ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 295-300

Batch-RLVQ

Barbara Hammer[1] and Thomas Villmann[2]

(1) University of Osnabrück, Department of Mathematics/Computer Science, Albrechtstraße 28, 49069 Osnabrück, Germany,
e-mail: hammer@informatik.uni-osnabrueck.de

(2) University of Leipzig, Clinic for Psychotherapy and Psychosomatic Medicine, Karl-Tauchnitz-Straße 25, 04107 Leipzig, Germany,
e-mail: villmann@informatik.uni-leipzig.de

Abstract. Recently a variation of learning vector quantization has been proposed in [1], which allows an automatic determination of relevance factors for the input dimensions: relevance learning vector quantization (RLVQ). RLVQ is heuristically motivated and may show instabilities for inappropriate data since it does not obey a gradient dynamics. Here we propose an energy function which describes the dynamics of RLVQ in the stable phase. It can be used to substitute the original dynamics for instable situations. Moreover, it yields to a batch version of RLVQ where hard competition can be substituted by soft clustering. Hence annealing schemes can be applied naturally in order to avoid local minima.

1. Introduction

Kohonen's learning vector quantization (LVQ) provides a prototype based supervised clustering algorithm which has been successfully applied in various different areas such as image classification, control, robotics, or data analysis [11]. There exist modifications which allow an adaptive number of prototypes, an optimized learning rate, or optimized decision borders, to name just a few [11]. The original LVQ algorithm is a very intuitive algorithm based on the metrical structure of the input space; commonly the standard Euclidian metric is used. Hence this metric should represent the internal structure of the data appropriately. In particular, the single data dimensions have to be of approximately the same importance for the clustering and they have to be scaled accordingly. Naturally, several approaches try to overcome this drawback by introducing an adaptive metric. They include methods which learn an appropriate scaling of the input dimensions like DSLVQ [13], RLVQ [1], or GRLVQ [7]; some methods allow a more complex metrical structure like the fuzzy clustering algorithms [4, 6], others allow an adaptive metric with respect to additional information like the unsupervised algorithms in [10, 16].

We will focus on the algorithm RLVQ because it is a very fast algorithm with an intuitive update which has been successfully tested [1]. The algorithm has been modified in [7] to generalized RLVQ (GRLVQ) due to the following reasons: RLVQ is only heuristically motivated and does not obey a gradient dynamics; for some data, it shows instabilities. These drawbacks are overcome by GRLVQ which minimizes a clear objective, however, GRLVQ involves factors which are no longer intuitive. Hence an objective which characterizes important cases of RLVQ and which might indicate in which cases RLVQ will show instabilities would be interesting. We will introduce an energy function which describes the behaviour of RLVQ in stable situations and which is similar to common energy functions for clustering [2]. This energy function can alternatively be solved in a batch mode. The corresponding update can be modified such that it yields soft clusterings instead of crisp decisions. Hence standard annealing schemes as proposed in [5, 14] can be applied to this batch version of RLVQ such that local minima due to a wrong initialization of the codebooks can be avoided. The more complex factors in GRLVQ prohibit the analogous application of the ideas to the energy function of GRLVQ.

2. RLVQ

Assume data $(\vec{x}^i, y^i) \in \mathbb{R}^n \times \{1,\dots,C\}$, $i = 1,\dots,p$, are given and we would like to learn the clustering induced by these training points. In LVQ, the clustering is provided by prototypes $\vec{w}^i \in \mathbb{R}^n$, $i = 1,\dots,c$, which are masked with labels $y(\vec{w}^i) \in \{1,\dots,C\}$. A point $\vec{x} \in \mathbb{R}^n$ is mapped to $y(\vec{w}^i)$ where the distance $|\vec{x} - \vec{w}^i|$ is minimal. Learning algorithms try to find prototypes such that (almost) all training points are mapped to their respective label. LVQ initializes the prototypes with random vectors and iteratively adapts as follows

```
repeat:  choose $(\vec{x}^i, y^i)$
         compute $\vec{w}^j$ such that $|\vec{x}^i - \vec{w}^j|$ is minimal
         $\vec{w}^j := \begin{cases} \vec{w}^j + \epsilon(\vec{x}^i - \vec{w}^j) & \text{if } y^i = y(\vec{w}^j) \\ \vec{w}^j - \epsilon(\vec{x}^i - \vec{w}^j) & \text{otherwise} \end{cases}$
```

