Federated Learning Vector Quantization

Johannes Brinkrolf and Barbara Hammer*

Machine Learning Group
Bielefeld University - Germany

Abstract. Prototype-based methods such as LVQ techniques combine discriminative and generative aspects by representing models in terms of representative locations in the data space which enable an intuitive nearest-neighbor based classification. This fact has already been used in the context of incremental learners for streaming data which might be subject to drift. In this contribution, we demonstrate that this intuitive representation enables a very simple strategy for federated learning.

1 Introduction

Recently, prototype-based models have gained increasing attention due to their ability for few shot and life-long learning [1, 2, 3] and their intuitive representation of models in terms of typical representatives [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]. Efficient learning schemes and their mathematical substantiation [11, 12] have been extended to incorporate additional functionality such as the possibility of model visualization [13], data privacy [14], sparsity [15], or reject options [16].

In practice, a large amount of data is produced by distributed devices which is expedited by the internet of things (IoT) [17]. These data eventually result in big data that can be vital in uncovering hidden patterns. Many decisions are directly made on edge devices for quickness, economy or security reason. In this context, federated ML plays a major role, i. e. learning schemes which enable an efficient, possibly privacy preserving integration of the information of local data produced by individual persons towards an integrated model [18]. Here, 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]. We will rely on the fact that LVQ models naturally offer a sufficient statistics of the model, a fact which has already been used in the context of drift-resistant incremental LVQ learning schemes for streaming data [19]. Based on this observation, we propose a novel method to fuse different LVQ models to enable distributed optimization also in the context of imbalanced classes.

Now, we will briefly introduce LVQ models and metric learning and explain how we are gaining slim models out of pre-trained models on different disjoint training sets. In the experiment we compare federated LVQ with the original version on three benchmarks and two different scenarios including imbalanced classes. In this contribution, we do not yet incorporate privacy, an extension based on the results from [14] would be immediate.

*Funding by Federal Ministry of Education and Research of Germany in the frame of the project MECH.ML (BMBF grant number 01IS18053E) is gratefully acknowledged.

2 Learning vector quantization

LVQ models offer a classification of $\mathbb{R}^d$ into c classes. One model is parameterized by a set of w labeled prototypes $\mathcal{W} = \{(\boldsymbol{w}_j, c(\boldsymbol{w}_j)) \in \mathbb{R}^d \times \{1, \ldots, c\} \mid j \in \{1, \ldots, w\}\}$, 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 or dissimilarity measure. Training in so-called *generalized learning vector quantization* (GLVQ) constitutes a stochastic gradient descent on the costs

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

where $d(\boldsymbol{x}, \boldsymbol{w}^\pm) = (\boldsymbol{x} - \boldsymbol{w}^\pm)^T(\boldsymbol{x} - \boldsymbol{w}^\pm)$ is the squared Euclidean distance, Φ a monotonic function (e.g. logistic one), and $\boldsymbol{w}^\pm$ refer to the closest prototype which belongs to the same class as the sample $\boldsymbol{x}$ (+) or to a different (-), respectively [20]. In principle, more than one prototype per class can be considered; we restrict to one prototype per class in this contribution since this is often sufficient in practice and simplifies accumulation in federated LVQ. Further, LVQ can be extended to differential privacy [14], we consider standard LVQ, since we are mainly interested in the model form rather than specific training mode.

In *metric learning* for LVQ, the squared Euclidean distance is substituted by a parameterized form which is adapted together with the prototype locations. *Generalized matrix LVQ* (GMLVQ) uses a positive semi-definite matrix Λ and the distance measure $d_\Lambda(\boldsymbol{x}, \boldsymbol{w}) = (\boldsymbol{x} - \boldsymbol{w})^T\Lambda(\boldsymbol{x} - \boldsymbol{w})$. *Local generalized matrix LVQ* (LGMLVQ) uses a distinct matrix Λ_j for each prototype $\boldsymbol{w}_j$ and the distance function is $d_{\Lambda_j}(\boldsymbol{x}, \boldsymbol{w}_j) = (\boldsymbol{x} - \boldsymbol{w}_j)^T\Lambda_j(\boldsymbol{x} - \boldsymbol{w}_j)$. Positive semi-definiteness of Λ and Λ_j is guaranteed by choosing $\Lambda = \Omega^T\Omega$ and $\Lambda_j = {\Omega_j}^T\Omega_j$ [11, 13].

