Advanced Machine Learning Methods for Autonomous Classification of Ground Vehicles with Acoustic Data

Xingchi Liu[1], Qing Li[2], Jiaming Liang[2], Jinzheng Zhao[3], Peipei Wu[3], Chenyi Lyu[1], Shidrokh Goudarzi[3], Jemin George[4], Tien Pham[4], Wenwu Wang[3], Lyudmila Mihaylova[1], and Simon Godsill[2]

[1]The University of Sheffield, UK
[2]The University of Cambridge, UK
[3]The University of Surrey, UK
[4]US Army Research Laboratory, USA

ABSTRACT

This paper presents a distributed multi-class Gaussian process (MCGP) algorithm for ground vehicle classification using acoustic data. In this algorithm, the harmonic structure analysis is used to extract features for GP classifier training. The predictions from local classifiers are then aggregated into a high-level prediction to achieve the decision-level fusion, following the idea of divide-and-conquer. Simulations based on the acoustic-seismic classification identification data set (ACIDS) confirm that the proposed algorithm provides competitive performance in terms of classification error and negative log-likelihood (NLL), as compared to an MCGP based on the data-level fusion where only one global MCGP is trained using data from all the sensors.

Keywords: Machine learning, Gaussian process, acoustic data, classification, surveillance

1. INTRODUCTION

A vehicle classification system captures the signals emitted from passing vehicles using varying types of sensors including RADAR, LIDAR, magnetic, and acoustic sensors[1], and identify the categories of the vehicles in terms of the captured signals. To solve the classification problem, the acoustic sensors which collect audio signals using microphone arrays have been an attractive option for several reasons. First, this modality is cost-effective as compared to other kinds of sensors. Second, audio data collection does not suffer from the existence of obstacles. Finally, this type of sensor is less invasive than other methods in terms of privacy[2].

However, since the performance of the acoustic sensors can be easily contaminated by environment noise, how to extract effective features that represent the characteristic of the audio signal is a non-trivial task. Multiple methods based on harmonic line association (HLA)[3], discrete wavelet transform[4], and wavelet packet transform[5] have been proposed for extracting features from acoustic signals. Besides the environment noise, the non-stationary phenomena during recording, including vehicle state, recording conditions, and testing sites, make the classification task even more challenging. Thus advanced classification algorithms are necessary to ensure reliable classification performance.

Various machine learning algorithms have been applied to solve the classification problem. For example, estimated harmonics' amplitudes have been used as the features and a multi-layer neural network was trained for decision making[6]. A fuzzy logic rule-based classifier was designed based on the HLA feature vector[7]. There are also other methods to solve the acoustic classification problem, such as using Gaussian mixture models[8], and support vector machine[9]. Recently, deep learning-based classification methods are also proposed for acoustic classification. For example, convolutional networks are used to decide the class of an audio clip, trained on log mel spectrogram[10].

Although the ground vehicle classification problem has been extensively studied in terms of feature extraction and decision making, there are few works about introducing probabilistic learning methods to solve the problem

E-mail: {xingchi.liu, clyu5, l.s.mihaylova}@sheffield.ac.uk, {ql289, jl809, sjg30}@cam.ac.uk, {s.goudarzi, j.zhao, p.wu, w.wang }@surrey.ac.uk, {jemin.george.civl, tien.pham1.civ}@army.mil

by producing outputs with uncertainty quantification. Multi-class Gaussian process (MCGP)[11,12], as a Bayesian non-parametric classification method, has been studied in the past few years and is a promising option for vehicle classification. Particularly, the recent works in distributed Gaussian process (DGP)[13,14] provide an approach to aggregate predictions from local GP classifiers, which can be treated as a promising way to achieve decision fusion in wireless sensor networks.

Inspired by these novel techniques[13–15], in this paper, a distributed MCGP algorithm is proposed to solve the ground vehicle classification problem using acoustic data. The proposed algorithm achieves a robust performance regardless of the vehicle speed, recording distance, and the test site. By introducing a tractable decision-level fusion, the classifier is designed to be resilient to environmental noise. In addition, via a sparse representation of GP and by considering training multiple local GP classifiers for each information source instead of a single and global one (based on the whole data set), the scalability of MCGP is further improved. The proposed algorithm is evaluated on the acoustic-seismic classification identification data set (ACIDS).

