PROTOTYPE-BASED FEATURE SELECTION WITH THE NAFES PACKAGE

Nana Abeka Otoo
nanaabekaotoo@authentic.network.com

Muhammad Abubakar
muhammad.abubakar@gwdg.de

ABSTRACT

This paper introduces Nafes as a prototype-based feature selection package designed as a wrapper centered on the highly interpretable and powerful Generalized Matrix Learning Vector Quantization (GMLVQ) classification algorithm and its local variant (LGMLVQ). Nafes utilizes the learned relevances evaluated by the mutation validation scheme for Learning Vector quantization (LVQ), which iteratively converges to selected features that relevantly contribute to the prototype-based classifier decisions.

Keywords Feature relevance · Feature selection · Feature ranking · Learning vector quantization

1 Introduction

Feature engineering is one of many vital steps in modeling learning machines for practical application. Feature selection remains a core aspect of feature engineering, which has an effectual bearing on the performance of machine learning models. Applied machine learning practitioners often need to gain prior knowledge of the relevant attributes in the feature space that effectively contribute to classifier decisions. Using an overly extensive feature space has high computational and performance costs that must be addressed [1]. A reduced feature space that maximizes classification performance based on concurrent and affirmatively learned relevances provides a technically sound answer to the aforementioned problem [2, 3, 4]. Most feature selection algorithms are black boxes or lack the requisite interpretability for outcomes. A wrapper algorithm centered on a matrix-GLVQ classifier evaluated for good fit by mutation validation (MV) with a full matrix of relevances can be utilized as a better alternative to the out-of-sample evaluated feature selection algorithms [5, 6]. This paper investigates a novel feature selection wrapper algorithm centered on the GMLVQ and LGMLVQ classification algorithms for mathematical comprehensibility, computational efficiency, high interpretability, and good generalization ability.

2 Learning Vector Quantization

Learning vector quantization (LVQ) is a highly interpretable prototype-based machine learning algorithm [7]. Classification in the LVQ family of algorithms is centered on prototypes, initialized within the feature space and subsequently updated to a converged point, allowing the data patterns to be typically represented by the prototypes for class assignments. The prototype updates follow an attraction of the correct matching prototype and repulsion of the incorrect matching prototypes to and from the training input under consideration. Since inference in LVQ is spirited on the nearest prototype classifier scheme, the minimum computed dissimilarity between test input and learned prototypes determines the classification thereof.

2.1 Generalized Matrix Learning Vector Quantization

Matrix-GLVQ [8] is a derivative of the GLVQ algorithm [9], which utilizes a matrix of relevances in the specification of a differentiable dissimilarity measure most often in applications chosen as the squared Euclidean distance [3]. Learning in Matrix-GLVQ follows the same scheme employed in GLVQ based on the cost function.

$$E_{\Omega}\left(\mathbf{s}, f_{\phi}\right)=\sum_{i=1}^{N} f_{\phi}\left(\mu_{\Omega}\left(\mathbf{s}\right)\right)=\sum_{i=1}^{N} f_{\phi}\left(\frac{d_{\Omega}\left(\mathbf{s}_i, \mathbf{w}^{+}\right)-d_{\Omega}\left(\mathbf{s}_i, \mathbf{w}^{-}\right)}{d_{\Omega}\left(\mathbf{s}_i, \mathbf{w}^{+}\right)+d_{\Omega}\left(\mathbf{s}_i, \mathbf{w}^{-}\right)}\right) \tag{1}$$

with the differentiable dissimilarity $d_{\Omega}^{+}\left(\mathbf{s}\right)=d_{\Omega}\left(\mathbf{s}, \mathbf{w}^{+}\right)$ and $d_{\Omega}^{-}\left(\mathbf{x}\right)=d_{\Omega}\left(\mathbf{x}, \mathbf{w}^{-}\right)$ signifying correct and incorrect matching distance respectively based on $\left\{\mathbf{w}^{+}, \mathbf{w}^{-}\right\} \in W$ that principally correctly and incorrectly [9] assign an input under consideration. Where f is the activation function conditioned to be monotonically increasing is given as