3 Federated Learning for LVQ

Federated learning (FL) aims for algorithmic realizations across multiple decentralized devices or servers with distributed data storage. Each server (or node) holds its own data samples. The task is to learn or improve an overarching model locally on every node without exchanging any data samples. In contrast to other techniques, such as more classical decentralized approaches, federated learning is not assuming identically distributed data samples on each node. Often, also privacy of data is realized, but this is out of the scope of this contribution.

For LVQ, federated learning can be based on the observation that LVQ models can be described by a sufficient statistics given by prototypes, metric and size of each receptive field. This information enables a merge of models by suitably accumulating these values [19]. For FL, we assume that each node i trains a separate LVQ model using only local data samples. We assume that the classes are consistent across all nodes, i. e. class labels are element of a common finite set. The model on a node i possesses a set of prototypes $\mathcal{W}_i$, where we assume that every class is represented by at most one prototype, and a projection matrix $\Omega^{(i)}$ in case of a global matrix or local projection matrices $\Omega_j^{(i)}$ for each prototype

$\boldsymbol{w}_j \in \mathcal{W}_i$ for local metrics. We obtain a new model by the following combination of the node information:

Prototypes: We calculate the position of a new prototype in class k via

$$\boldsymbol{w}_k = \frac{1}{N_k} \sum_{i=1}^{m} \sum_{\substack{\boldsymbol{w} \in \mathcal{W}_i: \\ c(\boldsymbol{w})=k}} \boldsymbol{w} \quad \forall\, k \in \{1, \ldots, c\} \tag{2}$$

where $N_k = |\{\boldsymbol{w} \mid \boldsymbol{w} \in \mathcal{W}_i : c(\boldsymbol{w}) = k,\ i \in \{1, \ldots, m\}\}|$ is the number of prototypes belonging to class k. Note, that this definition is consistent with the theoretical finding of stationary solutions of LVQ models as presented in [12]: it is shown that prototype $\boldsymbol{w}_j$ is given be the ratio of the expectations $\langle \Phi_j^i \boldsymbol{x}_i \rangle / \langle \Phi_j^i \rangle$ where Φ_j^i is ± 1 if $\boldsymbol{w}_j$ is winner and the label of $\boldsymbol{w}_j$ and $\boldsymbol{x}_i$ coincide/do not coincide, and 0 otherwise. Assuming similar values $\langle \Phi_j^i \rangle$ per node and prototype, a simple average of the prototypes approximates the complete solution.

Metric: The projection matrices Ω and Ω_j cannot be simply averaged because they are not unique, e. g., swapping two rows in Ω leads to the same solution. However, the relevance matrix itself is unique. Using N_k as above, we gain new relevance matrices Λ and Λ_k by

$$\Lambda = \frac{1}{m} \sum_{i=1}^{m} {\Omega^{(i)}}^T \Omega^{(i)} \text{ and } \Lambda_k = \frac{1}{N_k} \sum_{i=1}^{m} \sum_{\substack{\boldsymbol{w}_j \in \mathcal{W}_i: \\ c(\boldsymbol{w}_j)=k}} {\Omega_j^{(i)}}^T \Omega_j^{(i)} \quad \forall\, k \in \{1, \ldots, c\}. \tag{3}$$

Note, that this choice does not directly correspond to the stationary solution of metric learning for the full data set as described in [12], which would correspond to matrix induced by principal components of a variation of a local covariance matrix, which takes labeling into account. Hence, techniques similar to federated PCA/SVD would be required for exact solutions [21]. Here we take a different stance, because taking the mean of the relevance matrices is equivalent to calculating the mean of all distances because $\sum_{i=1}^{m} d_{\Lambda^{(i)}}(\boldsymbol{x}, \boldsymbol{w})/m = (\boldsymbol{x} - \boldsymbol{w})^T \left(\sum_{i=1}^{m} {\Omega^{(i)}}^T \Omega^{(i)}/m \right) (\boldsymbol{x} - \boldsymbol{w})$ and similar for local matrices. A computationally efficient realization for the fused model relies on projection matrices Ω and Ω_j with $\Lambda = \Omega^T \Omega$ and $\Lambda_j = {\Omega_j}^T \Omega_j$. These exist, since relevance matrices are symmetric and positive semi-definite by construction. Here, we use Schur's Algorithms to calculate the square root $\boldsymbol{A}^{1/2}$ of a matrix by a Schur decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^*$ where $\boldsymbol{T}$ is upper triangular and $\boldsymbol{Q}$ is unitary. We use the algorithm proposed in [22] which is efficient and aims for the unique *principal square root* $\boldsymbol{A}^{1/2}$ where all eigenvalues are non-negative.

