ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Sparse Metric Learning in Prototype-based Classification

Johannes Brinkrolf and Barbara Hammer*

Machine Learning Group
Bielefeld University - Germany

Abstract. Metric learning schemes can greatly enhance distance-based classifiers, and provide additional model functionality such as interpretability in terms of feature relevance weights. In particular for high dimensional data, it is desirable to obtain sparse feature relevance weights for higher efficiency and interpretability. In this contribution, a new feature selection scheme is proposed for prototype-based classification models with adaptive metric learning. More precisely, we integrate the group lasso penalty and a subsequent optimization of sparsity while leaving the mapping invariant. We evaluate the performance on a variety of benchmarks.

1 Introduction

Recently, prototype-based models have gained increasing attention due to their ability for few shot and life-long learning, on the one hand [1, 2, 3], and their intuitive representation of models in terms of typical representatives, on the other hand [4, 5]. Applications range from classification of hyperspectral data [6] and astrophysics [7], over intelligent tutoring systems [8], up to applications in bioinformatics [9] and medicine [10]. In addition, powerful algorithmic models for learning prototype-based models exist, including not only efficient learning schemes and their mathematical substantiation [11], but also extensions which enhance models by additional functionality or properties such as the possibility of model visualization [12], data privacy [13], or reject options [14].

In this contribution, we will focus on learning vector quantization (LVQ) as particularly robust training method, and its extensions to metric learning schemes as proposed in the work [11]. Metric learning schemes, which are based on a local or global adaptive quadratic form, lead to an enhanced model interpretability, by suggesting a feature relevance weighting scheme in terms of the diagonal entries of the matrix associated to the metric. Yet, in particular for high dimensional data, relevance weightings of the features are no longer easily interpretable nor efficient due to the sheer number of the involved features. Further, it has been pointed out in the work [15] that an increasing risk of spurious relevance terms is induced by possibly high feature correlations. In this work, we aim for efficient schemes which extend LVQ models to sparse models, which are based on as few input features as possible. Such models have the potential of increased interpretability, since a human observer needs to inspect only a subset of relevant features, as is also common in popular black box model interpretation methods [16]. Additionally, a restriction to a subset of features increases model efficiency and its suitability for edge computing, since only a small set of features needs to be measured and processed. It is well known that popular

*Funding by Federal Ministry of Education and Research of Germany in the frame of the project ITS.ML (BMBF grant number 01IS18041A) is gratefully acknowledged.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

matrix learning approaches are sparse in the sense that they tend to converge to low rank matrices, as shown e.g. in the work [17]. Yet, low rank matrices can still make use of number of features, unless the relevant directions are aligned with the coordinate axes of the original data model. One first approach tackling feature sparsity is already done in [18] using a differentiable approximation of the L_1-norm but getting intricate in the case of relevance matrices.

In this contribution, we will propose a novel model which enhances LVQ schemes to aim for sparse feature relevance vectors also for local metric learning. We will achieve this goal by a combination of two modeling steps: first, we enhance the cost function of LVQ schemes by an objective, which is based on group-lasso [19]. We will show that this technique already improves sparseness, but it does not enable a sparseness which is comparable to alternatives such as random forests [20]. Based on the insights as obtained in the work [15], we add a posterior optimization step, which can be solved approximately based on orthogonal matching pursuit (OMP) [21].