In brief, the main contributions of this work are the followings: 1) we adopt an MCGP method to classify noisy and non-stationary acoustic data from multiple classes of ground vehicles; 2) a distributed Gaussian process method is introduced to achieve the decision-level fusion which helps the classifier to be resilient to environmental noises and reduces computational costs; 3) the proposed algorithm is validated over a real data set. The distributed MCGP classification method offers competitive performance as compared to the MCGP based on the data-level fusion where only one global MCGP is trained using data from all the sensors

The remaining part of this paper is structured as follows. Section 2 gives a formulation of the considered MCGP classification problem as specified in.[15] Section 3 presents DGP methods which we use to achieve a decision-level fusion for ground vehicle classification, followed by performance evaluation in Section 4. Finally, Section 5 summarizes this paper and discusses future work.

2. MULTI-CLASS GAUSSIAN PROCESS CLASSIFICATION

Below we describe the MCGP classification framework, the formulation adopted in this paper is from[15].

2.1 Labeling Rule

Assume there is a data set of N instances. The input data can be defined as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)^\intercal$ and the corresponding target class label can be defined as $\mathbf{y} = (y_1, y_2, \cdots, y_N)^\intercal$. In a multi-class classification problem, denote C as the number of classes, for a target label i, we have $y_i \in (1, 2, \cdots, C)$. The aim is to train a classifier to predict the label y_* of any test input $\mathbf{x}_*$.

To formulate an MCGP classification problem, we first define C latent functions as $f_c(\cdot)$, with $c \in \mathcal{C} = \{1, 2, \cdots, C\}$. Each of them corresponds to a different class. Based on the latent functions, the labeling rule can be written as

$$y_i = \arg\max_{c \in \mathcal{C}} f_c(\mathbf{x}_i). \tag{1}$$

The rationale of assigning a label behind this rule is based on the value of the latent function which has the largest output at input $\mathbf{x}_i$.

Following the labeling rule, define $\mathbf{f}(\mathbf{x}_i) = (f_1(\mathbf{x}_i), f_2(\mathbf{x}_i), \cdots, f_C(\mathbf{x}_i))^\intercal$ as the set of values of C latent functions, the distribution of the label y_i conditional on $\mathbf{f}(\mathbf{x}_i)$ can be given by

$$p(y_i | \mathbf{f}(\mathbf{x}_i)) = \prod_{c \neq y_i} H(f_{y_i}(\mathbf{x}_i) - f_c(\mathbf{x}_i)), \tag{2}$$

where $H(\cdot)$ is a unit step function.

2.2 Gaussian Process for Classification

In order to solve the multi-class classification problem via GP, a GP prior is placed on each latent function, which can be represented as

$$f_c(\mathbf{x}) \sim \mathcal{GP}(0, k_{\mathbf{x},\mathbf{x}'}), \tag{3}$$

where $k_{\mathbf{x},\mathbf{x}'}$ denotes the covariance function.

Define $\mathbf{f} = \{\mathbf{f}_i\}_{i=1}^{N}$. To make predictions of the latent functions and further classify any test data, based on the likelihood (2) and the GP prior (3), assuming independence among the latent functions $\mathbf{f}_c$, we can derive the posterior distribution following Bayes' rule as

$$p(\mathbf{f}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) = \prod_{i=1}^{N} p(y_i|\mathbf{f}(\mathbf{x}_i)) \prod_{c=1}^{C} p(\mathbf{f}_c), \tag{4}$$

where $\mathbf{f}_c = (f_c(\mathbf{x}_1), f_c(\mathbf{x}_2), \cdots, f_c(\mathbf{x}_N))^\intercal$.

The likelihood presented in (2) is non-Gaussian, which leads to an intractable inference since computing the exact posterior is infeasible. Therefore, the posterior (4) needs to be approximated.

2.3 Sparse Gaussian Process

In addition to the non-Gaussian likelihood problem discussed in the previous section, another major difficulty in the GP-based MCGPC method is that the computational complexity related to prediction grows cubically in the number of data points due to the inversion and determinant of the prior covariance matrix. This limits the scalability of GP and makes it inefficient to solve classification and regression problems with large-scale data sets.

