PROTOTYPE-BASED SOFT FEATURE SELECTION PACKAGE

Nana Abeka Otoo
Authentic-Network
Chemnitz
nana.abekaotoo@authentic.network

Asirifi Boa
University of Applied Sciences
Mittweida
aboa@hs-mittweida.de

Muhammed Abubakar
Gesellschaft für wissenschaftliche
Datenverarbeitung mbH
Göttingen
muhammad.abubakar@gwdg.de

ABSTRACT

This paper presents a prototype-based soft feature selection package (Sofes) wrapped around the highly interpretable Matrix Robust Soft Learning Vector Quantization (MRSLVQ) and the Local MRSLVQ algorithms. The process of assessing feature relevance with Sofes aligns with a comparable approach established in the Nafes package, with the primary distinction being the utilization of prototype-based induction learners influenced by a probabilistic framework. The numerical evaluation of test results aligns Sofes' performance with that of the Nafes package.

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

1 Introduction

A crucial step in the process of building practical machine learning applications is feature engineering[1]. Among the various aspects of feature engineering, feature selection plays a pivotal role, significantly impacting the performance of machine learning models[1]. Practitioners in applied machine learning often need to acquire prior knowledge about which attributes in the feature space contribute to classifier decisions. Employing an excessively large feature space gives rise to the curse of dimensionality, leading to significant computational and performance burdens that require meticulous handling [2]. To address this challenge, a reduced feature space that maximizes classification performance can be achieved by considering concurrently and affirmatively learned relevances [3, 4, 5, 6]. Many feature selection algorithms lack step-by-step mathematical comprehensibility and thus obscure the understanding of their inner workings and outputs[1]. In this context, we propose a wrapper-based algorithm that focuses on the Matrix Robust Soft Learning Vector Quantization (MRSLVQ) and the Local MRSLVQ model(s) [7, 8] serving as an inductive learner(s) coupled with a target space robust evaluation scheme [9]. This paper presents a soft variant of the novel feature selection algorithm introduced in Nafes [1] by prioritizing mathematical simplicity, computational efficiency, high interpretability and good learning dynamics as crucial attributes for effective feature selection.

2 Learning Vector Quantization (LVQ)

A highly interpretable prototype-based supervised machine learning algorithm is Learning Vector Quantization (LVQ) [10]. Since the LVQ family of algorithms is a subset of nearest prototype classifiers, learning is spirited on prototypes $W = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$, selected within the attribute space $\mathbb{R}^n$ of the input vector $S = \{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots, \mathbf{s}_N\}$ and subsequently updated by an attraction and repulsion mechanism (2) based on the nearest prototype principle (1), allowing the data patterns to be typically represented by the prototypes for class assignments[11]. Inference in LVQ is determined by computing the nearest prototype to a given sample from the test set using a dissimilarity measure d

mostly chosen as the squared Euclidean distance. However, the choice of dissimilarity measure is not limited to the Euclidean distance [7, 10, 11, 12]. By the winner takes all rule guided by the nearest prototype principle, we have:

$$Q\left(\mathbf{s}\right) = \arg\min_{k}\ d\left(\mathbf{s}, \mathbf{w}_k\right), 1 \leq k \leq M \tag{1}$$

with M being the cardinality of the prototype set W. For a given input sample s, the reference vectors $\mathbf{w}_{Q(\mathbf{s})}$ is strengthened if $c\left(\mathbf{s}\right) = c\left(\mathbf{w}_{Q(\mathbf{s})}\right)$ and weakened if $c\left(\mathbf{s}\right) \neq c\left(\mathbf{w}_{Q(\mathbf{s})}\right)$ using the learning rule in (3) regarding (2) with learning rate β.

$$\zeta\left(c\left(\mathbf{s}\right), c\left(\mathbf{w}_{Q(\mathbf{s})}\right)\right) = \begin{cases} +1, & c\left(\mathbf{s}\right) = c\left(\mathbf{w}_{Q(\mathbf{s})}\right) \\ -1, & c\left(\mathbf{s}\right) \neq c\left(\mathbf{w}_{Q(\mathbf{s})}\right) \end{cases} \tag{2}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \beta\zeta\left(\mathbf{s} - \mathbf{w}_t\right); \quad \mathbf{w}_t = \mathbf{w}_{Q(\mathbf{s})}; \quad 0 < \beta \ll 1 \tag{3}$$