$$f_{\phi}\left(k\right)=\left(1+e^{(-\phi k)}\right)^{-1} \tag{2}$$

for stochastic gradient descent learning [9, 10, 11, 4]. The relevant dissimilarity measure with adaptation during learning [12] is given as

$$\begin{aligned} d_{\Omega}\left(\mathbf{s}, \mathbf{w}\right) &= \left(\Omega\left(\mathbf{s}-\mathbf{w}\right)\right)^{2} \\ &= \left(\mathbf{s}-\mathbf{w}\right)^{T} \Omega^{T} \Omega\left(\mathbf{s}-\mathbf{w}\right), \quad \Omega \in \mathbb{R}^{m \times n}, \quad m \leq n \end{aligned} \tag{3}$$

The activation function (2) and the classifier function $\mu_{\Omega} \in [-1, 1]$ effects the minimization of the classification error[13]. From equation (3), when $m=n$, the positive definite matrix $\Lambda=\Omega^{T} \Omega \in \mathbb{R}^{n \times n}$ offers insightful information for attribute space correlations which impacts learner class discrimination [13] and thus, the degree of contribution of attributes i and j to $d_{\Omega}\left(\mathbf{s}, \mathbf{w}\right)$ is measured by $\Lambda_{i,j}$ [8]. Since the matrix adaption follows similar updates of the prototypes, the elements of the $\Lambda_{i,j}$ matrix bears a true nature of the maximized classification performance [4] and Λ is not prone to data correlation matrix estimation related bias ([14, 15]). The contribution of stand-alone respective attributes j to the learner decisions is measured by $\Lambda_{j,j}=\sum_i \Omega_{ij} \Omega_{ij}=\sum_i \Omega_{ij}^{2}$ [8, 4].

3 Nafes Package

Nafes is an interface to a highly interpretable and robust prototype-based feature selection wrapper algorithm centralized on a matrix-generalized learning vector quantizer and its local variant. The wrapper algorithm herein presented is a prototype-based modification of [1] designed with a broader focus on interpretability and features that ensure a good fit. [1]

Algorithm 1 Prototype-based feature selection algorithm

Require: Training set $T=\{\mathbf{s}_n, c\left(\mathbf{s}_n\right)\}_{n=1}^{N} \in \{\mathbb{R}^{n}, \mathcal{C}\}^{N}$
1: **Learning Vector Quantizer:** Initialize a prototype-based classifier (GMLVQ or LGMLVQ) with $0<\epsilon_2 \leq \epsilon_1 \ll 1,\ 1 \leq nppc,\ 0<\varphi$ for termination criterion
2: **Validation Scheme:** Select either the holdout or mutation validation scheme for performance evaluation based on an appropriate metric (accuracy, precision etc.)
3: **Learning Iterations: for** $i=0,1,2, \ldots\ldots.$ **do**
4: **Optimal Search:** Run the learning vector quantizer and store the Ω matrix and performance evaluation scores ξ.
5: **Repeat:** Steps 2 and 3 with an increment of the number of prototypes per class by 1 for each iteration
6: **Convergence:** Compare if $\xi_i<\xi_{i+i}$ or using a convenient matrix norm if, $||\Omega_{i+1}-\Omega_i|| \leq \varphi$ **stop**
7: **Compute:** $\Lambda=\Omega^{T} \Omega \in \mathbb{R}^{n \times n}$ based on the learned full matrix of relevances Ω
8: **Ranking:** Rank by the magnitude of feature relevance $\Lambda_{j,j}$ for global Ω else compute ranks based on the number of hits from the local Λ respectively based on the local Ω matrices.
9: **Non-Rejection:** Select and rank features by order of magnitude for which $\sum_i \Lambda_{i,j} \neq 0$
10: **Rejection:** Reject features if $\sum_i \Lambda_{i,j}=0$
11: **Return:** Ranked selected, rejected features with corresponding hits
12: **End procedure**

[1]The prototype-based feature selection algorithm can be evaluated using either the holdout or the MV scheme[5]. In the case of the latter, the mutation degree (φ) must be small but positive and fixed. The parameters ϵ_1 and ϵ_2 are the respective prototype and omega-matrix adaption learning rates. The implementation of the algorithm in Python is made available in the Nafes package [16].

