

A manifold learning approach for gesture recognition from micro-Doppler radar measurements



E.S. Mason ^a, H.N. Mhaskar ^{b,*1}, Adam Guo ^c

^a Hawkeye 360, 196 Van Buren St #450, Herndon, VA 20170, USA

^b Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA 91711, USA

^c Mathematics/Computer Science Department, Pomona College, Claremont, CA 91711, USA

ARTICLE INFO

Article history:

Received 4 October 2021

Revised and accepted 21 April 2022

Available online 28 April 2022

Keywords:

Machine learning

Kernel methods

Micro-Doppler radar gesture recognition

ABSTRACT

A recent paper (Mhaskar (2020)) introduces a straightforward and simple kernel based approximation for manifold learning that does not require the knowledge of anything about the manifold, except for its dimension. In this paper, we examine how the pointwise error in approximation using least squares optimization based on similarly localized kernels depends upon the data characteristics and deteriorates as one goes away from the training data. The theory is presented with an abstract localized kernel, which can utilize any prior knowledge about the data being located on an unknown sub-manifold of a known manifold.

We demonstrate the performance of our approach using a publicly available micro-Doppler data set, and investigate the use of different preprocessing measures, kernels, and manifold dimensions. Specifically, it is shown that the localized kernel introduced in the above mentioned paper when used with PCA components leads to a near-competitive performance to deep neural networks, and offers significant improvements in training speed and memory requirements. To demonstrate the fact that our methods are agnostic to the domain knowledge, we examine the classification problem in a simple video data set.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

Identification of hand gestures is a subject of increasing research attention in the area of human–computer interaction, see for example, Ahmed, Kallu, Ahmed, and Cho (2021) and Skaria, Al-Hourani, and Evans (2020) for recent reviews. Applications include convenient device control, infection prevention in clinical settings, safer and quicker accessibility of features in automotive, and, in a more general sense, detection of unmanned aerial vehicles. In contrast to the commonly used hand gesture signal acquisition approaches such as camera, infra-red sensors, and ultrasonic sensors, radar sensors are observed to yield a higher performance in adverse lighting conditions and complex background. Among the common types of waveforms used by inexpensive K-band radar sensors – pulse, continuous waveform (CW), and frequency modulated continuous waveform (FMCW) – the FMCW radars provide an improved simultaneous estimation

of the range and the Doppler signatures. Many machine learning strategies are used for the classification of hand gestures, ranging from nearest neighbor and support vector machines (SVMs) to deep convolutional neural networks (CNNs) obtaining an accuracy as high as 95%. Except for deep networks, common machine learning approaches typically requires the extraction of hand crafted features based on domain knowledge. On the other hand, training a CNN is typically highly time consuming, and the mathematical theory of how, when, and why CNNs will produce the right output is still under development. In this paper, we demonstrate the use of manifold learning to classify hand gesture signals from FMCW radar data, agnostic of the fact that we are dealing with radar signals (i.e., independently of domain knowledge). Our general purpose method yields results that compare favorably with other methods in the literature that have been used for micro-Doppler gesture recognition, including CNNs.

The fundamental problem of machine learning is the following. We have (training) data of the form $\{(\mathbf{x}_j, f(\mathbf{x}_j) + \epsilon_j)\}$, where, for some integer $d \geq 1$, $\mathbf{x}_j \in \mathbb{R}^d$ are realizations of a random variable, f is an unknown real valued function, and the ϵ_j 's are realizations of a mean zero random variable. The distributions of both the random variables are not known. The problem is to approximate f with a suitable model, especially for the points which are not in the training data. In the context of classification problems, f is

* Corresponding author.

E-mail addresses: eric.mason@he360.com (E.S. Mason), hrushikesh.mhaskar@cgu.edu (H.N. Mhaskar), simagingadam.guo@pomona.edu (A. Guo).

¹ The research of HNM was supported in part by ARO, USA grant W911NF2110218 and NSF DMS, USA grant 2012355.

the labeling function defined as follows. If there are K classes, we define

$$\chi_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ belongs to class } i, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$f(\mathbf{x}) = \arg \min_{1 \leq i \leq K} \chi_i(\mathbf{x}).$$

A major theoretical problem in machine learning is the so called curse of dimensionality – the complexity of the approximating model increases exponentially with the dimension of the input. Another theoretical problem is that if the class boundaries are not smooth or not well separated then the labeling function is not smooth, resulting again in an increased model complexity.

Manifold learning tries to ameliorate this problem by assuming that the data lies on some unknown, low dimensional manifold, on which the classes are well separated. With this assumption, there are well developed techniques to learn various quantities related to the manifold, for example, the eigen-decomposition of the Laplace–Beltrami operator. The special issue (Chui & Donoho, 2006) gives a good introduction to the topic of diffusion geometry in this regard. In Chui and Mhaskar (2018) and Schmidt-Hieber (2019), the authors describe the construction of an atlas on the manifold which is then used for construction of deep networks. Since these objects must be learned from the data, these techniques are applicable in the supervised setting, in which we have all the data points \mathbf{x}_j available in advance.

Recently, we have discovered (Mhaskar, 2020a) a more direct method to approximate functions on unknown manifolds without trying to learn anything about the manifold itself, and giving an approximation with theoretically guaranteed errors on the entire manifold; not just the points in the original data set. Our approximation has the form

$$\mathbf{x} \mapsto \sum_j (f(\mathbf{x}_j) + \epsilon_j) \tilde{\Phi}_{N,q}(|\mathbf{x} - \mathbf{x}_j|_{2,Q}) \quad (1.1)$$

where $\tilde{\Phi}_{N,q}$ is a specially constructed kernel, Q is the dimension of the ambient space, q is the dimension of the manifold, and $|\cdot|_{2,Q}$ is the Euclidean metric on \mathbb{R}^Q (see Section 3.1 for details). In the case when $N = 1$, our kernel reduces to the oft-used Gaussian kernel. In this paper, we prove how a more classical approach of empirical risk minimization or generalization error measured in the least squares sense using the kernel introduced in Mhaskar (2020a) leads to errors that show a gradual deterioration as one moves away from the training data, thereby yielding a heuristic to determine the parameter N .

In many applications, we can assume some further knowledge of the data distribution; in particular, that the data is sampled from an unknown sub-manifold of a known manifold. It is then natural to conjecture that one might be able to replace the Euclidean distance in (1.1) by the geodesic distance on the known manifold. While the theory analogous to the one in Mhaskar (2020a) seems difficult at this point, we formulate our main theorems in a greater abstraction to allow the study of pointwise errors using least squares approximation based on the resulting kernels. We demonstrate experimentally the effect of such a construction by considering each element of the FMCW radar data as a time series, which in turn may be identified with a point on a Grassmann manifold using the ideas in Turaga, Veeraraghavan, Srivastava, and Chellappa (2011).

Although our main focus is on the detection of hand gestures based on FMCW radar data, we include an additional example about classification of simple video data in order to demonstrate

the fact that our method is agnostic to the domain knowledge feature extraction.

To summarize, the main contributions of this paper are as follows:

- We examine pointwise errors in function approximation using empirical risk minimization and generalization error measured with least squared loss with a generalized version of the kernel introduced in Mhaskar (2020a).
- We demonstrate the performance of our theory in the case of hand gesture recognition based on FMCW radar data. Our results compare favorably with other known techniques, when used with the right feature vectors. Additionally, this provides new benchmark results on the recently published Dop-Net data set (Ritchie, Capraru, & Fioranelli, 2020).
- There are many papers dealing with micro-Doppler gesture recognition using hand-crafted features. As far as we are aware, this is the first work in which we explore the use of the singular vectors and values of the micro-Doppler spectrogram as features.
- Our method is agnostic to the domain knowledge, a fact which we demonstrate using classification in a simple video data.

In Section 2, we summarize the connection between our work and other related works, although not exhaustively. The technical background that motivates our work in this paper is explained in Section 3. In Section 4, we state our main theoretical results. The experimental results are presented in Section 5. The proofs of the results in Section 4 are given in Section 6.

2. Related works

Micro-Doppler radar has found success in various applications related to classifying humans, animals and objects, such as activity monitoring, autonomous driving, and unmanned aerial vehicle detection (Molchanov, Harmanny, de Wit, Egiazarian, & Astola, 2014; Ritchie & Jones, 2019; Tahmoush, 2015; Zhao, Luo, Wang, Chen, & Wu, 2021; Zhou, Yang, Jiang, Wong, & Yang, 2020). The approach taken by practitioners is to use a time–frequency transform of the measured signal, which is then considered the raw data for subsequent processing. The Short-Time Fourier Transform (STFT) is the time–frequency transform commonly selected. Using the spectrograms, features are extracted or the spectrogram is used directly with a neural network.

Hand crafted features are usually inspired by physics and taken to be measurements of frequency and/or time, such as the bandwidth of the micro-Doppler signature, its time duration, energy content, and various spectrogram measurements (Björklund, Petersson, & Hendeby, 2015; Karabacak et al., 2015; Manfredi et al., 2021; Ritchie & Jones, 2019; Tahmoush, 2015). There have also been efforts to define new features through mathematical transforms, such as the Discrete Cosine Transform (DCT), Mel-Frequencies, S-transform, etc. (Erol, Seyfioğlu, Gürbüz, & Amin, 2018; Li et al., 2017; Li, Zhang, Ritchie, & Griffiths, 2018) and (Ma, Zhao, Liu, Kuang, & Al-qaness, 2019). The success of these approaches is usually determined by the application, radar system parameters, and quantity of training data. In Gürbüz, Erol, Çağlıyan, and Tekeli (2015), the authors study the effect of various micro-Doppler features, such as transmit frequency, range and Doppler resolution, antenna-target geometry, signal-to-noise ratio and dwell-time. While classifier performance on hand-crafted features undoubtedly depend on these parameters, the performance of more advanced feature extraction methods will also vary based on the size of the training set and distributional variation between the train and test data. For example, statistical variation within a class can depend on different measurement

environments and subjects, which was not addressed in Gürbüz et al. (2015), but explored here by means of features based on singular value decomposition (SVD) or principal component analysis (PCA) of preprocessed spectrograms.

Recently, deep learning architectures, such as CNNs have also been used for micro-Doppler applications (Abdulatif, Wei, Aziz, Kleiner, & Schneider, 2018; Brooks, Schwander, Barbaresco, Schneider, & Cord, 2018; Huijing et al., 2019; Zhu, Chen, & Ye, 2020). By design, CNNs are invariant to shifts in the two-dimensional image, obtained by using a combination of additive group convolutions and pooling operations. While relatively simple CNNs consisting of only three to five convolutional layers have been shown to be successful, deeper and more complex network approaches have also been investigated (Seyfioglu, Erol, Gürbüz, & Amin, 2019). Furthermore, more advanced deep learning approaches have also been taken to compensate for limited training data, such as generating data using generative adversarial networks and transfer learning (Park, Javier, Moon, & Kim, 2016; Tran, Griffin, Chetty, & Vishwakarma, 2020).

However, there are other works that indicate that the use of CNNs may not be optimal for this application. Firstly, it is a topic of current research to investigate in what applications CNNs are most successful, why, and how. The state of the art use of CNNs typically involves a network topology that is chosen on ad hoc basis. In Tran et al. (2020), the authors find that after mapping the micro-Doppler spectrograms to a one dimensional vector, a linear SVM outperformed various VGG architectures. Additionally, unlike CNNs, features defined on a specific manifold through an appropriate kernel function provide a clear advantage in being able to more precisely obtain invariance to well defined nuisance transformations known a priori, based on knowledge of the application and data. This is complemented by theory that gives insight into generalization performance.

In Zeng, Amin, and Shan (2020), the authors have demonstrated that simpler method like K-nearest neighbors (KNN) based on PCA components of the vectorized spectrograms yield results comparable to deep networks.

We will use our techniques both with SVD based features and PCA features (designed to overcome the statistical variations within-class), and compare the results with those obtained with CNNs. Our methods will sometimes outperform CNNs and sometimes not, but will be consistently faster on training. The main advantage though is that our techniques are well founded in theory while it is a major topic of research as to when, why, for what applications, and how CNNs will give what performance.

We will show in Section 4 that the process of empirical risk minimization with our kernel yields an exact reproduction of the label function for the training data. In this sense, the work is related to several recent works which have observed that it is possible to achieve a zero training error while keeping the test error under control. In the case of classification problems, Belkin, Hsu, and Mitra (Belkin, Hsu, & Mitra, 2018) analyze the “excess error” in least square fits by piecewise linear interpolants over that obtained by the optimal Bayes’ classifier. In Poggio et al. (2017) and Poggio et al. (2018), the question is analyzed from the perspective of the geometry of the error surface with respect to different loss functions near the local extrema. In particular, it is shown in Poggio et al. (2017) that substituting the rectified linear unit (ReLU) activation function by a polynomial approximation exhibits the same behavior as the original network. In Mhaskar and Poggio (2020) we have analyzed the question from the point of view of approximation theory so as to examine the intrinsic features of the data (rather than focusing on specific training algorithms) that allow this phenomenon. A crucial role in the proofs of the results in that paper is played by a highly localized kernel. In this paper, we focus on the properties of a localized kernel in a more general setting of a locally compact metric measure space that allow the results analogous to those in Mhaskar and Poggio (2020).

3. Technical background

In this section, we discuss some further technical details to motivate our main results in Section 4. In Section 3.1, we describe our constructions for approximation on an unknown sub-manifold of the Euclidean space. In Section 3.2, we provide a mathematical model for the Doppler radar measurements, leading to the need for examining our results in Section 3.1 to the case of a sub-manifold of a known manifold, namely, the Grassmann manifold. In Section 3.3, we describe the ideas in Turaga et al. (2011) that lead to the Grassmann manifold again from a different perspective of time series classification.

3.1. Approximation on manifolds

In this section, we describe our construction for a direct method for function approximation on a sub-manifold of a Euclidean space, without resorting to a two step procedure involving the estimation of quantities related to the manifold such as coordinate charts or eigen-decomposition of the Laplace–Beltrami operator. Our construction involves a localized kernel based on Hermite polynomials.