3 Matrix Robust Soft Learning Vector Quantization (MRSLVQ)

A probabilistic LVQ that utilizes a soft model predictor based on prototypical representation from the feature space that assumes centers of a Gaussian mixture model is introduced by [7] as Robust Soft LVQ (RSLVQ). Learning in RSLVQ follows probabilistic approach hence maximizes the mutual information between the predicted probability vector $p_W\left(\mathbf{s}\right) = \left(p_W\left(1|\mathbf{s}\right), p_W\left(2|\mathbf{s}\right), \ldots, p_W\left(C|\mathbf{s}\right)\right)^T$ and actual class target probability vector $p\left(\mathbf{s}\right) = \left(p_1\left(\mathbf{s}\right), \ldots, p_C\left(\mathbf{s}\right)\right)^T$ by minimizing the cross-entropy loss computed as:

$$E(S, W) = \sum_{r=1}^{N} \ln\left(\frac{P_W(\mathbf{s}_r, c_r)}{P_W(\mathbf{s}_r)}\right) \tag{4}$$

where

$$P_W\left(\mathbf{s}, c\right) = \sum_{j:c(\mathbf{w}_j)=c} \exp\left(\frac{-d_\Omega\left(\mathbf{s}, \mathbf{w}_j\right)}{2\sigma^2}\right) = \sum_{j:c(\mathbf{w}_j)=c} \exp\left(\frac{-\left(\Omega\left(\mathbf{s} - \mathbf{w}_j\right)\right)^2}{2\sigma^2}\right) \tag{5}$$

and

$$P_W\left(\mathbf{s}\right) = \sum_{l} \exp\left(\frac{-d_\Omega\left(\mathbf{s}, \mathbf{w}_l\right)}{2\sigma^2}\right) = \sum_{l} \exp\left(\frac{-\left(\Omega\left(\mathbf{s} - \mathbf{w}_l\right)\right)^2}{2\sigma^2}\right) \tag{6}$$

The soft model predictor for Matrix-RSLVQ is given by

$$p_W\left(c|\mathbf{s}\right) = \frac{P_W\left(\mathbf{s}, c\right)}{P_W\left(\mathbf{s}\right)} \tag{7}$$

$$p_{\mathbf{w}}\left(c|\mathbf{s}\right) = \frac{\sum_{j:c(\mathbf{w}_j)=c} \exp\left(-\frac{\left(\Omega\left(\mathbf{s} - \mathbf{w}_j\right)\right)^2}{2\sigma^2}\right)}{\sum_{l} \exp\left(-\frac{\left(\Omega\left(\mathbf{s} - \mathbf{w}_l\right)\right)^2}{2\sigma^2}\right)} \tag{8}$$

The dissimilarity measure follows the same relevance distance utilized in the Generalized Matrix Learning Vector Quantization [5, 4, 8] given by

$$\begin{aligned} \left(\Omega\left(\mathbf{s} - \mathbf{w}_j\right)\right)^2 &= \left(\Omega\left(\mathbf{s} - \mathbf{w}_j\right)\right)^T \Omega\left(\mathbf{s} - \mathbf{w}_j\right) \\ &= \left(\mathbf{s} - \mathbf{w}_j\right)^T \Omega^T \Omega\left(\mathbf{s} - \mathbf{w}_j\right); \quad \Omega \in \mathbb{R}^{m \times n},\ m \leq n \\ &= \left(\mathbf{s} - \mathbf{w}_j\right)^T \Lambda\left(\mathbf{s} - \mathbf{w}_j\right) \end{aligned} \tag{9}$$

Classification in MRSLVQ follows the RSLVQ analogy by learning a full linear transformation matrix of relevances Ω globally, and for Local MRSLVQ, each learned prototype comes with its respective matrix of relevances Ω.

4 Prototype-Based Soft Feature Selection (Sofes)

Sofes package serves as an interface to the highly interpretable soft feature selection wrapper algorithm built on the prototype-based inductive learner(s) MRSLVQ and LMRSLVQ. The wrapper algorithm introduced in this paper is a soft modification of the novel algorithm implemented in [1] designed with a broader focus on determining relevant features that ensure a good fit.

