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/.

On-edge adaptive acoustic models: an application to acoustic person presence detection

Lode Vuegen and Peter Karsmakers *

KU Leuven - Dept. of Computer Science (TC CS-ADVISE)
Kleinhoefstraat 4, B2440 Geel - Belgium
lode.vuegen@kuleuven.be

Abstract. This paper validates a machine learning framework that enables processing on resource limited devices. The discussed framework allows both inference and learning to be executed on the edge. More specifically, a Least-Squares Support Vector Machine (LS-SVM) framework with a time-recursive learning algorithm is evaluated in an application where person presence is estimated based on acoustic signals only. For this purpose, a real-life acoustical dataset of 555 hours was collected in an office environment for the evaluation of the on-edge machine learning framework.

1 Introduction

Recent advances in the domain of *'Internet of Things'* (IoT) have led to a wide range of off-the-shelf connected devices developed specifically for (indoor) monitoring applications. Today's IoT devices are known to be small, energy efficient and rather cheap allowing unobtrusive integrations in domestic and public environments at a low cost. As the amount of information an IoT device captures is increasing, shifting data processing closer to the sensor becomes more important and is known by *'on-edge processing'*. On-edge processing reduces the communication bandwidth and related power consumption, and comes with the advantage that no privacy sensitive data must be transmitted. This work evaluates a machine learning framework based on a Least-Squares Support Vector Machine (LS-SVM) that enables both inference and learning on the network edge where typically only a limited amount of resources are available. For learning, a time-recursive LS-SVM strategy is adopted. This implies that all model parameters are updated in a stepwise manner from small batches of newly collected samples only and thereby reducing the required memory footprint of the embedded platform.

The use of IoT-enabled devices to obtain an intelligent behaviour and to facilitate home automation gained a lot of research interest in recent years and is known by *'smart homes'* [1]. Research has shown that residential, public and commercial buildings account for 20% to 40% of the total energy demand in the developed countries [2]. The main energy consumers are the so-called lighting, heating, ventilation and air conditioning (L-HVAC) systems and are accountable for up to 70% of the total energy consumption [2]. Reliable presence detection could dynamically control the L-HVAC devices resulting in an overall reduced energy consumption and an improved user comfort.

*Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen.

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/.

Passive infrared (PIR) sensors are currently the most commonly used type of presence (motion) detectors due to their accurate detection performance, low power requirements, and are easy to install and calibrate [5]. However, the main disadvantages of PIR based presence detection are (i) the need of line-of-sight (LOS), (ii) limited operating range and field of view (FOV), (iii) the inability to detect (near) static persons, and (iv) that it cannot provide detailed context information about the environment being monitored [5]. Although acoustics are a rich source of information that glean useful insights about the monitoring context, the use of acoustics as a sensing modality for occupancy detection received limited attention in the research community. The proposed acoustic presence detection approaches in the literature are either focussing on classifying daily activities [3, 4] or on classifying human produced sounds such as speech [6, 7] and footsteps [8]. The development of an acoustic presence detector being able to discriminate *'presence'* from *'absence'* related sounds is to the best of our knowledge not yet investigated.

The remainder of this paper is organised as follows: Section 2 explains the used on-edge adaptive machine learning algorithm. Section 3 introduces the experimental specific details of the performed experiments regarding acoustic presence detection. The obtained results are discussed in Section 4 and are followed by the final concluding remarks given in Section 5.

2 On-edge learning and classification

The on-edge adaptive acoustic classifier model used in this work is based on a least-squares support vector machine (LS-SVM) framework. A LS-SVM basically preserves the SVM methodology but introduces a simplification via equality constraints and a least-squares optimisation [9]. Let us denote the dataset by $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$ with t being the current time step and T the total number of samples. Each sample $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^D$ is a column vector in the D-dimensional input space with $y_t \in \mathcal{Y} \subset \{-1, +1\}$ being its corresponding class label or target value. Hence, the objective of a kernelised LS-SVM can be expressed to find a decision function $f : \mathcal{X} \to \mathcal{Y}$ endowed with a kernel k, where $k : \mathcal{X} \times \mathcal{X} \to \mathcal{Y}$ is a symmetric positive definite function (i.e. Gaussian radial basis function), such that $J(f) = \sum_{t=1}^{T} \big(y_t - f(\mathbf{x}_t)\big)^2 + \gamma \|f\|_{\mathcal{H}_k}^2$, with γ as regularisation parameter, is minimised in the reproducing kernel Hilbert space (RKHS). The Representer theorem tells that any solution to this problem has a representation in the form $f(\cdot) = \sum_{t=1}^{T} \beta_t k(\mathbf{x}_t, \cdot)$, i.e. as a sum of kernels centred on the data [10]. Plugging this back into the original problem leads to