One popular and reliable approach to reduce the computational cost is to obtain a sparse approximation of the original $N \times N$ covariance matrix to summarize the dependence of the whole training data using M inducing points (also referred to as pseudo points)[16]. Define $\mathbf{Z}_c = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M)$ as a set of inducing points for each latent function, which will lie in the same space as the training input. Associated with the inducing points, we define the corresponding outputs (inducing variables) as $\mathbf{u}_c$. The value of the latent function at $\mathbf{x}_i$ can then be obtained from the predictive distribution of a Sparse GP as

$$p(\mathbf{f}_c \mid \mathbf{u}_c) = \mathcal{N}\left(\mathbf{f_c}|K^c_{\mathbf{X},\mathbf{Z}_c}(K^c_{\mathbf{Z}_c,\mathbf{Z}_c})^{-1}\mathbf{u}_c, K^c_{\mathbf{X},\mathbf{X}} - K^c_{\mathbf{X},\mathbf{Z}_c}(K^c_{\mathbf{Z}_c,\mathbf{Z}_c})^{-1}K^c_{\mathbf{Z}_c,\mathbf{X}}\right), \tag{5}$$

where $K^c_{\mathbf{X},\mathbf{Z}_c}$ is a $N \times M$ covariance matrix of function $(\mathbf{f}_c)$ between the values at the data inputs $\mathbf{X}$ and the inducing points $\mathbf{Z}_c$. $K^c_{\mathbf{Z}_c,\mathbf{Z}_c}$ is the covariance matrix among the inducing points. Importantly, now only the $M \times M$ covariance matrix $K^c_{\mathbf{Z}_c,\mathbf{Z}_c}$ needs to be inverted, and the computational complexity is reduced to $\mathcal{O}(NM^2)$, which is significantly low considering $M \ll N$. Under sparse GP, the prior distribution of $\mathbf{u}_c$ can be written as $p(\mathbf{u}_c) = \mathcal{N}(\mathbf{u}_c \mid 0, K^c_{\mathbf{Z}_c,\mathbf{Z}_c})$.

In practice, the values of the inducing variables $\mathbf{u}_c$ are unknown and are treated as latent variables which can be approximated as a Gaussian distribution $q(\mathbf{u}_c)$. The approximated distribution can be learned via variational inference, which will be described in Section 2.5.

2.4 Multi-Class GP Classification

Combining all the aspects discussed in the previous sections, the joint distribution of all the latent variables and observed variables can be written as

$$p(\mathbf{F}, \mathbf{U}, \mathbf{y}) = \prod_{i=1}^{N} p(y_i|\mathbf{f}(\mathbf{x}_i)) \prod_{c=1}^{C} p(\mathbf{f}_c|\mathbf{u}_c)p(\mathbf{u}_c), \tag{6}$$

where $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_N)^\intercal$ is the matrix of latent function values at inputs $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)^\intercal$. $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_C)^\intercal$ is the matrix of inducing variables, and $\mathbf{y}$ is the vector of labels.

The posterior distribution of the latent variables $\mathbf{F}$ and $\mathbf{U}$ can be acquired via Bayes's rule as

$$p(\mathbf{F}, \mathbf{U} \mid \mathbf{y}) = \frac{p(\mathbf{F}, \mathbf{U}, \mathbf{y})}{p(\mathbf{y})} \tag{7}$$

2.5 Posterior Approximation based on Variational Inference

In this section, the posterior distribution is approximated using variational inference. The approximation (which is also referred to as variational distribution) to the exact posterior (7) can be written as

$$q(\mathbf{F}, \mathbf{U}) = \prod_{c=1}^{C} p(\mathbf{f}_c|\mathbf{u}_c) q(\mathbf{u}_c), \tag{8}$$

where $q(\mathbf{u}_c)$ is Gaussian distributions.