where $\epsilon \in (0,1)$ is the so-called learning rate. RLVQ substitutes the standard Euclidian metric $|\vec{x}^i - \vec{w}^j|$ in the above algorithm by the weighted metric

$$|\vec{x}^i - \vec{w}^j|_{\vec{\lambda}} = \left(\sum_k \lambda_k^2 (x_k^i - w_k^j)^2 \right)^{1/2}$$

where $\lambda_k \geq 0$ constitute weighting factors with $\sum_k \lambda_k = 1$. They are adapted by Hebbian learning, *i.e.* the iterative update is accompanied by:

```
for all k:  $\lambda_k = \begin{cases} \lambda_k - \alpha\lambda_k (x_k^i - w_k^j)^2 & \text{if } y^i = y(\vec{w}^j) \\ \lambda_k + \alpha\lambda_k (x_k^i - w_k^j)^2 & \text{otherwise} \end{cases}$
normalize $\vec{\lambda}$
```

where $\alpha > 0$ is the learning rate for the weighting factors. This corresponds to Hebbian learning because precisely those values λ_k are increased which contribute to a correct classification if the normalization is taken into account [1]. In [1] the above update is related to simple Perceptron learning and its difficulties if provided with non separable data [9]. Indeed, RLVQ shows similar critical behaviour if trained for overlapping classes. For this reason a modification of RLVQ has been proposed in [7], GRLVQ, which obeys a stochastic gradient descent on the energy function

$$E_G = \sum_{\vec{x}_i} \operatorname{sgd}\left(\left(D_c^{\vec{\lambda}}(\vec{x}^i) - D_w^{\vec{\lambda}}(\vec{x}^i) \right) / \left(D_c^{\vec{\lambda}}(\vec{x}^i) + D_w^{\vec{\lambda}}(\vec{x}^i) \right) \right)$$

where $D_c^{\vec{\lambda}}(\vec{x}^i)$ is the squared weighted Euclidian distance to the closest correct prototype and $D_w^{\vec{\lambda}}(\vec{x}^i)$ is the squared weighted Euclidian distance to the closest wrong prototype. This method constitutes a very efficient generalization of GLVQ [15] for an adaptive metric. Unfortunately, the resulting formulas involve terms which are more complex than RLVQ and hence less intuitive. Can (approximate) energy functions be established for RLVQ, too?

3. Batch-RLVQ

An energy function for RLVQ would consist up to constant factors of two terms $E_R = E_R^+ + E_R^-$ where

$$E_R^+ = \sum_{\vec{x}^i} \sum_{\vec{w}^j, y^i = y(\vec{w}^j)} \chi_i(j) |\vec{x}^i - \vec{w}^j|^2_{\vec{\lambda}}, \quad E_R^- = -\sum_{\vec{x}^i} \sum_{\vec{w}^j, y^i \neq y(\vec{w}^j)} \chi_i(j) |\vec{x}^i - \vec{w}^j|^2_{\vec{\lambda}}$$

and $\chi_i(j) \in \{0,1\}$ yields 1 iff $\vec{x}^i$ is closest to $\vec{w}^j$ with respect to the weighted Euclidian distance. This energy function is highly discontinuous. Moreover, the minima of E_R^- lie at the borders, hence instabilities occur if the part E_R^- dominates the dynamics. Assumed we train nearly separable data, the part E_R^+ will determine the behaviour in the limit and the result will be stable. Hence we will consider only the part E_R^+ as an approximate energy function for RLVQ. *I.e.* we assume that the closest correct prototype is updated for each data point. The dynamics resulting from minimizing E_R^+ will be different from RLVQ iff data are inseparable and RLVQ would yield instable behaviour.