Algorithm 1 Soft Prototype-based feature selection algorithm

```
Require: Training set $T = \{\mathbf{s}_n, c(\mathbf{s}_n)\}_{n=1}^N \in \{\mathbb{R}^n, \mathcal{C}\}^N$
  Initialize: prototype-based classifier (3), $nppc \geq 1$, $\sigma^2 > 0$, $0 < \varphi$, and $\mathcal{E} > 0$
  Learning Iterations: at step $t = 0, 1, 2, .., :$ do
  Optimal Search: $\underset{t}{\text{minimize}}\, E(S, W)$ using (4) for the $\Omega_t$ and compute $\vartheta_t$ using (10).
  Complexity: increment $nppc_{t+1} = nppc_t + 1$ at each step
  Convergence: Compare if $\vartheta_{t+1} < \vartheta_t$ or using a convenient matrix norm if, $||\Omega_{t+1} - \Omega_t|| \leq \mathcal{E}$ stop
  Compute: $\Lambda = \Omega^T \Omega \in \mathbb{R}^{n \times n}$ based on the learned full matrix of relevances $\Omega_{t+1}$
  Ranking: Rank by the magnitude of feature relevance $\Lambda_{j,j}$ for global $\Omega$ else compute ranks based on the number
    of hits from the local $\Lambda$ respectively based on the local $\Omega$ matrices using (12) and (13)
  Non-Rejection: Select and rank features by order of magnitude for which $\sum_i \Lambda_{i,j} \neq 0$
  Rejection: Reject features if $\sum_i \Lambda_{i,j} = 0$
  Return: Ranked significant, insignificant and tentative features
  End procedure
```

In order to evaluate the performance of Sofes, we opt for the mutation validation scheme (ϑ) for LVQ[9], as a pragmatic alternative to cross-validation and holdout schemes.

$$\vartheta = (1 - 2\varphi)\,\frac{1}{\#T} \sum_{(\mathbf{s},c(\mathbf{s}))\in T} \phi(s) + \left[\frac{1}{\#T} \sum_{(\mathbf{s},\hat{c}(\mathbf{s}))\in \hat{T}} 1 - \phi(s) - \frac{1}{\#\hat{T}} \sum_{(\mathbf{s},\hat{c}(\mathbf{s}))\in \hat{T}} -\phi(s)\right] + \varphi \tag{10}$$

$$\phi\left(\mathbf{s}\right) = \begin{cases} +\,1, & c(\mathbf{s}) = c(\mathbf{w}_{Q(\mathbf{s})}) \\ \quad 0, & c(\mathbf{s}) = \hat{c}(\mathbf{w}_{Q(\mathbf{s})}) \\ -\,1, & \hat{c}(\mathbf{s}) = \hat{c}(\mathbf{w}_{Q(\mathbf{s})}) \end{cases} \tag{11}$$

Where the mutation degree (φ) is a small positive yet constant parameter and the mutated set $\hat{T}$ is represented as $\{\mathbf{s}_n, \hat{c}\left(\mathbf{s}_n\right)\}_{n=1}^N \in \{\mathbb{R}^n, \mathcal{C}\}^N$. The number of significant prototype hits per feature space is defined as:

$$P_{i^*} = \#\{\mathbf{w}^L \in W|\ i^L \in \mathcal{F} \wedge \sum_i \Lambda_{ij}^L \neq 0\} \tag{12}$$

and the corresponding number of insignificant prototype hits per feature space is given by:

$$P_{i_*} = \#\{\mathbf{w}^L \in W|\ i^L \in \mathcal{F} \wedge \sum_i \Lambda_{ij}^L = 0\} \tag{13}$$

where L is local, $\mathcal{F}$ is the feature space, i_* and i^* are the significant (*S*) and insignificant (*I*) features under consideration respectively. If $P_{i^*} = P_{i_*}$, the feature i is designated as tentative (*T*) observing that $|\mathcal{F}| = |\{S \cup T \cup I\}|$.

The feature selection process within the Sofes framework involves considering two settings: global relevance and local relevances. Due to the complexity associated with local relevances, rejection strategies outlined in (12) and (13) are introduced. The purpose is to align the behavior of local relevances with that of the global setting, aiming at enhancing interpretability, simplicity, understandability, comparability, and easy usability. In the context of Sofes, akin to Nafes[1], employing the LMRSLVQ inductive learner within the feature subset selection algorithm 1 emerges as a highly viable option for applications, particularly in the domain of machine learning pipelines designed for prototype-based feature selection.