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^T} J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{K}\boldsymbol{\beta}\|^2 + \gamma \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta}, \tag{1}$$

with $\mathbf{y} \in \{-1, +1\}^T$ the label vector, $\mathbf{K} \in \mathbb{R}^{T \times T}$ the dense kernel matrix, i.e. $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ of pairwise similarities, and $\boldsymbol{\beta} \in \mathbb{R}^T$ the parameter vector. From (1) the parameters $\boldsymbol{\beta}$ can obtained by solving

$$\boldsymbol{\beta} = (\mathbf{K}^\top \mathbf{K} + \gamma \mathbf{K})^{-1} \mathbf{K}^\top \mathbf{y} \tag{2}$$

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/.

due to $\mathbf{K}$ being symmetric and positive definite. Solving (2) requires $\mathcal{O}(T^3)$ operations and becomes computationally expensive in case of large-scale datasets.

The subset-of-regressors method (SR) approximates the kernel function on arbitrary points through linear combinations of kernels selected from a set of prototypes ($\mathcal{PV}$) [11]. Consider that $\mathcal{PV}$ is given by $\{\tilde{\mathbf{x}}_m\}_{m=1}^{M}$, with $M \ll T$, and is defined by the first M samples in $\{\mathbf{x}_t\}_{t=1}^{T}$. Hence, the true kernel is approximated by $\mathbf{K} \approx \mathbf{K}_{(TM)}\mathbf{K}_{(MM)}^{-1}\mathbf{K}_{(TM)}^{\top}$ with $\mathbf{K}_{(MM)} \in \mathbb{R}^{T\times T}$ being the kernel matrix corresponding to the prototypes (i.e. the upper left $M \times M$ submatrix of $\mathbf{K}$) and $\mathbf{K}_{(TM)} \in \mathbb{R}^{T\times M}$ being the first M columns of $\mathbf{K}$. The problem given by (1) can now be rewritten into

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^M} J(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{K}_{(TM)}\boldsymbol{\beta}\|^2 + \gamma\boldsymbol{\beta}^{\top}\mathbf{K}_{(MM)}\boldsymbol{\beta}, \tag{3}$$

which has as solution

$$\boldsymbol{\beta} = (\mathbf{K}_{(TM)}^{\top}\mathbf{K}_{(TM)} + \gamma\mathbf{K}_{(MM)})^{-1}\mathbf{K}_{(TM)}^{\top}\mathbf{y}. \tag{4}$$

Despite its similarity with (2), the complexity has now been reduced to $\mathcal{O}(MT^2)$ operations. In addition, in [12] a time-recursive variant of (3) is proposed where both the parameter vector $\boldsymbol{\beta}$ and the set of prototypes in $\mathcal{PV}$ can be updated from the newly observed input-output pair $(\mathbf{x}_{t+1}, y_{t+1})$ only. In this work, we will adopt the method of [12] in the following two operating modes.

Fixed-model architecture learning: evaluates the framework when $\mathcal{PV}$ is initialised in advance by randomly drawing M samples from the entire training set resulting in a fixed-model architecture. The parameter vector $\boldsymbol{\beta}$ is updated recursively every time a new input-output pair $(\mathbf{x}_{t+1}, y_{t+1})$ is observed by using the normal step recursive least-squares (RLS) updates as proposed in [12].

Adaptive-model architecture learning: evaluates the framework when both $\mathcal{PV}$ and $\boldsymbol{\beta}$ are updated from the incoming data. This learning strategy allows the model architecture to adapt over time resulting in an adaptive-model behaviour. More specifically, the model starts with an empty set of prototypes and updates both $\mathcal{PV}$ and $\boldsymbol{\beta}$ from $(\mathbf{x}_{t+1}, y_{t+1})$ using the RLS updates of the normal, growing and pruning step as proposed in [12].

Deciding whether $\mathbf{x}_{t+1}$ must be included as prototype or not is done by the two-part criterion as proposed in [12]. Skipping the mathematical derivations and equations in [12], the first part basically measures the *'novelty'* (Δ_{nov}) of the current sample, i.e. whether it is sufficiently different from those already stored in $\mathcal{PV}$, and the second part determines the *'usefulness'* (Δ_{use}) of the candidate prototype, i.e. the reduction in regularised cost. Only the samples having a $\Delta_{nov}\Delta_{use} > \tau$ will be added as prototype.

3 Experimental setup

In this work the previously discussed on-edge learning and classification framework is evaluated in the context of presence detection based on acoustic signals.

