Channel-Robust Classifiers

Hyrum S. Anderson, *Student Member, IEEE,* Maya R. Gupta, *Senior Member, IEEE,*
Eric Swanson, *Student Member, IEEE,* and Kevin Jamieson, *Student Member, IEEE*

***Abstract*—A key assumption underlying traditional supervised learning algorithms is that labeled examples used to train a classifier are drawn i.i.d. from the same distribution as test samples. This assumption is violated when classifying a test sample whose statistics differ from the training samples because the test signal is the output of a noisy linear time-invariant system, e.g., from channel propagation or filtering. We assume that the channel impulse response is unknown, but can be modeled as a random channel with finite first and second-order statistics that can be estimated from sample impulse responses. We present two kernels, the expected and projected RBF kernels, that account for the stochastic channel. Compared to the strategy of virtual examples, an SVM trained with the proposed kernels requires dramatically less training time, and may perform better in practice. We also extend the joint quadratic discriminant analysis (joint QDA) classifier, which also accounts for a stochastic channel, to a local version that reduces model bias. Results show the proposed methods achieve state-of-the-art performance and significantly faster training times.**

I. Introduction

THERE are many applications in which the operating environment of a classifier differs from the environment in which training samples are acquired. For example, the acoustic environment in which automatic speech recognition systems operate may differ drastically from training conditions, and methods must be employed to ensure robustness to the test environment [1], [2]. In underwater acoustics, training features may be acquired in deep ocean water where multipath is negligible, but the classifier may be deployed on test signals that are corrupted by propagating through a shallow-water multipath environment [3], [4]. Face recognition methods may be trained on a database of high-resolution training images, but used on test images from blurred or low-quality video stills [5]. These examples violate a fundamental assumption in traditional machine learning, that a test sample $\mathbf{x}$ and its true label y are drawn independently and identically (i.i.d.) from the same joint distribution p_{XY} as the training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. A mismatch between the test distribution and training distributions is known as *dataset shift*.

The above scenarios can be modeled as an unknown linear time-invariant channel and additive noise inducing a different distribution at test time than at training time. Training samples $\{\mathbf{x}_i\}_{i=1}^N$ are extracted from sampled time signals $\{x_i[n]\}_{i=1}^N$, but a test sample $\mathbf{z}$ is extracted from the signal

$$z[n] = h[n] * x[n] + w[n], \tag{1}$$

Authors are with the Dept. of Electrical Engineering at the University of Washington. E-mail: gupta@ee.washington.edu.

where $*$ denotes convolution, $h[n]$ is the unknown impulse response of a corrupting channel, $x[n]$ is the unknown signal of interest, and $w[n]$ is a realization of a zero-mean Gaussian white noise process with known variance. Features $\mathbf{x}$ of $x[n]$ are unknown, however, it is assumed that $\mathbf{x}$ and its true label y are drawn i.i.d. from the same distribution as the training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. In addition, we assume that a finite set of auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$ are available, and that the unknown features $\mathbf{h}$ of $h[n]$ are drawn i.i.d. from the same joint distribution as $\{\mathbf{h}_i\}_{i=1}^M$. Clearly, though, the test feature vector $\mathbf{z}$ is in general not drawn i.i.d. from the same distribution as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Throughout, bold-face $\mathbf{x}$ denotes a vector, regular-face x denotes a scalar; random vectors and scalars are uppercase, $\mathbf{X}$ and X, respectively; see Table I for notation.

This paper proposes and compares algorithmic solutions to this problem. We adapt the support vector machine (SVM) to account for the stochastic noisy channel by constructing channel-robust kernel functions. We propose two kernel definitions that account for dataset shift (the *expected kernel* and the *projected kernel*), and provide closed-form derivations of these kernels for the cases in which the features of interest are either the discrete-time signal itself or the energy in certain frequency subbands. We also investigate an approach to make it possible to train a classifier once for many different environments, in which an SVM discriminant function trained with a standard (not channel-robust) kernel is modified at test time to incorporate a channel-robust kernel by retraining only the bias of the discriminant function. We propose a local extension of the joint QDA classifier in [3], which also adapts for a stochastic channel. Experimental comparisons with real and simulated data demonstrate the effectiveness of the proposed algorithms. The proposed *expected kernel* was first presented in a conference publication [6]; this manuscript provides a richer development and expanded results.