4 Experiments

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

- *Image Segmentation* (IMS): [23] It consists of 2310 samples representing small patches from outdoor images with seven different classes. Each sample consists of 19 real-valued image descriptors.

Table 1: Mean and standard deviation in parentheses of the class-wise F1-score over a 5-fold cross validation for the first scenario in columns *1st scenario* and for removing training samples belonging some classes in columns *2nd scenario*

		GMLVQ			LGMLVQ		
data	class	1st scenario	2nd scenario	vanilla	1st scenario	2nd scenario	vanilla
HAR	1	0.957 (0.0048)	0.945 (0.0148)	0.956 (0.0057)	0.972 (0.0047)	0.970 (0.0054)	0.972 (0.0054)
	2	0.932 (0.0072)	0.915 (0.0173)	0.931 (0.0083)	0.953 (0.0071)	0.950 (0.0073)	0.951 (0.0047)
	3	0.936 (0.0107)	0.923 (0.0114)	0.934 (0.0107)	0.956 (0.0084)	0.954 (0.0054)	0.955 (0.0071)
	4	0.795 (0.0101)	0.805 (0.0124)	0.801 (0.0124)	0.855 (0.0050)	0.857 (0.0089)	0.856 (0.0061)
	5	0.859 (0.0045)	0.802 (0.0447)	0.861 (0.0034)	0.873 (0.0056)	0.875 (0.0060)	0.871 (0.0053)
	6	0.984 (0.0039)	0.992 (0.0047)	0.986 (0.0050)	0.997 (0.0018)	0.997 (0.0017)	0.997 (0.0022)
	avg	0.910 (0.0069)	0.897 (0.0175)	**0.911** (0.0076)	**0.934** (0.0054)	0.934 (0.0058)	0.934 (0.0051)
Image Segmentation	1	0.908 (0.0234)	0.918 (0.0310)	0.901 (0.0282)	0.991 (0.0113)	0.986 (0.0075)	0.988 (0.0176)
	2	0.995 (0.0037)	0.989 (0.0185)	0.994 (0.0058)	1 (0)	1 (0)	1 (0)
	3	0.843 (0.0270)	0.824 (0.0276)	0.888 (0.0184)	0.887 (0.0212)	0.882 (0.0355)	0.901 (0.0215)
	4	0.866 (0.0242)	0.820 (0.0088)	0.880 (0.0294)	0.914 (0.0485)	0.897 (0.0118)	0.917 (0.0452)
	5	0.684 (0.0615)	0.644 (0.0464)	0.753 (0.0145)	0.824 (0.0274)	0.806 (0.0230)	0.844 (0.0239)
	6	0.964 (0.0176)	0.921 (0.0205)	0.979 (0.0142)	0.978 (0.0047)	0.975 (0.0149)	0.992 (0.0067)
	7	0.992 (0.0096)	0.995 (0.0037)	0.992 (0.0049)	0.995 (0.0061)	0.995 (0.0037)	0.995 (0.0061)
	avg	0.893 (0.0239)	0.873 (0.0224)	**0.913** (0.0165)	0.941 (0.0170)	0.935 (0.0138)	**0.948** (0.0173)
Digits	0	0.988 (0.0096)	0.990 (0.0018)	0.986 (0.0075)	0.993 (0.0036)	0.993 (0.0046)	0.993 (0.0022)
	1	0.884 (0.0109)	0.871 (0.0218)	0.888 (0.0145)	0.969 (0.0119)	0.975 (0.0078)	0.976 (0.0082)
	2	0.945 (0.0134)	0.935 (0.0156)	0.946 (0.0187)	0.986 (0.0065)	0.985 (0.0061)	0.988 (0.0061)
	3	0.959 (0.0176)	0.943 (0.0268)	0.944 (0.0201)	0.972 (0.0192)	0.973 (0.0120)	0.971 (0.0179)
	4	0.943 (0.0155)	0.937 (0.0249)	0.946 (0.0139)	0.980 (0.0078)	0.980 (0.0110)	0.982 (0.0069)
	5	0.953 (0.0129)	0.950 (0.0250)	0.950 (0.0133)	0.971 (0.0177)	0.973 (0.0136)	0.972 (0.0124)
	6	0.963 (0.0079)	0.955 (0.0100)	0.953 (0.0162)	0.987 (0.0058)	0.988 (0.0022)	0.988 (0.0058)
	7	0.972 (0.0059)	0.972 (0.0155)	0.973 (0.0089)	0.986 (0.0086)	0.987 (0.0116)	0.985 (0.0060)
	8	0.917 (0.0278)	0.911 (0.0149)	0.914 (0.0287)	0.951 (0.0208)	0.957 (0.0129)	0.963 (0.0174)
	9	0.915 (0.0247)	0.903 (0.0230)	0.908 (0.0225)	0.957 (0.0170)	0.960 (0.0080)	0.964 (0.0182)
	avg	**0.944** (0.0146)	0.937 (0.0179)	0.941 (0.0164)	0.975 (0.0119)	0.977 (0.0090)	**0.978** (0.0101)