These are defined by the Rodrigues’ formula (cf. Szegö (1975, Eqns. (5.5.3), (5.5.1))

$$h_k(x) = \frac{(-1)^k}{\pi^{1/4} 2^{k/2} \sqrt{k!}} \exp(x^2) \frac{d^k}{dx^k} \exp(-x^2).$$

However for relatively small values of k and $|x| < \sqrt{2k}$, they are most efficiently computed using the recurrence relations:

$$\begin{aligned} h_k(x) &:= \sqrt{\frac{2}{k}} x h_{k-1}(x) - \sqrt{\frac{k-1}{k}} h_{k-2}(x), \quad k = 2, 3, \dots, \\ h_0(x) &:= \pi^{-1/4}, \quad h_1(x) := \sqrt{2\pi}^{-1/4} x. \end{aligned} \quad (3.1)$$

We write

$$\psi_k(x) := h_k(x) \exp(-x^2/2), \quad x \in \mathbb{R}, \quad k \in \mathbb{Z}_+. \quad (3.2)$$

The functions $\{\psi_k\}_{k=0}^\infty$ are an orthonormal set with respect to the Lebesgue measure on \mathbb{R} . In the sequel, we fix an infinitely differentiable function $H : [0, \infty) \rightarrow [0, 1]$, such that $H(t) = 1$ if $0 \leq t \leq 1/2$, and $H(t) = 0$ if $t \geq 1$. Further, let $Q \geq q \geq 1$ be fixed integers. We define for $x \in \mathbb{R}$, $m \in \mathbb{Z}_+$:

$$\mathcal{P}_{m,q}(x) := \begin{cases} \pi^{-1/4} (-1)^m \frac{\sqrt{(2m)!}}{2^m m!} \psi_{2m}(x), & \text{if } q = 1, \\ \frac{1}{\pi^{(2q-1)/4} \Gamma((q-1)/2)} \sum_{\ell=0}^m (-1)^\ell \\ \times \frac{\Gamma((q-1)/2 + m - \ell)}{(m - \ell)!} \frac{\sqrt{(2\ell)!}}{2^\ell \ell!} \psi_{2\ell}(x), & \text{if } q \geq 2, \end{cases} \quad (3.3)$$

and the kernel $\tilde{\Phi}_{N,q}$ for $x \in \mathbb{R}$, $N > 0$ by

$$\begin{aligned} \tilde{\Phi}_{N,q}(x) &:= \sum_{m=0}^{\lfloor N^{2/2} \rfloor} H\left(\frac{\sqrt{2m}}{N}\right) \mathcal{P}_{m,q}(x) \\ &= \frac{1}{\pi^{q/2} \Gamma((q-1)/2)} \sum_{\ell=0}^{\lfloor N^{2/2} \rfloor} \left\{ \sum_{m=\ell}^{\lfloor N^{2/2} \rfloor} H\left(\frac{\sqrt{2m}}{N}\right) \right. \\ &\quad \left. \times \frac{\Gamma((q-1)/2 + m - \ell)}{(m - \ell)!} \right\} \psi_{2\ell}(0) \psi_{2\ell}(x). \end{aligned} \quad (3.4)$$

The kernel $\tilde{\Phi}_{N,q}(x)$ can be computed easily and efficiently using the second equation in (3.4), and the Clenshaw algorithm (Gautschi, 2004, p. 79).

Next, let \mathbb{X} be a q dimensional, compact, connected, orientable, sub-manifold of \mathbb{R}^q , ρ be the geodesic distance on \mathbb{X} , μ^* be its volume element. We will abbreviate our notation and write $\Phi_N^M(\mathbf{x}) = \tilde{\Phi}_{N,q}(|\mathbf{x}|_{2,Q})$, $\mathbf{x} \in \mathbb{X}$.²

Let $C(\mathbb{X})$ denote the class of all continuous real valued functions on \mathbb{X} equipped with the supremum norm $\|f\|_\infty := \max_{\mathbf{x} \in \mathbb{X}} |f(\mathbf{x})|$. Using the kernel Φ_N^M , we define an integral operator on $C(\mathbb{X})$, analogous to a convolution operator, by

$$\sigma_N^M(f)(\mathbf{x}) := \int_{\mathbb{X}} \Phi_N^M(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mu^*(\mathbf{y}), \quad N > 0. \quad (3.5)$$

(In the notation of Mhaskar (2020a), $\sigma_N^M(f) = \sigma_{N,1}(\mathbb{X}; f)$.) We are interested in approximation of Lipschitz continuous functions on \mathbb{X} ; i.e., functions $f \in C(\mathbb{X})$ for which

$$\|f\|_{\text{Lip}} := \sup_{\substack{\mathbf{x}, \mathbf{x}' \in \mathbb{X} \\ \mathbf{x} \neq \mathbf{x}'}} \frac{|f(\mathbf{x}) - f(\mathbf{x}')|}{\rho(\mathbf{x}, \mathbf{x}')} < \infty. \quad (3.6)$$

A consequence of Mhaskar (2020a, Theorem 8.1) is the following Theorem 3.1. Before stating this theorem, we state first a convention regarding constants.

Constant convention

In the sequel, the notation $A \lesssim B$ (equivalently, $B \gtrsim A$) will mean that $A \leq cB$ for some positive constant c depending only on fixed objects under discussion such as \mathbb{X} , ρ , μ^* , q , Q , the parameter S to be introduced later, and the manifold/spaces/measures/distances to be described later. The notation $A \sim B$ means $A \lesssim B$ and $B \lesssim A$. In particular, the constants do not depend upon the target function, the points at which the approximation is desired or on which it is based, and the index N of the family of kernels.

Theorem 3.1. Let $f \in \text{Lip}(\mathbb{X})$. For $N \geq 1$, we have

$$\|f - \sigma_N^M(f)\|_\infty \lesssim \frac{\|f\|_{\text{Lip}}}{N}. \quad (3.7)$$

An important ingredient in the proof of Theorem 3.1 is the following proposition proved in Mhaskar (2020a, Corollary 6.1), and reformulated using the fact that $|\mathbf{x} - \mathbf{y}|_{2,Q} \sim \rho(\mathbf{x}, \mathbf{y})$ for $\mathbf{x}, \mathbf{y} \in \mathbb{X}$ (Mhaskar, 2020a, Corollary 8.1).

Proposition 3.1. Let $S > q$. The kernel Φ_N^M defined in (3.4) satisfies each of the following properties.

$$|\Phi_N^M(\mathbf{x} - \mathbf{y})| \lesssim \frac{N^q}{\max(1, (N\rho(\mathbf{x}, \mathbf{y}))^S)}, \quad \mathbf{x} \in \mathbb{R}^q, \quad (3.8)$$

$$\begin{aligned} |\Phi_N^M(\mathbf{0})| &\sim N^q, \quad |\Phi_N^M(\mathbf{x})| \lesssim N^q, \\ |\Phi_N^M(\mathbf{x}) - \Phi_N^M(\mathbf{y})| &\lesssim N^{q+1} |\mathbf{x}|_{2,Q} - |\mathbf{y}|_{2,Q}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^q. \end{aligned} \quad (3.9)$$

Motivation 1: In practice, of course, one needs to use a discretization of the integral in (3.5). Obviously, an accurate discretization would require the knowledge of the values of f at a sufficiently large number of points in \mathbb{X} . In micro-Doppler radar applications, one has only a small amount of data, so that the above theorem cannot be used effectively in a direct manner.

3.2. Micro-Doppler radar signals

In this section we denote the radar location by $\mathbf{x} \in \mathbb{R}^3$ and the target location by $\mathbf{y} \in \mathbb{R}^3$. We assume that at $t = 0$ the object is initially located at $\mathbf{y}_0 \in \mathbb{R}^3$. During the collection period $t \in [0, T]$, the target undergoes motion defined by a displacement

² We note that when $N = 1$, $\Phi_N^M(\mathbf{x})$ reduces to just a constant multiple of $\exp(-|\mathbf{x}|^2/2)$.

with velocity $\mathbf{v} \in \mathbb{R}^3$ and a rotation given by the matrix $\mathbf{R}(t)$. Thus, at time t the location of the object is

$$\mathbf{y}(t) = \mathbf{R}(t)\mathbf{y}_0 + \mathbf{v}t. \quad (3.10)$$

Assuming the radar transmits in free space, and that there are only a finite number K of isotropic scattering locations,³ we may model the received signal as (cf. (3.12))

$$s(t) = \sum_{k=1}^K p(t - \|\mathbf{x} - \mathbf{R}_k(t)\mathbf{y}_{0,k} - \mathbf{v}_k t\|_2) \rho_k + n(t), \quad (3.11)$$

where for $k = 1, \dots, K$, each $\mathbf{y}_{0,k}$ represents the initial scattering locations of the object that contribute to the signal measured at the receiver, $\mathbf{R}_k(t)$ and \mathbf{v}_k are the corresponding rotations and velocities, $\rho_k \in \mathbb{R}^+$ is the amplitude of the scatterer, and $n(t)$ represents noise, clutter, and small amplitude interfering signals when the isotropic point scattering assumption does not hold precisely. Typically, $p(t)$ is a frequency modulation continuous (FMCW), defined as

$$p(t) = e^{i(\omega_c t + \frac{1}{2}\alpha t^2)}, \quad (3.12)$$

where ω_c is the carrier frequency and α is the chirp rate. The data set introduced in Ritchie et al. (2020) and used in our experiments, were collected with a FMCW waveform.

For gesture recognition K will be small since there are only a small number of dominant scattering locations. Thus, the received signal $s(t)$ will lie in a low-dimensional subspace given by the span of the individual received signals for each scatterer, of maximum dimension K . This is due to the fact that the scattered signals are approximately orthogonal for large ω_c used for micro-Doppler radars, and motivates our following choice of features.

Generally, micro-Doppler signal classification is accomplished using the spectrogram of (3.11), defined as

$$D(\omega, \tau) = \int_{\mathbb{R}} s(t + \tau) w(t) e^{-i\omega(t+\tau)} dt, \quad (3.13)$$

where w is a smooth window function. In practice, (3.11) will be sampled to form a discrete time-series, and one uses the discrete version of (3.13):

$$\mathbf{D} = \mathbf{F} \mathbf{W} \mathbf{S}, \quad (3.14)$$

where \mathbf{F} is a partial Fourier matrix, $\mathbf{W} = \text{diag}(\mathbf{w})$ is a diagonal matrix consisting of the elements of \mathbf{w} , and $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N]$. The columns of \mathbf{D} correspond to the discrete Doppler frequencies $\omega_{-M/2}, \dots, \omega_{M/2}$, and the rows correspond to the discrete time points τ_0, \dots, τ_N .

Let the singular value decomposition of \mathbf{D} be given by

$$\mathbf{D} = \mathbf{U} \Sigma \mathbf{V}^H, \quad (3.15)$$

where \mathbf{U} , \mathbf{V} are unitary matrices, and Σ is a diagonal matrix consisting of the singular values. From (3.15) we can define a number of distance functions based on using left singular vectors and singular values, each of which defines a sub-manifold of the data.

3.3. Representation of a general time series

The autoregressive-moving-average (ARMA) is a well-known dynamic model for time series data that parametrizes a signal $f(t)$ by the equations

$$f(t) = Cz(t) + w(t), \quad w(t) \sim \mathcal{N}(0, R) \quad (3.16)$$

³ Isotropic scattering means the signal is reflected in all directions, a reasonable assumption for small scattering locations on the human hand.

$$z(t+1) = Az(t) + v(t), \quad v(t) \sim \mathcal{N}(0, Q) \quad (3.17)$$

where $z \in \mathbb{R}^d$ is the hidden state vector, $f : \mathbb{R} \rightarrow \mathbb{R}^p$, $d \leq p$ is the hidden state dimension, and $\mathcal{N}(0, \sigma)$ denotes a normal distribution with mean zero and standard deviation σ (Turaga et al., 2011). There are widely-used closed form solutions for estimating the parameters A and C in terms of the singular value decomposition $[f(1), \dots, f(\tau)] = U\Sigma V^T$, namely,

$$C = U, \quad A = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1}, \quad (3.18)$$

where

$$D_1 = \begin{pmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{pmatrix}, \quad D_2 = \begin{pmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

It can be shown (Turaga et al., 2011) that the expected observation sequence is given by

$$O_\infty = \mathbb{E} \left[\begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \end{pmatrix} \right] = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} z(0) \quad (3.19)$$

Thus, by truncating the matrix O_∞ up to m th block for some m , one can represent the time series f as point on the Grassmann manifold $\mathcal{G}(d, mp)$.

We consider time-frequency images, in this case spectrograms generated from the radar data as multivariate timeseries, and similarly for the frames of the video data. In the former case, one way to view the spectrogram is as a multivariate time series, where the columns are multi-dimensional vectors formed by the taking the Discrete Fourier Transform of the time segment, the collection of these vectors implies the spectrogram is a multivariate time series. As a consequence we can represent these spectrograms using an orthonormal basis corresponding to a point on the Grassmann manifold. Independently, as explained in Section 3.2, the physics of radar imaging leads to treating the left singular vectors of the spectrograms as feature vectors, which in turn, are points on a Grassmann manifold. These features capture the frequency content of the micro-Doppler signature while imposing invariance with respect to the underlying time-series, which can better represent the discriminant attributes common to the entire data set. For example, we expect multiple samples corresponding to the same gesture can be represented by a collection of orthonormal basis matrices that are close under a particular metric. Furthermore, the subspace corresponding to a data sample can be viewed as subspace of a larger subspace spanned by all the samples of the class.

Motivation 2: Clearly, the set of time series of interest, such as the low rank representations of the micro-Doppler radar signals, is only an unknown subset of the Grassmann manifold, rather than the entire manifold. Assuming that this subset is a sub-manifold, it is interesting to examine the span of $\{\tilde{\Phi}_{N,q}(\rho(\circ, \mathbf{y}_j))\}_{j=1}^M$, where ρ is the geodesic distance on the Grassmann manifold.

4. Theoretical results

Our goal in this section is to formulate results for approximation based on a small amount of data. We will examine two tools for this purpose: empirical risk minimization, and theoretical least square loss. In each case, we will obtain pointwise error estimates which indicate how the error deteriorates as the input variable moves away from the training data. Our estimates will give a deeper insight into which features of the data give rise to what accuracy *intrinsically*, as opposed to many results in the literature that depend upon the method used to solve the

optimization problems involved. In view of the motivation given in Section 3.3, we will state our results in greater abstraction than in the context of the motivation given in Section 3.1.

Let \mathbb{X} be a locally compact metric measure space, with metric ρ , and a distinguished measure μ^* . As before, $C(\mathbb{X})$ denotes the space of bounded and uniformly continuous real valued functions on \mathbb{X} , equipped with the supremum norm: $\|f\|_\infty = \sup_{x \in \mathbb{X}} |f(x)|$. The class $\text{Lip}(\mathbb{X})$ of Lipschitz functions comprises $f \in C(\mathbb{X})$ for which

$$\|f\|_{\text{Lip}} := \sup_{\substack{x, x' \in \mathbb{X} \\ x \neq x'}} \frac{|f(x) - f(x')|}{\rho(x, x')} < \infty. \quad (4.1)$$

We recall the constant convention from Section 3.1.

Definition 4.1. Let $S > q \geq 1$ be integers. A family of kernels $\{\Phi_N : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}\}$ is called **(q, S) -localized** if each Φ_N is symmetric, and

$$|\Phi_N(x, y)| \lesssim \frac{N^q}{\max(1, (N\rho(x, y))^S)}, \quad (4.2)$$

where the constant involved in \lesssim may depend upon q and S but not on N , x , or y . With an abuse of terminology we will say that Φ_N is (q, S) -localized.