Section II introduces basic notation for SVMs, kernels and features used throughout the paper II. We review related work on invariant classifiers in Section III. The proposed channel-robust kernels are presented in Section IV, including derivations for RBF kernels for discrete-time and subband energy features. The local joint QDA classifier is presented in VIII. We describe simulated and real-data classification experiments in Sec. IX, discuss the results in X, and conclude in Section XI with observations and open questions.

II. Background and Notation

The SVM classifies a feature vector $\mathbf{x}$ based on the sign of the discriminant function

$$f(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i), \tag{2}$$

where the label of the ith training sample is $y_i \in \{-1, 1\}$, $K(\cdot,\cdot)$ is a kernel function, and the weights $\{\alpha_i\}_{i=1}^N$ and bias b are chosen to minimize regularized training error over $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ [7], which is an $O(N^3)$ time computation[1]. The support vectors are those $\mathbf{x}_i$'s for which $0 < \alpha_i \leq C$, where C is the maximum penalty assigned for misclassifying $\mathbf{x}_i$.

A kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ measures the similarity of its two arguments, and any kernel can be formulated as an inner product of an implicit mapping $\phi(\cdot)$ such that $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ [7]. In this paper, we focus on the popular Gaussian radial basis function (RBF) :

$$K_{\text{rbf}}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}\left(\mathbf{x}_i; \mathbf{x}_j, \gamma^{-1} I\right), \tag{3}$$

where $\mathcal{N}(\cdot)$ denotes the Gaussian function, and γ is the bandwidth parameter. The RBF kernel is commonly defined as $K_{\text{rbf}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2}\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$ so that $K_{\text{rbf}}(\mathbf{x}, \mathbf{x}) = 1$ and since the Gaussian normalization factor represents an arbitrary global scaling of similarity measure, but for mathematical convenience, we use the definition in (3).

In some applications (e.g., image classification), it is convenient to train a classifier using the sampled signal (e.g., pixels) as features. Let $\mathbf{x}$, $\mathbf{h}$, $\mathbf{w}$ and $\mathbf{z}$ be vectors whose elements contain the samples of the discrete-time signals $x[n]$, $h[n]$, $w[n]$ and $z[n]$, respectively. Then, (1) can be written concisely as $\mathbf{z} = \mathbf{h} * \mathbf{x} + \mathbf{w}$, where $*$ denotes discrete convolution.

In other applications, features extracted from the discrete-time signals better discriminate the different classes. In particular, *subband energy features* are a useful and frequently utilized feature choice in many signal processing classification applications. Let $x^f[k]$ denote the kth bin of the discrete Fourier transform of $x[n]$, and let $w^f[k]$ be a realization of a zero-mean proper complex Gaussian white noise process with known variance. The subband energy of $u_z[k] = \left|z^f[k]\right|^2$ is given by

$$u_z[k] = u_h[k]u_x[k] + u_w[k] + 2\,\text{Re}\left\{x^f[k]h^f[k]w^{f*}[k]\right\},$$

where $w^{f*}[k]$ is the complex conjugate of $w^f[k]$. Consider a feature vector of subband energies at d frequency bins $k \in \{k_1, k_2, \ldots, k_d\}$. The relationship of the observed vector $\mathbf{u}_z \in \mathbb{R}^d$ and the (unknown) vector $\mathbf{u}_x \in \mathbb{R}^d$ can be written concisely as

$$\mathbf{u}_z = \mathbf{u}_h \cdot \mathbf{u}_x + \mathbf{u}_w + 2\,\text{Re}\left\{\mathbf{x}^f \cdot \mathbf{h}^f \cdot \mathbf{w}^{f*}\right\}, \tag{4}$$

where $\cdot$ denotes the Hadamard (element-wise) product.