Commonly, the function E_R^+ has local optima and the result of a stochastic gradient descent critically depends on the initial condition. One can avoid local minima if E_R^+ is minimized via a cooling schedule starting from simple energy landscapes. We substitute E_R^+ by a version including soft competition of the prototypes where the degree of fuzziness is determined by the temperature T:

$$E_R^+(S,T) = \sum_{\vec{x}^i} \left(\sum_{\vec{w}^j, y^i = y(\vec{w}^j)} p_i(j) |\vec{x}^i - \vec{w}^j|^2_{\vec{\lambda}} + T Ent(\vec{x}^i) \right) + S \sum_i \left(\lambda_i - \frac{1}{n} \right)^2 .$$

Here $p_i(j) \in [0,1]$ are new assignment parameters which are optimized under the condition $\sum_{\vec{w}^j, y(\vec{w}^j) = y^i} p_i(j) = 1$. They come from a soft competition of the prototypes masked with y^i. $Ent(\vec{x}^i) = \sum_{\vec{w}^j, y(\vec{w}^j) = y^i} p_i(j) \ln p_i(j)$ is the entropy. The parameters λ_k fulfill $\sum_k \lambda_k = 1$. The last summand determines the influence of these parameters in comparison to standard vector quantization. The magnitude of S regulates to which extend the weighting terms can deviate from the Euclidian setting which is reached for $S \to \infty$. The latter summands in $E_R^+(S,T)$ are minimal if $p_i(j)$ is equal for all j or λ_i is equal for all i, respectively. Hence $E_R(S,T)$ has a unique minimum for $T \to \infty$ where $p_i(j) = 1/|\{\vec{w}^k \mid y(\vec{w}^k) = y(\vec{w}^j)\}|$ for $y(\vec{w}^j) = y^i$. For $T, S \to 0$, we obtain the original energy. The above equation can be minimized by an EM approach, iteratively minimizing with respect to $p_i(j)$ or with respect to $\vec{w}^j$ and λ_k. It follows like in [8, 12] that this procedure converges. The minima of $E_R^+(S,T)$

for fixed $p_i(j)$ or fixed $\vec{w}_j$ and λ_k, respectively, can be explicitely computed using Lagrange multipliers. We obtain for $y(\vec{w}^j) = y^i$

$$p_i(j) = \frac{\exp\left(-|\vec{x}^i - \vec{w}^j|^2_{\vec{\lambda}}/T\right)}{\sum_{\vec{w}^k, y(\vec{w}^k)=y^i} \exp\left(-|\vec{x}^i - \vec{w}^k|^2_{\vec{\lambda}}/T\right)}\,.$$

In addition, we define $p_i(j) := 0$ for $y^i \neq y(\vec{w}^j)$. The minimum with respect to $\vec{w}^j$ yields

$$\vec{w}^j = \frac{\sum_{\vec{x}^i} p_i(j)\vec{x}^i}{\sum_{\vec{x}^i} p_i(j)}\,.$$

Minimizing $E_R^+(S,T)$ with respect to λ_k under the constraint $\sum_k \lambda_k = 1$ yields

$$\lambda_k = \left(\sum_l \frac{S + \sum_{\vec{x}^i,\vec{w}^j} p_i(j)(x^i_k - w^j_k)^2}{S + \sum_{\vec{x}^i,\vec{w}^j} p_i(j)(x^i_l - w^j_l)^2}\right)^{-1}\,.$$

Hence in the limit $S \to 0$, λ_k is maximal for those dimensions where the relative variance of the data points with respect to the codebooks is small. One can now iteratively update the assignments $p_i(j)$ and the prototypes and weights for fixed S and T. The limit $S \to 0$ allows to study the influence of different weightings. In order to obtain global optima, the temperature T can be annealed from high values with a single optimum to $T \to 0$ which corresponds to the original energy. The cooling schedule has to be very careful at phase transitions of the clustering in order to allow the prototypes to separate [5]. One possibility is an exponential schedule with slight increasing of the temperature and slower cooling if a phase transition is observed.