2 Learning vector quantization

LVQ models offer a classification of $\mathbb{R}^d$ into K classes. The model is parameterized by w labeled prototypes $(\boldsymbol{w}_j, c(\boldsymbol{w}_j)) \in \mathbb{R}^d \times \{1, \ldots, K\}, j \in \{1, \ldots, w\}$, which induce a winner takes all classification of a new data point $\boldsymbol{x} \mapsto c(\boldsymbol{x}) := c(\boldsymbol{w}_l)$ with $l = \arg\min_{j \in \{1, \ldots, w\}} d(\boldsymbol{w}_j, \boldsymbol{x})$ where $d(\cdot, \cdot)$ is an appropriate distance measure. Training in so-called *generalized learning vector quantization* (GLVQ) constitutes a stochastic gradient descent on the costs

$$E_{\mathrm{GLVQ}} = \sum_i \Phi \left(\frac{d^+(\boldsymbol{x}_i) - d^-(\boldsymbol{x}_i)}{d^+(\boldsymbol{x}_i) + d^-(\boldsymbol{x}_i)} \right) \tag{1}$$

where $d(\boldsymbol{x})$ is the squared Euclidean distance, Φ a monotonic function such as the logistic one or the identity, and the indices $+$ and $-$ refer to the closest prototype with correct or incorrect label, respectively [22].

In *metric learning* for LVQ, the squared Euclidean distance is substituted by a parameterized form, and metric parameters are adapted together with the prototype locations. More specifically, generalized matrix LVQ (GMLVQ) uses a positive semi-definite matrix Λ and the distance measure $d(\boldsymbol{x}, \boldsymbol{w}_j) = (\boldsymbol{x} - \boldsymbol{w}_j)^T \Lambda (\boldsymbol{x} - \boldsymbol{w}_j)$. Local GMLVQ uses a different matrix Λ_j for each prototype $\boldsymbol{w}_j$ such that $d(\boldsymbol{x}, \boldsymbol{w}_j) = (\boldsymbol{x} - \boldsymbol{w}_j)^T \Lambda_j (\boldsymbol{x} - \boldsymbol{w}_j)$. Positive semi-definiteness of Λ and Λ_j is guaranteed by the representations $\Lambda = \Omega^T \Omega$ and $\Lambda_j = {\Omega_j}^T \Omega_j$, respectively [11]. Instead of using a squared matrix of size $d \times d$, a rectangle matrix can be chosen $\Omega, \Omega_j \in \mathbb{R}^{m \times d}$ [12]. This reduces the complexity of the model. Note that a quadratic form corresponds to a linear data transformation $d(\boldsymbol{x}, \boldsymbol{w}_j) = (\Omega(\boldsymbol{x} - \boldsymbol{w}_j))^2$; we will later use this representation in Section 5.

3 Group lasso

Group lasso [23] aims for a selection of features which come in J predefined groups of sizes $p_j,\ j \in \{1, \ldots, J\}$, whereby either all or none of the features of a group can be chosen. For a linear model, group lasso enhances the squared error

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

by a penalty: $\frac{1}{2}\left\|y - \sum_{j=1}^{J} \boldsymbol{X}_j \boldsymbol{\beta}_j\right\|^2 + S \cdot \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_{\boldsymbol{K}_j}$ where $\|z\|_{\boldsymbol{K}_j} = (z^T \boldsymbol{K}_j z)^{0.5}$ and $\boldsymbol{K}_j$ are positive definite matrices, often $\boldsymbol{K}_j = p_j \boldsymbol{I}$ where p_j is the size of the jth group, and the usual design matrix $\boldsymbol{X}$ and covariate vector $\boldsymbol{\beta}$ is replaced by a collection $\boldsymbol{X}_j$ and $\boldsymbol{\beta}_j$ for each group. Defining $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_J)$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \ldots, \boldsymbol{\beta}_J^T)^T$ yields the sum squared error (SSE), $\|y - \boldsymbol{X}\boldsymbol{\beta}\|^2$, which we will substitute by GLVQ cost later on. Since $p_j = d$ will be constant we choose $\boldsymbol{K}_j$ as identity matrix, which yields the simplification of the regularizer $S \cdot \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|$ where the Euclidean norm $\|z\| = (z^T z)^{0.5}$ is used.