- *Human Activity Recognition* (HAR): [24] It 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.
- *Digit* (DIG): [23] The optical recognition of handwritten digits data set consists of small filter method on binary images of handwritten digits from zero to nine. The number of samples is 5620 with 64 dimensions.

All data sets are balanced. We evaluate the results by their performance in a five-fold cross-validation. For each training set, we use feature-wise z-transform and for the latter two benchmarks also a PCA projection into 30 dimension reducing the complexity without loosing much information. Both transforms are applied to the test sets in the evaluation. For federated LVQ we train a separate model on a disjoint subset and obtain a final model by Eqs. (2) and (3). We compare our new method with a centralized version of (L)GMLVQ using the toolbox in [13] and all training data. Two different scenarios are tested. For the first one the training set is randomly split into five disjoint sets one for each node. In the second an imbalanced subsets are generated by removing all samples belonging to class i for node i.

In Table 1 we report the averaged class-wise F1-scores on the test sets and the mean values over all classes in the rows labeled *avg* besides their standard deviations. The results for the first scenario and for both LVQ variants are in the columns named *1st scenario* and *vanilla*. All scores of the federated versions are comparable to those of the centralized ones, except for three classes of the IMS and GMLVQ. All other differences are less than 0.0150 and smaller than half of the standard deviation. The drops for the three classes, 3, 4, and 5 of

Table 2: Mean of the class-wise F1-score over a 5-fold cross validation for each node and class for the second scenario. Cells of missing classes are left in blank.

		GMLVQ						LGMLVQ					
data	class	Node 1	Node 2	Node 3	Node 4	Node 5	max	Node 1	Node 2	Node 3	Node 4	Node 5	max
HAR	1		0.758	0.874	0.951	0.932	0.951		0.810	0.879	0.970	0.959	0.970
	2	0.691		0.720	0.927	0.909	0.927	0.708		0.733	0.945	0.943	0.945
	3	0.785	0.781		0.938	0.921	0.938	0.805	0.774		0.948	0.951	0.951
	4	0.796	0.785	0.808		0.645	0.808	0.844	0.828	0.845		0.648	0.845
	5	0.849	0.837	0.860	0.700		0.860	0.864	0.860	0.864	0.692		0.864
	6	0.986	0.981	0.988	0.964	0.991	0.991	0.997	0.997	0.995	0.983	0.996	0.997
	avg	0.822	0.829	0.850	0.896	0.880		0.844	0.854	0.863	0.908	0.900	
Image Segmentation	1		0.955	0.972	0.930	0.941	0.972		0.976	0.979	0.878	0.982	0.982
	2	0.990		0.981	0.986	0.991	0.991	1		1	0.925	1	1.000
	3	0.858	0.791		0.866	0.737	0.866	0.871	0.809		0.860	0.740	0.871
	4	0.699	0.715	0.850		0.757	0.850	0.605	0.638	0.839		0.776	0.839
	5	0.652	0.749	0.641	0.706		0.749	0.815	0.745	0.650	0.704		0.815
	6	0.945	0.955	0.942	0.735	0.948	0.955	0.965	0.972	0.979	0.897	0.962	0.979
	7	0.973	0.809	0.991	0.992	0.947	0.992	0.988	0.992	0.994	0.995	0.983	0.995
	avg	0.853	0.829	0.896	0.869	0.887		0.874	0.855	0.907	0.877	0.907	
Digits	0		0.977	0.986	0.984	0.950	0.986		0.992	0.977	0.993	0.968	0.993
	1	0.866		0.796	0.884	0.808	0.884	0.950		0.792	0.952	0.875	0.952
	2	0.930	0.929		0.839	0.944	0.944	0.854	0.922		0.950	0.985	0.985
	3	0.936	0.934	0.785		0.940	0.940	0.966	0.951	0.878		0.954	0.966
	4	0.948	0.934	0.938	0.946		0.948	0.962	0.857	0.972	0.974		0.974
	5	0.854	0.943	0.932	0.878	0.935	0.943	0.931	0.958	0.970	0.898	0.952	0.970
	6	0.858	0.935	0.942	0.973	0.786	0.973	0.952	0.972	0.978	0.985	0.869	0.985
	7	0.969	0.918	0.958	0.943	0.940	0.969	0.979	0.978	0.950	0.963	0.898	0.979
	8	0.837	0.727	0.835	0.860	0.901	0.901	0.905	0.781	0.869	0.892	0.933	0.933
	9	0.775	0.845	0.874	0.753	0.870	0.874	0.768	0.899	0.909	0.742	0.861	0.909
	avg	0.886	0.905	0.894	0.896	0.897		0.919	0.923	0.922	0.928	0.922	