To achieve an analytical approximation, the variational distribution needs to be as similar to the target posterior as possible. This problem is solved by minimizing the Kullback-Leibler (KL) divergence between the variational distribution $q(\mathbf{F}, \mathbf{U})$ and the target posterior distribution $p(\mathbf{F}, \mathbf{U}|\ \mathbf{y})$. This can be solved equivalently by maximizing the evidence lower bound (ELBO) [17], which can be written as

$$\text{ELBO} = \mathbb{E}_q \left[\log \frac{p(\mathbf{F}, \mathbf{U}, \mathbf{y})}{q(\mathbf{F}, \mathbf{U})} \right] = \sum_{i=1}^{N} \mathbb{E}_q \left[\log p(y_i|\mathbf{f}(x_i)) \right] - \sum_{c=1}^{C} \text{KL}(q(\mathbf{u}_c)|p(\mathbf{u}_c)), \tag{9}$$

where $\text{KL}(\cdot|\cdot)$ is the KL divergence and $\mathbb{E}_q[\cdot]$ is the mathematical expectation operation. Please see Villacampa-Calvo et al.[15] for further details about how to compute a stochastic estimate of the ELBO and how to use the posterior approximation for target class prediction.

3. DISTRIBUTED GAUSSIAN PROCESS

The previous section describes building an MCGP to solve the classification problem. In practice, the data can be from multiple sources and fusing all the data to train a global classifier can incur high computational costs and may overlook the potential local features of individual information sources. In this section, inspired by the idea of divide-and-conquer[13], the distributed GP (DGP) methods are introduced to achieve a decision-level fusion, by first training local MCGPs based on data from each information source and then aggregating predictions from local MCGPs to a more reliable high-level prediction. In addition, DGP can also help to reduce the computational cost since each local GP only needs to deal with a part of the data.

The first type of DGP schemes is the product-of-experts (PoEs)[18]. The idea of this approach is to multiply the local predictive distributions for an overall decision. Given the data $D^{(s)}$ collected by sensor s, the PoE predicts a latent function value $f(\mathbf{x}_*)$ at a corresponding test input $\mathbf{x}_*$ according to

$$p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}) = \prod_{s=1}^{S} p_s(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}^{(s)}), \tag{10}$$

where S is the number of local GPs (sensors). Moreover, $\mu_i(\mathbf{x}_*)$ and $\sigma_i^2(\mathbf{x}_*)$ represent the predicted mean and variance of the s-th local GP. Since the product of these Gaussian predictions is proportional to a Gaussian distribution, the closed form of the aggregated predicted mean and variance can be calculated as

$$\mu_*^{\text{PoE}} = (\sigma_*^{\text{PoE}})^2 \sum_{s=1}^{S} \sigma_s^{-2}(\mathbf{x}_*) \mu_s(\mathbf{x}_*), \tag{11}$$

$$(\sigma_*^{\text{PoE}})^{-2} = \sum_{s=1}^{S} \sigma_s^{-2}(\mathbf{x}_*). \tag{12}$$

The PoE model provides a straightforward way to aggregate local predictions and sidesteps the weight assignment problem in other local approximated GP models such as the mixture-of-expert model[19]. However, this model becomes overconfident when making predictions, especially in regions without any training data.

The generalised product-of-experts (GPoE) model[20] improves PoE by adding weights β to the local predictions, which can reflect the contributions of different local GPs. In this work, we consider using the number of training instances per class as the weight, which is then normalized by the size of the training data. This ensures $\sum_i^S \beta_i = 1$. The GPoE predicts a function value $f(\mathbf{x}_*)$ at a test input $\mathbf{x}_*$. The predicted distribution and the closed forms of the aggregated predicted mean and variance can be written as

$$p(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}) = \prod\nolimits_{s=1}^{S} p_s^{\beta_s}(f(\mathbf{x}_*) \mid \mathbf{x}_*, \mathcal{D}^{(s)}), \tag{13}$$

$$\mu_*^{\text{GPoE}} = (\sigma_*^{\text{GPoE}})^2 \sum\nolimits_{s=1}^{S} \beta_s \sigma_s^{-2}(\mathbf{x}_*)\mu_s(\mathbf{x}_*), \tag{14}$$

$$(\sigma_*^{\text{GPoE}})^{-2} = \sum\nolimits_{s=1}^{S} \beta_s \sigma_s^{-2}(\mathbf{x}_*). \tag{15}$$

The next section presents results from real data and performance validation of the algorithms.

