Analysis of Robust Soft Learning Vector Quantization and an application to Facial Expression Recognition

— Extended Abstract —

Gert-Jan de Vries[1] and Michael Biehl[2]

[1] Philips Research Europe, User Experiences
High Tech Campus 34, NL-5656 AE Eindhoven, The Netherlands
GJ.de.Vries@philips.com
[2] University of Groningen, Inst. of Mathematics and Computing Science
P.O. Box 407, 9700 AK Groningen, The Netherlands
M.Biehl@rug.nl

Keywords. Learning Vector Quantization, Analysis, Facial Expression Recognition

1 Introduction

Learning Vector Quantization (LVQ) [1] is a popular method for multiclass classification. Several variants of LVQ have been developed recently, of which Robust Soft Learning Vector Quantization (RSLVQ) [2] is a promising one. Although LVQ methods have an intuitive design with clear updating rules, their dynamics are not yet well understood. In simulations within a controlled environment RSLVQ performed very close to optimal. This controlled environment enabled us to perform a mathematical analysis as a first step in obtaining a better theoretical understanding of the learning dynamics. This extended abstract provides the outline of our theoretical analysis and its results. Moreover, we will focus on the practical application of RSLVQ to a real world data set containing extracted features from facial expression data.

2 LVQ

Learning Vector Quantization (LVQ), originally posed by Kohonen [1,3] and known by the name LVQ1, is a method of online supervised competitive learning. Many variations on the basic scheme of LVQ1 have been suggested, among which LVQ2.1 and LVQ3 [3,4], GLVQ [5] and RSLVQ [6,2], with the aim of obtaining better generalization capacity.

During learning, data samples and their class labels are presented sequentially, or so called 'on-line'. From a set of prototype vectors, defined in the same (potentially high dimensional) space as the data, the closest (set of) prototype(s) is determined and updated such that if the class label coincides with the class label of the data sample, the prototype is attracted to the data, otherwise repelled. The data, carrying labels of different classes, is assumed to be distributed

Dagstuhl Seminar Proceedings 09081
Similarity-based learning on structures
http://drops.dagstuhl.de/opus/volltexte/2009/2035

around a specified number of prototypes. Note that there can be more than one prototype per class, enabling a good fit of prototypes to data that contains highly complex class boundaries. After training, classification is done by determining the closest of all prototypes and returning the class label corresponding to this winning prototype. The decision boundaries between the prototypes can also be considered as the Voronoi tessellation of the feature space.

The variations of LVQ-algorithms mainly differ in which specific prototypes are updated (for example only the closest conflicting prototypes or the closest prototype with corresponding and closest with conflicting label) and how these prototypes are updated. The generic structure of an LVQ algorithm can be expressed in the following way:

$$\begin{aligned}\boldsymbol{w_l}^{\mu} &= \boldsymbol{w_l}^{\mu-1} + \Delta\boldsymbol{w_l}^{\mu},\\ &= \boldsymbol{w_l}^{\mu-1} + \frac{\eta}{N} f(\{\boldsymbol{w_i}^{\mu-1}\}, \boldsymbol{\xi}^{\mu}, \sigma^{\mu}, \ldots)(\boldsymbol{\xi}^{\mu} - \boldsymbol{w_l}^{\mu-1}) \\ &\quad \text{with } l, i = 1 \ldots c, \mu = 1, 2, \ldots\end{aligned} \tag{1}$$

Where w_l^{μ} is prototype w_l (of class l) at time step μ, η is the so-called learning rate, N is the dimensionality of the system. The specific form of the update strength function f is determined by the specific LVQ variant.

2.1 RSLVQ

LVQ variants as LVQ2.1 or Learning From Mistakes (LFM) [7] apply updates with update strength 0 or 1, referred to as a hard or crisp update. Robust Soft LVQ uses the relative distance between a data sample and the prototypes to soften this update strength. The update formula of RSVLQ, as defined by Seo and Obermayer [2] is as follows:

$$\begin{aligned}\boldsymbol{w}_{\tilde{l}}^{\mu} &= \boldsymbol{w}_{\tilde{l}}^{\mu-1} + \Delta\boldsymbol{w}_{\tilde{l}}^{\mu}\\ &= \boldsymbol{w}_{\tilde{l}}^{\mu-1} + \tilde{\eta}\begin{cases}\left(P_l(\tilde{l}|\boldsymbol{\xi}^{\mu}) - P(\tilde{l}|\boldsymbol{\xi}^{\mu})\right)\frac{\partial f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})}{\partial \boldsymbol{w}_{\tilde{l}}^{\mu}} & \text{if } l = \sigma^{\mu}\\ -P(\tilde{l}|\boldsymbol{\xi}^{\mu})\frac{\partial f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})}{\partial \boldsymbol{w}_{\tilde{l}}^{\mu}} & \text{if } l \neq \sigma^{\mu}\end{cases}\end{aligned} \tag{2}$$