[1]

[1] $nppc, \sigma^2$ and $\mathcal{E}$ herein used in Sofes are user-defined parameters denoted respectively as the number of prototypes per class, variance and a small but positive number for termination. The source code for the Sofes algorithm is available at `https://github.com/naotoo1/Prototype-Based-Soft-Feature-Selection-Package`

5 Experimentation

To illustrate the performance of the proposed soft prototype-based feature selection wrapper algorithm used in the Sofes package, two real datasets, namely, the Ozone Level [13] and Wisconsin Breast Cancer (WDBC) [14] datasets, were used for all experiments. The datasets have 569 and 2536 inputs with corresponding 30 and 73 attributes, respectively.

Table 1: Classification accuracies for WDBC dataset using Nafes[1] and Sofes feature selection Packages

	GMLVQ[1]	LGMLVQ[1]	MRSLVQ	LMRSLVQ
Method	$1\|1 \xrightarrow{1,2} 3\|3,3\|3$	$1\|1 \xrightarrow{1,2} 2\|2,3\|3$	$1\|1 \xrightarrow{1,2} 5\|5,6\|6$	$1\|1 \xrightarrow{1,2} 3\|3,5\|5$
^{1}MV($\varphi = 0.1$)	94.64%	$94.89\% \overset{r}{\to} 94.89\%$	91.44%	$96.92\% \overset{r}{\to} 96.92\%$
^{2}MV($\varphi = 0.2$)	94.65%	$94.20\% \overset{r}{\to} 94.90\%$	80.14%	$97.84\% \overset{r}{\to} 97.84\%$
^{1}# features.	$30 \to 17$	$30 \to 29 \overset{r}{\to} 23$	$30 \to 30$	$30 \to 29 \overset{r}{\to} 20$
^{2}# features.	$30 \to 17$	$30 \to 29 \overset{r}{\to} 23$	$30 \to 30$	$30 \to 29 \overset{r}{\to} 21$

Observations from the results in Tables (1,2) show the GMLVQ learner performs better than the MRSLVQ learner when employed for the relevance feature selection of both the WDBC and Ozone level datasets. GMLVQ reduced the feature space of the WDBC dataset by way of relevant feature selection from 30 to 17 as compared to MRSLVQ, which weighted all features with equal relevance.

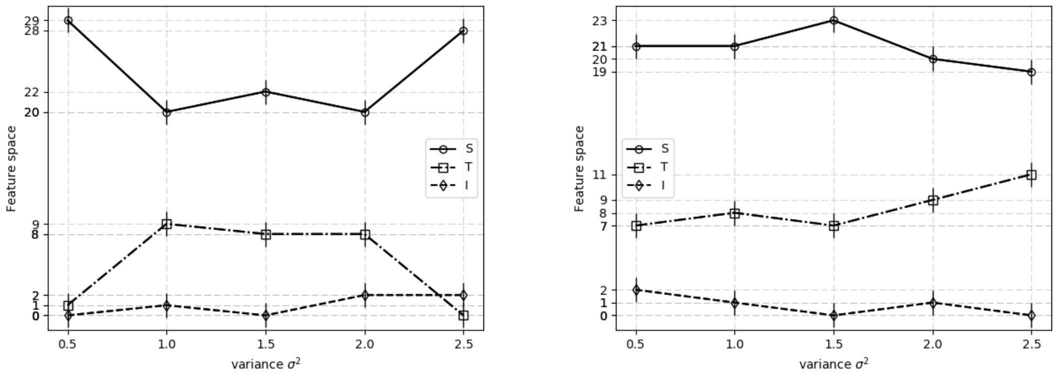
Table 2: Classification accuracies for Ozone Level dataset using Nafes[1] and Sofes feature selection Package

	GMLVQ[1]	LGMLVQ[1]	MRSLVQ	LMRSLVQ
Method	$1\|1 \xrightarrow{1,2} 2\|2,3\|3$	$1\|1 \xrightarrow{1,2} 2\|2,2\|2$	$1\|1 \xrightarrow{1,2} 2\|2,2\|2$	$1\|1 \xrightarrow{1,2} 3\|3,3\|3$
^{1}MV($\varphi = 0.1$)	93.07%	$93.21\% \overset{r}{\to} 93.21\%$	91.94%	$92.74\% \overset{r}{\to} 92.74\%$
^{2}MV($\varphi = 0.2$)	94.65%	$94.20\% \overset{r}{\to} 94.90\%$	94.33%	$93.30\% \overset{r}{\to} 93.30\%$
^{1}# features.	$72 \to 42$	$72 \to 66 \overset{r}{\to} 43$	$72 \to 72$	$72 \to 72 \overset{r}{\to} 40$
^{2}# features.	$72 \to 42$	$72 \to 21 \overset{r}{\to} 13$	$72 \to 72$	$72 \to 67 \overset{r}{\to} 49$

