ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

Implementation of Multi-Matrix Median Generalized Learning Vector Quantization

Lukas Bader[1,2] and Ina Terwey-Scheulen[1] and Dietlind Zühlke[2] *

1- August Rüggeberg GmbH - Marketing
Marienheide - Germany

2- Technische Hochschule Köln - Faculty of Engineering and Computer Science
Gummersbach - Germany

Abstract. We present Multi-Matrix Median Generalized Learning Vector Quantization (M^3GLVQ), a median-based LVQ method for multiple heterogeneous proximity matrices. The model combines class-wise medoid prototypes with a learnable simplex-constrained relevance vector that determines each matrix's contribution. Prototype positions are updated via the established greedy hill-climbing procedure of median GLVQ, while matrix weights are adapted through a normalized gradient step followed by simplex projection, ensuring stable, scale-independent updates. This alternating optimization operates directly on proximity data and requires no embedding or feature representation. Experiments on industrial customer data with four complementary proximity sources show that M^3GLVQ leads to higher recall than standard MGLVQ.

1 Introduction and motivation

In industrial customer analytics, predicting whether a customer purchases consumable industrial tools requires models capable of handling heterogeneous profiling data. Such data can consist of categorical features such as industry sector, produced products, processed materials or manufacturing applications used, complemented by a few numerical attributes like company size or location.

The Median Generalized Learning Vector Quantization (MGLVQ) method is suitable for non-vectorial categorical data due to its median-based structure and interpretable through representative prototypes. [1]. However, the classical implementation of MGLVQ is based on a single proximity matrix derived from categorical dissimilarities [2]. In scenarios with several heterogeneous feature groups, multiple proximity matrices could be constructed, but standard MGLVQ only allows a pre-weighted aggregation of these matrices as input. This aggregation must be defined before training and cannot be adapted during learning.

The proposed extension addresses this limitation by enabling MGLVQ to process multiple proximity matrices simultaneously. In addition to optimizing prototype positions, the model dynamically learns the relative importance of each matrix during training. This allows it to capture varying influences of different feature

*This contribution has been developed in the project PLan_CV. Within the funding Programme FH-Personal, the project PLan_CV (reference number 03FHP109) is funded by the German Federal Ministry of Research, Technology and Space (BMFTR) and Joint Science Conference (GWK).

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

sets on purchasing behavior while maintaining the high interpretability required in industrial applications where the credibility and explainability of predictive models are essential.

2 Related work

The Median Generalized Learning Vector Quantization (MGLVQ) framework [1, 3] extends prototype-based learning to proximity data. Instead of using feature vectors and dissimilarity functions to identify prototypes that can move through the entire feature space, a matrix of inter-data-point proximities is used to identify those data points that are best suited as representative prototypes. Despite being computationally intensive, the method is effective for categorical or heterogeneous data, but current formulations are restricted to a single proximity matrix that must capture all information in a static manner.

For vectorial data, several LVQ variants exist incorporating adaptive metric components, such as Generalized Matrix LVQ (GMLVQ) and its local extensions [4], which adjust the weights of different proximity components, and thus the relevance of the features during training. These matrix-based formulations naturally support multi-metric representations but are limited to continuous feature spaces. Further studies on soft, kernelized, and multi-view LVQ [5, 6] explored the combination of multiple representations. In these cases, categorical data is mapped to a vector space using Kernels. However, finding a good kernel is a challenging task, especially in areas where you have a large number of categories that are in a specific relation to each other. Furthermore, these methods assume a single proximity for each feature, which is sometimes not useful as the features might be organized in groups that need to be handled together to ensure valid semantics. For these cases, Heterogeneous-LVQ was developed [7], which allows intergration of several feature groups with their corresponding dissimilarity measures.

To our knowledge, no median-based LVQ method has yet been formulated to jointly process and adapt multiple proximity matrices. The proposed approach fills this gap by extending the median-LVQ paradigm towards multi-matrix learning on proximity data using the ideas of the Heterogeneous-LVQ.

3 Multi-Matrix Extension of MGLVQ (M^{3}GLVQ)

Classical MGLVQ operates on a single proximity matrix $D \in \mathbb{R}^{m \times m}$. M^{3}GLVQ generalizes this to a set $\mathcal{D} = \{D^{(v)}\}_{v=1}^{V}$ of heterogeneous proximity matrices and introduces two additional parameters: the number of matrices V and a weight vector $\boldsymbol{\alpha} \in \mathbb{R}^V$ with $\alpha_v \geq 0 \quad \forall v, \qquad \sum_{v=1}^{V} \alpha_v = 1$ ensuring regularization. All proximity evaluations now use a dynamically aggregated matrix that replaces the static single-matrix input of the original algorithm and is recomputed in every training iteration:

$$\tilde{D} = \sum_{v=1}^{V} \alpha_v \, D^{(v)},$$

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

where $\tilde{D}$ denotes the aggregated proximity matrix, $D^{(v)}$ the v-th input matrix, and α_v its associated weight.
The prototype updates are still done using Expectation Maximization in a greedy hill-climbing scheme as used by Paassen [2]. Data points in the local neighborhood of the current prototype are evaluated and accepted whenever they decrease the loss. After each accepted prototype move, the matrix weights are updated by a normalized additive step followed by simplex projection:

$$g_v := \sum_i \mu_i^{(v)}, \qquad \boldsymbol{g} \leftarrow \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_1}, \qquad \alpha_v \leftarrow \Pi_\Delta\big(\alpha_v - \eta\, g_v\big)$$

