENCES

- [1] **Al-Kahnal** and **Al-Hendi**, *Vector Quantization of Arabic Phonemes*, Graduate thesis, King Saud University, 1992
- [2] **Haykin Simon**, *Artificial Neural Networks: A Comprehensive Foundation 2 ed.*, Addison Wesley, 1999
- [3] **Prasad T. V.** and **Ahson S. I.**, *Analysis of Microarray Gene Expression Data*, International Conference on Application of Artificial Intelligence in Engineering and Technology, Universiti Malaysia Sabah, 2004

APPENDIX

TABLE 2: COMPARATIVE EFFICIENCY OF LVQ1 ALGORITHM APPLIED TO DIFFERENT DATASETS

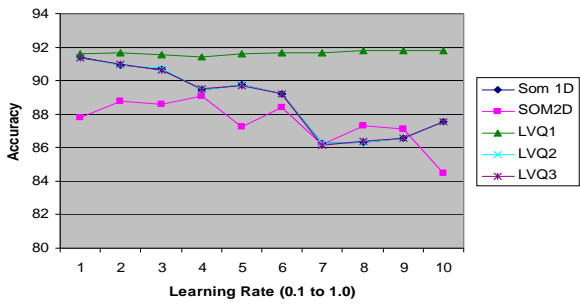
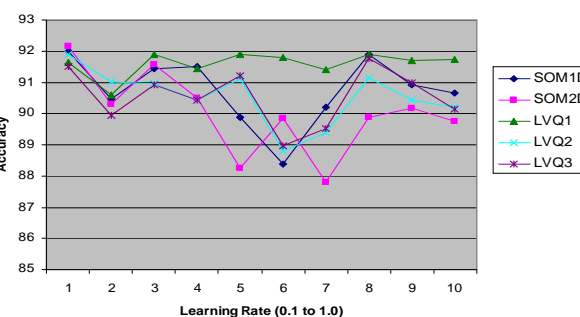
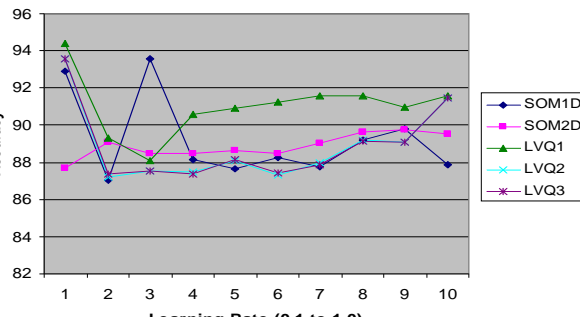
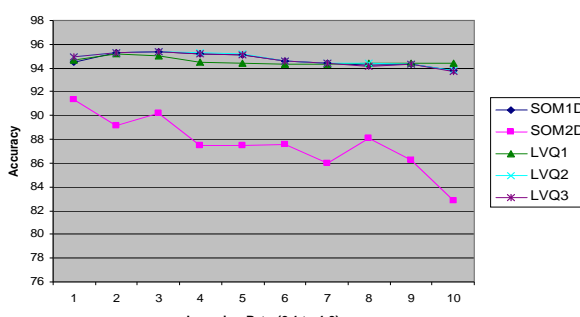
<p><i>Figure 1: Breast cancer dataset - LVQ1 algorithm produced highly consistent and accurate results in the range [91, 92] compared to the other variants as well as that of SOM. For LVQ2 and LVQ3, the accuracy remained in the range [86, 91] percent</i></p>	<p style="text-align: center;">Breast Cancer - Accuracy produced by SOM and LVQ</p> 
<p><i>Figure 2: Mus musculus muscle injury and contraction dataset - The trend of LVQ1 dominating other algorithms continued, as the accuracy was in the range [90.5, 92]</i></p>	<p style="text-align: center;">MIFLC - Accuracy produced by SOM and LVQ</p> 
<p><i>Figure 3: Mus musculus dataset – the data was preprocessed zero filling the values of genes whose expression value was null; further duplicate genes were merged together by suitably substituting with row mean; no log transform was done for the dataset - LVQ1 again produced the best accuracy of all, except when learning rate was in the range [0.2, 0.4]. The accuracy was in the range [88, 94.5]</i></p>	<p style="text-align: center;">Mus musculus - Accuracy produced by SOM and LVQ</p> 
<p><i>Figure 4: Sugarcane dataset - An excellent outcome using LVQ algorithms, with accuracy in the range [94, 95.5]; SOM produced the poorest clustering with accuracy in the range [81-91] percent</i></p>	<p style="text-align: center;">Sugarcane - Accuracy produced by SOM and LVQ</p> 

Figure 5: Homo sapiens dataset - LVQ1 algorithm was very consistent throughout compared to all other applications; accuracy was in the range [90, 91]