4 Experiments

This section illustrates the proposed prototype-based feature selection wrapper algorithm using Wisconsin Breast Cancer data (WDBC)[17] and Ozone Level dataset[18] from the UCI Machine Learning Repository. The WDBC dataset has a cardinality of 569 cases for binary classes, namely infectious and non-infectious, with 30 attributes. The Ozone Level dataset has 73 features for 2536 cases with two classes.

Table 1: classification accuracies for WDBC and Ozone Level datasets using Nafes feature selection Package

	WDBC		Ozone	
	GMLVQ	LGMLVQ	GMLVQ	LGMLVQ
Method	$1\|1 \xrightarrow{1,2} 3\|3, 2\|2$	$1\|1 \xrightarrow{1,2} 3\|3, 2\|2$	$1\|1 \xrightarrow{1,2} 2\|2, 2\|2$	$1\|1 \xrightarrow{1,2} 2\|2, 2\|2$
[1]MV($\varphi = 0.2$)	94.65%	$94.20\% \xrightarrow{r} 94.90\%$	93.04%	$93.16\% \xrightarrow{r} 92.90\%$
[2]Holdout	94.15%	$91.81\% \xrightarrow{r} 88.89\%$	93.50%	$93.50\% \xrightarrow{r} 93.50\%$
[1]# param.	$960 \rightarrow 1.1$k	1.9k$\rightarrow$ 5.6k	5.3k$\rightarrow$ 5.5k	10.5k$\rightarrow$ 21.5k
[2]# param.	$960 \rightarrow 1.0$k	1.9k$\rightarrow$ 3.7k	5.3k$\rightarrow$ 5.5k	10.5k$\rightarrow$ 21.5k
[1]# features.	$30 \rightarrow 17$	$30 \rightarrow 29 \xrightarrow{r} 23$	$72 \rightarrow 42$	$72 \rightarrow 66 \xrightarrow{r} 41$
[2]# features.	$30 \rightarrow 21$	$30 \rightarrow 21 \xrightarrow{r} 13$	$72 \rightarrow 45$	$72 \rightarrow 70 \xrightarrow{r} 48$

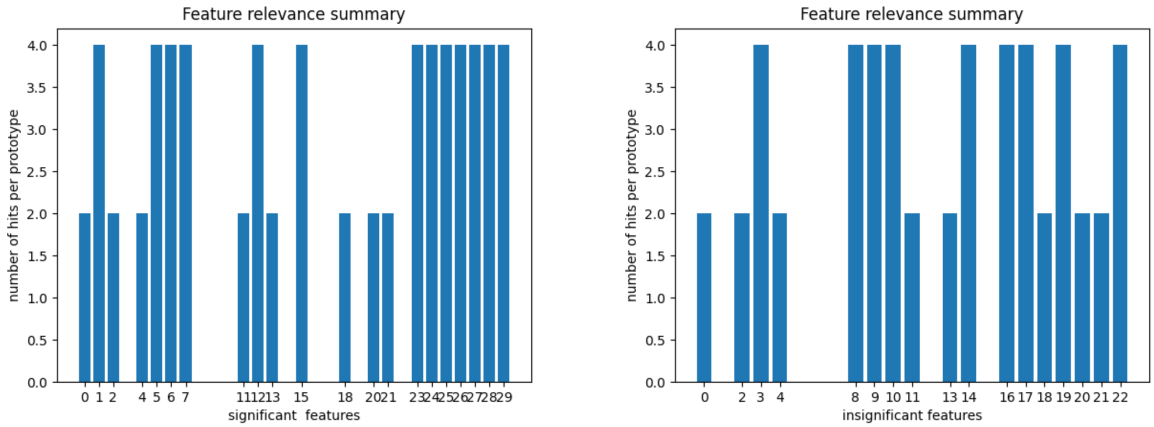
The results from Table (1) shows a stable performance for the induction learners (leaning vectors quantizers) employed by the proposed prototype-based features selection wrapper algorithm. Utilizing the global and local relevances of the matrix-generalized learning quantizer allows the NAFES package to reduce the attribute space of the WDBC and Ozone Level datasets.