4 Orthogonal Matching Pursuit

Matching pursuit is an approximation algorithm which enables an approximation of a signal $\boldsymbol{f}$ in terms of a sparse linear combination of elements $\boldsymbol{g}$ from a given dictionary: $\boldsymbol{f} \approx \hat{\boldsymbol{f}}_N = \sum_{n=1}^{N} a_n \boldsymbol{g}_n(t)$ where $\boldsymbol{g}_n$ is the nth column (element) of the dictionary and a_n the scaling factor, and the coefficient vector $\boldsymbol{a}$ should be sparse. Solving this problem exactly is NP-hard. Matching pursuit constitutes a greedy approximation where dictionary elements are iteratively chosen to minimize the reconstruction error by adding a single element and suitable scaling. Orthogonal matching pursuit (OMP) extends this scheme such that the orthogonal projection of the signal onto the subspace spanned by the set of already selected base elements is calculated [21].

5 Sparse LGMLVQ

We are interested in sparse models for matrix learning. A local adaptive matrix Λ_k does not use feature j iff the column j of the linear transformation Ω_k equals a vector $\mathbf{0}$, and $\Lambda_k = {\Omega_k}^T \Omega_k$. Hence we can form groups of elements $(\Omega_\bullet)_{\bullet j}$ for fixed j, and we can penalize such groups by extending the LGMLVQ-costs

$$E_{\text{SLGMLVQ}} = E_{\text{LGMLVQ}} + S \cdot \sum_{j=1}^{d} \left(\sum_{k=1}^{K} \sum_{i=1}^{m} |(\Omega_k)_{ij}|^2 \right)^{0.5} \tag{2}$$

The gradients for $\boldsymbol{w}$ are not affected by the penalization term; the gradients for the projection matrices become $\nabla_{\Omega_j} E_{\text{SLGMLVQ}} = \nabla_{\Omega_j} E_{\text{LGMLVQ}} + \boldsymbol{S}$ with

$$\boldsymbol{S}_{hl} = \begin{cases} 0 & \text{for } (\Omega_j)_{hl} = 0 \\ S \cdot \dfrac{(\Omega_j)_{hl}}{\left(\sum_{k=1}^{K} \sum_{i=1}^{m} |(\Omega_k)_{il}|^2 \right)^{0.5}} & \text{otherwise} \end{cases} \tag{3}$$

This extension shrinks the column values of Ω_j close to zero.

In practice, however, we do not necessarily observe values exactly 0, and the choice of a suitable cutoff-threshold might be problematic. Therefore, we add an efficient heuristic to choose those columns which contribute most and delete those which are close to $\mathbf{0}$. The motivation is based on the observation that the model is not changed if the linear projection induced by Ω is not altered on the data. The work [15] provides an exact characterization of the null space of this

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

mapping. Here, we aim for a substitution of Ω by a sparse matrix Ω' which leaves the data projection invariant, i.e. $\Omega'(\boldsymbol{x}_i - \boldsymbol{w}_j) \approx \Omega(\boldsymbol{x}_i - \boldsymbol{w}_j) \quad \forall \boldsymbol{x}_i,\ j \in \{1, \ldots, w\}$ where the sparsity s is measured by the number of columns of only zeros

$$s = 1 - \frac{1}{d} \left\| \boldsymbol{s} \right\|_0, \quad (\boldsymbol{s})_l = \sum_{i=1}^{m} |(\Omega)_{il}| \quad \forall l \in \{1, \ldots, d\}.$$