III. Related Work

A standard signal-processing approach for classifying a channel-corrupted signal $z[n]$ is to first estimate a clean test signal $\hat{x}[n]$ via blind deconvolution, then apply a standard classifier [8], [9], [10]. However, blind deconvolution is ill-posed, and in practice the prior knowledge on which a deconvolution scheme is based may not match real-world conditions, for example, the sparsity of multipath channel impulse responses [3]. Alternatively, classifiers can be made robust to the effects of the channel by using one of three general strategies that we discuss in the following subsections: using invariant features, creating virtual examples that model the conditions at test time, and designing classifiers that have robustness to test conditions built in.

[1]Theoretically, solving the SVM problem is $O(N^3)$, but empirically, performance with fast SVM solvers often approaches $O(N^2)$.

TABLE I
Notation

$x[n]$	discrete-time signal
$\mathbf{x}$	vector
$x^f[k]$	DFT of $x[n]$
$\mathbf{x}^f$	DFT signal vector
$u_x[k]$	subband energy $\|x^f[k]\|^2$
$\mathbf{u}_x$	subband energy vector
$\mathbf{X}$	random vector
$\bar{\mathbf{X}}$	mean of $\mathbf{X}$
N	# of training samples
M	# of auxiliary channel examples
d	# of feature dimensions
$\mathcal{N}(\mathbf{x}; \mathbf{m}, A)$	Gaussian in $\mathbf{x}$ with parameters $\mathbf{m}, A$
$A * *B$	two-dimensional convolution of A with B
$A \cdot B$	Hadamard (component-wise) product of A and B
$\frac{[A]}{[B]}$	Hadamard division

A. Invariant Features

One technique to building a classifier to be robust to channel propagation effects is to select features that are invariant to the channel [11], [12], [13]. For example, Okopal and Loughlin derived damping-invariant and dispersion-invariant features in [12]. However, the utility of channel-invariant features for discrimination depends heavily on the classification task.

B. Virtual Examples

The idea of augmenting a training set with "virtual examples" (VEs) dates back to at least 1990 [14]. For example, to build a handwritten digit classifier that is robust to various rotations, one can augment the original training set with artificial examples of rotated digits [15]. The transformed VEs are included with the original training examples to form an expanded training set. The choice of transformation applied to generate the VEs is based on prior knowledge about the perturbations that may be expected in the test features. Typically, the VEs are generated from a discrete and deterministic set of transformations, for example, single pixel translations in the four principal directions of the image plane.

Lorens et al. have employed virtual examples to train SVMs to classify targets from their acoustic signatures [4]. High quality recorded signatures are artificially corrupted by simulating their propagation through an acoustic channel to produce virtual examples that better represent the test distribution p_{ZY}. Since the original training examples are not representative of the test distribution, they are discarded.

Like Lorens et al., we implement the VE method by propagating each of the N training signals through M example channels, resulting in a training data set size of $M \times N$. This approach has the disadvantage of $O\left(M^3N^3\right)$ complexity in training an SVM.

A variant of VEs is the method of *virtual support vectors* (VSV), which trains an SVM on an uncorrupted training

set, then generates virtual examples from only the support vectors [16]. While the VSV method has been shown to reduce the overall cost of training an SVM, we found that for the RBF kernel—which is known to select many training points as support vectors—the VSV method did not substantially decrease training time, and often exhibited worse performance in preliminary experiments on our datasets.

C. Prior Robust Classifiers

Classifiers can be designed to be invariant to conditions one would expect at test time. Decoste and Schölkopf employed *jittering kernels* in an SVM to build a classifier robust to slight translations and rotations of handwritten digits and showed previously unmatched error rates on the MNIST benchmark dataset of handwritten digits [15]. A *jittering kernel* $K_{\text{jitter}}(\mathbf{x}, \mathbf{x}_i)$ takes some some kernel function $K(\cdot,\cdot)$, and measures similarity as the maximum similarity $K(t(\mathbf{x}), \tilde{t}(\mathbf{x}_i))$ over $t, \tilde{t} \in \mathcal{T}$ where $\mathcal{T}$ is a finite set of perturbations (jitters) that one might expect at test time. Jittering kernels for SVMs have the advantage over VEs for SVM in that the jittering kernel SVM scales linearly with the number of jitters $|\mathcal{T}|$, whereas the VE SVM is cubic in the number of jitters. However, jittering kernels assume there exists a finite set of complete-invariances $\mathcal{T}$, and do not take into account the relative probability of transformations in the set, and thus are not applicable to the kind of probabilistic transformations imposed by random channels or additive noise. Invariant kernels have been further studied by Haasdonk and Burkhardt [17].