4. PERFORMANCE EVALUATION AND VALIDATION

4.1 Acoustic-Seismic Classification Identification Data Set

The proposed method is evaluated on ACIDS, which is an ideal data set for developing and training acoustic classification/identification algorithms. This data set contains acoustic and seismic time series data collected from 9 different types of ground vehicles as they pass by a fixed location. A three-element equilateral triangular microphone array and a seismic sensor located at the center of the array are used to record the sound from each passing vehicle. The recordings were collected from four different test sites including desert, arctic, and mid-Atlantic environments, with vehicle speeds ranging from 5 to 40 km/hour, and closest point of arrival distances to the array ranging from 25 to 100 meters. Due to the varying data collection conditions, the length of the recording ranges from 56 to 420 seconds.

We use the acoustic data from the three microphones only. The characteristic of the ACIDS is presented in Table 1.

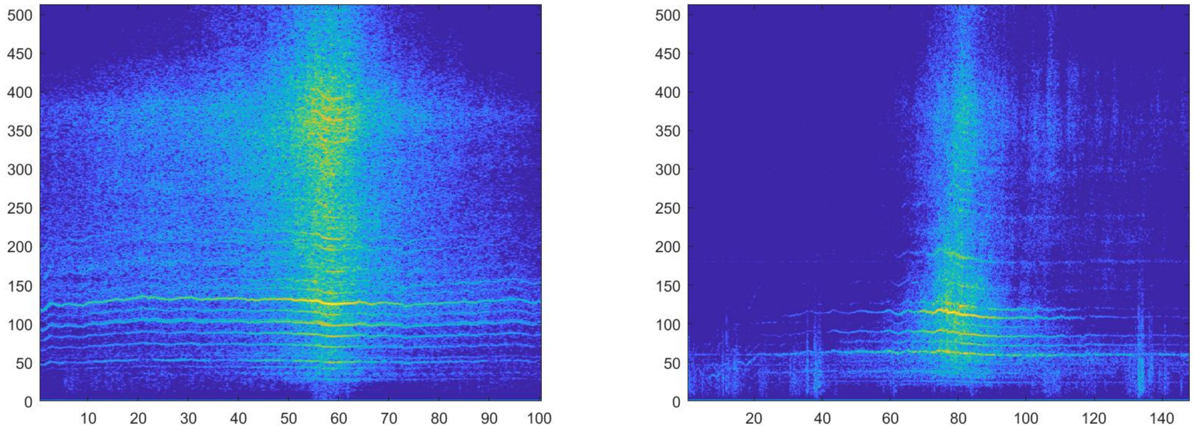
Table 1: The number of acoustic records for each type of vehicle

Vehicle type	Class label	Number of recordings
Heavy-tracked	1	62
	2	37
	8	35
	9	21
Light-tracked	4	27
Heavy-wheeled	3	9
	5	39
Light-wheeled	6	35
	7	7

The following subsection 4.2 describes the harmonics model for features extraction.

4.2 Harmonic Structure Analysis

The harmonic model[21] can extract the acoustic signal features for classification and we adopt this model here. The extracted input features include the fundamental frequency, the number of harmonic components, and the magnitude of each harmonic component. In this paper, we utilize a fast algorithm in Nielsen et al.[22] for computing the non-linear least squares estimate (NLSE) of the fundamental frequency and the number of harmonic components, based on which we can locate the harmonic frequency and its magnitude for each component.



(a) Time-frequency response of acoustic recording 6 (b) Time-frequency response of acoustic recording 29

Figure 1: Time (second)-frequency (Hz) responses of two acoustic recordings from class label 1 (i.e. heavy-tracked vehicle).

4.2.1 Data Segmentation

The harmonic structure estimated from the recording is non-stationary since the distance between the vehicle and the microphones is changing and the signal-to-noise ratio (SNR) of the data is varying. Therefore, it is impractical to utilize the whole set of data as one training instance. Instead, inspired by Wu and Mendel[7], the estimated parameters are grouped into vectors representing a number of instances, with the length of 5 seconds and 40% overlapping between contiguous blocks. Each block is treated as one training instance.