Remark 4.1. Kernels satisfying (4.2) are known in many contexts. The kernel denoted by ϕ_N^M in Section 3.1 is one example. Many other constructions in different contexts are discussed in Mhaskar (2020b) and references therein. ■

Let $M \geq 1$ be an integer, $\mathcal{C} = \{y_1, \dots, y_M\} \subset \mathbb{X}$. We are interested in studying approximation of Lipschitz functions on \mathbb{X} from the space

$$\mathcal{V}(\mathcal{C}) = \text{span}\{\Phi_N(\circ, y_j)\}_{j=1}^M. \quad (4.3)$$

Remark 4.2. If $\Phi_N(x, y)$ admits a Mercer expansion of the form $\sum \lambda_k \phi_k(x) \phi_k(y)$, and the set \mathcal{C} is sufficiently dense in \mathbb{X} , then in a number of cases as in Mhaskar (2020b), the space $\mathcal{V}(\mathcal{C})$ is the same as $\text{span}\{\phi_k : \lambda_k \lesssim N\}$. However, our interest in this paper is when \mathcal{C} has a large minimal separation rather than being dense. ■

We will study the behavior of approximations defined by the solutions of two minimization problems: The **empirical risk** minimizer is defined to be

$$P_E(\mathcal{V}(\mathcal{C}); f) := \arg \min_{P \in \mathcal{V}(\mathcal{C})} \sum_{k=1}^M |f(y_j) - P(y_j)|^2. \quad (4.4)$$

If τ is a probability measure on \mathbb{X} , the **theoretical least square loss** minimizer is defined by

$$P_T(\tau, \mathcal{V}(\mathcal{C}); f) := \arg \min_{P \in \mathcal{V}(\mathcal{C})} \int_{\mathbb{X}} |f(y) - P(y)|^2 d\tau(y). \quad (4.5)$$

We need to assume some further conditions on the kernels $\{\Phi_N\}$.

Definition 4.2. Let $S > q \geq 1$ be integers. A family of kernels $\{\Phi_N\}$ is called **admissible** if it is (q, S) -localized and satisfies each of the following properties.

$$|\Phi_N(y, y)| \sim N^q, \quad y \in \mathbb{X}. \quad (4.6)$$

$$\begin{aligned} |\Phi_N(x, y) - \Phi_N(x', y)| &\lesssim N^{q+1} \rho(x, x'), \quad x, x', y \in \mathbb{X}, \\ |\Phi_N(x, y) - \Phi_N(x, y')| &\lesssim N^{q+1} \rho(y, y'), \quad x, y, y' \in \mathbb{X}. \end{aligned} \quad (4.7)$$

With an abuse of terminology as before, we refer to any member Φ_N of an admissible family as an admissible kernel.

Remark 4.3. The kernel denoted by Φ_N^M in Section 3.1 is one example of admissible kernels (cf. Proposition 3.1). Several other examples are available in the literature, for example, based on eigenfunctions of the Laplace–Beltrami operator on manifolds (Filbir & Mhaskar, 2010, 2011). ■

Remark 4.4. In the sequel, q and S will be treated as fixed parameters. All constants may depend upon these. Moreover, we will omit their mention from the notation; e.g., say that Φ_N is localized rather than (q, S) -localized. ■

In the study of approximation properties of both the empirical risk minimizer and theoretical least squares loss minimizer, a crucial role is played by the minimal separation among the points in \mathcal{C} , defined by the minimal separation among the points defined by

$$\eta(\mathcal{C}) = \min_{1 \leq j \neq k \leq M} \rho(y_j, y_k), \quad (4.8)$$

In order to state our main theorems, we need to make some further assumptions.

For $x \in \mathbb{X}$, $r > 0$, we write

$$\mathbb{B}(x, r) = \{y \in \mathbb{X} : \rho(x, y) \leq r\}, \quad \Delta(x, r) = \mathbb{X} \setminus \mathbb{B}(x, r). \quad (4.9)$$

We assume that there exists $q \geq 0$ such that

$$\mu^*(\mathbb{B}(x, r)) = \mu^*(\{y \in \mathbb{X} : \rho(x, y) < r\}) \lesssim r^q, \quad x \in \mathbb{X}, \quad r > 0, \quad (4.10)$$

and

$$\mu^*(\mathbb{B}(x, r)) \gtrsim r^q, \quad x \in \mathbb{X}, \quad 0 < r \leq 1. \quad (4.11)$$

It is shown in Mhaskar (2020b) that in the case when \mathbb{X} is a manifold as described earlier, the condition (4.11) is satisfied in fact for all $r \leq \text{diam}(\mathbb{X})$. The condition (4.10) is satisfied in many cases when \mathbb{X} is a smooth compact connected q -dimensional manifold, ρ is the Riemannian metric, and μ^* denotes the volume measure.

Our first theorem derives pointwise error estimates for the empirical risk minimizer.

Theorem 4.1. Let $\{\Phi_N\}$ be a family of admissible kernels, and (4.10) and (4.11) be satisfied. Let $x \in \mathbb{X}$, $\delta(x) = \min_{1 \leq j \leq M} \rho(x, y_j)$, and $\tilde{\eta} = \min(1, \eta(\mathcal{C}))$. Let $F \in \text{Lip}(\mathbb{X})$. There exists a constant $C > 0$ independent of F , N , and \mathcal{C} such that if $N \geq C\eta(\mathcal{C})^{-1}$, then there exists a unique empirical risk minimizer $P_E(\mathcal{V}(\mathcal{C}); F)$ that satisfies

$$P_E(\mathcal{V}(\mathcal{C}); F)(y_j) = F(y_j), \quad j = 1, \dots, M. \quad (4.12)$$

We have the following error estimates. If $\delta(x) > \tilde{\eta}/3$ then

$$|P_E(\mathcal{V}(\mathcal{C}); F)(x)| \lesssim \frac{\|F\|_\infty}{(N\tilde{\eta})^S}. \quad (4.13)$$

If $\delta(x) \leq \tilde{\eta}/3$ then

$$|P_E(\mathcal{V}(\mathcal{C}); F)(x) - F(x)| \lesssim (N + \|f\|_{\text{Lip}})\delta(x) + \|F\|_\infty(N\tilde{\eta})^{q-S}. \quad (4.14)$$

Here, the constants involved are independent of F and x .

To describe the analogue of Theorem 4.1 for the minimizer of the theoretical least square loss, we need some further notation relevant to the normal equations for the minimization problem. We define

$$\Psi_N(x, y) = \int_{\mathbb{X}} \Phi_N(x, z)\Phi_N(y, z)d\tau(z), \quad N > 0, \quad x, y \in \mathbb{X}, \quad (4.15)$$

and for $f \in C(\mathbb{X})$,

$$\sigma_N(f)(x) = \int_{\mathbb{X}} f(y)\Phi_N(x, y)d\mu^*(z). \quad (4.16)$$

We note that if τ is absolutely continuous with respect to μ^* with the Radon–Nikodym derivative f_0 , then the normal equations for the theoretical least squares loss minimization are

$$\sum_{\ell=1}^M a_\ell \Psi_N(y_\ell, y_j) = \sigma_N(f_0 F)(y_j), \quad j = 1, \dots, M. \quad (4.17)$$

Theorem 4.2. We assume the set up as in Theorem 4.1. Let τ be absolutely continuous with respect to μ^* with $d\tau = f_0 d\mu^*$ for some $f_0 \in C(\mathbb{X})$, $f_0(z) \geq m(f_0) > 0$ for all $z \in \mathbb{X}$. There exists a constant $C > 0$ independent of F , N , and \mathcal{C} such that if $N \geq C\eta(\mathcal{C})^{-1}$, then there exists a unique theoretical least square loss risk minimizer $P_T(\tau, \mathcal{V}(\mathcal{C}); F)$ that satisfies the following error estimates.

If $\delta(x) > \tilde{\eta}/3$ then

$$|P_E(\mathcal{V}(\mathcal{C}); F)(x)| \lesssim \frac{\|f_0 F\|_\infty}{(N\tilde{\eta})^S}. \quad (4.18)$$

If $\delta(x) = \rho(x, y_\ell) \leq \tilde{\eta}/3$ then

$$\begin{aligned} & \left| P_T(\tau, \mathcal{V}(\mathcal{C}); F)(x) - \frac{\Phi_N(x, x)}{\Psi_N(y_\ell, y_\ell)} f_0(x) F(x) \right| \\ & \lesssim (N\eta)^{q-S} \|f_0 F\|_\infty + \|f_0 F - \sigma_N(f_0 F)\|_\infty \\ & \quad + (N + \|f_0 F\|_{\text{Lip}})\delta(x). \end{aligned} \quad (4.19)$$

Here, the constants involved are independent of F and x .

Remark 4.5. In the case of the kernels Φ_N^M , the quantity $\|f_0 F - \sigma_N(f_0 F)\|_\infty$ is estimated by Theorem 3.1. Similar estimates exist also in the other examples described in Mhaskar (2020b) with other conditions on f_0 and F . ■

Remark 4.6. We note that (4.13) and (4.18) show that one does not expect a good approximation to F as $N \rightarrow \infty$ in the region away from the points \mathcal{C} . On the other hand, (4.14) and (4.19) show that the approximation actually deteriorates if $N \rightarrow \infty$. The balance of the various terms on the right hand side of these estimates gives us some insight into the choice of N as well as how the approximation error deteriorates as x is further and further away from \mathcal{C} . This phenomenon is caused by the localization of the kernels, resulting in overfitting the data if N is very large. For example, in our experiments in Section 5.1.4, we will need to use $N = 1$; the higher values of N led to overfitting. ■

5. Experimental results

In this section, we describe the use of the localized kernels discussed in Sections 3 and 4 in the context of gesture recognition with micro-Doppler radar readings (Section 5.1), and to demonstrate the fact that the theory is applicable agnostically to the domain knowledge, also in the context of a small video data set (Section 5.2). As remarked before, we use the kernel $\tilde{\Phi}_{N,q}(\rho(x, y))$ with different distances $\rho(x, y)$, corresponding to different types of manifolds, which depend on the feature representation.

5.1. Gesture recognition

In the case of micro-Doppler gesture recognition experiments, we use two kinds of features of the measured spectrogram: features based on the singular value decomposition of the spectrogram viewed as a matrix, and features based on the PCA components of a vectorized version of the spectrogram. For comparison, we include the results based on two CNNs, which worked with the same preprocessed data as the other methods. We do not consider hand-crafted features here, because both the SVD and PCA features yield substantially better results.

Table 1
The definition of various kernels used in our experiments.

Kernel type	Expression	Parameters
Grassmann kernel	$\exp(-\gamma(r - \ U_1^T U_2\ _F^2))$	$\gamma = 0.2$
Laplace kernel	$\exp(-\alpha\ U_1 - U_2\ _F - \beta\ \Sigma_1 - \Sigma_2\ _F)$	$\alpha = 0.2, \beta = 0.0042$
Gaussian kernel	$\exp(-\alpha\ U_1 - U_2\ _F^2 - \beta\ \Sigma_1 - \Sigma_2\ _F^2)$	$\alpha = 0.2, \beta = 0.12$
Localized kernel (3.4)	$\tilde{\Phi}_{N,q}(\gamma x)$	$\gamma = 0.8, q = 2 (n = 16) \text{ or } q = 18 (n = 64)$

In view of the results described in more detail in Usevich, Emiya, Brie, and Chaux (2018), the first set of features we considered are the first r columns of \mathbf{U} to be a feature vector \mathbf{U}_r for the signal \mathbf{D} in (3.13) (cf. (3.15)). We note that this feature vector is a point on the Grassmann manifold $\mathcal{G}(r, N)$, with geodesic approximations that can be calculated efficiently. As a variant of this set of features, we relax the assumption that subspaces of gesture spectrograms are suitable for discrimination, and consider incorporating the singular values by defining distances using both \mathbf{U} and Σ , which we refer to as the SVD manifold.

Lastly, we relax any assumptions about the discriminative capability of the SVD components between individual samples, and instead use Principle Component Analysis (PCA) to calculate a common low-dimensional subspace to project the data on. Our methods are guided by the paper (Zeng et al., 2020) by Zeng, Amin, and Shan. Precise definitions of the metrics and kernels used in these approaches are listed in Table 1.

We describe the data set in Section 5.1.1. The data involves 4 gestures made by 6 persons. We focus first on recognition of gestures independently of the person making the gesture. The preprocessing step is described in Section 5.1.2. Section 5.1.3 describes the various kernels which will be used in our experiments. The methodology of the experiments as well as the results are described in Section 5.1.4 for the case of SVD features, Section 5.1.5 for the case of PCA features, and Section 5.1.6 for the case of CNNs. The effect of choosing the dimension of the manifold is discussed in Section 5.1.7. The effect of choosing various percentage of the training data is described in Section 5.1.8. In Section 5.1.9, we examine the results of training on all the gestures made by 5 of the subjects, and predicting the results for the corresponding gesture by the 6th subject.

5.1.1. Data set

We use the publicly available micro-Doppler data set *Dop-NET* distributed by University College London Radar group (Ritchie et al., 2020). The data set consists of complex spectrograms formed using motion compensated, measured micro-Doppler signatures of five people performing four different hand gestures: snap, wave, pinch, click. The data is obtained using a linear FMCW at 24 GHz center and 750 MHz bandwidth. While a training and test set are provided, no ground truth values are provided for test set. Thus, we split the provided training set into a training and test set using a 80/20% split. This train/test split is used for all our experiments, except those in Section 5.1.8 where we investigate the effect of the size of the training set. In each of our experiments we report the mean and variance of 5 trials of independent sampling of the train/test set.

In Fig. 1, an example spectrogram is shown for each class. Since the time duration of each gesture varies, so does the number of columns in each spectrogram, which can vary by up to an order of magnitude in terms of number of samples. Furthermore, we note that there are visible similarities between the classes; for example, the pinch, swipe and click gesture are both visually similar in terms of frequency and time support, while the wave gesture lasts for longer time duration and demonstrates oscillatory behavior. Clearly, these similarities make this classification problem particularly difficult, and motivate the use of features that are robust to the within class variation of time and Doppler frequencies.

5.1.2. Data preprocessing

Prior to classification, we perform data preprocessing to reduce noise and clutter; then normalize to account for variation in the dynamic range. To suppress the effects of noise and clutter we investigated the use of various binary thresholding algorithms to segment the signal from background, including the envelope detection approach (Zeng et al., 2020). It was determined that Yen's threshold method worked best, and was used for all of our experiments (Yen, Chang, & Chang, 1995). We can explicitly summarize this process as

$$\tilde{X}_k = \mathcal{T} \{20\log_{10}(|X_k|)\} \quad (5.1)$$

where X_k is a sample from the data set, \mathcal{T} is thresholding function described in Yen et al. (1995).

Following thresholding, we normalize the data to account for the variation in the dynamic range. We consider two approaches to this normalization which are both used in the experiments. The first approach is to normalize the data point \tilde{X}_k such that each element is in the range [0, 1]. Our second approach is to convert the threshold spectrogram to a binary representation where each element is in the set {0, 1}.