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/.

3.1 Dataset

A real-life acoustic dataset was recorded in an office environment with dimensions 7.2×6.6 meter. The acoustic sensor was placed next to the door entrance at an height of 1.2 meter and the used sampling frequency was set to 32 kHz with a 16 bit resolution. In total 555 hours of data, i.e. 105 hours of presence and 450 hours of absence, were recorded over a period of 28 days and was labelled on-the-fly (i.e. person count) by means of pressing a button press when entering or leaving.

In total four independent folds were generated for the experiments with a ratio of 75% for training and 25% for testing. Next, the training sets were balanced to an equal number of presence and absence instances in order to eliminate the potential influence of class imbalance during training.

3.2 Acoustic feature extraction

The used acoustic features are the well-known Mel-Frequency Cepstral Coefficients (MFCC). An energy based sound activity detector was used to determine the acoustical relevant parts in the data, and only the MFCCs are extracted when the data contains sufficient energy. In this work, default 14^{th} order MFCCs are computed from the data on a 25 ms window basis with a frame shift of 10 ms.

Next, the MFCCs are processed into a statistical representation containing the cepstral means and standard deviation in order to reduce the number of feature vectors per time instance. The latter is done on a 500 ms window basis (50 frames) with an overlap of 250 ms (25 frames).

3.3 Evaluation score

The used evaluation score for tuning the hyperparameters (i.e. RBF-kernel bandwidth and regularisation parameter) and to analyse the obtained results is based on the *'true positive rate'* (tpr) and *'false positive rate'* (fpr). More specifically, the score is defined by $d = \sqrt{fpr^2 + (1 - tpr)^2}$ and needs to be minimised. The motivation to use this score is that we want to find a solution such that the Euclidean distance to the optimal condition, i.e. $tpr = 1$ and $fpr = 0$, is minimised since missed detections cause considerable user inconvenience, e.g. turning off the L-HVAC system when someone is still present in the environment, while a large number of false presence detections leads to lower energy savings.

4 Results

The obtained results for both the fixed-model and adaptive-model architecture are shown in Figure 1 and are compared to an offline SVM solution where all model parameters are estimated in batch mode (i.e. no recursive updates). The relation between the max. number of prototypes (M) in $\mathcal{PV}$ and the amount of seen data is examined regarding presence detection performance. All hyperparameters, i.e. RBF-kernel bandwidth, regularisation parameter and τ (in case of

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/.

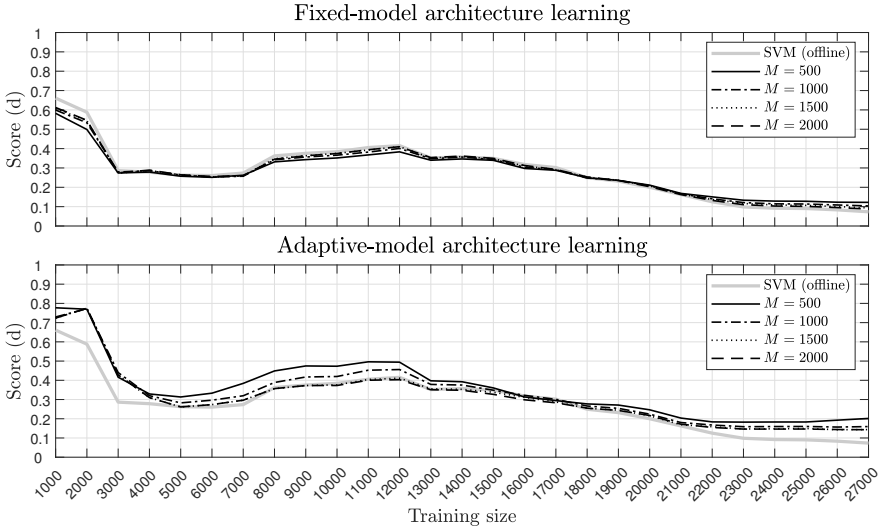

Fig. 1: The obtained presence detection results in function of the number of seen data for both operating modes.

adaptive-model architecture learning), were tuned in advance by a grid-search on all data. Note that an evaluation on the test set is done after each 1000 updates.