In addition, due to the different conditions under which the acoustic data is collected (e.g. different travelling speeds and different environmental conditions), the length of each recording varies. When the vehicle is far away from the microphone array, the acoustic data mainly consists of background noise, whereas when the vehicle is closer to the array, namely in the middle part of a run, the data consist of acoustic emissions of the ground vehicle as well as the noise. To eliminate the impact of background noise, we limit the training and testing data to a 40-second window centered on the midpoint of the recording data.

4.2.2 Feature extraction

Based on the data segmentation in Section 4.2.1, we extract the features over each overlapping block. Specifically, we take a Maximum a Posterior (MAP) estimate of the harmonic order over all overlapping blocks introduced in Section 4.2.1, and then this MAP harmonic order is taken as input for the estimation of fundamental frequency in each overlapping block. After obtaining the fundamental frequency, the Maximum likelihood (ML) estimation of the linear weights of each harmonic component in the harmonic model is obtained, by which the magnitude at each component can be calculated.

For some of the recordings, the applied harmonic analysis method can only identify a limited number of harmonic components. As an example, here we present two time-frequency responses of the 6-th and 29-th acoustic recordings from the same heavy-tracked ground vehicle with class label 1. We can see from Figure. 1 that the left recording's quality is better than the right one, and it is also shown in the estimation result where the 29-th recording has an underestimated harmonic order of 2, while the 6-th recording can identify 15 harmonic components. Therefore, recordings with poor quality are excluded from the MGPC training and test process. In this paper, by adjusting the threshold (number of harmonic components) of the exclusion, three data sets are built for performance evaluation and the corresponding threshold value is set to be 5, 8, and 10.

4.3 Numerical Results

The data is collected from each sensor and preprocessed via the discussed method; then we spilt it equally into the training and the test set. Since ACIDS contains three microphones, three local GP classifiers are trained independently using their local training data sets. After that, the three local classifiers make predictions based on three different test data sets (from three microphones), respectively. The aggregation method is then applied to fuse the local predictions into a global prediction. Table 2 shows the training error and the corresponding negative log-likelihood (NLL) of each local classifier and a global MCGP classifier based on data from all the sensors. The used training and test data are based on selecting recordings from the original ACIDS which meet the exclusion threshold of having at least 5 harmonics. The test errors and NLLs of each local MCGP classifier, the distributed classifier based on two aggregation methods, and the global MCGP classifier are presented in Table 3. We observe that as compared to the predictions from the local classifiers, the aggregated prediction has a lower NLL and competitively well prediction performance. For the third test set, the aggregation result outperforms the local classifier. In addition, by considering the weight of each local classifier, GPoE outperforms PoE with both lower test error and test NLL. Finally, as compared to the case when only a global MCGP is trained based on data of all the sensors, the proposed method performs competitively well in terms of the test error and even achieves a smaller NLL. This demonstrates that when the data is relatively less representative, DGP-based MCGP can achieve reliable classification with lower computational cost.

Table 2: Average training error and NLL: harmonic threshold=5

	Training set 1	Training set 2	Training set 3	Global MCGP
Training error	0.1814	0.2099	0.1908	0.2290
Training NLL	1.5071	1.8486	1.5741	1.8225

Table 3: Average test error and NLL: harmonic threshold=5

	Test data set 1		Test data set 2		Test data set 3	
	Test error	NLL	Test error	NLL	Test error	NLL
Local MCGP 1st sensor	0.3179	2.1751	0.3013	2.1423	0.3014	2.0817
Local MCGP 2nd sensor	0.3298	2.4501	0.3319	2.3512	0.2972	2.2491
Local MCGP 3rd sensor	0.2982	2.0780	0.2782	2.0357	0.2965	1.9826
Distributed MCGP-PoE	0.3003	1.7653	0.2831	1.6427	0.2851	1.6082
Distributed MCGP-GPoE	0.2989	1.7600	0.2810	1.4555	0.2766	1.4467
Global MCGP	0.2968	2.1574	0.2789	2.0651	0.2645	1.9978

Table 4: Average training error and NLL: harmonic threshold=8

	Training set 1	Training set 2	Training set 3	Global MCGP
Training error	0.0746	0.0711	0.0539	0.0699
Training NLL	0.3933	0.4844	0.4361	0.4817