Harnessing the local but much more powerful and complex variant (LGMLVQ), practitioners can access information regarding tentative features by analyzing the number of prototype hits per relevant and irrelevant features. This information is used along with a reject strategy in (1,2, 4,3) to further reduce the feature space relevantly without loss in performance. This approach allows applied machine learning practitioners, by way of interpretation, to account for relevant, irrelevant and tentative features of the attribute space that effects classification decisions.

The performance of the proposed prototype-based feature selection wrapper algorithm was evaluated using the holdout and mutation validation schemes [5]. The MV scheme ensures the use of the entire dataset, focusing on the goodness of fit of the learning vector quantizers used. It hence accurately represents the entire feature space through the reduced feature space. The results from Table (1) indicate that algorithm (1) combined with the MV scheme offers a significant feature space reduction with comparable and improved performance compared to the holdout validation scheme.

Feature selection herein experimented is based on the global and local setting. The latter is comparatively complex; hence, we introduce a rejection strategy that moves the behavior of the local setting to one akin to the global setting. The rejection strategy employed by prototype-based feature selection algorithm (1) entails two steps: (a) dropping features with significant designation if their recorded prototype hits are less than the prototype hits of same features with insignificant designation (b) moving features with equalized prototype hits for significant and insignificant designations to the tentative state. [2]

[2]The rejection strategy employed in (1) entails two steps. We, however, stressed that if the second step is foregone, a relaxed rejection strategy that permits the tentativeness of some features is attained. The decision regarding the choice or the form of a rejection strategy for the local setting depends on the user. This paper recommends that users try both relaxed and full forms of the rejection strategy and choose the best performance.

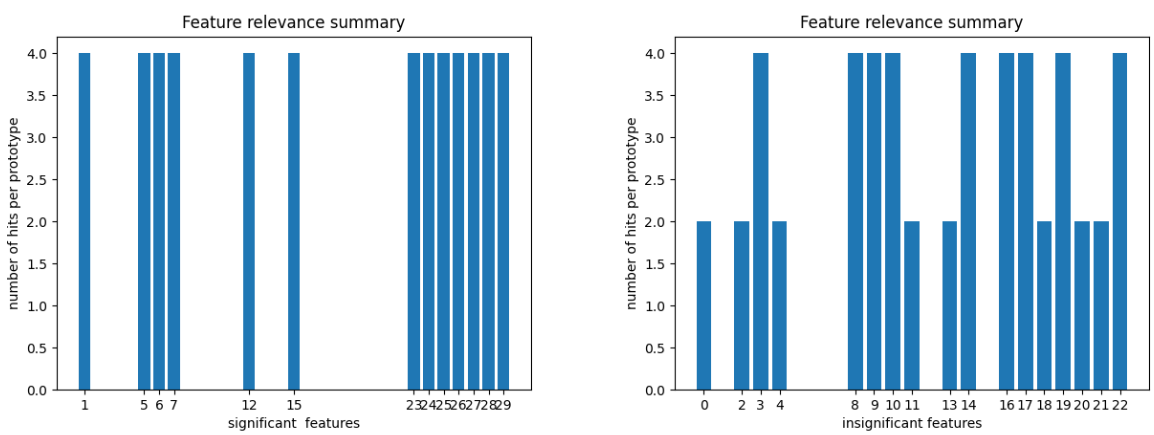


(a) LMGLVQ

(b) LGMLVQ

Figure 1: Feature relevance without reject strategy evaluated with a holdout scheme against the number of hits per prototype for the WDBC dataset.

A comparison of the results based on the WDBC dataset shows that global relevances from GLMVQ reduced the feature space by way of relevance ranking from 30 to 21 for holdout evaluation scheme and 30 to 17 for the mutation validation scheme. The analysis of the local relevances from the LGMLVQ learner initially reduced the features space to 29 and 21 respectively for the holdout and MV scheme. An optimal reduction is attained by applying the full rejection strategy that detects and removes tentative features from the significant feature space. The effect of the rejection strategy advanced a further significant reduction to 23 and 13 features, respectively, for the MV and holdout validation schemes. Nonetheless, a better performance evaluation score was attained for the MV scheme than the holdout scheme. Hence, the significant features selected based on the MV scheme and the conclusion from [5] are the significant features with a good fit.