By analysing the fixed-model architecture results it can be clearly seen that similar presence detection scores are obtained as with the offline SVM setting. However, the main advantage of the fixed-model architecture solution is (i) that the parameter vector $\boldsymbol{\beta}$ can be updated recursively without the need of recalling all past seen samples and (ii) that the model complexity can be directly controlled by the number of prototypes in $\mathcal{PV}$. The final presence detection scores yield a true positive rate (tpr) of 93.5% and a false positive rate (fpr) of 10.3% when $M = 500$. Increasing the number of prototypes to $M = 2000$ further improves the presence detection performance to a tpr of 94.6% and a fpr of 6.9% but comes with a higher computational complexity. Note that the dynamical nature of the learning can be clearly seen in the region between 7000 and 12000 training samples. During this period of time a specific type of background noise was recorded, i.e. sounds related to renovation works in the building next to our office, that reduced the model performance. When the construction works were finished the model was automatically adapted to converge to a solution that has similar performance as that of the offline SVM. The adaptive-model architecture results on the other hand are slightly less accurate compared to the fixed-model architecture solution but it comes with the advantage that we do not need to initialise $\mathcal{PV}$ in advance. The reason for the decreased presence detection performance is basically due to higher number of model parameters needing to be learned from the same amount of data (i.e. more complex learning task). The final obtained presence detection results are a tpr of 82.9% and a fpr of 10.3% for $M = 500$, and a tpr of 92.2% and a fpr of 11.8% for $M = 2000$.

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/.

5 Conclusion

This paper discusses a Least-squares Support Vector Machine (LS-SVM) learning framework that can be deployed near the sensor on the extreme edge. Two different operating modes were examined, i.e. a fixed-model architecture and an adaptive-model architecture, employing a time-recursive learning strategy and were compared to an offline SVM solution acting as a baseline. The introduced framework was validated in the application of acoustic presence detection on 555 hours of real-life data collected in an office environment. The obtained results indicate that the fixed-model architecture operating mode achieves similar presence detection scores compared to the offline SVM solution. The adaptive-model architecture results on the other hand are slightly less accurate, but comes with the advantage that no initialisation is required. Future research will mainly focus on (i) further improving the adaptive-model architecture operating mode, (ii) the use of a multi-modal dataset to further improve the overall detection performance and (iii) the development of a real-life demonstrator embedded on a microcontroller (i.e. ARM Cortex M7).

References

[1] B. L. R. Stojkoska and K. V. Trivodaliev, A review of Internet of Things for smart home: Challenges and solutions, *Journal of Cleaner Production*, 140:1454-1464, Elsevier, 2017.

[2] L. Pérez-Lombard, J. Ortiz and C. Pout, A review on buildings energy consumption information, *Journal of Energy and Buildings*, 40:394-398, Elsevier, 2008.

[3] J. Mun Sim, Y. Lee and O. Kwon, Acoustic Sensor Based Recognition of Human Activity in Everyday Life for Smart Home Services, *International Journal of Distributed Sensor Networks*, 11:1-11, 2015.

[4] S. Ntalampiras, I. Potamitis and N. Fakotakis, Acoustic Detection of Human Activities in Natural Environments, *Journal of the Audio Engineering Society*, 60:686-695, 2012.

[5] T. Labeodan, W. Zeiler, G. Boxem and Y. Zhao, Occupancy measurement in commercial office buildings for demand-driven control applications - A survey and detection system evaluation, *Journal of Energy and Buildings*, 93:303-314, Elsevier, 2015.

[6] Q. Huang, Occupancy-Driven Energy-Efficient Buildings Using Audio Processing with Background Sound Cancellation, *Journal of Buildings*, 8:78-94, MDPI, 2018.

[7] S. Chen, J. Epps, E. Ambikairajah and P. N. Le less, An Investigation of Crowd Speech for Room Occupancy Estimation, *Proceedings of INTERSPEECH 2017*, August 20-24, Stockholm (Sweden), 2017.

[8] J. M. Sabatier and A. E. Ekimo, A Review of Human Signatures in Urban Environments Using Seismic and Acoustic Methods, *2008 IEEE Conference on Technologies for Homeland Security*, May 12-13, Waltham (Massachusetts, USA), 2008.

[9] J. A. K. Suykens and J. Vandewalle, Least Squares Support Vector Machine Classifiers, *Journal of Neural Processing Letters*, 9(3):293-300, Springer, 1999.

[10] Y. Engel, S. Mannor and R. Meir, The kernel recursive least squares algorithm, *IEEE Transactions on Signal Processing*, 52(8):2275?2285, IEEE, 2004.

[11] A. J. Smola and B. Schölkopf, Sparse greedy matrix approximation for machine learning, *Proceedings of International Conference on Machine Learning* (ICML 2000), June 29 - July 2, Stanford (California, USA), 2000.

[12] T. Jung and D. Polani, Sequential Learning with LS-SVM for Large-scale Data Sets, *Proceedings of Artificial Neural Networks* (ICANN 2006), pages 381-390, Sept. 10-14, Athens (Greece), 2006.