Let $({\boldsymbol{\omega}_1}^T, \ldots, {\boldsymbol{\omega}_m}^T) = \Omega$ the m rows of the projection matrix. Then, we are seeking for a sparse $\boldsymbol{\omega}_i'$ with ${\boldsymbol{\omega}'_i}^T \boldsymbol{X} \approx \boldsymbol{\omega}_i^T \boldsymbol{X}$ where $\boldsymbol{X} = (\boldsymbol{x}_1 - \boldsymbol{w}_1, \ldots, \boldsymbol{x}_m - \boldsymbol{w}_1, \ldots, \boldsymbol{x}_1 - \boldsymbol{w}_w, \ldots, \boldsymbol{x}_m - \boldsymbol{w}_w)$ is the matrix containing all pairwise differences of all samples and prototypes. We can use OMP to approximate the problem $\boldsymbol{\omega}_i^T \boldsymbol{X} =: y \approx \sum_{n=1}^{N} a_n (\boldsymbol{X}')_j$ where $(\boldsymbol{X}')_j$ is the jth column of the data matrix $\boldsymbol{X}$, which is normalized as required by OMP, and the (sparse) coefficients a_n are determined by OMP; $\boldsymbol{\omega}_i'$ results from the normalization factors. For local matrices, we can concatenate all local matrices, and apply the same procedure.

6 Experiments

We evaluate the method based on the following three data sets:

- *FRI* [24] includes an artificially generated data set of 3000 points. Linearly separable data are enriched by ten redundant features (linear combinations of the first three) and 37 uniformly distributed irrelevant features.
- *Human Activity Recognition* (HAR) [25] data set consists of time and frequency domain variables. Activities of daily living (walking, walking upstairs, walking downstairs, sitting, standing, laying) were captured by a smartphone on the waist. It consists of 5744 samples with 561 dimensions.
- *Gisette* [26] is a high dimensional data set generated for a feature selection challenge. The classification task is to distinguish between the 4 and 9. It consists of 7000 samples with 5000 dimensional feature vectors.

All data sets are balanced. We use feature-wise z-transformation and evaluate the results by their accuracy in a five-fold cross-validation. We compare the following techniques (based on scikit-learn [27] and the LVQ Matlab Toolbox [12]):

- *Logistic regression (LR)* with lasso penalty with increasing weight of the $L1$-penalty;
- *Random forests (RF)* where an increasing number of features is dropped according to the relevance given by random forests;
- *LGMLVQ* with low-rank matrices; this is obtained by applying PCA first and projecting back to full prototypes;
- *LGMLVQ (RF)* where LGMLVQ is trained on sparse features as determined by random forests and different thresholds for features selection;
- *OMP* for sparse LGMLVQ with group lasso penalty, whereby the penalty is increased with $S \in [0, 10]$ and low-rank matrices as for LGMLVQ are used, with subsequent OMP for feature selection and short retraining.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

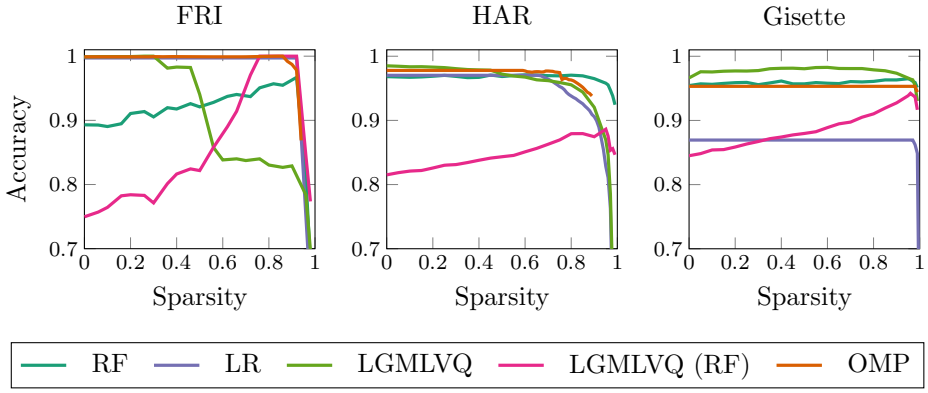

Fig. 1: Sparsity vs. accuracy of all three data sets and a fix projection dimension $m = 5$

Fig. 1 plots accuracy versus sparsity of each method and each data set. In Table 1 accuracies and standard deviations are given for five sparsity values. Some methods do not enable a high degree of sparsity, indicated by an empty entry. Interestingly, for all data sets the accuracy does stay the same up to sparsity values up to 0.9. For random forests, there are some cases where a higher sparsity improves the performance of the model. In all cases extremely sparse models lead to a high accuracy loss. At this point strongly relevant features are deleted and a good classification is not possible anymore.

