ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

AGLVQ - Making Generalized Learning Vector Quantization Aware of Context

Torben Graeber[1,2] and Sebastian Vetter[2] and Sascha Saralajew[2]
and Michael Unterreiner[2] and Dieter Schramm[1]

1 - University of Duisburg-Essen - Chair of Mechatronics
Lotharstraße 1, MD 222-230, 47057 Duisburg, Germany

2 - Dr. Ing. h.c. F. Porsche AG
Porschestrasse 911, 71287 Weissach, Germany

Abstract. Generalized Learning Vector Quantization methods are a powerful and robust approach for classification tasks. They compare incoming samples with representative prototypes for each target class. While prototypes are physically interpretable, they do not account for changes in the environment. We propose a novel framework for the incorporation of context information into prototype generation. We can model dependencies in a modular way ranging from polynomials to neural networks. Evaluations on artificial and real-world datasets show an increase in performance and meaningful prototype adaptations.

1 Introduction

Prototype-based learning algorithms like the Generalized Learning Vector Quantization (GLVQ) [1] constitute a powerful approach to classification tasks. They feature a transparent decision-making process through a vivid distance metric and physically interpretable prototypes. Furthermore, they tend to be robust because of their inherent hypothesis margin maximization [2]. Nonetheless, prototypes may give away representability to achieve better classification performance [3]. Thus, this approach still has drawbacks. We will show that the gap between interpretability and classification performance can be closed further with the use of *adaptive prototypes*. It is intuitive to us as humans, that our internal representation of the world changes in dependence on context. When we classify elephants against tigers, our visual representation of an elephant might change in light of the country we live in. Asian elephants e.g. are smaller than African elephants. Therefore, we propose an extension of the GLVQ algorithm that utilizes prior knowledge about the context to adapt the representation for target classes. This dependency can be modeled by any differentiable function. We make use of simple polynomes to keep a level of interpretability and neural networks (NN) as general approximators in the sense of [4].

The gap between interpretable but underfitting and non-interpretable but highly performant models is a problem for safety-critical applications. The use of NNs still constitutes a great challenge in many applications [5]. Several publications approach this problem by vigorously testing NNs, but capturing every corner case is nearly impossible [6]. Work on adversarial attacks makes the brittleness of their performance even more apparent [7]. On the other hand,

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

prototype-based learning methods show greater robustness to attacks [8]. The proposed approach offers a design choice with a high degree of interpretability while allowing for highly non-linear dependencies on environmental factors.

Saralajew et al. combine GLVQ algorithms with NNs [8]. They propose different uses of GLVQ in neural network structures, e.g. as a final classification layer. This approach uses the feature extraction capabilities of neural networks but has the downside that prototypes are not physically interpretable. In contrast, the approach presented in this paper restricts prototypes to be physically interpretable by direct comparison to incoming samples.

In the following sections, we introduce GLVQ algorithms and present the theory of our new approach as well as remarks regarding model structure and training process. We show the effectiveness of the proposed extension by conducting experiments on different datasets. The results show an improvement in classification performance and increased representability of the prototypes.

2 Introduction GLVQ

For the introduction to Generalized Learning Vector Quantization we define a prototype $\mathbf{w} = [w_1, w_2, ..., w_n]$ for each target class. In this paper we assume one prototype per class for readability. Samples are compared with the prototypes by a distance metric $d = d(\mathbf{x}, \mathbf{w})$ (e.g. Euclidean or Mahalanobis). Furthermore, we define a generalized distance $\mu = (d^+ - d^-)/(d^+ + d^-)$, where $d^+ = d(\mathbf{x}, \mathbf{w}^+)$ is the distance to the closest matching prototype with class $C = C(\mathbf{x})$ and analogously $d^- = d(\mathbf{x}, \mathbf{w}^-)$ to the closest prototype with the wrong class. We formulate loss function $S(\mu)$ for N samples