where $P_l(\tilde{l}|\boldsymbol{\xi}^{\mu})$ and $P(\tilde{l}|\boldsymbol{\xi}^{\mu})$ are assignment probabilities:

$$\begin{aligned}P_l(\tilde{l}|\boldsymbol{\xi}^{\mu}) &= \frac{p(\tilde{l}) \exp\left(f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})\right)}{\sum_{\tilde{l}=\sigma^{\mu}} p(\tilde{l}) \exp\left(f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})\right)}\\ P(\tilde{l}|\boldsymbol{\xi}^{\mu}) &= \frac{p(\tilde{l}) \exp\left(f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})\right)}{\sum_{\tilde{l}} p(\tilde{l}) \exp\left(f(\boldsymbol{\xi}, \boldsymbol{w}_{\tilde{l}}^{\mu})\right)}\end{aligned} \tag{3}$$

$P_l(\tilde{l}|\boldsymbol{\xi}^{\mu})$ describes the posterior probability that the data sample $\{\boldsymbol{\xi}^{\mu}, \sigma^{\mu}\}$ is assigned to prototype $\boldsymbol{w}_{\tilde{l}}$ of class l, given that the data sample was generated by the correct class. $P(\tilde{l}|\boldsymbol{\xi}^\mu)$ describes the posterior probability that the data sample is assigned to prototype $\boldsymbol{w}_{\tilde{l}}$ of all prototypes of all classes. $f(\boldsymbol{\xi}, \boldsymbol{w}^\mu_{\tilde{l}})$ describes the assumed distribution of the data around the prototypes in such a way that $K(\tilde{l})exp\big(f(\boldsymbol{\xi}, \boldsymbol{w}^\mu_{\tilde{l}})\big)$ gives the probability that the data vector $\boldsymbol{\xi}^\mu$ is assigned to prototype $\boldsymbol{w}_{\tilde{l}}$.

In our analysis we assume a Gaussian distribution, i.e., $K(\tilde{l}) = (2\pi v_{\tilde{l}})^{N/2}$ and $f(\boldsymbol{\xi}, \boldsymbol{w}^\mu_{\tilde{l}}) = -(\boldsymbol{\xi} - \boldsymbol{w}^\mu_{\tilde{l}})^2/2v_{\tilde{l}}$, and equal width and strength of all prototypes, i.e., the variances and priors are equal: $\forall S :: v_S = v_{soft}, p(S) = \frac{1}{\#\boldsymbol{w}_S}$. Furthermore we consider a two class data set to be represented with one prototype per class.

With these assumptions the update rule (2) becomes:

$$\boldsymbol{w_l}^\mu = \boldsymbol{w_l}^{\mu-1} + \frac{\eta}{N v_{soft}} (\delta_{l\sigma^\mu} - \Omega_l)(\boldsymbol{\xi}^\mu - \boldsymbol{w_l}^{\mu-1}) \tag{4}$$

Where $\delta_{l\sigma^\mu}$ is the Kronecker delta and $\Omega_l = (1 + \exp(\frac{d^\mu_l - d^\mu_{-l}}{2v_{soft}}))^{-1}$.

3 Analysis

In order to study the algorithm analytically, we restrict ourselves to a model of artificial data, as described in more detail in [8]. It considers a mixture of two Gaussian clusters which overlap fully in all dimensions, however there exists a linear projection to a plane in which there is controllable less overlap. Seven characteristic quantities are chosen such that they characterize the full system and are used to visualize the learning system evolving over time.

Conceptually, the analysis consists of the following steps:

1. Description of the development of characteristic quantities in terms of recurrence relations
2. Reformulation of recurrence relations as differential equations (ODE), yielding a coupled system of 7 ODE's
3. Performing averages on the ODE's to describe generalization ability

The update formula of RSLVQ (4) cannot be integrated analytically, as will be needed in the analysis. We use the observation that $\frac{1}{1+\exp(x)}$ is very similar to $\Phi(\frac{-x}{c})$, where $c \in \mathbb{R}$ is a constant which controls the slope of the Φ-function (we derived $c = \frac{4}{\sqrt{2\pi}}$ for the best fit). This adaptation allows an analytical solution of the ODE's in the limit $N \to \infty$ and $\eta \to 0$.