The SVD features are based on r left singular vectors and corresponding singular values in the decomposition $\tilde{X}_k = U_k \Sigma_k V_k^T$, $k = 1, 2, \dots$, for various values of r .

To obtain the PCA features, the spectrograms are zero padded to the same size, then vectorized. Zero padding was also used for the inputs to the CNN. The PCA is made on the entire training data, and the projection of any datum on the first r PCA vectors was taken as the representation of that datum in this feature space.

5.1.3. Kernels

The set of kernels used in our experiments together with the values of the parameters are summarized in Table 1. The notation U , Σ refers to the left singular vectors and singular values of the data corresponding to the r largest singular values obtained from the spectrograms as described in Section 5.1.2, and $\|\cdot\|_F$ denotes the Frobenius norm. The three kernels considered are the *Grassmann kernel*, which uses the projection distance, an approximation of the geodesic on the Grassmann manifold (Ye & Lim, 2016). The *Laplace kernel* and *Gaussian kernel* use a distance function that incorporates both the singular vectors and singular values of the spectrograms. We note that the Gaussian kernel corresponds to the case where $N = 1$ and $q = 1$ in (3.4). The general kernel as defined in (3.4) is referred to as the *localized kernel*.

5.1.4. Performance accuracy for SVD features

We evaluate the performance of the proposed algorithms in terms of classification accuracy and measured computational time, as these are major considerations in transitioning technology. The results are organized by the type of preprocessing and feature type, where preprocessing refers to whether the data is normalized or binary, and feature type refers to methods that use components of the SVD decomposition as features or PCA projections. In this section, we focus on the SVD features. We use values of r given in Table 2, which were found to perform best for each SVD-based method in testing.

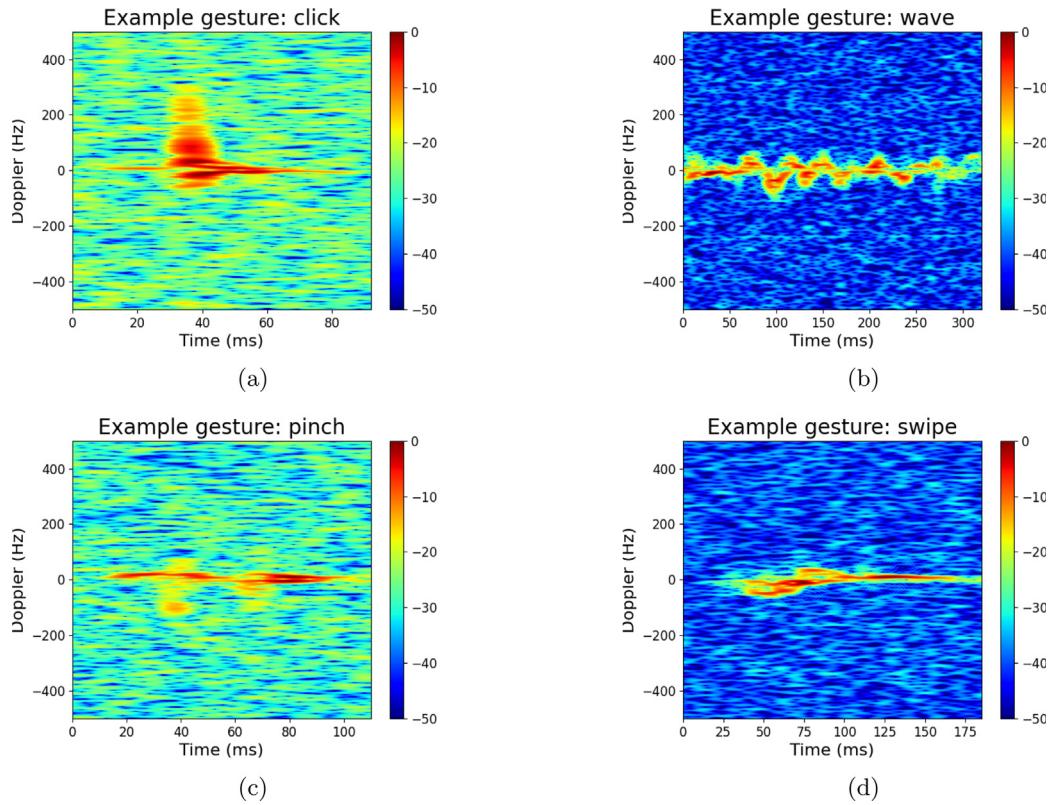


Fig. 1. A data sample selected from each of the four gestures, provided in the Dop-Net data set. Note that while the figures are scaled to be the same size, the time axis varies between different gestures. (a) Click. (b) Wave. (c) Pinch. (d) Swipe.

Table 2

The values of r (number of selected singular vectors and values) used for each of the SVD SVM methods.

Method	r
Gaussian SVD SVM (binary)	4
Grassmann SVD SVM (binary)	5
Laplace SVD SVM (binary)	11
Gaussian SVD SVM (normalized)	3
Grassmann SVD SVM (normalized)	7
Laplace SVD SVM (normalized)	19

To perform classification using the kernels we trained a support vector machine classifier, which was implemented in Python using SciKit-Learn (Pedregosa et al., 2011). In the discussion of these methods and the corresponding results we use the notation $**$ SVD SVM to denote the SVM based on SVD features and using the kernel $**$. For example, Grassmann SVD SVM refers to the SVM based on SVD features and the Grassmann kernel.

The results of these experiments are displayed in Table 3. The kernel methods based on the binary features performed in the range of 83.16% to 89.94%, with the Grassmann SVD SVM performing worst, and the Gaussian SVD SVM the best.

When the normalized samples are used, the Gaussian and Laplace SVD SVMs see a performance decrease over binary samples by 1.36% and 0.89%, respectively. The opposite is true for the Grassmann SVD SVM which increases by 3.49%. We believe the reduction in performance when using SVD derived features may be a result of an undesired invariance to magnitude and time variation, a consequence of using the singular vectors as a feature, and suggests that the singular values provide more useful information in the case of binary data. It is possible that the left singular vectors fail to represent certain non-stationary statistical behavior, as the time ordering is lost when calculating

Table 3

The performance accuracy, training time, and testing time with variance (written as accuracy \pm variance) for the kernel methods using SVD features are presented for the different preprocessing approaches. (a) Results for binary preprocessing. (b) Results for normalized preprocessing.

(a)			
Approach (Binary)	Average accuracy (%)	Test time (s)	Train time (s)
Gaussian SVD SVM	89.94 \pm 0.0	17.8 \pm 0.0	48.15 \pm 0.27
Grassmann SVD SVM	83.16 \pm 0.02	19.58 \pm 0.14	54.15 \pm 0.92
Laplace SVD SVM	87.02 \pm 0.01	21.47 \pm 0.01	61.8 \pm 0.04
(b)			
Approach (Normalized)	Average accuracy (%)	Test time (s)	Train time (s)
Gaussian SVD SVM	88.58 \pm 0.0	17.9 \pm 0.0	48.1 \pm 0.01
Grassmann SVD SVM	86.65 \pm 0.01	20.92 \pm 0.01	58.85 \pm 0.67
Laplace SVD SVM	85.13 \pm 0.02	25.12 \pm 0.02	76.41 \pm 1.29

the singular vectors. We leave a detailed analysis of these feature types for future work.

5.1.5. Performance accuracy for PCA features

In this section, we focus on the results based on the PCA features described in Section 5.1.2.

We used a 5-nearest neighbor algorithm (KNN) as well as SVM trained with the localized kernel $\tilde{\phi}_{N,q}$ defined in (3.4), which we denote as PCA LocSVM*, where * denotes the degree of the polynomial in (3.4); i.e., $2\lfloor N^2/2 \rfloor$. We note that an expression of the form $\sum_j a_j \tilde{\phi}_{N,q}(|\mathbf{x} - \mathbf{y}_j|_Q)$ resembles an average of a_j at \mathbf{x} , weighted depending upon the distance of \mathbf{x} from the point \mathbf{y}_j . Here Q is the number of singular values used in our computation; i.e., $Q = 30$. The optimal value of q is shown empirically to be 2 for degree $n = 16$ and 18 for degree $n = 64$. This

suggests that in the 30-dimensional feature space, the data “lives” on an 18-dimensional sub-manifold. In view of the localization property (3.8), many terms in this expression will be close to 0, and there are only a few terms involved in this average for any \mathbf{x} depending upon its distance from various points. So in some sense, this is also a nearest neighbor estimate. However, rather than prescribing how many nearest neighbors to use, we allow the parameter N to control this number effectively. Thus, how many neighbors are counted effectively depend upon how many points \mathbf{y}_j are there in a neighborhood of \mathbf{x} , the radius of this neighborhood controlled by N . We note that the weights involved in this averaging are not all positive. This may look unusual, but is an essential requirement to get to good theoretical results on the accuracy of approximation.

The binary results are shown in Table 4a, and the normalized results in Table 4b.

In the binary case, the PCA KNN performed identically with PCA LocSVM64 obtaining an accuracy score of 95.81% with very small variance. The smaller degree polynomial kernel PCA LocSVM16 performed slightly worse by 0.98%. There is 2.48% variation when using the normalized data, shown in Table 4b. The best performing method is PCA LocSVM64 achieving 96.81% accuracy. The PCA KNN achieved 95.36% which is 0.45% below the binary normalized. Based on these results, we conclude the overall performance between binary and normalized preprocessing techniques are nearly equivalent, with the binary preprocessing reducing variance between methods. The fact that the high order localized kernel based SVM method was able to accurately classify the PCA transformed data to a level comparable to a 5 nearest-neighbor classifier suggests that in the reduced space the data is dense, with the classes well separated. This property allows the kernel to be highly localized, but generalize well.

In terms of computational time, PCA KNN and LocSVM16 performed similarly in both train and testing, performing inference on the entire test set in less than a second, whereas the PCA LocSVM64 was an order of magnitude slower taking 5.69 s on the test set, and three times slower taking 62.27 s to train. We note however that our localized kernel is a very newly introduced kernel, and our computation is a brute force evaluation. It is a subject of future research to develop fast algorithms involving Hermite polynomial expansions based at the so-called scattered data,⁴ and in particular, the fast evaluation of the kernels taking into account their localization.

5.1.6. Performance accuracy for CNNs

Another popular approach to micro-Doppler gesture recognition are CNNs. We compare the proposed methods with two different mid-size CNN architectures used for gesture recognition (Kim & Toomajian, 2016; Kulhandjian, Sharma, Kulhandjian, & D'Amours, 2019), which we refer to as CNN1 and CNN2, respectively. The input to both of these networks is the preprocessed spectrograms (5.1) as described in Section 5.1.2, which were also used to extract the SVD and PCA features. This ensures that every approach uses the same preprocessed data, which is further zero padded in the case of the CNNs and PCA features. Both networks consist of 4 convolutional layers with ReLU activation followed by 2 : 1 max pooling, and the prediction is made using a single dense layer. For CNN1, the first layer consisted of 20 channels of 8×8 filters, the second layer is 10 channels of 16×16 filters, and the final layer is 5 channels of 32×32 filters. The three layers of CNN2 each consist of 5 channels with 5×5 , 4×4 , 2×2 sized filters. Both networks were implemented using PyTorch (Paszke et al., 2019) and trained using cross-entropy loss

⁴ This term refers to data points whose location is not prescribed in a specific way, for example, zeros of Hermite polynomials.

Table 4

The performance accuracy, training time, and testing time with variance (written as accuracy \pm variance) for the kernel methods using the different PCA feature based methods considered and different preprocessing schemes. Results are presented also for the two CNNs (Section 5.1.6). (a) Results for binary preprocessing. (b) Results for normalized preprocessing.

(a)			
Approach (Binary)	Average accuracy (%)	Test time (s)	Train time (s)
PCA KNN	95.81 \pm 0.0	0.42 \pm 0.0	22.49 \pm 0.23
PCA LocSVM16	94.83 \pm 0.0	0.72 \pm 0.0	24.89 \pm 0.05
PCA LocSVM64	95.81 \pm 0.0	5.69 \pm 0.0	62.27 \pm 0.14
CNN1	90.06 \pm 0.01	12.69 \pm 0.01	14770.88 \pm 749.08
CNN2	88.38 \pm 0.01	4.79 \pm 0.01	2162.59 \pm 952.77

(b)			
Approach (Normalized)	Average accuracy (%)	Test time (s)	Train time (s)
PCA KNN	95.36 \pm 0.0	0.53 \pm 0.02	23.51 \pm 0.57
PCA LocSVM16	93.72 \pm 0.01	0.73 \pm 0.0	25.61 \pm 0.1
PCA LocSVM64	96.18 \pm 0.0	5.97 \pm 0.15	64.53 \pm 1.5
CNN1	92.85 \pm 0.01	12.7 \pm 0.01	14746.82 \pm 342.77
CNN2	91.38 \pm 0.01	4.75 \pm 0.0	2133.72 \pm 307.46

function with the adaptive moment estimation (Adam) optimizer for 100 epochs, at which point the models had converged. A mini-batch size of 16 and learning rate 0.001 was used. We also experimented using networks with increased feature dimension, but this led to over-fitting due to the small size of the data set. Therefore, pursuing deeper networks was unlikely to yield improved results. Though it is worth noting that due to the small training set, CNN1 and CNN2 are still significantly over parameterized.

The results are given together with those for the PCA features in Table 4. With both the binary and normalized spectrograms, the PCA KNN as well as the methods PCA LocSVM16 and PCA LocSVM64 based on our localized kernels clearly outperform both the CNNs both in terms of accuracy and in terms of training time. The testing time for CNNs is comparable with that of PCA LocSVM64.

In comparison with methods based on SVD features, we observe that the Gaussian SVM surpasses CNN2 by 1.56% in the case of binary preprocessing. When normalized features are used, CNN1 and CNN2 see accuracy improvements of 1.79% and 3.00%, respectively. The improvement in CNN performance is clearly a result of the increased information representation in the dynamic range of the spectrograms, where as only the support is available in the binary case.

In terms of time complexity in comparison with SVD based features, CNN2 performs inference on the entire data set in 4.7–4.8 seconds, but at the cost of 35 minutes to train. CNN1 is a larger network and saw inference time increase to 12.7 s and more than 4 h to train. In contrast, the kernel methods inference time on the test ranged from 12.69 to 25.12 s and training times in the range 48.15 to 76.41 s.

While the CNNs maintain a slight classification accuracy and inference time increase over the SVD kernel methods, it takes significantly larger amount of time to train. While the CNNs appear to be faster than the kernel methods, the variation in inference time is most likely a result of the fact that Pytorch is a C++ library wrapped in Python, whereas Sci-Kit Learn is pure python, which is much slower.