Table 5: Average test error and NLL: harmonic threshold=8

	Test data set 1		Test data set 2		Test data set 3	
	Test error	NLL	Test error	NLL	Test error	NLL
Local MCGP 1st sensor	0.1153	0.5593	0.1283	0.6188	0.1633	0.8157
Local MCGP 2nd sensor	0.1294	0.5859	0.1446	0.6508	0.138	0.755
Local MCGP 3rd sensor	0.1211	0.6764	0.1242	0.7028	0.1263	0.7191
Distributed MCGP-PoE	0.1111	0.5815	0.1225	0.6024	0.1271	0.7104
Distributed MCGP-GPoE	0.1095	0.5720	0.1062	0.5068	0.1263	0.6556
Global MCGP	0.0887	0.5247	0.1013	0.5666	0.1128	0.6677

Table 6: Average training error and NLL: harmonic threshold=10

	Training set 1	Training set 2	Training set 3	Global MCGP
Training error	0.0248	0.0262	0.0250	0.0263
Training NLL	0.2189	0.2086	0.2056	0.1870

Table 7: Average test error and NLL: harmonic threshold=10

	Test data set 1		Test data set 2		Test data set 3	
	Test error	NLL	Test error	NLL	Test error	NLL
Local MCGP 1st sensor	0.1102	0.5585	0.1024	0.6087	0.1125	0.6436
Local MCGP 2nd sensor	0.0923	0.3907	0.0958	0.4251	0.0861	0.4049
Local MCGP 3rd sensor	0.1074	0.5163	0.1273	0.5972	0.0903	0.3784
Distributed MCGP-PoE	0.0964	0.4935	0.1024	0.5169	0.0819	0.4213
Distributed MCGP-GPoE	0.0895	0.4809	0.0853	0.4679	0.0611	0.3586
Global MCGP	0.0606	0.3066	0.0656	0.3340	0.0583	0.2831

Table 8: Training time in second

	Harmonic threshold		
	5	8	10
Training set 1	987.69	842.45	513.93
Training set 2	993.40	847.02	533.40
Training set 3	1028.01	1173.35	505.81
Global MCGP	3026.36	2615.32	1536.58

The training and test results based on a new data set with the exclusion threshold of 8 are presented in Tables 4 and 5. This data set selects fewer recordings for training and testing as compared to the previous experiments, which means this time fewer undesirable recordings are involved. From the results, we can find that overall the training and test errors are greatly reduced due to less undesirable data being used. This also helps to improve the aggregation process since now GPoE achieves the lowest test errors with all three test sets. The training and test results based on a data set with the exclusion threshold of 10 are presented in Tables 6 and 7.

Based on all the three data sets built with different harmonic thresholds, the training time for the proposed distributed classification algorithm and the global MCGP-based scheme is presented in Table 8. We can find that the proposed decision-level fusion achieves a shorter training time for each training set as compared to the case that the data from multiple microphones are first fused and then used for training a global MCGP. Due to the fact that DGP can be implemented in parallel[23], the computational efficiency can be greatly improved as compared to the global MCGP-based scheme.

5. CONCLUSION

This paper proposes a distributed MCGP classification algorithm for ground vehicle identification using acoustic signals. The harmonic structure analysis is used feature extraction. The predictions from local classifiers are then aggregated into a high-level prediction to achieve the decision-level fusion. Simulations based on the ACIDS confirm that the proposed algorithm can outperform local classifier in terms of classification error and NLL. Particularly, it performs competitively well as compared to a global MCGP classifier based on the data-level fusion.

ACKNOWLEDGMENTS

This research is sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing under the SIGNeTS project. It is accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government or the U.K. Government. The U.S. Government and U.K. Government are authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

1 Guo, B., Nixon, M. S., and Raju Damarla, T., "Acoustic information fusion for ground vehicle classification," in [*Proc. of the 2008 11th International Conference on Information Fusion*], 1–7 (2008).

2 Ntalampiras, S., "Moving vehicle classification using wireless acoustic sensor networks," *IEEE Transactions on Emerging Topics in Computational Intelligence* **2**(2), 129–138 (2018).