$$S = \sum_{i=1}^{N} f(\mu(x_i)), \tag{1}$$

where $f(\mu)$ is a monotonic activation function. During the training process, prototypes are adapted by gradient-based optimization techniques such as SGD or RMSprop. Hammer et al. introduced the addition of a relevance vector to the distance calculation (GRLVQ) to allow the weighting of features [9]. The distance calculation changes to

$$d_\lambda(\mathbf{x}, \mathbf{w}) = \sum_{t=1}^{n} \lambda_t (x_t - w_t)^2 \tag{2}$$

with the relevance vector λ that is updated alongside the prototypes. Features that allow a good separation between the target classes will be weighted higher. While each feature weight is independent their sum is normalized to prevent degeneration. Schneider et al. expand the relevance vector from GRLVQ to a full importance matrix (GMLVQ) [10]. Therefore, the distance calculation is extended with the symmetric and positive definite matrix Δ:

$$d_\Delta(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^T \Delta (\mathbf{x} - \mathbf{w}) \tag{3}$$

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

If Δ is learned to be the covariance of the data, it represents the Mahalanobis distance. Analogously to GRLVQ, the matrix Δ is updated and has to be normalized afterwards that the condition $\sum_i \Delta_{ii} = 1$ holds.

3 Proposed Extension

To incorporate changes in the environment, we propose a formulation of prototype $\mathbf{w} = [w_1, ..., w_n]$ in dependence of an environment vector $\mathbf{c} = [c_1, ..., c_m]$. We write $\mathbf{w} = \mathbf{w}(\mathbf{c}) = \mathbf{w}_s + \mathbf{w}_a(\mathbf{c})$ with static prototype $\mathbf{w}_s$ and a delta $\mathbf{w}_a$. The prototypes, therefore, are no longer constant but adaptive. Thus, a generative path is added to the GLVQ algorithm to adjust the static prototypes. Figure 1 illustrates the extension of the conventional GLVQ algorithm to the proposed adaptive one. Extensions are drawn as dashed lines, original model elements with solid lines. The extension can be combined with all GLVQ classifiers (e.g. GRLVQ or GMLVQ) by adding importances to the distance calculation.

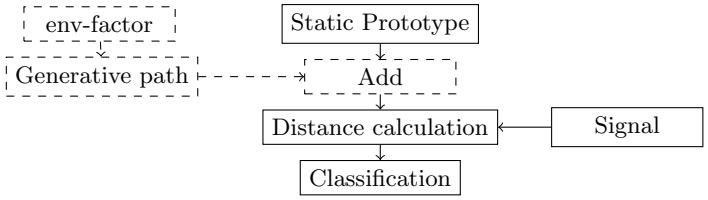


Fig. 1: GLVQ (solid) and additional steps of the adaptive extension (dashed)

We propose two variants for the newly added generative part: one with a polynomial (PN) and one with a NN. The PN variant constitutes a version with lower computational cost, while the complex NN is able to achieve the best performance. We formulate the polynomes[1] with degree l as $\mathbf{w}_a = \sum_{i=1}^{m} \mathbf{w}_{a,i} = \sum_{i=1}^{m} (\mathbf{a}_i * c_i + \mathbf{b}_i * c_i^2 + ... + \mathbf{l}_i * c_i^l)$ with learned coefficient vectors $\mathbf{a}_i$, $\mathbf{b}_i$, ..., $\mathbf{l}_i$ for each environment factor c_i and the NN as $\mathbf{w}_a = NN(\mathbf{c})$[2]. The output $\mathbf{w}_a$ is added to the static prototypes.