Similar behavior is also witnessed with the Ozone level dataset where the GMLVQ relevantly reduces the feature space from 72 to 42 as compared with the equalized relevance weighting by the MRSLVQ learner. Furthermore, MRSLVQ, when wrapped inductively in algorithm (1), records higher complexity (more prototypes per class) as compared with the GMLVQ inductive learner.

Due to the equalization of relevances recorded with MRSLVQ, we apply a rejection strategy based on (12,13) as shown in Figures (1,2,3) to the local variants for effective feature reduction. The rejection strategy involves identifying potential features, which are subsequently excluded from the set of significant attributes based on the number of prototype hits. Importantly, this elimination does not lead to a significant decline in performance. As a result, it allows for categorizing the feature space into three distinct groups: significant, insignificant, and tentative.

The outcomes presented in Tables (1,2) suggest that LGMLVQ[1] and LMRSLVQ induction learners exhibit a similar level of performance when it comes to reducing the relevance of features in the WDBC dataset. However, in the case of the Ozone level dataset, the LGMLVQ approach, as employed in [1], notably surpasses LMRSLVQ in feature space reduction. The performance discrepancy in LMRSLVQ compared to LGMLVQ can be attributed to the probabilistic framework utilized in LMRSLVQ, which imposes specific conditions on the input space to achieve optimal results. Key prerequisites dictate that the input space should exhibit a Gaussian distribution with minimal to no noise, as described in [15], and that the user-defined variance hyperparameter σ^2 must be finely tuned for optimal performance, as emphasized in [7] and shown in Figures(1,2,3). These conditions, however, do not strictly apply to LGMLVQ prototype-based induction learners.

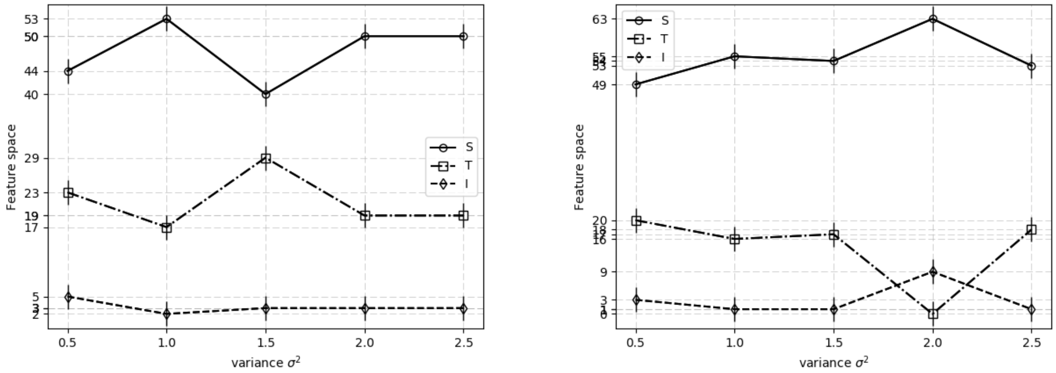


(a) LMRSLVQ ($\varphi = 0.1$)

(b) LMRSLVQ ($\varphi = 0.2$)

Figure 1: Visualization of significant '*S*', tentative '*T*' and insignificant '*I*' features with reject strategy evaluated using MV scheme against the variance σ^2 for the WDBC dataset. The *solid*, *broken* and *short-dashed lines* are indicative of the behaviour of the LMRSLVQ learner regarding the feature space designations.

Furthermore, the performance evaluation scores of the suggested prototype-based soft feature selection wrapper algorithm align with the principles of the MV scheme. The aligned behavior is particularly evident when identifying features that lead to effective learning for classification decisions. We note that soft feature selection is efficient, as explored in this study, within the context of prototype-based induction models. Efficient feature selection is especially true when combined with the necessary data preprocessing and transformations, which serve as an initial step for utilizing probabilistic-inspired learners.