Overall, we found that using PCA transformed data was a better feature selection than those based on the singular vectors and values of the individual data points. In fact, it outperformed both CNNs best performance by an average of 3% accuracy. Based on performance, we conclude that the LocSVM64 kernel method performs very closely to PCA KNN and best overall.

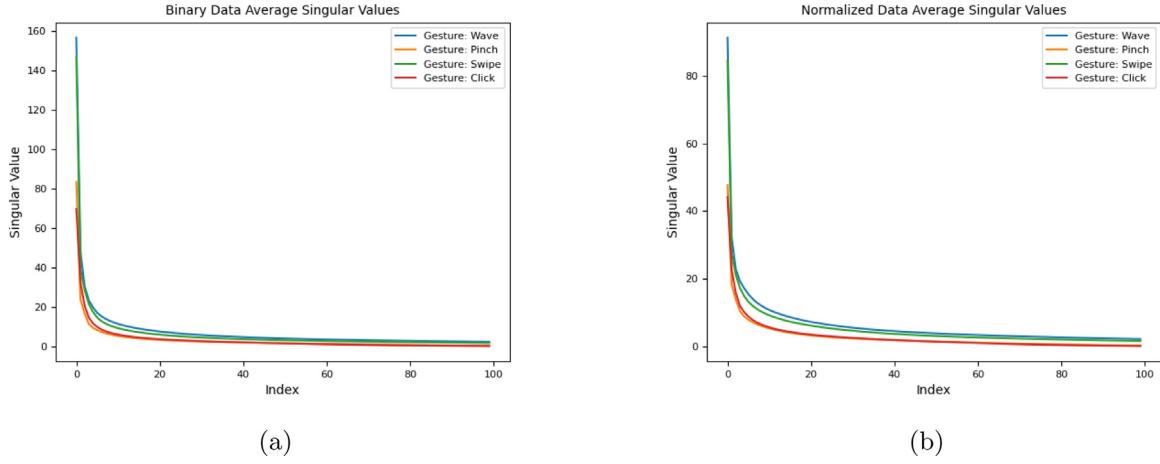


Fig. 2. Plot of the first 100 average singular values for each gesture. (a) Binary preprocessing. (b) Normalized preprocessing.

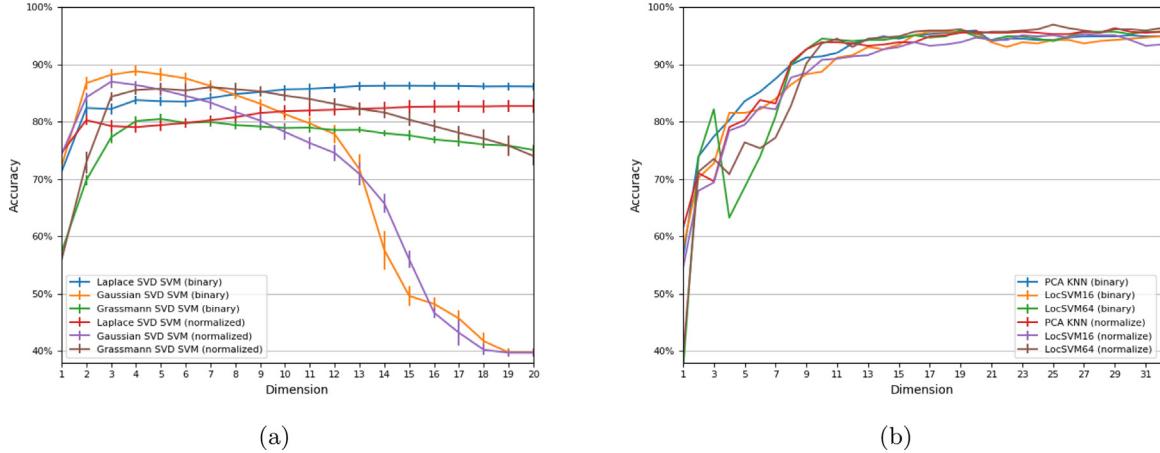


Fig. 3. Classification accuracy averaged vs. the dimension r of the feature manifold for each method considered on both binary and normalized preprocessing schemes. (a) SVD-based methods. (b) PCA-based methods. In this plot the y-axis is 40% to 100%.

5.1.7. Sensitivity to feature dimension

In this section, we investigate how well the proposed kernels perform when the total number of left singular vectors r changes. We carry out this experiment by using only the top r singular values and corresponding singular vectors of the data. This can be viewed as function of information captured by the features used to construct the kernels defined in Table 1. Ideally, r should be chosen large enough to retain a sufficient amount of information required for optimal classifier performance. Since the received signal spans a low-dimensional subspace, mentioned in Section 3.2, we expect that the different methods will plateau after a certain dimension due to the maximal amount of information being captured.

In Fig. 2, we show the singular values averaged over all realizations of each gesture. Clearly, for each gesture and preprocessing technique there is a sharp drop before $r = 20$ and the rest of the singular values are near zero. This suggests that the dimension of the subspace features need not be large, this is expected based on the micro-Doppler structure described in Section 3.2.

In Fig. 3(a) we plot the average classification error of the methods based on SVD features for $r = 1, \dots, 20$ and the different preprocessing techniques. For $r = 1$ all the SVD kernels out-perform the Grassmann kernels by approximately 18%. This discrepancy in performance reduces as r increases to 3–4, at which point the performance of the Gaussian kernel begins to drop, while the Grassmann and Laplace kernels perform relatively constant with a few percent variance. At this time, we are not able

to explain the drop in performance of the Gaussian kernels, which we find interesting.

The proposed approaches using PCA features, are shown in Fig. 3(b). Here, the number r of PCA components is the dimension of the ambient space, denoted by Q in (3.4). For the LocSVM tests, we adjust the manifold dimension q to ensure that it is no greater than the feature space dimension r . Specifically, for values of r at or above 18, q is set to 18 for LocSVM64. For values of r less than 18, q is set to r . Since LocSVM16 was found to perform best for $q = 2$, it did not need adjusting to remain below r , except in the case when $r = q = 1$. The performance of all methods and preprocessing techniques perform similarly, achieving accuracy in the mid 90% for sufficiently large r . Choice of preprocessing seemingly has little effect on performance, observed in the prior experiments as well. There is also a large drop off in performance as the dimension reduces from $r = 10$. The KNN PCA and LocSVM64 degrade similarly, with classification performance collectively falling by 30%. The LocSVM16 performs worse, with accuracy dropping over 70% as r reduces from 10 to 1. This drop is expected since for very small r only a small amount of discriminative information of the data is preserved.

5.1.8. Robustness to limited training data

In many radio-frequency machine learning applications, training data is difficult to collect and will likely be the limiting factor of the model's performance. In the case of gesture recognition, not only is cost and effort high, but the data collection

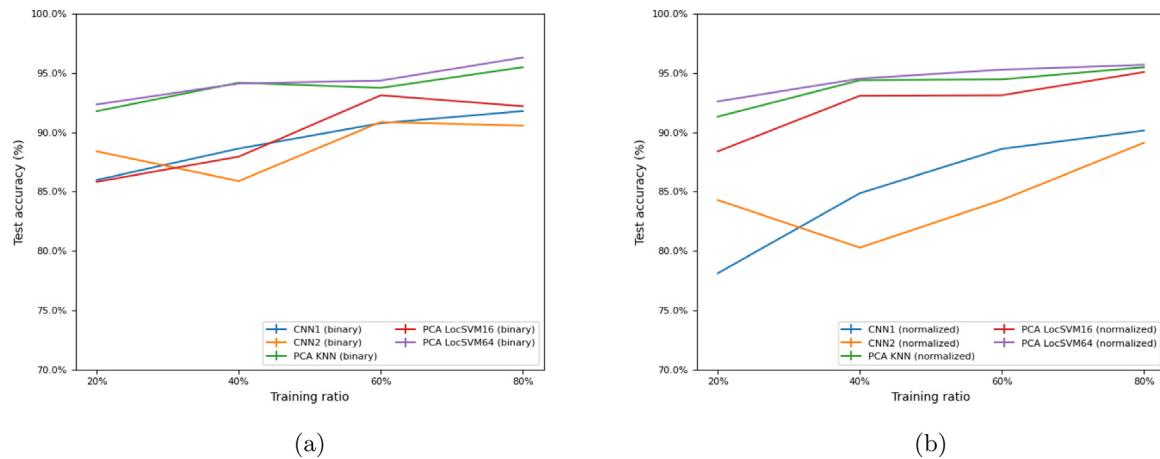


Fig. 4. Test error vs. training size for each PCA-based method and preprocessing approach considered. (a) Binary preprocessing. (b) Normalized preprocessing. In this plot the y-axis is 70% to 100%.

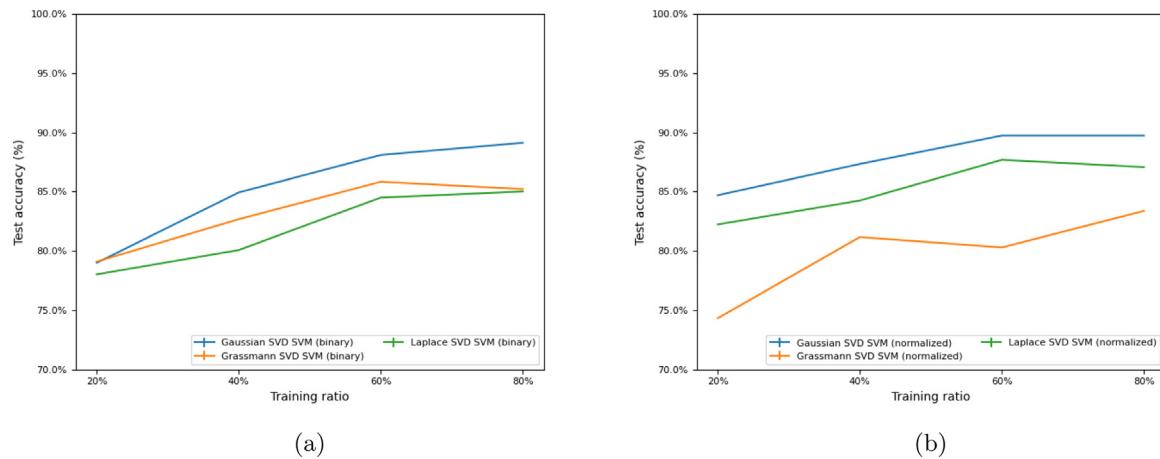


Fig. 5. Test error vs. training size for each and preprocessing approach based on SVD. (a) Binary preprocessing is used. (b) Normalized preprocessing is used. In this plot the y-axis is 70% to 100%.

involves human subjects (Gürbüz & Mason, 2020). Thus, it is important to develop an approach that is robust to limited data. As a result of the complexity of data collection, what is considered a large data set in micro-Doppler recognition problems will pale in comparison to those used as benchmarks in the Deep Learning community, such as ImageNet used in object recognition (Deng et al., 2009). For example, the Dop-net data set consists of 2433 training samples and probably less testing samples,⁵ while ImageNet has 14 million.

We evaluate each approach on {20, 40, 60, 80}% of the training data for five trials and average the results. Figs. 4 and 5 show the classification accuracy for the methods based on SVD and PCA features for different training set sizes, respectively. The total amount of data used for training varied between 487 and 1947 samples.

Fig. 4(a) shows the results for the PCA based features with binary preprocessing. As expected, the classification accuracy increases with the size of the data set, this is also the case for normalized data, shown in Fig. 4(b). For both the binary and normalized data preprocessing the PCA-KNN and PCA LocSVM64 demonstrate equivalent performance over the range of training data. In the binary case, all the methods and training sizes classification accuracy are in the range 85.82% to 96.30%, and 88.39%

to 95.65% for normalized preprocessing. It is notable that in the case of normalized preprocessing, PCA LocSVM64 does marginally outperform PCA KNN by 0.14%–1.28%, while the PCA LocSVM16 under performs in every case, and by a larger margin when binary preprocessing is used.

Ignoring the under performance of PCA LocSVM16, the performance range reduces to 91.78% to 96.30% for binary preprocessing and up to 91.32% and 95.69% for normalized preprocessing. Considering the tight performance range and inconsistency of any of the methods to consistently outperform, we conclude that the methods are robust to limiting the amount of training data. While this does not necessarily guarantee robustness when the data set size gets significantly larger, in the case of PCA LocSVM64 and PCA KNN we expect this observed robustness will hold. The reason is that when using PCA features the data manifold in this case allows a greater separation among the classes than that based on SVD features. Therefore, the PCA LocSVM64 can better approximate the data with fewer samples. In theory, of course, the main problem is to find the right feature space; the straightforward approximation procedure developed in Mhaskar (2020a) obviates the need to do any training at all.

The SVD features result in more variation of classification accuracy and are shown in Fig. 5. With the exception of CNN2, which has an accuracy drop at 40% train/test ratio, all the methods demonstrate a clear monotonic increase in accuracy with increasing training data. The SVD features in the binary preprocessing are shown in Fig. 5(a), where the range of classification

⁵ Since the test set is not publicly released, we are assuming the test set is small compared to the training set, which is typical ML practice.

accuracy is 78.02% to 90.14%. The normalized preprocessed SVD feature performance are shown in [Fig. 5\(b\)](#), with classification accuracy in the range of 78.02% to 91.79%. Overall, there is a clear reduction in performance when using SVD features opposed to PCA. The PCA LocSVM64 outperformed the SVD features by 6.16% and 3.86% for the binary and normalized preprocessing methods.

In the binary case, CNN1 and CNN2 outperformed the methods using SVD features, except for CNN2 when the train/test ratio was 40%. Following the CNNs in classification performance was the Gaussian SVD SVM, Laplace SVD SVM and the Grassmann SVD SVM, listed in order of descending performance. With normalized preprocessing the Gaussian SVM outperforms all other methods, except for CNN1 with 80% train/test ratio, in which case CNN1 only outperforms by 0.41%.

Clearly, the proposed manifold learning approach appears robust to using just a few hundred samples. We attribute this property to a number of factors, first of which is micro-Doppler radar data can be very sensitive to multiple scattering and environmental effects. The choice of preprocessing clearly plays a role in suppressing these nuisance variables, which is then improved by the fact that using only r singular vectors as features further reduces uninformative information.

Furthermore, the r dimensional subspace representation is able to capture the time-varying properties of the data, which correspond to important discriminative features in the original time-series and improves generalization. This is further motivated, by noting that the relationship between the subspace and received radar signals, which span a low-dimensional subspace. To understand this, we observe that in [\(3.14\)](#) the rank of the spectrogram is determined only by the rank of the matrix \mathbf{S} . Therefore, since the micro-Doppler signature consists of a linear combination of delayed and Doppler shifted signals from the dominant scattering points, the rank of the resulting matrix \mathbf{S} is small, thus, it is well captured with only a small number of singular vectors. Furthermore, the proposed kernel provides good localization at data points on the manifold, which we suspect helps to isolate nuisance variables in the data.