3.1 Practical remarks and extension of the loss function

We introduce an auxiliary loss d_{aux} to ensure the representability of prototypes. The extension is a mean-squared-error between prototypes $\mathbf{w}$ and samples $\mathbf{x}$:

$$d = f_{mar} d_{mar} + f_{aux} d_{aux} \qquad d_{aux} = \frac{1}{n} \sum_{i=1}^{n} (x_i - w_i)^2 \tag{4}$$

[1] Polynomes in our case: one per feature and environment factor (not multivariate)
[2] Architecture NN in our case: several dense layers with batch normalization

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

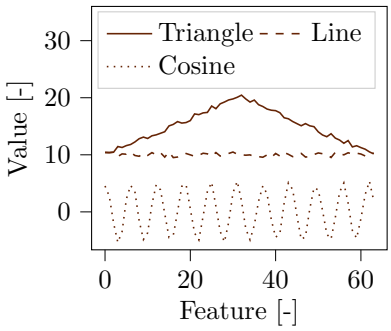
with weighting factors f_{mar}, f_{aux} and maximum margin loss d_{mar}. The results show an improvement of convergence and stability of the training process. The factors f_{mar}, f_{aux} provide adjustable parameters to balance between maximum margin and the level of representability.

Furthermore, the training process consists of two optimization phases: a pretraining to reach representative prototypes ($f_{mar} = 0, f_{aux} = 1$) and margin maximization training ($f_{mar} > 0$). During margin maximization training, we have to set a good balance between margin maximization and auxiliary loss.

4 Experiments

4.1 Artificial Datasets

We make use of three datasets with each three classes ($n = 64$) for demonstration purposes: Constant signals (CS), variable signals with linear influence (VSL), and variable signals with non-linear influence (VSNL). Figure 2a visualizes samples of the classes of dataset CS. Class one follows a cosine function, class two a straight line, and class three the upper half of a triangle. Uniform noise is added to all samples. For datasets VSL/VSNL we introduce an artificial environment factor $\mathbf{c} = [c_1]$ ($m = 1$) over which the samples change their shape (class one: amplitude, class two: offset, class three: slope). In the case of VSL, this change is linear w.r.t. to $\mathbf{c}$, in the case of VSNL non-linear.



(a) Classes of data set CS

Model	CS	VSL	VSNL
GLVQ	100	84.78	63.69
GRLVQ	100	93.89	76.67
GMLVQ	100	95.11	80.22
A-GLVQ (PN)	100	100	65.11
A-GRLVQ (PN)	100	100	79.89
A-GMLVQ (PN)	100	100	80.56
A-GLVQ (NN)	100	100	100
A-GRLVQ (NN)	100	100	100
A-GMLVQ (NN)	100	100	100

(b) Results on data sets CS, VSN and VSNL (in percent)

Table 2b compares the results of conventional and novel classifiers. All classifiers reach 100% on dataset CS. The conventional classifiers are not able to fully separate dataset VSL since samples from different classes overlap (for different c). The novel adaptive classifiers can learn this dependency and still reach 100%. On dataset VSNL only the adaptive prototypes with NN are able to capture the non-linear dependencies. The PN underfits but can still improve its results compared with the conventional algorithms. The experiments highlight the importance of designing the right dependency on environment $\mathbf{c}$.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

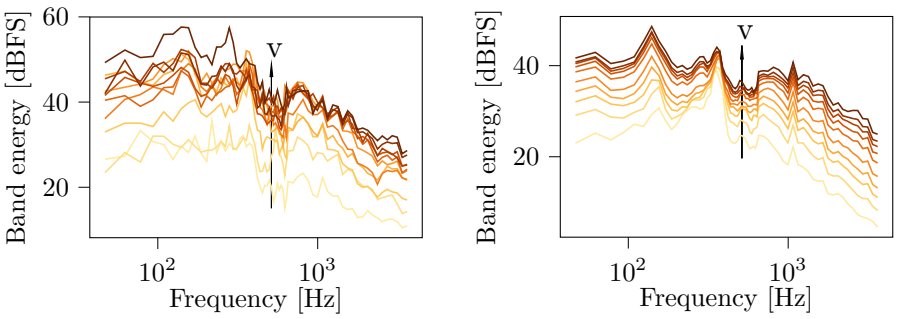
4.2 Real world data

Tires for passenger cars emit different sounds at different tread depths. This sound can be captured with a microphone in the wheel arc. The target for this dataset is to classify the tread depth in three levels: new, half-worn, worn. Raw acoustic signals are filtered and transformed into the frequency domain via an FFT (logarithmic energy spectrums, $n = 64$). We performed a clean training-test-split. Figure 3a shows how the features are highly dependent on vehicle speed v (higher energies with higher speeds), so we define $\mathbf{c} = [v]$ ($m = 1$). The speed-adaptive prototypes $\mathbf{w}(v)$ of trained classifiers resemble a smoothed version of the real-world data (figure 3b). Table 1 compares the performance of trained classifiers. We observe a great increase in classification performance by the introduction of adaptive prototypes. For our example, tests with A-