Anderson and Gupta proposed taking channel and noise statistics into account when building a generative classifier from clean training samples [3]. They derived a quadratic discriminant analysis (QDA) classifier termed *joint QDA* for two cases: classifying a discrete-time signal $z[n]$, or subband energies $u_z[k]$ at several frequency bins. Joint QDA learns a Gaussian approximation for each $p(\mathbf{z}|y)$ from clean training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and second-order statistics of a stochastic channel. For several sonar-related binary classification problems, joint QDA showed superior performance over other approaches.

IV. Channel-Robust Kernels and SVMs

Given stochastic models for the linear time-invariant channel and noise in (1), the VE method may be used to train a channel-robust SVM. However, as previously noted, training an SVM using the VE method has $O(M^3N^3)$ complexity. Rather than increase the dataset by a factor of M, we introduce two approaches—the *expected kernel* and the *projected RBF kernel*—that incorporate a stochastic channel model into the kernel definition. For both approaches, we map the ith training sample $\mathbf{x}_i$ to a probability distribution $p_{Z_i|x_i}$ over the domain of noisy channel-corrupted signals. Then we define a kernel that acts on two probability densities in the noisy domain. The two approaches differ in how the samples are mapped to probability densities, and how the kernels are defined on the densities.

In the following sections, we derive closed-form solutions for the proposed kernels for discrete-time features and for subband energy features by employing Gaussian assumptions. We also show how to create analytic kernels for any feature using VEs and a Gaussian assumption. The Gaussian assumption is motivated by the fact that the Gaussian distribution is the maximum entropy distribution over $\mathbb{R}^d$ given only the mean and covariance. For non-negative subband energy features, it may seem more suitable to instead consider the maximum entropy distribution over the positive orthant $\mathbb{R}_+^d$. However, the maximum entropy distribution over $\mathbb{R}_+^d$ is the multivariate truncated normal distribution, which requires cumbersome multi-dimensional lookup tables of the cumulative distribution function [18]. Another option for modeling the distribution of subband energy features is the multivariate Rayleigh model, but that is analytically and computationally even more challenging. Conversely, Gaussian functions are mathematically tractable.

V. Expected Kernels

Consider the random signal resulting from propagating the features $\mathbf{x}_i$ of the ith training signal through a random noisy channel, and let the feature vector computed from that random signal be the random feature vector $\mathbf{Z}_i \sim p_{Z_i|x_i}$. Then the channel can be taken into account by training an SVM with a kernel that acts on the random feature vectors $\{\mathbf{Z}_i\}_{i=1}^N$ as training examples. Specifically, given any kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, we define the expected kernel K_{exp} to be the following functional of two distributions:

$$K_{\text{exp}}(p_{Z_i|x_i}, p_{Z_j|x_j}) \triangleq \mathrm{E}_{\mathbf{Z}_i,\mathbf{Z}_j|\mathbf{x}_i,\mathbf{x}_j}\left[K\left(\mathbf{Z}_i, \mathbf{Z}_j\right)\right], \quad (5)$$
$$= \iint p_{Z_i|x_i}(\mathbf{z}_i) p_{Z_j|x_j}(\mathbf{z}_j) K(\mathbf{z}_i, \mathbf{z}_j) d\mathbf{z}_i d\mathbf{z}_j.$$

The expected training kernel can be interpreted as averaging the similarity of all possible channel corruptions of $\mathbf{x}_i$ and $\mathbf{x}_j$ weighted by their probability density. To compute the kernel between a training sample and a test sample, let the probability density of the test sample be a Dirac delta distribution with all of its support on the feature vector $\mathbf{z}$ computed from the test signal $z[n]$, that is, $p_{Z|z}(\mathbf{z}') = \delta(\mathbf{z}' - \mathbf{z})$. The expected kernel given in (5) is a legitimate kernel because it is an inner product between its two inputs, where the inner product is weighted by the positive definite function $K(\cdot,\cdot)$, analogous to a discrete inner product of the form $< a, b >_K = a^T K b$ for some positive definite matrix K.