From these results, we can conclude that LGMLQ alone does not enable us to obtain sparse models by means of deleting features according to the relevance weighting, but both, explicit sparsity terms or prior feature selection using random forests enable us to obtain sparse models. Surprisingly, random forest features yield a good accuracy only for a sparse number of features for both, random forests itself as well as a subsequent use by LVQ models. OMP enables a decent classification accuracy and reasonable sparsity, but in two cases it does not enable an extreme sparsity of only 2% of used features. A linear classifier is naturally restricted for nonlinear classification tasks such as HAR and Gisette.

Table 1: Mean and standard deviation in parentheses of the accuracy for the data sets out of a 5-fold cross validation and a fix projection dimension $m = 5$

data	sparsity	RF	LR	LGM	LGM (RF)	OMP
FRI	0.5	0.921 (0.0168)	0.997 (0.0035)	0.94 (0.1320)	0.822 (0.0238)	0.999 (0.0015)
	0.75	0.947 (0.0045)	0.997 (0.0035)	0.84 (0.0763)	0.993 (0.0070)	0.999 (0.0009)
	0.9	0.963 (0.0080)	0.997 (0.0035)	0.829 (0.0866)	1 (0)	0.987 (0.0241)
	0.94	0.891 (0.0125)	0.875 (0.0024)	0.799 (0.0709)	0.923 (0.0079)	0.869 (0.0765)
	0.98	0.677 (0.0065)	0.631 (0.0005)	0.698 (0.0195)	0.774 (0.0125)	—
HAR	0.5	0.968 (0.0023)	0.97 (0.0025)	0.972 (0.0035)	0.843 (0.0116)	0.978 (0.0036)
	0.75	0.969 (0.0021)	0.953 (0.0052)	0.959 (0.0026)	0.87 (0.0084)	0.974 (0.0025)
	0.9	0.965 (0.0033)	0.906 (0.0081)	0.92 (0.0083)	0.875 (0.0034)	0.938 (0.0045)
	0.94	0.959 (0.0020)	0.854 (0.0059)	0.88 (0.0108)	0.884 (0.0076)	—
	0.98	0.938 (0.0155)	0.594 (0.0311)	0.646 (0.0089)	0.856 (0.0331)	—
Gisette	0.5	0.957 (0.0052)	0.869 (0.0053)	0.981 (0.0037)	0.88 (0.0114)	0.953 (0.0086)
	0.75	0.96 (0.0058)	0.869 (0.0053)	0.981 (0.0044)	0.905 (0.0140)	0.953 (0.0086)
	0.9	0.963 (0.0083)	0.869 (0.0053)	0.974 (0.0066)	0.927 (0.0120)	0.953 (0.0086)
	0.94	0.965 (0.0059)	0.869 (0.0053)	0.968 (0.0062)	0.936 (0.0130)	0.953 (0.0086)
	0.98	0.959 (0.0058)	0.862 (0.0260)	0.954 (0.0072)	0.937 (0.0110)	0.953 (0.0086)

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

7 Conclusions

We have introduced an approach to obtain sparse LGMLVQ models by interpreting the relevance matrix as a projection matrix and searching for sparse approximations. In the results, we showed that it becomes possible to remove more than 90% of the features this way without deteriorating performance. Such sparse models are particularly interesting for its efficient realization in hardware, on edge devices, or real-time classification models. Furthermore a smaller model is easier to study and interpret.

References

[1] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4077–4087, 2017.

[2] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3474–3482, 2018.

[3] Viktor Losing, Barbara Hammer, and Heiko Wersing. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274, 2018.