Modelvariant	**Training** [%]	**Test** [%]
GMLVQ	86.69	72.78
A-GMLVQ (PN)	95.06	80.83
A-GMLVQ (NN)	95.97	80.40

Table 1: Results on real world dataset

GLVQ and A-GRLVQ have not shown a similar increase in performance but comparable prototypes like A-GMLVQ. The relevance vector collapsed to only use one feature (only one non-zero element).



(a) Spectrums from real world data

(b) Prototypes on real world data

5 Conclusion

This paper proposes an extension of GLVQ variants that incorporate changes of context in the classification process. An additional generative path transfers learned static prototypes into different contexts. While arbitrary generative paths are possible, we introduce a polynomial and NN variant. The adaptive GLVQ variants have the advantage that they offer physical interpretability

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

of learned prototypes while allowing complex adaptions to different contexts. Model comparisons show improved performances for classification tasks with changing environments, such as acoustic profile tread depth classification. Evaluations on artificial data sets narrow the advantages down to specific data set properties. The performance increase of the novel approach in comparison to non-adaptive GLVQ algorithms is steep for our application. Especially the A-GMLVQ variant shows robust convergence properties.

We propose further research with the following focus. For one, the importance of different features could also vary with the environment. Thus, we propose to learn an adaptive projection in the sense of relevance learning $\lambda = \lambda(\mathbf{c})$ or matrix learning $\Delta = \Delta(\mathbf{c})$. A second approach is the improvement of the combination with relevance or tangent learning. While we did not reach satisfying results so far, we are convinced that fitting regularization techniques could lead to improvements. We provided our code as an open-source toolbox[3] and propose the application of the novel framework to other data sets and the reproduction of performance increases.

References

[1] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In MIT Press, editor, *Proceedings of the 8th International Conference on Neural Information Processing Systems*, pages 423–429. MIT Press, Cambridge, MA, 1995.

[2] Koby Crammer, Ran Gilad-bachrach, Amir Navot, and Naftali Tishby. Margin analysis of the lvq algorithm. In MIT Press, editor, *Advances in Neural Information Processing Systems*, volume 15, pages 462–469. MIT Press, Cambridge, MA, 2003.

[3] Barbara Hammer, David Nebel, Martin Riedel, and Thomas Villmann. Generative versus discriminative prototype based classification. In Thomas Villmann, Frank-Michael Schleif, Marika Kaden, and Mandy Lange, editors, *Advances in Self-Organizing Maps and Learning Vector Quantization*, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham and s.l., 2014.

[4] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[5] Rick Salay, Rodrigo Queiroz, and Krzysztof Czarnecki. An analysis of iso 26262: Machine learning and safety in automotive software. In SAE International, editor, *WCX World Congress Experience*. 2018.

[6] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. *Commun. ACM*, 62(11):137–145, 2019.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[8] Sascha Saralajew, Lars Holdijk, Maike Rees, and Thomas Villmann. Prototype-based neural network layers: Incorporating vector quantization. *CoRR*, 2018, 2018.

[9] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. *Neural networks : the official journal of the International Neural Network Society*, 15:1059–1068, 2002.

[10] Petra Schneider, Michael Biehl, and Barbara Hammer. Adaptive relevance matrices in learning vector quantization. In MIT Press, editor, *Neural Computation*, volume 21, pages 3532–3561. MIT Press, Cambridge, MA, 2009.

[3] A-GLVQ Python implementation: https://github.com/graebe/aglvq