The per-sample, per-matrix term $\mu_i^{(v)}$ is

$$\mu_i^{(v)} = \frac{\alpha_v\,(d_{i,v}^+ d_i^- - d_{i,v}^- d_i^+)}{(d_i^+ + d_i^- + \varepsilon)^2}.$$

Here, d_i^+ and d_i^- are the aggregated dissimilarities of sample i to its closest correct and incorrect prototype, and $d_{i,v}^+$, $d_{i,v}^-$ are the corresponding per-matrix dissimilarities. The constant ε prevents division by zero. The projection Π_Δ enforces non-negativity and unit-sum constraints on the weight vector. The L1-normalization of the gradient stabilizes the update by removing scale effects: regardless of the magnitude of the raw gradient, the effective step size is controlled solely by η, making the learning rate intuitively usable and preventing excessively large weight shifts. The model adapts alternatingly prototype positions and the relative relevance of each proximity matrix.
During prediction, the aggregated proximity $\tilde{D} = \sum_{v=1}^{V} \alpha_v D^{(v)}$ is used to assign each sample to its nearest prototype. This maintains consistency with training and allows inspection of matrix relevances via α_v. The computational cost grows linearly with V, while model interpretability remains unchanged.
The original MGLVQ implementation [2] allowed only a fixed and equal number for all classes in the data set. We extended it to support class-specific prototype numbers and user-defined initial matrix weights [8].

4 Experiments and Results

The experiments evaluate how heterogeneous proximity data, learned matrix weights, and prototype allocation affect classification performance, and compare MGLVQ with its multi-matrix extension.

4.1 Data Description

The setup uses 2,000 industrial customers, represented by four proximity matrices generated from distinct profiling feature groups. For every customer, a binary purchase indicator for one product – determined by strategic focus – defines the classification target, with the positive class at a rate of 0.27. Each proximity matrix uses a structure-adapted metric specific to its profiling feature shown in Table 1 and [8].

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

ID	Proximity matrix	Features	Metric
D_1	Industry	Hierarchical NAICS codes	Weighted Hamming
D_2	Produced Products	HS code sets	Weighted Dice
D_3	Material/Application	DIN/ISO code sets	Weighted Dice
D_4	Geographic	Coordinates, country, subregion, continent	Metric, adjacency adjustment

Table 1: Proximity matrices, feature groups, and metrics.

4.2 Two-Stage Parameter Optimization

The central objective of M^3GLVQ is to determine the optimal matrix-weight combination. Related models show strong variability in how their internal weight matrices evolve, even under constant hyperparameter settings [9]. Weight discovery and hyperparameter tuning were decoupled for reproducibility and domain-consistent results, forming a two-stage optimization process. Balanced accuracy and recall serve as primary quality metrics. Balanced accuracy compensates class imbalance in sales data. Recall captures the model's ability to identify new buyers, a core marketing requirement.

Bayesian Search for Weight Combination In the first stage, the model was given a maximally flexible search space, allowing the internal matrix weights to evolve freely. A Bayesian optimization using the Optuna framework was applied to explore the number of prototypes per class, the learning rate η, and different initial weight settings [10]. Across 200 trials evaluated with 5-fold stratified cross-validation, balanced accuracy served as the primary assessment metric, with the top 10% of runs retained as high-performing. From these runs, the final weight vectors were extracted and clustered using k-means, resulting in three stable weight configurations (mean balanced accuracy > 0.70) shown in Table 2. Cluster 1 is notable for its strong drift toward v_4 (location), which contradicts domain knowledge. Cluster 2, in contrast to Cluster 1, has fewer members and an almost uniform weight distribution, offering little domain-specific structure. Therefore, all further experiments are based on Cluster 0. We also observed that highly unbalanced initial weights can cause a single dominant component to trap the model in a local minimum, preventing meaningful adjustment of the remaining weights.

Cluster	Members	Mean bal. acc.	v_1	v_2	v_3	v_4
0	52	0.714	0.100	0.145	0.620	0.135
1	23	0.713	0.013	0.012	0.010	0.965
2	26	0.715	0.248	0.281	0.271	0.199

Table 2: Cluster statistics for weight mixtures (v_1–v_4).

Grid Search with Fixed Weights With fixed weights and disabled η, the grid search varied only prototype counts for Class 0 and Class 1. For MGLVQ, the

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

same grid search was applied, using an input matrix constructed from domain-knowledge-based weights. The best-performing prototype configurations for both models in terms of balanced accuracy and recall are summarized in Table 3.