4. Experiments

Usually, batch-RLVQ will not obtain the same classification results as GRLVQ since its objective is a minimization of the overall distance of the data from their nearest prototypes and not optimum classification. However, for many problems these two objectives will be correlated. Moreover, batch-RLVQ can avoid local minima if an appropriate annealing schedule is applied in comparison to GRLVQ which would require other global minimization methods. Note that an analogous batch formulation for GRLVQ would not yield explicit formulas for the values $p_i(j)$, $\vec{w}^j$ and λ_k in each iteration; rather, an additional numerical minimization would be necessary in each step. Of course, GRLVQ could be run starting with the solution provided by batch-RLVQ in order to optimize the results. In the following, we compare the results of batch-RLVQ to RLVQ and GRLVQ for various data sets. In each case the classification error of a randomly selected training and test set is measured. Since they differ by at most 2% in all cases, we only report one value.

Artificial data without overlap: This data set is as in [1]. It comprises three almost separated classes with two clusters each. The intrinsic data dimension is two. However eight dimensions are added to these first two dimensions. Four of them comprise slightly disturbed copies of the first dimension with increasing

	data 1	data 2	iris
LVQ	0.81 - 0.89	0.56 - 0.7	0.94-0.96
RLVQ	0.9 - 0.96	0.79 - 0.86 (instable)	0.95-0.97
GRLVQ	0.93 - 0.97	0.83 - 0.86	0.96-0.98
batch	0.91-0.95	0.7-0.75	0.91-0.97

Table 1: Portion of correctly classified patterns for various data and the clustering provided by the training algorithms, obtained in several runs

noise, the remaining four dimensions comprise pure noise. We train a clustering with two codebooks for each class. See Table 1 for the respective accuracies. Typical weighting vectors are

$$\begin{aligned}\lambda_{\text{RLVQ}} &= (0.13, 0.12, 0.12, 0.11, 0.1, 0.09, 0.1, 0.08, 0.07, 0.06),\\ \lambda_{\text{GRLVQ}} &= (0.49, 0.4, 0.07, 0.02, 0, 0.02, 0, 0, 0, 0),\\ \lambda_{\text{batch}} &= (0.28, 0.39, 0.09, 0.09, 0.04, 0.05, 0.02, 0.03, 0.01, 0).\end{aligned}$$

Hence the separation of batch-RLVQ clearly indicates the importance of the first two dimensions. The ranking of the weighting factors lies between RLVQ and GRLVQ. The classification accuracy of batch-RLVQ is comparable to RLVQ, however, the result is much more stable. For RLVQ, pretraining with LVQ was mandatory for good results, whereas for batch-RLVQ a simultaneous annealing of both parameters, S and T does not lead to instabilities.

Artificial data with overlap: This data set, introduced in [1] again, has the same form as data 1 with the difference that the classes overlap considerably. Typical weighting vectors which result for the algorithms are

$$\begin{aligned}\lambda_{\text{RLVQ}} &= (0.11, 0.12, 0.11, 0.09, 0.09, 0.09, 0.09, 0.09, 0.1, 0.1),\\ \lambda_{\text{GRLVQ}} &= (0.28, 0.38, 0.3, 0.05, 0, 0, 0, 0, 0, 0),\\ \lambda_{\text{batch}} &= (0.35, 0.21, 0.07, 0.09, 0.11, 0.1, 0.01, 0.03, 0.02, 0.02).\end{aligned}$$

The classification result of batch-RLVQ is a bit worse compared to the other methods since it does not optimize the decision borders. However, it is still better than simple LVQ. Moreover, the separation of the first two dimensions as important is clearly indicated by batch-RLVQ. Simple RLVQ showed instabilities in about a quarter of the runs if it was not pretrained with LVQ and obtained a ranking with values $\lambda_1 >> \lambda_k$ for $k \neq 1$. In particular, the informative dimension 2 is neglected. This corresponds to a local optimum of the energy function $E_R^+(S,T)$ which is reached by batch-RLVQ for fast annealing. Then, the two codebooks for classes 2 and 3, respectively, do not separate and hence the underlying distribution is not fitted appropriately with weighting $\lambda_{\text{batch}} = (0.3, 0.06, 0.3, 0.14, 0.07, 0.05, 0.01, 0.02, 0.05, 0.04)$.