3 Succi, G. P., Pedersen, T. K., Gampert, R., and Prado, G., "Acoustic target tracking and target identification: recent results," **3713**, 10 – 21, SPIE (1999).

4 Khandoker, A. H., Lai, D. T. H., Begg, R. K., and Palaniswami, M., "Wavelet-based feature extraction for support vector machines for screening balance impairments in the elderly," *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **15**(4), 587–597 (2007).

5 Karlsen, R. E., Meitzler, T. J., Gerhart, G. R., Gorsich, D. J., and Choe, H. C., "Comparative study of wavelet methods in ground vehicle signature analysis," in [*Wavelet Applications III*], **2762**, 314 – 324, SPIE (1996).

6 William, P. E. and Hoffman, M. W., "Classification of military ground vehicles using time domain harmonics' amplitudes," *IEEE Transactions on Instrumentation and Measurement* **60**(11), 3720–3731 (2011).

7 Wu, H. and Mendel, J. M., "Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers," *IEEE Transactions on Fuzzy Systems* **15**(1), 56–72 (2007).

8 Mesaros, A., Heittola, T., Eronen, A., and Virtanen, T., "Acoustic event detection in real life recordings," in [*2010 18th European Signal Processing Conference*], 1267–1271 (2010).

9 Uzkent, B., Barkana, B. D., and Cevikalp, H., "Non-speech environmental sound classification using svms with a new set of features," *International Journal of Innovative Computing, Information and Control* **8**(5), 3511–3524 (2012).

10 Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D., "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 2880–2894 (2020).

11 Hernández-lobato, D., Hernández-lobato, J., and Dupont, P., "Robust multi-class gaussian process classification," in [*Advances in Neural Information Processing Systems*], **24** (2011).

12 Hensman, J., Matthews, A., and Ghahramani, Z., "Scalable Variational Gaussian Process Classification," in [*Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*], *Proceedings of Machine Learning Research* **38**, 351–360 (2015).

13 Liu, H., Ong, Y.-S., Shen, X., and Cai, J., "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems* **31**(11), 4405–4423 (2020).

14 Deisenroth, M. and Ng, J. W., "Distributed gaussian processes," in [*Proceedings of the 32nd International Conference on Machine Learning*], *Proceedings of Machine Learning Research* **37**, 1481–1490 (2015).

15 Villacampa-Calvo, C., ZaldÃvar, B., Garrido-MerchÃ¡n, E. C., and HernÃ¡ndez-Lobato, D., "Multi-class gaussian process classification with noisy inputs," *Journal of Machine Learning Research* **22**(36), 1–52 (2021).

16 Titsias, M., "Variational learning of inducing variables in sparse gaussian processes," in [*Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*], **5**, 567–574 (Apr 2009).

17 Blei, D. M., Kucukelbir, A., and McAuliffe, J. D., "Variational inference: A review for statisticians," *Journal of the American Statistical Association* **112**(518), 859–877 (2017).

18 Hinton, G. E., "Training products of experts by minimizing contrastive divergence," *Neural Computation* **14**(8), 1771–1800 (2002).

19 Yuksel, S. E., Wilson, J. N., and Gader, P. D., "Twenty years of mixture of experts," *IEEE Transactions on Neural Networks and Learning Systems* **23**(8), 1177–1193 (2012).

20 Cao, Y. and Fleet, D. J., "Generalized product of experts for automatic and principled fusion of Gaussian process predictions," *arXiv preprint arXiv:1410.7827* (2014).

21 Godsill, S. and Davy, M., "Bayesian harmonic models for musical pitch estimation and analysis," in [*2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*], **2**, II–1769, IEEE (2002).

22 Nielsen, J. K., Jensen, T. L., Jensen, J. R., Christensen, M. G., and Jensen, S. H., "Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient," *Signal Processing* **135**, 188–197 (2017).

23 Liu, H., Cai, J., Wang, Y., and Ong, Y. S., "Generalized robust Bayesian committee machine for large-scale Gaussian process regression," in [*Proceedings of the 35th International Conference on Machine Learning*], **80**, 3131–3140 (Jul 2018).