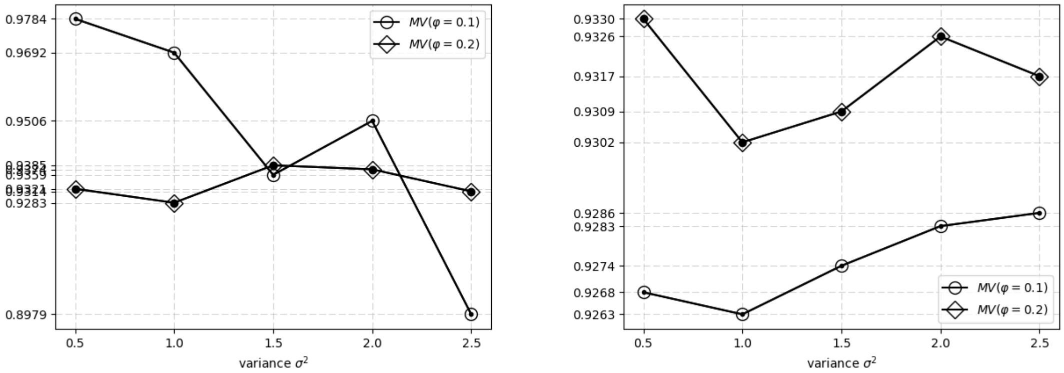
(a) LMRSLVQ($\varphi = 0.1$)

(b) LMRSLVQ($\varphi = 0.2$)

Figure 2: Visualization of significant '*S*', tentative '*T*' and insignificant '*I*' features with reject strategy evaluated using MV scheme against the variance σ^2 for the Ozone level dataset. The *solid*, *broken* and *short-dashed lines* are indicative of the behaviour of the LMRSLVQ learner regarding the feature space designations.

The Figures (1,2) depict the optimal exploration of the feature space $\mathcal{F}$, considering designations $\{S, T, I\}$, in terms of the variance σ^2 evaluated by the MV scheme ϑ based on φ. However, an interesting observation emerges: the tentative class of attributes mirrors the significant class for both the WDBC and Ozone level datasets. This mirroring effect in the tentative attribute class suggests that our rejection strategy can identify and categorize features that may appear significant but, in reality, do not significantly contribute to classification decisions. These tentative features may seem of minor use to prototype-based soft learners, indicating the need for further investigation to confirm this observation. [2]

[2] However, based on the observations in Figures (1,2,3), we propose that if the feature space is excessively large, posing challenges in handling and complexity, then the tentative class of attributes might be included in the insignificant class. Conversely, if the feature space is too small, the tentative class of attributes might be considered as part of the significant class through test ascertainment.



(a) LMRSLVQ

(b) LMRSLVQ

Figure 3: Visualization of the accuracies from the MV scheme against the variance σ^2 for LMRSLVQ with reject strategy for the WDBC and Ozone Level datasets respectively.

6 Discussion

The analysis of the numerical assessment of test results reveals that the proposed soft feature selection wrapper algorithm presents an understandable method for feature selection, akin to the Nafes package. The primary distinction lies in Sofes' utilization of prototype-based induction learners inspired by a probabilistic framework. For optimal performance of efficient feature space reduction using Sofes, practitioners should employ the local version with the optimal rejection strategy and necessary data transforms. The proposed soft feature selection algorithm is characterized by its simplicity, consistency, non-heuristic nature, and interpretability while demonstrating strong generalization capabilities. The Sofes package offers an intuitive interface, making it a valuable tool for applied machine learning practitioners and domain managers when building effective machine learning pipelines.

7 Conclusion

In this paper, a pioneering prototype-based probabilistic feature selection algorithm is introduced. The paper elucidates the mathematical underpinnings of the Sofes algorithm and exhibits experimental results derived from real-world datasets. The numerical assessment of the experiments indicates that the proposed algorithm meets the requirements of a wrapper-based feature selection approach that is highly interpretable. The validation scheme employed in the study is designed to guarantee a well-suited reduced feature set.

8 Acknowledgment and Disclosure

The researchers assert that there are no conflicts of interest associated with this work.

References

[1] Nana Abeka Otoo and Muhammad Abubakar. Prototype-based-feature-selection-with-the-nafes-package, 2023.

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

[3] 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.

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

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

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

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

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

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

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

[11] 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.

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

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

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

[15] Jason Brownlee. *Probability for machine learning: Discover how to harness uncertainty with Python*. Machine Learning Mastery, 2019.