We compared the analytical results with simulations and found that they coincide. The Φ-replacement however causes for slight deviations from the original RSLVQ behavior. The generalization error (the parameter of main interest), however, is not affected and thereby well described by the ODE's. Note that the differential equations are determined in the limit $N \to \infty$. The simulations were performed with $N = 100$, which is obviously sufficient to match the theory for $N \to \infty$, see [8] for a discussion of finite N corrections.

We studied the effect of the hyperparameter of RSLVQ, v_{soft}. While simulations with a large number of epochs show a slightly increasing error for larger softness, the analytical results show that there is no effect on the softness (indicating that in the simulations the system had, because of finite training time, not converged fully for large softness settings).

4 Practical application: Facial Expression Recognition

We applied RSLVQ to a real world data set consisting of features extracted from photos of facial expressions from the Cohn-Kanade database [9]. The features extracted are so termed Local Binary Patterns (LBP), which are composed as follows: Per grey valued pixel (i_c) the LBP value is calculated by comparing the pixel to its eight neighbors, resulting in a binary string of which the decimal value is taken, according to:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \tag{5}$$

Where s is the Heaviside step function. The $2^8 - 1$ possible outcomes are reduced to $L = 59$ by regarding only those LBP values with at most 2 bitwise transitions, as proposed by Ojala et al. [10].

The images are divided into (42) regions R_j, where per region a histogram $H_i = \sum_{x,y} \delta_{LBP(x,y),i}$ (with $(x, y) \in R_j, i = 0, \ldots, L - 1$) is built. These histograms are placed next to each other, forming a single vector of length $N = 42 * 59 = 2478$. Another variant uses overlapping regions, resulting in a vector of length $N = 8437$.

The data set contains the facial expressions of 95 university students, showing 7 emotions: Anger, Disgust, Fear, Joy, Sadness, Surprise, Neutral. In total there are 1214 instances which we used in 10-fold cross validation, in such a way that the test set contains not only unseen expressions, but unseen faces.

The results, given in Table 1, indicate that RSLVQ competes well with Support Vector Machine (SVM) [11], a leading technique in facial expression classification. LVQ2.1 showed severe stability issues, therefore its performance listed should not be given too much credit.

Table 1. Performance on facial expression data (10 fold cross validation)

	LBP	**LBP with overlap**
SVM (linear kernel) [12]	90.9 ± 5.6%	92.9 ± 5.0%
RSLVQ	92.2 ± 2.0%	93.2 ± 2.6%
LVQ2.1	83.2% (stability issues)	

The influence of the softness parameter on the performance of RSLVQ turned out to be quite small and showed similar patterns as found in the mathematical analysis. The influence of choosing more than one prototype per class was only marginal for this data set.

5 Conclusions

In conclusion we have shown the outline of our mathematical analysis of RSLVQ, enabling an analytical exploration of the behavior and performance in terms of generalization ability within a controlled environment. The analysis showed that there is no influence of the hyperparameter on the asymptotic generalization error, however in practice (with finite training time), larger softness values tend to lead to larger generalization error. Moreover, we showed that in a practical application on facial expression data, RSLVQ competes well with the leading classifier in the field, SVM.

6 Acknowledgements

The authors' gratitude goes to Caifeng Shan from the Video Processing & Analysis group at Philips Research for providing the facial expression data and reviewing this paper. We also thank Igor Berezhnoy and Rob Takken for their helpful comments to improve this paper.

References

1. Kohonen, T. In: Learning vector quantization. MIT Press, Cambridge, MA, USA (1995) 537–540
2. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural computation **15** (2003) 1589–1603
3. Kohonen, T.: Improved versions of learning vector quantization. Procedings of the International Joint conference on Neural Networks **1** (1990) 545–550
4. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1997)
5. Sato, A., Yamada, K. In: Generalized learning vector quantization. Volume 7. (1995) 423–429
6. Seo, S., Bode, M., Obermayer, K.: Soft nearest prototype classification. IEEE Transactions on Neural Networks **13** (2003) 390–398
7. Ghosh, A., Biehl, M., Hammer, B.: Dynamical analysis of lvq type learning rules. In: Workshop on the Self-Organizing-Map WSOM05. Univ. de Paris (I. (2005)
8. Biehl, M., Gosh, A., Hammer, B.: Dynamics and generalization ability of lvq algorithms. Journal of Machine Learning Research **8 (Feb)** (2007) 323–360
9. Kanade, T., Cohn, J., Tian, Y.: Comprehensive database for facial expression analysis. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00) (2000) 46–53
10. Ojala, T., Pietikinen, M., Menp, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 971–987
11. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
12. Gritti, T., Shan, C., Jeanne, V., Braspenning, R.: Local features based facial expression recognition with face registration errors. FG (2008)