Since the distribution $p_{Z_i|x_i}$ is a function of the training point $\mathbf{x}_i$, for notational simplicity, we write $K_{\text{exp}}(\mathbf{x}_i, \mathbf{x}_j)$ for (5), and $K_{\text{exp}}(\mathbf{z}, \mathbf{x}_i)$ for the corresponding kernel between $\delta(\mathbf{z}' - \mathbf{z})$ and $p_{Z_i|x_i}$.

A. Expected Kernel SVM Compared to VE SVM

Let $\{(\mathbf{z}_{ij}, y_{ij})\}_{i,j=1}^{N,M}$ be VE training pairs generated from the powerset of $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ and $\{\mathbf{h}_j\}_{j=1}^M$. Let $L(f(\mathbf{x}), y) = (1 - yf(\mathbf{x}))_+$ be the hinge loss, and λ be a regularization parameter. In the noiseless case (so that $\mathrm{E}_{\mathbf{H}}[\cdot] = \mathrm{E}_{\mathbf{Z}|\mathbf{x}}[\cdot]$), and ignoring the bias b, take the limit of the VE SVM objective function as the number of auxiliary channel samples M tends

TABLE II
TRAINING AND TEST RBF KERNELS FOR SUBBAND ENERGY FEATURES

	Training Kernel $K(\mathbf{u}_{x_i}, \mathbf{u}_{x_j})$	**Test Kernel** $K(\mathbf{u}_z, \mathbf{u}_{x_i})$
Expected RBF	$\mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$
Expected RBF (clean)	$\mathcal{N}\left(\mathbf{u}_{x_i}; \mathbf{u}_{x_j}, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$
Projected RBF	$\mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, R_{u_{z_i}} + R_{u_{z_j}}\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right)$
Projected RBF (clean)	$\mathcal{N}\left(\mathbf{u}_{x_i}; \mathbf{u}_{x_j}, \gamma^{-1} I\right)$	$\mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right)$

where,

$\bar{\mathbf{U}}_{z_i} = \mathbf{u}_{x_i} \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1}$

$\Sigma_{u_{z_i}} = \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right)$

$R_{u_{z_i}} = \dfrac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_{z_i}}$

$\tilde{R}_{u_z} = \dfrac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_h} \cdot \left(\dfrac{[\mathbf{u}_z - \sigma_w^2 \mathbf{1}]}{[\bar{\mathbf{U}}_h]}\right) \left(\dfrac{[\mathbf{u}_z - \sigma_w^2 \mathbf{1}]}{[\bar{\mathbf{U}}_h]}\right)^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\mathbf{u}_z - \sigma_w^2 \mathbf{1}\right)$

$$\lim_{M\to\infty} \arg\min_{\{\alpha_{ij}\}} \frac{1}{MN} \sum_{i=1}^{N} \sum_{m=1}^{M} L\left(\sum_{j=1}^{N} \sum_{m'=1}^{M} \alpha_{jm'} y_{jm'} K(\mathbf{z}_{im}, \mathbf{z}_{jm'}), y_i\right) + \lambda \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{N} \sum_{m'=1}^{M} \alpha_{im} y_{im} \alpha_{jm'} y_{jm'} K(\mathbf{z}_{im}, \mathbf{z}_{jm'}), \tag{6}$$

$$\xrightarrow{p} \lim_{M\to\infty} \arg\min_{\{\alpha_{ij}\}} \frac{1}{N} \sum_{i=1}^{N} \mathrm{E}_{\mathbf{Z}_i|\mathbf{x}_i}\left[L\left(\sum_{j=1}^{N} \sum_{m'=1}^{M} \alpha_{jm'} y_{jm'} K(\mathbf{Z}_i, \mathbf{z}_{jm'}), y_i\right)\right] + \lambda \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{N} \sum_{m'=1}^{M} \alpha_{im} y_{im} \alpha_{jm'} y_{jm'} K(\mathbf{z}_{im}, \mathbf{z}_{jm'}). \tag{7}$$