[4] Michael Biehl, Barbara Hammer, and Thomas Villmann. Prototype-based models in machine learning. *WIREs Cognitive Science*, 7(2):92–11, 2016.

[5] David Nova and Pablo A. Estévez. A review of learning vector quantization classifiers. *Neural Computing and Applications*, 25(3-4):511–524, 2014.

[6] Thomas Villmann, Marika Kästner, Andreas Backhaus, and Udo Seiffert. Processing hyperspectral data in machine learning. In *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium*, 2013.

[7] Gert-Jan de Vries, Steffen C. Pauws, and Michael Biehl. Insightful stress detection from physiology modalities using learning vector quantization. *Neurocomputing*, 151:873–882, 2015.

[8] Benjamin Paaßen, Bassam Mokbel, and Barbara Hammer. Adaptive structure metrics for automated feedback provision in intelligent tutoring systems. *Neurocomputing*, 192:3–13, 2016.

[9] Lukas Pfannschmidt, Christina Göpfert, Ursula Neumann, Dominik Heider, and Barbara Hammer. Fri-feature relevance intervals for interpretable and interactive data exploration. In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2019, Siena, Italy, July 9-11, 2019*, pages 1–10, 2019.

[10] Rick van Veen, L. Talavera Martinez, R. V. Kogan, Sanne K. Meles, Deborah Mudali, Jos B. T. M. Roerdink, F. Massa, M. Grazzini, Jose A. Obeso, Maria C. Rodriguez-Oroz, Klaus Leonard Leenders, Remco J. Renken, J. J. G. de Vries, and Michael Biehl. Machine learning based analysis of FDG-PET image data for the diagnosis of neurodegenerative diseases. In *Applications of Intelligent Systems - Proceedings of the 1st International APPIS Conference 2018, Las Palmas de Gran Canaria, Spain, 8-12 January 2018*, pages 280–289, 2018.

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

[12] Kerstin Bunte, Petra Schneider, Barbara Hammer, Frank-Michael Schleif, Thomas Villmann, and Michael Biehl. Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Netw.*, 26:159–173, 2012.

[13] Johannes Brinkrolf, Christina Göpfert, and Barbara Hammer. Differential privacy for learning vector quantization. *Neurocomputing*, 342:125–136, 2019.

[14] Lydia Fischer, Barbara Hammer, and Heiko Wersing. Optimal local rejection for classifiers. *Neurocomputing*, 214:445–457, 2016.

[15] Alexander Schulz, Bassam Mokbel, Michael Biehl, and Barbara Hammer. Inferring feature relevances from metric learning. In *IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015*, pages 1599–1606. IEEE, 2015.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

[17] Michael Biehl, Barbara Hammer, Frank-Michael Schleif, Petra Schneider, and Thomas Villmann. Stationarity of matrix relevance LVQ. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–8, 2015.

[18] Martin Riedel, Fabrice Rossi, Marika Kästner, and Thomas Villmann. Regularization in relevance learning vector quantization using l1-norms. In *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium*, 2013.

[19] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[20] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[21] Geoffrey M. Davis, Stephane G. Mallat, and Zhifeng Zhang. Adaptive time-frequency decompositions. *Optical Engineering*, 33(7):2183 – 2191, 1994.

[22] Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 423–429. MIT Press, 1995.

[23] Ming Yuan and Yi Lin. Model Selection and Estimation in Regression With Grouped Variables. *Journal of the Royal Statistical Society Series B*, 68:49–67, 2006.

[24] Christina Göpfert, Lukas Pfannschmidt, and Barbara Hammer. Feature Relevance Bounds for Linear Classification. In *25th European Symposium on Artificial Neural Networks, ESANN 2017, Bruges, Belgium*, 2017.

[25] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition using Smartphones. In *21st European Symposium on Artificial Neural Networks, ESANN 2013, Bruges, Belgium*, 2013.

[26] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result Analysis of the NIPS 2003 Feature Selection Challenge. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.