It is very surprising that the CNN continues to perform well with limited training samples. While this may seem impressive, it is difficult to generalize this claim to all micro-Doppler data sets as the literature shows varying performance when similar models are used on different data sets. We think this may be a result of the fact that the training set is already so small to begin with, so training either the full data set or only 20% does not make a significant difference in performance due to the very large over-parameterization of the neural networks. This is in contrast to our kernel approach where the model does not over-parameterize the data. Furthermore, this peculiar performance may also be a result of robustness added by the preprocessing used prior to input into the CNN.

5.1.9. Cross-subject generalization

Generalization is an important aspect in any machine learning task and seeks to ensure that the algorithm will perform well on out-of-sample data. In micro-Doppler radar, there are a number of factors that can cause variation of the in- and out-sample distributions, like environmental variations leading to changes in the electromagnetic (EM) wave propagation medium, changes in material from which the EM waves scatter off, and non-stationary clutter and multiple scattering effects. For example, in synthetic aperture radar target classification, it has been observed that deep learning classifiers will fail to generalize from synthetic data to measured data. This is attributed to the fact that neural networks tend to learn background surface statistics instead of true features of the object of interest ([Jo & Bengio, 2017](#)). In this case, the problem is that simulating the scattering off foliage is

computationally intractable. These types of challenges have also been observed in classification of communication signals, where algorithms will fail to generalize to data collected on different days.

Since the data set consists of collections from six different individuals, it is an interesting question as to whether the manifold learning approach is robust to testing on individuals it was not trained on. To investigate this, we train on five of the six individuals and test on the remaining one. We present these results in [Fig. 6](#). From these results, it is interesting to see that there is a 40% to 50% variation in performance between different combinations of methods and the person held out of training. This suggests that there must be characteristics between each person not necessarily captured by the others as possible environmental effects are mitigated by the fact that the data is collected in a lab experiment.

For example, in the case of PCA features, all the methods perform most poorly on Person *B*. In the binary case, PCA LocSVM16 performs worse with accuracy score 6.17% and PCA KNN performs best, scoring 24.20% accuracy. The performance increased by 20% to 40% in the case of Person *A* and increased by 40% to 60%, in the case of either preprocessing choice. There is no consistent pattern of out performance by any particular approach, but they do follow the same trend of when they do and do not generalize well.

In the case of SVD features, the overall performance is lower, ranging from 18.77% for the Grassmann SVD SVM when tested on Person *B*, up to 81.21% for CNN2 with normalized preprocessing when tested on person *D*. Similarly for PCA features, there is no approach that consistently outperforms all other methods for each person. An interesting behavior is how the Grassmann SVD SVM performs worse than the Gaussian SVD SVM and Laplace SVD SVM on Person *A* and *B*, but then outperforms on person *C*, and comparably for Persons *D*, *E* and *F*.

Based on these results it is clear that the choice of preprocessing does not have a meaningful impact in generalization across different persons, though there is a slight out performance of normalization preprocessing. Clearly, for some persons the methods are able to generalize well, whereas in some cases they do not. We suspect this may have to do with variation in how different persons make specific gestures and/or body size, which can effect the distance and orientation of the scattering centers to the radar. This could effect both the amplitude and support of the spectrogram and no longer have the same distribution as the training data.

While these results do not provide a clear pattern, it motivates the requirement for a more in depth and controlled study as to what physical effects of the micro-Doppler data are important for achieving good generalization. Thus, we conclude that there is relationship between understanding the data from a physics perspective and what are appropriate approaches. Overall this suggests that optimal performance is as much a function of the ML algorithm, as it is the data collection approach and preprocessing.

5.2. Performance on Video-Data

The purpose of this section is to demonstrate that our kernel can be used agnostically to the domain knowledge about the data. Toward this goal, we examine a small and simple video data set (without the audio component) where the objective is to identify which digit was spoken in the video. In Section [5.2.1](#), we describe the data set. Our methodology and results are described in Section [5.2.2](#).

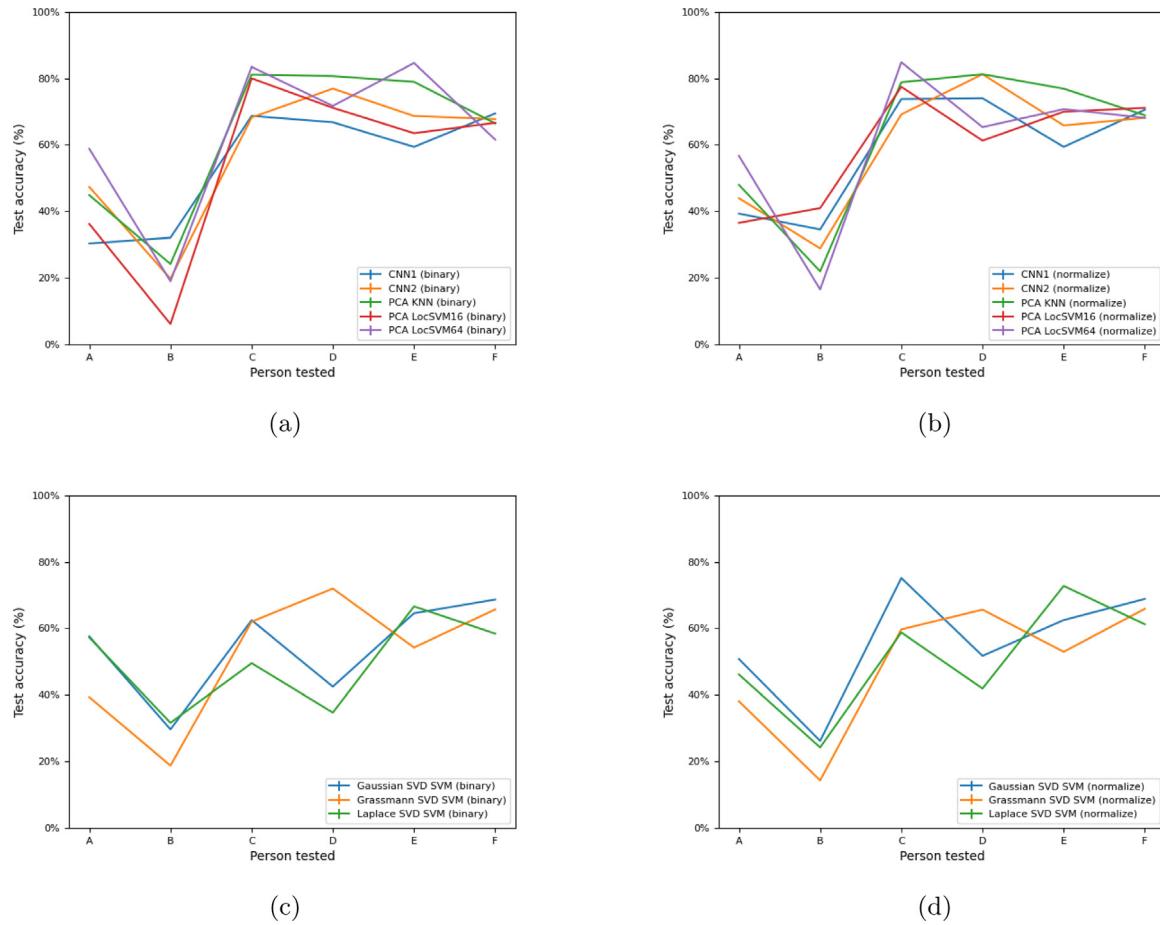


Fig. 6. Test accuracy for each algorithm and preprocessing approach when the methods are trained on data from five people and tested on the sixth. (a) PCA based features with binary preprocessing. (b) PCA based features with normalized preprocessing. (c) SVD features with binary preprocessing. (d) SVD features with normalized preprocessing.

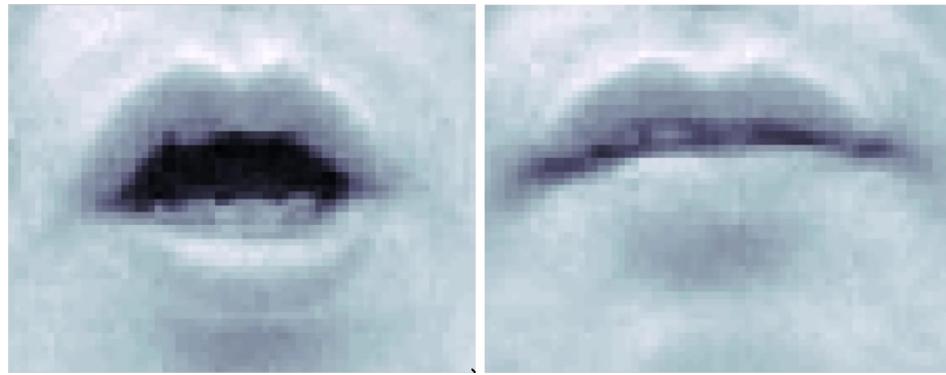


Fig. 7. Two example frames from lip video data set.

5.2.1. Data set

We use the data described in [Lieu and Saito \(2011\)](#). The data consists of video clips of the same person pronouncing digits from 1–5 in English. Each of the five digits is recorded ten times, yielding 50 total videos. Each video is cropped to a 70×55 area around the mouth, with the position of the nose held constant, and then converted to grayscale. The problem is to read what digit between one and five the subject is pronouncing in each video clip, effectively a signal classification problem among 5 classes (see Fig. 7).

5.2.2. Methodology and results

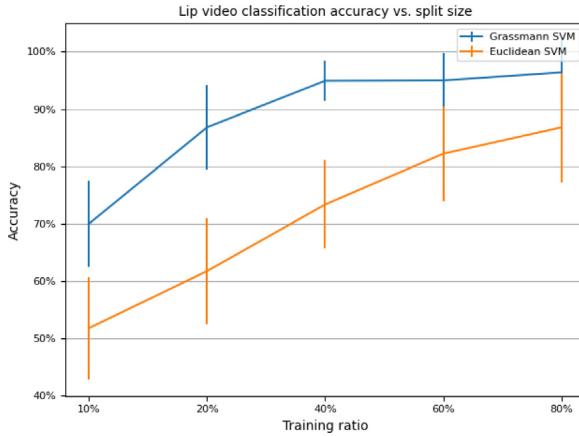
For this data set, we test the two kernels: the Grassmann kernel as used above as well as a standard radial-basis function (RBF) kernel, also referred to as the Gaussian kernel on Euclidean space, to establish a performance baseline using a support vector machine to perform classification. Both the kernels and the corresponding hyperparameters are defined in [Table 5](#). Each frame of the video clips is flattened into a 3850×1 row vector, yielding a $3850 \times \tau$ signal for each clip, where τ denotes the number of frames in the clip. For the RBF kernel, we further flatten the

Approach	Average Accuracy (%)	Train Time (s)	Test Time (s)
Grassmann SVM	95.84±3.29	3.06±0.39	3.15±0.48
Euclidean SVM	75.20±8.31	2.43±0.28	2.51±0.47
PCA-EMD [22]	94.70		
PCA-HD [22]	90.60		

Fig. 8. Classification accuracy of three kernels on lip video data set.**Table 5**

The definition of various kernels used for lip video data set.

Kernel type	Expression	Parameters
Grassmann kernel	$\exp(-\gamma(r - \ U_1^T U_2\ _F^2))$	$\gamma = 0.2$
Euclidean kernel	$\exp(-\gamma\ x - x'\ ^2)$	$\gamma = 2.1 \cdot 10^{-7}$

**Fig. 9.** Classification accuracy vs. training/test data set split ratio on lip video data set. In this plot the y-axis is 40%–100%.

signal into a $(3850 \times \tau) \times 1$ row vector. In accordance to the experiments in [Lieu and Saito \(2011\)](#), the training set consists of 50% data and the test data is the remaining 50%. Specifically, we randomly choose 5 videos from each of the five digit classes to use as training data, totaling 25 videos. We run each test 25 times with randomly-chosen splits each time and average the results below. We compare the performance of our methods to the PCA-EMD algorithm used by [Lieu and Saito \(2011\)](#), where the data is transformed into a lower-dimensional representation using principal component analysis, then the earth-mover's distance is calculated and used for nearest-neighbor classification (see [Fig. 8](#)).

The Grassmann kernel performs the best at 95.84% accuracy, outperforming the two methods (PCA-EMD and PCA-HD) originally used on the data set, as well as the basic Euclidean kernel.

As above, we perform the experiment using differently-sized subsets of training data for the classifiers to evaluate the robustness of each approach to small training sets. Each approach is evaluated on {10, 20, 40, 60, 80}% of the training data for 25 trials each, and each train/test split retains an equal ratio of each class. The results are shown in [Fig. 9](#). Similar to the DopNet data set, performance does not vary significantly within 40%–80% training size using the Grassmann SVM method, suggesting that the Grassmann kernel has strong predictive power with just 20 samples in the training set.

6. Proofs

The proofs of both [Theorems 4.1](#) and [4.2](#) involve the solution of an interpolation problem. Accordingly, our first objective is

to study this problem when Φ_N is an admissible kernel (cf. [Definition 4.2](#)). We assume the set up as in [Section 4](#).

For integer $M \geq 2$, distinct points $y_1, \dots, y_M \in \mathbb{X}$, and real numbers f_1, \dots, f_M , we consider the interpolation problem

$$\sum_{k=1}^M a_k \Phi_N(y_j, y_k) = f_j. \quad (6.1)$$

Our first goal is to investigate when the collocation/normal matrix $[\Phi_N(y_j, y_k)]_{j,k=1}^M$ is invertible and to estimate the norm of this matrix.

Theorem 6.1. *Let $M \geq 2$ be an integer, $\mathcal{C} = \{y_1, \dots, y_M\} \subset \mathbb{X}$, $\eta = \eta(\mathcal{C})$ be defined as in [\(4.8\)](#), $\tilde{\eta} = \min(1, \eta/3)$. We assume that Φ_N is a (q, S) -localized kernel that satisfies [\(4.6\)](#), and further that [\(4.10\)](#) and [\(4.11\)](#) are satisfied. There exists a positive constant C such that if $N \geq C\tilde{\eta}^{-1}$, then the system of equations [\(6.1\)](#) has a solution $\{a_k^*\}$ satisfying*

$$\max_{1 \leq k \leq M} |a_k^*| \lesssim N^{-q} \max_{1 \leq j \leq M} |f_j|, \quad (6.2)$$

and

$$\max_{1 \leq k \leq M} |a_k^* \Phi_N(y_k, y_k) - f_k| \lesssim (N\eta)^{q-S} \max_{1 \leq j \leq M} |f_j|. \quad (6.3)$$

It is convenient to formulate certain technical details of the proof in terms of the notion of a regular measure. If ν is a (positive or signed) measure on \mathbb{X} , we denote by $|\nu|$ its total variation measure. If $d \geq 0$, we say that ν is d -regular if

$$\|\nu\|_d = \sup_{\substack{x \in \mathbb{X} \\ r > 0}} \frac{|\nu|(\mathbb{B}(x, r))}{(r + d)^q} < \infty. \quad (6.4)$$

For example, μ^* itself is a 0-regular measure, with $\|\mu^*\|_0$ being the constant involved in [\(4.10\)](#). Another important example is given in the following lemma.