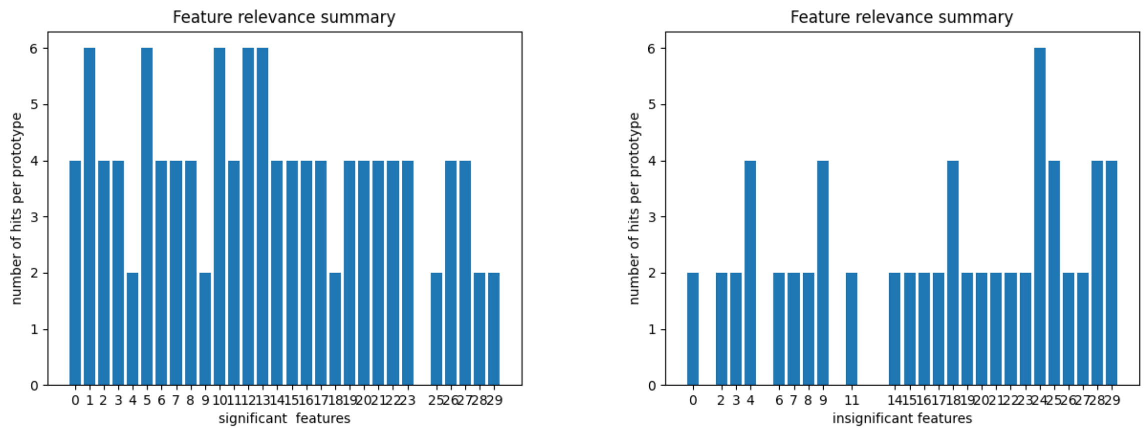


(a) LMGLVQ

(b) LGMLVQ

Figure 2: Feature relevance with reject strategy evaluated with holdout scheme against the number of hits per prototypes for the WDBC dataset.

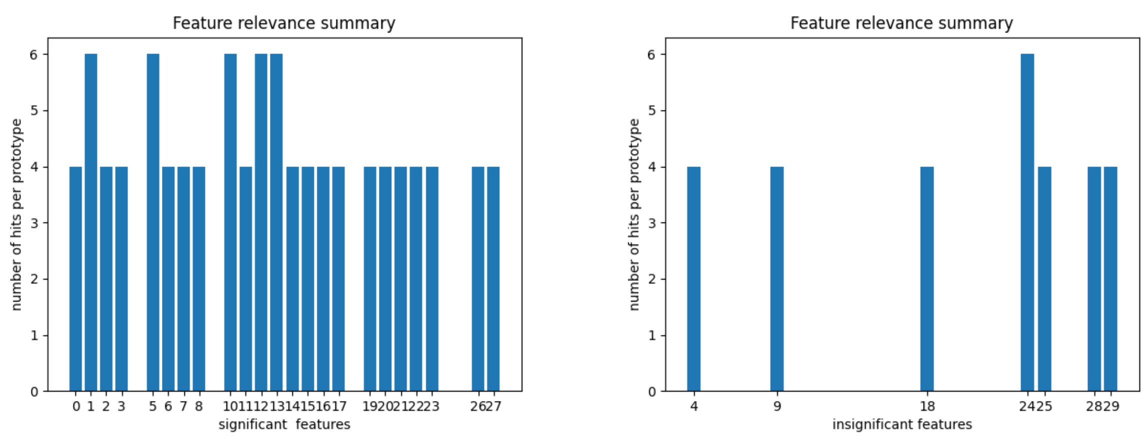
The numerical evaluation of the result in Table (1) for the Ozone Level dataset depicts a similar stable behavior for both learning vector quantizers utilized by the proposed algorithm in (1) as induction learners. The global GMLVQ reduced the feature space from 72 to 42 and 45 respectively for the MV and holdout schemes. The local variant with the aforementioned reject strategy truncated the features space from an initial reduction of 66 to 41 and 70 to 48 for the MV and holdout scheme herein respectively considered.



(a) LGMLVQ (b) LGMLVQ

Figure 3: Feature relevance without reject strategy evaluated using MV scheme against the number of hits per prototype for the WDBC dataset.