$$\arg\min_{\{\alpha_i\}} \frac{1}{N} \sum_{i=1}^{N} L\left(\sum_{j=1}^{N} \alpha_j y_j \, \mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j|\mathbf{x}_i, \mathbf{x}_j}\left[K(\mathbf{Z}_i, \mathbf{Z}_j)\right], y_i\right) + \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j \, \mathrm{E}_{\mathbf{Z}_i, \mathbf{Z}_j|\mathbf{x}_i, \mathbf{x}_j}\left[K(\mathbf{Z}_i, \mathbf{Z}_j)\right]. \tag{8}$$

towards infinity, as in (6), which converges in probability by the law of large numbers to (7). However, the expected kernel SVM solves the objective function in (8). A comparison of (7) and (8) shows that the VE SVM is not asymptotically equivalent to an SVM using the expected kernel: the VE SVM asymptotically minimizes expected loss, while the expected kernel SVM minimizes the loss with respect to the expected similarities. The regularization terms also differ.

B. Expected RBF Kernel for Discrete-time Signals

Model the impulse response of a stochastic channel as the random vector $\mathbf{H}$ with mean $\bar{\mathbf{H}}$ and covariance Σ_h, and model random vector $\mathbf{W}$ as zero mean with covariance $\sigma_w^2 I$. Then, a deterministic vector $\mathbf{x}$ propagated through the stochastic channel results in a random observation $\mathbf{Z} = \mathbf{H} * \mathbf{x} + \mathbf{W}$. Model $\mathbf{Z} \sim p_{Z|x}(\mathbf{z}|\mathbf{x})$ as Gaussian distributed with mean $\bar{\mathbf{Z}}$ and covariance Σ_z:

$$\bar{\mathbf{Z}} = \bar{\mathbf{H}} * \mathbf{x}, \tag{9}$$

$$\Sigma_z = \Sigma_h ** \left(\mathbf{x}\mathbf{x}^T\right) + \sigma_w^2 I, \tag{10}$$

where $A ** B$ denotes two dimensional convolution of matrices A and B.

To derive the expected RBF training kernel, map $\mathbf{x}_i$ and $\mathbf{x}_j$ to $\mathbf{Z}_i$ and $\mathbf{Z}_j$, which are modeled as independent Gaussians with means and covariances as prescribed in (9) and (10). Then, evaluate the integral in (5) for the RBF kernel in (3) using the product-of-Gaussians rule given in (25) twice to produce

$$K_{\exp}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}\left(\bar{\mathbf{Z}}_i; \bar{\mathbf{Z}}_j, \Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I\right).$$

Similarly, the expected RBF test kernel is also derived using the product-of-Gaussians rule in (25):

$$\begin{aligned} K_{\exp}(\mathbf{z}, \mathbf{x}_i) &= \int p(\mathbf{z}_i|\mathbf{x}_i) K(\mathbf{z}, \mathbf{z}_i) \, d\mathbf{z}_i \\ &= \mathcal{N}\left(\mathbf{z}; \bar{\mathbf{Z}}_i, \Sigma_{z_i} + \gamma^{-1} I\right). \end{aligned}$$

C. Expected RBF Kernel with Subband Energy Features

Model the subband energy feature vector $\mathbf{u}_x$ after propagation through a stochastic channel as a Gaussian random vector $\mathbf{U}_z$, where, from (4)

$$\mathbf{U}_z = \mathbf{U}_h \cdot \mathbf{u}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{x}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f\,*}\right\}.$$

Model $\mathbf{W}^f$ as a proper Gaussian complex random vector with $p(\mathbf{w}^f) = \mathcal{N}\left(0, \sigma_w^2 I\right)$, and random subband energy vector $\mathbf{U}_h$ as having mean $\bar{\mathbf{U}}_h$ and covariance Σ_{u_h} (no additional assumptions are needed about the distribution of $\mathbf{H}^f$).

The expected RBF kernel for subband energy features (with test kernel as a special case) is given by:

$$K_{\exp}(\mathbf{u}_{x_i}, \mathbf{u}_{x_j}) = \mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} I\right),$$
$$K_{\exp}(\mathbf{u}_z, \mathbf{u}_{x_i}) = \mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, \Sigma_{u_{z_i}} + \gamma^{-1} I\right)$$

where

$$\bar{\mathbf{U}}_{z_i} = \mathbf{u}_{x_i} \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1},$$
$$\Sigma_{u_{z_i}} = \Sigma_{u_h} \cdot \mathbf{u}_{x_i} \mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right).$$

The covariance $\Sigma_{u_{z_i}}$ is derived from (48) in the appendix by substituting $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$ and $\Sigma_{u_x} = 0$, since we condition on $\mathbf{U}_x = \mathbf{u}_{x_i}$.