Lemma 6.1. *Let $\mathcal{C} = \{y_1, \dots, y_M\}$ be distinct points in \mathbb{X} , $\eta = \eta(\mathcal{C})$ be as in [\(4.8\)](#), $\tilde{\eta} = \min(1, \eta/3)$, and ν be the counting measure; i.e., the measure that associates the mass 1 with each y_j . Then*

$$\|\nu\|_{\tilde{\eta}} \lesssim \frac{1}{\tilde{\eta}^q}. \quad (6.5)$$

Proof. Let $x \in \mathbb{X}$, $r > 0$, and J be the number of points in $\mathcal{C} \cap \mathbb{B}(x, r)$. By re-arranging the indices, we may assume without loss of generality that $\mathcal{C} \cap \mathbb{B}(x, r) = \{y_1, \dots, y_J\}$. Then the balls $\mathbb{B}(y_j, \tilde{\eta})$ are pairwise disjoint and the union of these balls is a subset of $\mathbb{B}(x, r + \tilde{\eta})$. Therefore, using the conditions [\(4.10\)](#) and [\(4.11\)](#) on measures of balls, we deduce that

$$\begin{aligned} J\tilde{\eta}^q &\lesssim \sum_{j=1}^J \mu^*(\mathbb{B}(y_j, \tilde{\eta})) \leq \mu^*\left(\bigcup_{j=1}^J \mathbb{B}(y_j, \tilde{\eta})\right) \\ &\leq \mu^*(\mathbb{B}(x, r + \tilde{\eta})) \lesssim (r + \tilde{\eta})^q. \end{aligned}$$

This implies [\(6.5\)](#). ■

Lemma 6.2. *Let $d \geq 0$ and ν be a d -regular measure. Let Φ_N be a (q, S) -localized kernel. If $N \geq 1$ and $r \geq 1/N$, then*

$$\sup_{x \in \mathbb{X}} \int_{\Delta(x, r)} |\Phi_N(x, y)| d|\nu|(y) \lesssim \|\nu\|_d (Nr)^{q-S} (1 + d/r)^q. \quad (6.6)$$

Hence,

$$\sup_{x \in \mathbb{X}} \int_{\mathbb{X}} |\Phi_N(x, y)| d|\nu|(y) \lesssim \|\nu\|_d (1 + Nd)^q, \quad (6.7)$$

and

$$\sup_{x \in \mathbb{X}} \int_{\mathbb{X}} |\Phi_N(x, y)|^2 d|\nu|(y) \lesssim \|\nu\|_d N^q (1 + Nd)^q \quad (6.8)$$

In particular, taking $d\nu = gd\mu^*$, where $g \in C(\mathbb{X})$, we have $d = 0$, $\|\nu\|_0 \lesssim \|g\|_\infty$, and obtain for $N > 0$

$$\sup_{x \in \mathbb{X}} \int_{\mathbb{X}} |\Phi_N(x, y)| |g(y)| d\mu^*(y) \lesssim \|g\|_\infty. \quad (6.9)$$

Proof. Without loss of generality, we may replace $|\nu|/\|\nu\|_d$ by ν , and thereby, assume that ν is a positive measure with $\|\nu\|_d = 1$. In this proof only, let $x \in \mathbb{X}$, $A_k = \{y \in \mathbb{X} : 2^k r \leq \rho(x, y) < 2^{k+1}r\}$. Clearly, (6.6) implies that $\nu(A_k) \lesssim 2^{kq} r^q (1 + d/r)^q$. Therefore, using (4.2), we deduce that

$$\begin{aligned} \int_{\Delta(x, r)} |\Phi_N(x, y)| d|\nu|(y) &= \sum_{k=0}^{\infty} \int_{A_k} |\Phi_N(x, y)| d|\nu|(y) \\ &\lesssim N^q \sum_{k=0}^{\infty} \int_{A_k} \frac{d\nu(y)}{(N\rho(x, y))^S} \\ &\lesssim N^{q-S} r^{-S} \sum_{k=0}^{\infty} 2^{-ks} \int_{A_k} d\nu(y) \\ &\lesssim (Nr)^{q-S} (1 + d/r)^q \sum_{k=0}^{\infty} 2^{-k(S-q)}. \end{aligned}$$

This proves (6.6). To prove (6.7), we observe that

$$\begin{aligned} \int_{\mathbb{B}(x, 1/N)} |\Phi_N(x, y)| d\nu(y) &\lesssim N^q \nu(\mathbb{B}(x, 1/N)) \lesssim N^q (1/N + d)^q \\ &= (1 + Nd)^q. \end{aligned}$$

The same estimate is obtained for the integral over $\Delta(x, 1/N)$ by using (6.6) with $r = 1/N$. We arrive at (6.7) by adding these estimates. Since (4.2) shows that $|\Phi_N(x, y)| \lesssim N^q$ for all $x, y \in \mathbb{X}$, we get from (6.7) that for any $x \in \mathbb{X}$,

$$\int_{\mathbb{X}} |\Phi_N(x, y)|^2 d|\nu|(y) \lesssim N^q \int_{\mathbb{X}} |\Phi_N(x, y)| d|\nu|(y) \lesssim N^q (1 + Nd)^q. \quad \blacksquare$$

Proof of Theorem 6.1. In this proof, we simplify our notation, and write η in place of $\tilde{\eta}$. We consider first the measure ν as in Lemma 6.1, so that with $d = \eta$, $\|\nu\|_d \lesssim \eta^{-q}$. If $N\eta \geq 1$, then we may use (6.6) with $r = \eta$ to obtain

$$\begin{aligned} \max_{1 \leq k \leq M} \sum_{j \neq k} |\Phi_N(y_j, y_k)| &\leq \int_{\Delta(y_j, \eta)} |\Phi_N(y_j, y)| d\nu(y) \\ &\lesssim N^q (N\eta)^{-S} (1 + N\eta)^q \lesssim N^q (N\eta)^{q-S}. \end{aligned} \quad (6.10)$$

In view of our assumption (4.6), $\Phi_N(y_j, y_j) \gtrsim N^q$. Further, we recall that $S > q$. So, if $N\eta \gtrsim 1$, the matrix $\mathbf{L} = [\Phi_N(y_k, y_k)]_{j,k=1}^M$ satisfies

$$\max_{1 \leq k \leq M} \sum_{j \neq k} |\Phi_N(y_j, y_k)| \leq (1/2) \Phi_N(y_k, y_k), \quad k = 1, \dots, M.$$

The facts that \mathbf{L} is invertible, and (6.2) holds, now follow from well known facts in linear algebra, e.g., Mhaskar (2010, Proposition 6.1). The estimate (6.3) is clear from (6.2) and (6.10). \blacksquare

Proof of Theorem 4.1. The conditions of Theorem 4.1 ensure those in Theorem 6.1 are satisfied with the kernel Φ_N . Hence, a solution of the equation in (4.12) exists and therefore, is automatically the minimizer of the empirical risk. It remains to obtain the

other bounds in the theorem as indicated. Once more, we denote $\tilde{\eta}$ by η . We consider first the case when $\delta(x) \geq \eta/3$. Then using Lemma 6.2 with ν being the measure as in Lemma 6.1, and then using (6.2) we obtain

$$\begin{aligned} \left| \sum_{k=1}^M a_k^* \Phi_N(x, y_k) \right| &\lesssim \|F\|_\infty N^{-q} \sum_{k=1}^M |\Phi_N(x, y_k)| \\ &\lesssim \|F\|_\infty N^{-q} \int_{\Delta(x, \eta/3)} |\Phi_N(x, y)| d\nu(y) \\ &\lesssim \|F\|_\infty (N\eta)^{-S}. \end{aligned} \quad (6.11)$$

Next, we consider the case when $\delta(x) \leq \eta/3$. In this case, there is a unique y_ℓ with $\rho(x, y_\ell) = \delta(x)$, and $y_k \in \Delta(x, \eta/3)$ for all $k \neq \ell$. Arguing as before, we see that

$$\left| a_\ell^* \Phi_n(x, y_\ell) - \sum_{k=1}^M a_k^* \Phi_n(x, y_k) \right| \lesssim \|F\|_\infty (N\eta)^{-S}. \quad (6.12)$$

Therefore, using the Lipschitz conditions (4.7), (6.2), (6.3), we conclude that

$$\begin{aligned} \left| \sum_{k=1}^M a_k^* \Phi_n(x, y_k) - F(x) \right| &\lesssim |a_\ell^* \Phi_n(x, y_\ell) - F(x)| + \|F\|_\infty (N\eta)^{-S} \\ &\leq |a_\ell^* \Phi_n(x, y_\ell) - a_\ell^* \Phi_n(y_\ell, y_\ell)| \\ &\quad + |a_\ell^* \Phi_n(y_\ell, y_\ell) - F(y_\ell)| + |F(y_\ell) - F(x)| \\ &\quad + \|F\|_\infty (N\eta)^{-S} \\ &\lesssim (N + \|F\|_{\text{Lip}}) \delta(x) + \|F\|_\infty (N\eta)^{q-S}. \quad \blacksquare \end{aligned}$$

In order to prove Theorem 4.2, we prove first the following lemmas, which will enable us to apply Theorem 4.1 with $\sigma_N(f_0 F)$ (cf. (4.16)) in place of F and Ψ_N defined in (4.15) in place of Φ_N .

Lemma 6.3. Let $\{\Phi_N\}$ be a (q, S) -localized family of kernels, and Ψ_N is defined by (4.15). If $d\tau = f_0 d\mu^*$ for some $f_0 \in C(\mathbb{X})$, then $\{\Psi_N\}$ is a (q, S) -localized family of kernels.

Proof. In this proof, let $y \neq x \in \mathbb{X}$, and $\delta = \rho(x, y)/3$. Then $\mathbb{B}(x, \delta) \subseteq \Delta(y, \delta)$ and

$$\begin{aligned} |\Psi_N(x, y)| &= \left| \int_{\mathbb{X}} \Phi_N(x, z) \Phi_N(y, z) d\tau(z) \right| \\ &\leq \int_{\Delta(x, \delta)} |\Phi_N(x, z) \Phi_N(y, z)| d\tau(z) \\ &\quad + \int_{\mathbb{B}(x, \delta)} |\Phi_N(x, z) \Phi_N(y, z)| d\tau(z) \\ &\leq \int_{\Delta(x, \delta)} |\Phi_N(x, z) \Phi_N(y, z)| d\tau(z) \\ &\quad + \int_{\Delta(y, \delta)} |\Phi_N(x, z) \Phi_N(y, z)| d\tau(z). \end{aligned} \quad (6.13)$$

In view of (4.2),

$$\begin{aligned} |\Phi_N(x, z)| &\lesssim \frac{N^q}{\max(1, (N\rho(x, z))^S)} \lesssim \frac{N^q}{\max(1, (N\delta)^S)}, \\ z \in \Delta(x, \delta). \end{aligned} \quad (6.14)$$

The estimates (6.9), (6.14) and (6.13) lead to the fact that Ψ_n is (q, S) -localized.

Next, using (4.2) with $x = y$, we obtain

$$\Psi_N(x, x) = \int_{\mathbb{X}} |\Phi_N(x, z)|^2 d\tau(z) \lesssim N^q \int_{\mathbb{X}} |\Phi_N(x, z)| |f_0(z)| d\mu^*(z).$$

Using (6.9), we deduce that the last integral is $\lesssim \|f_0\|_\infty$. Therefore, $\Psi_N(x, x) \lesssim N^q \|f_0\|_\infty$. \blacksquare

Lemma 6.4. Let $\{\Phi_N\}$ be an admissible family of kernels, and Ψ_N is defined by (4.15). If $d\tau = f_0 d\mu^*$ for some $f_0 \in C(\mathbb{X})$, and $f_0(\mathbf{x}) \geq \mathfrak{m}(f_0) > 0$ for $\mathbf{x} \in \mathbb{X}$ then $\{\Psi_N\}$ is an admissible family of kernels.

Proof. We have already proved in Lemma 6.3 that $\{\Psi_N\}$ is (q, S) -localized. In order to prove that (4.6) holds with Ψ_N replacing Φ_N , we need only to prove that

$$\Psi_N(x, x) \gtrsim N^q, \quad x \in \mathbb{X}. \quad (6.15)$$

Since Φ_N is admissible, we deduce from (4.7) that there exists $c > 0$ such that if $\rho(x, z) \leq c/N$ then

$$|\Phi_N(x, x) - \Phi_N(x, z)| \lesssim N^{q+1} \rho(x, z) \leq (1/2) |\Phi_N(x, x)|;$$

i.e., $|\Phi_N(x, z)| \geq (1/2) |\Phi_N(x, x)| \gtrsim N^q$.

Using (4.6), we obtain

$$\begin{aligned} \Psi_N(x, x) &= \int_{\mathbb{X}} |\Phi_N(x, z)|^2 f_0(z) d\mu^*(z) \\ &\geq \mathfrak{m}(f_0) \int_{\mathbb{B}(x, c/N)} |\Phi_N(x, z)|^2 d\mu^*(z) \\ &\gtrsim N^{2q} \mathfrak{m}(f_0) \mu^*(\mathbb{B}(x, c/N)). \end{aligned}$$

In view of (4.11), this leads to (6.15).