IMS are those, where also the vanilla version performs worse. Here, the number of training samples is an issue because we end up in having only roughly 370 samples per node (slightly more than 50 per class). The local versions do not exhibit such behavior.

For the second scenario the averaged F1-scores are given in the columns *2nd scenario*. For the GMLVQ models the scores drop slightly for the federated as well as for the vanilla version if for the latter the union of all node's trainings sets is used which contains less data. The mean of the class-wise F1-scores drops down to 0.906 (HAR), 0.909 (IMS), and 0.935 (DIG), respectively. Again, this effect does not emerge when using local relevance matrices. If looking at the performances of each single trained model for each node in Table 2, one can see that the impact of missing classes plays a big role for only a few classes on these models. For example, removing class 1, 2, or 3 in HAR, the scores of the other two classes drop due to a lower precision and therefore a lower F1-score. The same holds for class 4 and 5. Also for the other two data sets some scores drop significantly, e. g., the scores drops by up to 0.15 for digit 8 if class 1 is missing, and for digit 9 if there are no samples of class 3 in the training set. Comparing the class-wise F1-scores of our method with the best scores reached on one single node, we observe that the scores for the federated GMLVQ are comparable to the maximum obtainable score of all nods. For LGMLVQ these scores are actually at least as high as the maximum.

5 Conclusions

We have introduced an approach to obtain (L)GMLVQ models from trained models across multiple decentralized nodes for distributed optimization. In our experiments, we showed that for the homogeneous scenario our models perform as good as those models which used all the training samples. In the second scenario where classes are removed on some nodes, we show that the performances of the GMLVQ models are approximating and for the LGMLVQ the models are as good as these from the first scenario. Such models are particularly interesting for its efficient realization in hardware, on edge devices, or real-time classification models. Also, it enables training of a model on a decentralized server and optimize the models with the help of other models on different servers with new data samples without transmitting and sharing potentially sensitive data.

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] 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. IEEE, 2015.

[13] 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.

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

[15] Johannes Brinkrolf and Barbara Hammer. Sparse metric learning in prototype-based classification. In Michel Verleysen, editor, *Proceedings of the 28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2020)*. i6doc.com, 2020.

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

[17] Minhaj Ahmad Khan and Khaled Salah. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.*, 82:395–411, 2018.

[18] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2), January 2019.

[19] Xibin Zhu, Frank-Michael Schleif, and Barbara Hammer. Patch processing for relational learning vector quantization. In Jun Wang, Gary G. Yen, and Marios M. Polycarpou, editors, *Advances in Neural Networks - ISNN 2012 - 9th International Symposium on Neural Networks, Shenyang, China, July 11-14, 2012. Proceedings, Part I*, volume 7367 of *Lecture Notes in Computer Science*, pages 55–63. Springer, 2012.

[20] 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.

[21] Andreas Grammenos, Rodrigo Mendoza-Smith, Cecilia Mascolo, and Jon Crowcroft. Federated PCA with adaptive rank estimation. *CoRR*, abs/1907.08059, 2019.

[22] Edvin Deadman, Nicholas J. Higham, and Rui Ralha. Blocked Schur Algorithms for Computing the Matrix Square Root. In *Applied Parallel and Scientific Computing - 11th International Conference, PARA 2012, Helsinki, Finland, June 10-13, 2012, Revised Selected Papers*, pages 171–182, 2012.

[23] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[24] 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.