D. Expected Kernel for Arbitrary Features

In the previous two subsections, we analytically derived the Gaussian distribution $p_{Z_i|x_i}(\mathbf{z}_i)$, but such analytic derivations do not exist for all possible feature definitions that may be useful in classification problems. One solution is to approximate the expected kernel using Monte Carlo sampling. Alternatively, one can use the VE methodology and a Gaussian assumption to easily compute an expected RBF kernel for any feature: For each sample $\mathbf{x}_i$, a Gaussian distribution can be fit to its M VE's $\{\mathbf{z}_{ij}\}_{j=1}^M$ to form a Gaussian approximation for $p_{Z_i|x_i}(\mathbf{z}_i)$. For the RBF kernel, the product of Gaussians rule given in (25) can then be used to compute a closed-form solution to (5).

This approach still requires computing the features for each of the MN VE's, however, the SVM is trained on N samples rather than the MN samples used for the VE SVM, drastically reducing the SVM training time. Unlike the traditional VE SVM, in computing the expected kernel using VE's, we need not assume that all VE's $\{\mathbf{z}_{ij}\}_{i=1}^{MN}$ are i.i.d. samples of a single distribution—a poor assumption. Rather, only the original training samples $\{\mathbf{x}_i\}_{i=1}^N$ are assumed to be i.i.d. samples, and the the VE's $\{\mathbf{z}_{ij}\}_{i=1}^M$ corresponding to each $\mathbf{x}_i$ are assumed to be i.i.d. samples of the Gaussian density $p_{Z_i|x_i}(\mathbf{z}_i)$.

E. Unscaled Expected RBF Kernel

Commonly, the standard RBF kernel is implemented without the Gaussian normalization factor, as $K_{\text{rbf}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2}\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$ so that $K_{\text{rbf}}(\mathbf{x}, \mathbf{x}) = 1$. The inclusion of the Gaussian normalization factor $\left(\frac{\gamma}{2\pi}\right)^{d/2}$ is arbitrary, since it represents a global scaling of the similarity measure. The expected RBF kernels, however, have a bandwidth and scaling that are data-dependent, so that it is necessary to include the scaling factor.

The *unscaled expected RBF kernel* K_{uexp} is defined to be the expected RBF kernel without Gaussian normalization, that is, as only the exponential part of the Gaussian. This has the appeal that $K_{\text{uexp}}(\mathbf{x}, \mathbf{x}) = 1$, and does not require calculating the matrix determinant. Moreover, we have found experimentally that the entries of the kernel matrix are more sensitive to small errors in the scale factor than in small errors in the exponent, which has practical implications. For example, we found that the performance of the expected RBF kernel (with Gaussian normalization factor) degrades quickly when channel statistics are poorly estimated; in contrast, the unscaled expected RBF kernel was more robust to channel estimation errors. For precisely these reasons, the experiments in Section IX use the unscaled expected RBF kernel, and unless specifically stated when referring to the "expected RBF kernel" we mean specifically the unscaled version.

The unscaled expected RBF kernel $K_{\text{uexp}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{S(\mathbf{x}_i, \mathbf{x}_j)} K_{\exp}(\mathbf{x}_i, \mathbf{x}_j)$, where $S(\mathbf{x}_i, \mathbf{x}_j)$ is the Gaussian scale factor, is in fact positive definite since $\frac{1}{S(\cdot,\cdot)}$ and $K_{\exp}(\cdot,\cdot)$ are both positive definite. To verify that $\frac{1}{S(\cdot,\cdot)}$ is positive definite, note that the Gaussian scale factor $S(\cdot,\cdot)$ is positive definite, since it can be written as the inner product in (5) with $p_{Z_i|x_i}(\mathbf{z}_i) \triangleq \mathcal{N}(\mathbf{z}_i; 0, \Sigma_{z_i})$ and $p_{Z_j|x_j}(\mathbf{z}_j) \triangleq \mathcal{N}(\mathbf{z}_j; 0, \Sigma_{z_j})$, resulting in