4.3 Evaluation and Statistical Comparison

Both models were evaluated using a stratified 100-fold procedure with all parameters fixed, allowing direct performance comparison. Balanced accuracy does not differ significantly between the two models (Wilcoxon signed-rank, p=0.111), whereas recall shows a significant improvement for M^{3}GLVQ (p=0.009).

	MGLVQ	M^{3}GLVQ
K	8	$\{0:3,\ 1:7\}$
T	100	100
v	[0.300, 0.250, 0.350, 0.100]	[0.100, 0.145, 0.620, 0.135]
Balanced Accuracy	0.640 ± 0.135	0.661 ± 0.120
Recall	0.504 ± 0.264	0.575 ± 0.248

Table 3: Comparison of MGLVQ and M^{3}GLVQ (100-fold mean $\pm$ std).

4.4 Visual Inspection

Figure 1 shows a t-SNE projection of the weighted aggregation of the proximity matrices. The M^{3}GLVQ weights, obtained through the two-stage optimization, generate a representation with more clearly separated class structures. The MGLVQ baseline, initialized with expert-defined weights, produces only partial separation and exhibits stronger class mixing within clusters. This visualizes the performance advantage also reported in Table 3. M^{3}GLVQ further provides asymmetric prototype allocation: three prototypes suffice for Class 0, whereas seven prototypes capture the more complex structure of Class 1.

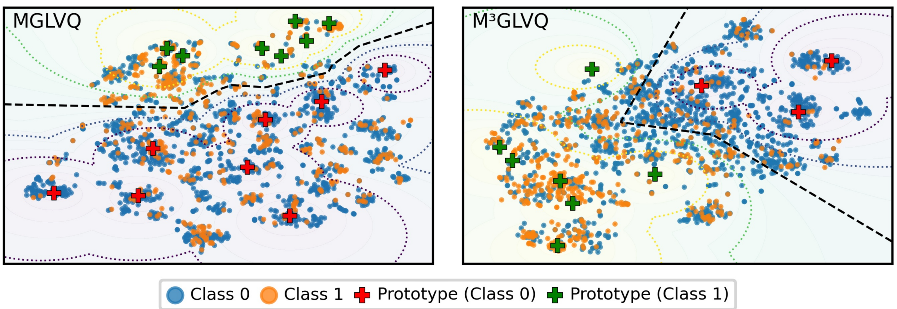

Fig. 1: Visual comparison of MGLVQ and M^{3}GLVQ using t-SNE

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

5 Interpretation

The results show that the adaptive proximity variant has a slightly higher balanced accuracy as compared to the original MGLVQ variant and a higher recall, which is the more important quality measure in this use case.
The identified weights are significantly different from the distribution proposed by the domain expert given to MGLVQ. V_3 being rated significantly higher indicates that the feature capturing material–application combinations is substantially more important for the product than the expert initially assumed.

6 Discussion and future work

Besides extending the data set to include additional products and customers and conducting an extensive evaluation with domain experts, there are potential mathematical extensions of the method. As mentioned, the weights identified by M^3GLVQ varied significantly; thus, decoupling the weight identification from the determination of the ideal number of prototypes was chosen. However, current extensions of LVQ variants [9, 11] explicitly address weight variance to enable combined optimisation. The respective concepts should be applied to M^3GLVQ.

References

[1] David Nebel, Barbara Hammer, Kathleen Frohberg, and Thomas Villmann. Median variants of learning vector quantization for learning of dissimilarity data. *Neurocomputing*, 169:295–305, 2015.

[2] Benjamin Paaßen. proto-dist-ml: `proto_dist_ml/mglvq.py` (median generalized learning vector quantization). GitLab repository, 2019. GNU GPLv3. Accessed 2025-11-12.

[3] David Nebel, Barbara Hammer, and Thomas Villmann. A median variant of generalized learning vector quantization. In *International Conference on Neural Information Processing*, pages 19–26. Springer, 2013.

[4] Petra Schneider, Michael Biehl, and Barbara Hammer. Adaptive relevance matrices in learning vector quantization. *Neural computation*, 21(12):3532–3561, 2009.

[5] Sambu Seo and Klaus Obermayer. Soft learning vector quantization. *Neural computation*, 15(7):1589–1604, 2003.

[6] Daniela Hofmann and Barbara Hammer. Kernel robust soft learning vector quantization. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 14–23. Springer, 2012.

[7] Dietlind Zühlke, Frank-Michael Schleif, Tina Geweniger, Sven Haase, and Thomas Villmann. Learning vector quantization for heterogeneous structured data. In *Proc. ESANN 2010*, April 2010.

[8] Lukas Bader. M^3GLVQ. GitHub repository, 2025. Accessed: 2025-11-19.

[9] Sofie Lövdal and Michael Biehl. Iterated relevance matrix analysis (IRMA) for the identification of class-discriminative subspaces. *Neurocomputing*, 577, April 2024.

[10] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM, 2019.

[11] Marco Canducci, Lida Abdi, Alessandro Prete, Roland J. Veen, Michael Biehl, Wiebke Arlt, and Peter Tino. Discriminative subspace emersion from learning feature relevances across different populations. Workingpaper, arXiv, April 2025.