Iris data: In the well known iris data, the task is to separate three classes of plants based on four numerical values. Vector quantization with six codebooks yields the results reported in Table 1. Weighting factors in the respective cases are $\lambda_{\text{RLVQ}} = (0.04, 0.05, 0.03, 0.87)$, $\lambda_{\text{GRLVQ}} = (0, 0, 0.4, 0.6)$, and $\lambda_{\text{batch}} = (0.1, 0.12, 0.53, 0.25)$. Batch-RLVQ yields comparable classification results. It clearly indicates that the last two dimensions are important as proposed by GRLVQ and rule extraction algorithms like [3].

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 295-300

5. Conclusions

We have proposed an approximate energy function and a batch version of RLVQ, an intuitive generalization of LVQ to an adaptive weighting scheme of the input dimensions. This approach provides theoretical insight into the behaviour of RLVQ: it describes the limiting behaviour for stable situations; moreover, it gives hints which local minima might cause bad classification accuracies for RLVQ. The batch formulation can be combined with annealing schedules such that local minima can be avoided. Hence the result of batch-RLVQ can be used as a good starting point for modifications which directly optimize the classification accuracy such as GRLVQ. Note that the objective of batch-RLVQ itself is minimization of the overall distances which is usually related to the classification task but might yield worse classification results. Still in our cases the results are comparable to RLVQ and GRLVQ and better than simple LVQ due to the adaptive metric.

References

[1] T. Bojer, B. Hammer, D. Schunk, and K. Tluk von Toschanowitz, Relevance determination in learning vector quantization, in M.Verleysen (ed.), Proceedings of ESANN'01, D-facto publications, 271-276, 2001.

[2] J. M. Buhmann and H. Kühnel, Vector quantization with complexity costs, IEEE Transactions on Information Theory 39, 1133-1145, 1993.

[3] W. Duch, R. Adamczak, and K. Grąbcweski, Extraction of crisp logical rules using constrained backpropagation networks, in M. Verleysen (ed.), Proceedings of ESANN'97, D-facto Publications, 1997.

[4] I. Gath and A.B. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Analysis and Machine Intelligence 11, 773-791, 1989.

[5] T. Graepel, M. Burger, and K. Obermayer, Self-organizing maps: generalizations and new optimization techniques, Neurocomputing, 21, 173–190, 1998.

[6] D.E. Gustafson and W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, Proceedings of IEEE CDC'79, 761-766, 1979.

[7] B. Hammer and T. Villmann, Estimating relevant input dimensions for self-organizing algorithms, in N.Allison, H.Yin, L.Allinson, J.Slack (eds.), Advances in Self-Organizing Maps, Springer, 173-180, 2001.

[8] T. Heskes, Self-organizing maps, vector quantization, and mixture modeling, to appear in IEEE Transactions on Neural Networks.

[9] K.-U. Höffgen, H.-U. Simon, and K. VanHorn, Robust trainability of single neurons, Journal of Computer and System Sciences, 50, 1995.

[10] S. Kaski, Bankruptcy analysis with self-organizing maps in learning metrics, to appear in IEEE Transactions on Neural Networks.

[11] T. Kohonen, Self-Organizing Maps, Springer, 1997.

[12] R. Neal and G. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, in M. Jordan (ed.), Learning in Graphical Models, Kluwer, 355-368, 1998.

[13] M. Pregenzer, G. Pfurtscheller, and D. Flotzinger, Automated feature selection with distinction sensitive learning vector quantization, Neurocomputing 11, 19-29, 1996.

[14] K. Rose, E. Gurewitz, and G. C. Fox, Statistical mechanics and phase transitions in clustering, Physical Review Letters 65(8), 945-948, 1990.

[15] A. S. Sato and K. Yamada, Generalized learning vector quantization, in G. Tesauro, D. Touretzky, and T. Leen (eds.), Advances in Neural Information Processing Systems 7, 423-429, MIT Press, 1995.

[16] J. Sinkkonen and S. Kaski, Clustering based on conditional distribution in an auxiliary space, to appear in Neural Computation.