Next, if $x, x', y \in \mathbb{X}$, then using (4.7) and (6.9), we obtain

$$\begin{aligned} |\Psi_N(x, y) - \Psi_N(x', y)| &\leq \int_{\mathbb{X}} |\Phi_N(x, z) - \Phi_N(x', z)| |\Phi_N(z, y)| \\ &\quad \times |f_0(z)| d\mu^*(z) \lesssim \|f_0\|_{\infty} N^{q+1} \rho(x, x'). \end{aligned}$$

This proves (4.7) with Ψ_N in place of Φ_N . ■

Proof of Theorem 4.2. Since Ψ_N is an admissible kernel, we may apply Theorem 6.1 with Ψ_N replacing Φ_N and $\sigma_N(f_0 F)(y_j)$ in place of f_j (cf. (4.16)) to obtain $\{a_k^*\}$ such that

$$\begin{aligned} \sum_{k=1}^M a_k^* \Psi_N(y_k, y_j) &= \sigma_N(f_0 F)(y_j) = \int_{\mathbb{X}} \Phi_N(y_j, y) F(y) f_0(y) d\mu^*(y), \\ j &= 1, \dots, M, \end{aligned} \quad (6.16)$$

and moreover, (cf. (6.2) and (6.3))

$$\max_{1 \leq k \leq M} |a_k^*| \lesssim N^{-q} \max_{1 \leq j \leq M} |\sigma_N(f_0 F)(y_j)|, \quad (6.17)$$

and

$$\max_{1 \leq k \leq M} |a_k^* \Psi_N(y_k, y_k) - \sigma_N(f_0 F)(y_k)| \lesssim (N\eta)^{q-S} \max_{1 \leq j \leq M} |\sigma_N(f_0 F)(y_j)|. \quad (6.18)$$

We note that the minimizer $P_T(\tau; \mathcal{V}(C); F)$ of the theoretical least square loss is given by

$$P_T(\tau; \mathcal{V}(C); F)(x) = \sum_{k=1}^M a_k^* \Phi_N(x, y_k). \quad (6.19)$$

Further, in view of (6.9),

$$\|\sigma_N(f_0 F)\|_{\infty} \lesssim \|f_0 F\|_{\infty}. \quad (6.20)$$

Let $x \in \mathbb{X}$ and $\delta(x) > \eta/3$. Then using the measure ν as in Lemma 6.1, we obtain from (6.17), (6.6), and (6.20) that

$$\begin{aligned} |P_T(\tau; \mathcal{V}(C); F)(x)| &\leq \sum_{k: y_k \in \Delta(x, \eta/3)} |a_k^*| |\Phi_N(x, y_k)| \\ &\lesssim N^{-q} \|\sigma_N(f_0 F)\|_{\infty} \int_{y \in \Delta(x, \eta/3)} |\Phi_N(x, y)| d\nu(y) \\ &\lesssim N^{-q} \|\sigma_N(f_0 F)\|_{\infty} \eta^{-q} (N\eta)^{q-S} \lesssim \|f_0 F\|_{\infty} (N\eta)^{-S}. \end{aligned} \quad (6.21)$$

This proves (4.18).

Next, let $\delta(x) \leq \eta/3$. Then there exists a unique ℓ such that $\delta(x) = \rho(x, y_{\ell})$. So, using the fact that $\Psi_N(y_{\ell}, y_{\ell}) \sim \Phi_N(y_{\ell}, y_{\ell}) \sim N^q$, (6.18), and (6.20), we obtain

$$\begin{aligned} &\left| a_{\ell}^* \Phi_N(y_{\ell}, y_{\ell}) - \frac{\Phi_N(y_{\ell}, y_{\ell})}{\Psi_N(y_{\ell}, y_{\ell})} \sigma_N(f_0 F)(y_{\ell}) \right| \\ &\lesssim (N\eta)^{q-S} \|\sigma_N(f_0 F)\|_{\infty} \lesssim (N\eta)^{q-S} \|f_0 F\|_{\infty}, \end{aligned}$$

and hence,

$$\begin{aligned} &\left| a_{\ell}^* \Phi_N(y_{\ell}, y_{\ell}) - \frac{\Phi_N(y_{\ell}, y_{\ell})}{\Psi_N(y_{\ell}, y_{\ell})} F(y_{\ell}) f_0(y_{\ell}) \right| \\ &\lesssim \|f_0 F - \sigma_N(f_0 F)\|_{\infty} + (N\eta)^{q-S} \|f_0 F\|_{\infty}. \end{aligned} \quad (6.22)$$

Since

$$|F(y_{\ell}) f_0(y_{\ell}) - F(x) f_0(x)| \leq \|f_0 F\|_{\text{Lip}} \delta(x),$$

we deduce from (6.22), (4.7), (6.17), and (6.20) that

$$\begin{aligned} &\left| a_{\ell}^* \Phi_N(x, y_{\ell}) - \frac{\Phi_N(y_{\ell}, y_{\ell})}{\Psi_N(y_{\ell}, y_{\ell})} F(x) f_0(x) \right| \\ &\lesssim (N + \|f_0 F\|_{\text{Lip}}) \delta(x) + \|f_0 F - \sigma_N(f_0 F)\|_{\infty} + (N\eta)^{q-S} \|f_0 F\|_{\infty}. \end{aligned} \quad (6.23)$$

Arguing as in (6.21), we get

$$\sum_{k \neq \ell} |a_k^* \Phi_N(x, y_k)| \lesssim \|f_0 F\|_{\infty} (N\eta)^{-S}.$$

Hence, (6.23) leads to (4.19). ■

7. Conclusions

In Mhaskar (2020a), HNM had developed a very simple method to approximate functions on unknown manifolds without making any effort to learn the manifold itself (e.g., by estimating an atlas or eigen-decomposition of the Laplace–Beltrami operator). This method involves a simple matrix vector multiplication using a specially constructed localized kernel. However, the approach requires that the training data be dense on the manifold. In this paper, we examine the accuracy of the approximation if the training data is sparse instead, and we use either empirical risk minimization or the theoretical square loss minimization. We study this question in a very general setting of a locally compact metric measure space, thereby initializing a theme for further research where the unknown manifold is known to be a submanifold of a known manifold rather than just a high dimensional Euclidean space. In practice, the problem arises, for example, in analysis of time series, for which the domain knowledge indicates that the ambient space is a Grassmann manifold. We present a detailed experimental study where we use different variations of the localized kernel (3.4) to the classification of hand gestures using micro-doppler radar data – a problem of interest in its own right. Our results show that the SVMs trained with our proposed localized kernel and PCA components of zero-padded vectorized spectrograms outperform existing methods for micro-Doppler gesture recognition, including some CNNs by a 3%–6% margin of accuracy, but with a much shorter training time. To demonstrate the fact that our theory is general purpose, we use similar techniques for the classification of spoken digits from a video data set, and demonstrate how an embedding of the data set onto an unknown submanifold of a Grassmann manifold yields superior results.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Professor Dr. Naoki Saito at University of California, Davis for providing us with the data set used in [Lieu and Saito \(2011\)](#). We thank the referees for their valuable suggestions for the improvement of this paper.

References

Abdulatif, S., Wei, Q., Aziz, F., Kleiner, B., & Schneider, U. (2018). Micro-doppler based human-robot classification using ensemble and deep learning approaches. In *2018 IEEE radar conference (RadarConf18)* (pp. 1043–1048).

Ahmed, S., Kallu, K. D., Ahmed, S., & Cho, S. H. (2021). Hand gestures recognition using radar sensors for human-computer-interaction: A review. *Remote Sensing*, 13(3), 527.

Belkin, M., Hsu, D. J., & Mitra, P. (2018). Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. In *Advances in neural information processing systems* (pp. 2300–2311).

Björklund, S., Petersson, H., & Hendeby, G. (2015). Features for micro-doppler based activity classification. *IET Radar, Sonar & Navigation*, 9(9), 1181–1187.

Brooks, D. A., Schwander, O., Barbaresco, F., Schneider, J.-Y., & Cord, M. (2018). Temporal deep learning for drone micro-doppler classification. In *2018 19th international radar symposium (IRS)* (pp. 1–10).

Chui, C. K., & Donoho, D. L. (2006). Special issue: Diffusion maps and wavelets. *Applied and Computational Harmonic Analysis*, 21(1).

Chui, C. K., & Mhaskar, H. N. (2018). Deep nets for local manifold learning. *Frontiers in Applied Mathematics and Statistics*, 4, 12.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). Ieee.

Erol, B., Seyfoglu, M. S., Gürbüz, S. Z., & Amin, M. (2018). Data-driven cepstral and neural learning of features for robust micro-Doppler classification. In K. I. Ranney, & A. Doerry (Eds.), *Radar sensor technology XXII*, Vol. 10633 (pp. 163–170). International Society for Optics and Photonics, SPIE.

Filbir, F., & Mhaskar, H. N. (2010). A quadrature formula for diffusion polynomials corresponding to a generalized heat kernel. *Journal of Fourier Analysis and Applications*, 16(5), 629–657.

Filbir, F., & Mhaskar, H. N. (2011). Marcinkiewicz-Zygmund measures on manifolds. *Journal of Complexity*, 27(6), 568–596.

Gautschi, W. (2004). *Orthogonal polynomials: Computation and approximation*. Oxford University Press on Demand.

Gürbüz, S. Z., Erol, B., Çağlıyan, B., & Tekeli, B. (2015). Operational assessment and adaptive selection of micro-doppler features. *IET Radar, Sonar & Navigation*, 9(9), 1196–1204.

Gürbüz, S. Z., & Mason, E. (2020). Prologue: Perspectives on deep learning of rf data. In S. Z. Gurbuz (Ed.), *Deep neural network design for radar applications*. IET.

Huizing, A., Heiligers, M., Dekker, B., de Wit, J., Cifola, L., & Harmanny, R. (2019). Deep learning for classification of mini-uavs using micro-doppler spectrograms in cognitive radar. *IEEE Aerospace and Electronic Systems Magazine*, 34(11), 46–56.

Jo, J., & Bengio, Y. (2017). Measuring the tendency of cnns to learn surface statistical regularities. arXiv preprint [arXiv:1711.11561](#).

Karabacak, C., Gurbuz, S. Z., Gurbuz, A. C., Guldogan, M. B., Hendeby, G., & Gustafsson, F. (2015). Knowledge exploitation for human micro-doppler classification. *IEEE Geoscience and Remote Sensing Letters*, 12(10), 2125–2129.

Kim, Y., & Toomajian, B. (2016). Hand gesture recognition using micro-doppler signatures with convolutional neural network. *IEEE Access*, 4, 7125–7130.

Kulhandjian, H., Sharma, P., Kulhandjian, M., & D'Amours, C. (2019). Sign language gesture recognition using doppler radar and deep learning. In *2019 IEEE globecom workshops (GC Wkshps)* (pp. 1–6).

Li, F., Yang, C., Xia, Y., Ma, X., Zhang, T., & Zhou, Z. (2017). An adaptive s-method to analyze micro-doppler signals for human activity classification. *Sensors*, 17(12).

Li, G., Zhang, R., Ritchie, M., & Griffiths, H. (2018). Sparsity-driven micro-doppler feature extraction for dynamic hand gesture recognition. *IEEE Transactions on Aerospace and Electronic Systems*, 54(2), 655–665.

Lieu, L., & Saito, N. (2011). Signal ensemble classification using low-dimensional embeddings and earth mover's distance. In J. Cohen, & A. I. Zayed (Eds.), *Applied and numerical harmonic analysis, Wavelets and multiscale analysis: Theory and applications* (pp. 227–256). Boston: Birkhäuser.

Ma, X., Zhao, R., Liu, X., Kuang, H., & Al-qaness, M. A. A. (2019). Classification of human motions using micro-doppler radar in the environments with micro-motion interference. *Sensors*, 19(11).

Manfredi, G., Hinostroza, I. D. S., Menelle, M., Saillant, S., Ovarlez, J.-P., & Thirion-Lefevre, L. (2021). Measurements and analysis of the doppler signature of a human moving within the forest in uhf-band. *Remote Sensing*, 13(3).

Mhaskar, H. N. (2010). Eignets for function approximation on manifolds. *Applied and Computational Harmonic Analysis*, 29(1), 63–87.

Mhaskar, H. N. (2020a). A direct method for function approximation on data defined manifolds. *Neural Networks*, 132, 253–268. <http://dx.doi.org/10.1016/j.neunet.2020.08.018>, ArXiv preprint [arXiv:1908.00156](#).

Mhaskar, H. N. (2020b). Kernel-based analysis of massive data. *Frontiers in Applied Mathematics and Statistics*, 6, 30.

Mhaskar, H. N., & Poggio, T. (2020). An analysis of training and generalization errors in shallow and deep networks. *Neural Networks*, 121, 229–241.

Molchanov, P., Harmanny, R. I., de Wit, J. J., Egiazarian, K., & Astola, J. (2014). Classification of small uavs and birds by micro-doppler signatures. *International Journal of Microwave and Wireless Technologies*, 6(3–4), 435–444.

Park, J., Javier, R. J., Moon, T., & Kim, Y. (2016). Micro-doppler based classification of human aquatic activities via transfer learning of convolutional neural networks. *Sensors*, 16(12).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830.

Poggio, T., Kawaguchi, K., Liao, Q., Miranda, B., Rosasco, L., Boix, X., et al. (2017). Theory of deep learning III: explaining the non-overfitting puzzle. arXiv preprint [arXiv:1801.00173](#).

Poggio, T., Liao, Q., Miranda, B., Banburski, A., Boix, X., & Hidary, J. (2018). Theory IIb: Generalization in deep networks. arXiv preprint [arXiv:1806.11379](#).

Ritchie, M., Capraru, R., & Fioranelli, F. (2020). Dop-net: a micro-doppler radar data challenge. *Electronics Letters*, 56, 568–570(2).

Ritchie, M., & Jones, A. M. (2019). Micro-doppler gesture recognition using doppler, time and range based features. In *2019 IEEE radar conference (RadarConf)* (pp. 1–6). Boston, MA, USA: IEEE.

Schmidt-Hieber, J. (2019). Deep ReLU network approximation of functions on a manifold. arXiv preprint [arXiv:1908.00695](#).

Seyfoglu, M. S., Erol, B., Gürbüz, S. Z., & Amin, M. G. (2019). Dnn transfer learning from diversified micro-doppler for motion classification. *IEEE Transactions on Aerospace and Electronic Systems*, 55(5), 2164–2180.

Skaria, S., Al-Hourani, A., & Evans, R. J. (2020). Deep-learning methods for hand-gesture recognition using ultra-wideband radar. *IEEE Access*, 8, 203580–203590.

Szegő, G. (1975). Orthogonal polynomials. In *Colloquium publications/American mathematical society*, Vol. 23. Providence.

Tahmoush, D. (2015). Review of micro-doppler signatures. *IET Radar, Sonar & Navigation*, 9(9), 1140–1146.

Tran, K. T., Griffin, L. D., Chetty, K., & Vishwakarma, S. (2020). Transfer learning from audio deep learning models for micro-doppler activity recognition. In *2020 IEEE international radar conference (RADAR)* (pp. 584–589).

Turaga, P., Veeraraghavan, A., Srivastava, A., & Chellappa, R. (2011). Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11), 2273–2286.

Usevich, K., Emiya, V., Brie, D., & Chaux, C. (2018). Characterization of finite signals with low-rank stft. In *2018 IEEE statistical signal processing workshop (SSP)* (pp. 393–397).

Ye, K., & Lim, L.-H. (2016). Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3), 1176–1197.

Yen, J.-C., Chang, F.-J., & Chang, S. (1995). A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3), 370–378.

Zeng, Z., Amin, M. G., & Shan, T. (2020). Automatic arm motion recognition based on radar micro-doppler signature envelopes. *IEEE Sensors Journal*, 20(22), 13523–13532.

Zhao, C., Luo, G., Wang, Y., Chen, C., & Wu, Z. (2021). Uav recognition based on micro-doppler dynamic attribute-guided augmentation algorithm. *Remote Sensing*, 13(6).

Zhou, T., Yang, M., Jiang, K., Wong, H., & Yang, D. (2020). Mmw radar-based technologies in autonomous driving: A review. *Sensors*, 20(24), 7283.

Zhu, J., Chen, H., & Ye, W. (2020). A hybrid cnn-lstm network for the classification of human activities based on micro-doppler radar. *IEEE Access*, 8, 24713–24720.