It can also be observed from the results in (1) that local relevances offer an insightful and interpretable view regarding the behavior of global relevances. The local view of how relevant features are determined based on the number of prototype hits combined with global interpretations of learned relevances provides access to crucial information for practitioners and domain managers.



(a) LMGLVQ (b) LGMLVQ

Figure 4: Feature relevance with reject strategy evaluated using MV scheme against the number of hits per prototype for the WDBC dataset.

5 Discussion

The numerical evaluation of test outcomes shows that the proposed feature selection wrapper algorithm offers an interpretable feature selection scheme. The prototype-based induction learners utilized by the feature selection wrapper algorithm reasonably attain equivalent and comparable results with high interpretability. Practitioners should opt for global selection or use the local option with an efficient rejection strategy to avoid performance evaluation penalization. The procedure described in the proposed prototype-based feature selection algorithm is devoid of heuristics and hence possesses simplicity, consistency, interpretability, good generalization ability and robustness from inception to completion. The Nafes package provides an interface to the proposed algorithm. It is a valuable tool for applied machine learning practitioners and domain managers to prepare the feature space for effective machine learning pipelines.

6 Conclusion

A new prototype-based feature selection algorithm has been introduced in this paper. The mathematical background for formulating the proposed algorithm and experiments using real-world datasets has been presented. Numerical evaluation of the experimental outcomes indicates that the proposed algorithm meets the requirements of a highly interpretable wrapper-based feature selection algorithm with a validation scheme designed to ensure a good fit for the reduced feature set. Future work will entail the application of soft prototype-based model predictors for interpretable feature selection.

7 Acknowledgment and Disclosure

The authors declare no conflict of interest in this work.

References

[1] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[2] Roland Nilsson, José M Pena, Johan Björkegren, and Jesper Tegnér. Consistent feature selection for pattern recognition in polynomial time. *The Journal of Machine Learning Research*, 8:589–612, 2007.

[3] Barbara Hammer, Marc Strickert, and Thomas Villmann. On the generalization ability of grlvq networks. *Neural Processing Letters*, 21:109–120, 2005.

[4] Michael Biehl. Matrix learning in learning vector quantization. 2006.

[5] Nana Abeka Otoo. Mutation validation for learning vector quantization, 2023.

[6] Jie M. Zhang, Mark Harman, Benjamin Guedj, Earl T. Barr, and John Shawe-Taylor. Model validation using mutated training labels: An exploratory study, 2021.

[7] Teuvo Kohonen. Improved versions of learning vector quantization. In *1990 ijcnn international joint conference on Neural networks*, pages 545–550. IEEE, 1990.

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

[9] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. *Advances in neural information processing systems*, 8, 1995.

[10] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[11] Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer, 2007.

[12] Petra Schneider, Michael Biehl, and Barbara Hammer. Distance learning in discriminative vector quantization. *Neural computation*, 21(10):2942–2969, 2009.

[13] Thomas Villmann, Andrea Bohnsack, and Marika Kaden. Can learning vector quantization be an alternative to svm and deep learning?-recent trends and advanced variants of learning vector quantization for classification learning. *Journal of Artificial Intelligence and Soft Computing Research*, 7(1):65–81, 2017.

[14] Jianqing Fan, Yuan Liao, and Han Liu. An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, 19(1):C1–C32, 2016.

[15] Marika Kaden, Katrin Sophie Bohnsack, Mirko Weber, Mateusz Kudła, Kaja Gutowska, Jacek Blazewicz, and Thomas Villmann. Learning vector quantization as an interpretable classifier for the detection of sars-cov-2 types based on their rna sequences. *Neural Computing and Applications*, 34(1):67–78, 2022.

[16] Nana Abeka Otoo. Prototype-based-feature-selection-with-the-nafes-package. `https://github.com/naotoo1/Prototype-based-feature-selection-with-the-Nafes-Package`, 2023.

[17] Catherine Blake. Uci repository of machine learning databases. *http://www. ics. uci. edu/~ mlearn/MLRepository. html*, 1998.

[18] Kun Zhang, Wei Fan, and XiaoJing Yuan. Ozone level detection. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5NG6W.