$$S(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_{z_i} + \Sigma_{z_j} + \gamma^{-1} I|^{\frac{1}{2}}},$$

where Σ_{z_i} depends on $\mathbf{x}_i$ and Σ_{z_j} depends on $\mathbf{x}_j$. For a set of any N samples, let S be the $N \times N$ positive definite matrix produced by evaluating the kernel $S(\cdot,\cdot)$ for all pairs of the N samples. Since S is positive definite, the Hadamard inverse $S^{\circ -1} = \left[\frac{1}{S_{ij}}\right]$ is also positive definite [19, p. 397]. The positive definite matrix $S^{\circ -1}$ precisely corresponds to the kernel matrix formed by $\frac{1}{S(\mathbf{x}_i, \mathbf{x}_j)}$. Then, we conclude that since the Hadamard product of two positive definite matrices is also positive definite [19], the unscaled expected RBF kernel matrix $K_{\text{uexp}} = S^{\circ -1} \cdot K_{\exp}$ is positive definite.

F. Modeling Channel Dependency

The definition (5) assumes that the random feature vectors $\mathbf{Z}_i$ and $\mathbf{Z}_j$ are independent, which implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ are corrupted by different random channels and noise. An alternative model is to treat them as being corrupted by the same random channel and noise, which produces a joint distribution $p_{Z_i, Z_j|x_i, x_j}(\mathbf{z}_i, \mathbf{z}_j)$, and then the expectation in (5) becomes

$$\iint p_{Z_i, Z_j|x_i, x_j}(\mathbf{z}_i, \mathbf{z}_j) K(\mathbf{z}_i, \mathbf{z}_j)\, d\mathbf{z}_i\, d\mathbf{z}_j. \tag{11}$$

Unfortunately, it is not clear under what conditions (11) will be a legitimate kernel. We compared the resulting classifier with the expected kernel classifier for the experiments detailed in Section IX and found no statistically significant differences between the two in any of the experiments.

VI. Projected RBF Kernels

We propose another channel-robust kernel that is motivated by a recent interpretation of the RBF kernel. Jebara et al. introduced the *probability product kernel*, which essentially replaces training samples with random variables, $\mathbf{x}_i \mapsto \mathbf{X}' \sim p(\mathbf{x}'|\mathbf{x}_i)$, and defines a positive definite kernel as the inner product of these distributions [20]:

$$K_{\text{prob}}(\mathbf{x}_i, \mathbf{x}_j) \stackrel{\triangle}{=} \int p(\mathbf{x}'|\mathbf{x}_i)p(\mathbf{x}'|\mathbf{x}_j)\, d\mathbf{x}'. \tag{12}$$

Jebara et al. noted that the standard RBF kernel with bandwidth parameter γ in (3) can be derived as a special case of (12) by letting $p(\mathbf{x}'|\mathbf{x}_i) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}_i, \frac{\gamma^{-1}}{2} I\right)$ and $p(\mathbf{x}'|\mathbf{x}_j) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}_j, \frac{\gamma^{-1}}{2} I\right)$, and applying the product of Gaussians identity in (25). In order for the RBF kernel to have have same bandwidth parameter γ at test time, the test feature vector $\mathbf{x}$ must also be replaced with a random variable with density $p(\mathbf{x}'|\mathbf{x}) = \mathcal{N}\left(\mathbf{x}'; \mathbf{x}, \frac{\gamma^{-1}}{2} I\right)$.

To extend (12) to account for a stochastic channel, we also consider $\mathbf{x}_i$ to be mapped to a Gaussian random feature vector $\mathbf{X}'$, and then propagate $\mathbf{X}'$ through the stochastic channel, resulting in the random vector $\mathbf{Z}'$. The resulting distribution $p(\mathbf{z}'|\mathbf{x}_i)$ of $\mathbf{Z}'$ is not necessarily Gaussian; however, for mathematical tractability we project $p(\mathbf{z}'|\mathbf{x}_i)$ to the nearest Gaussian using the following lemma whose proof is given in the appendix.

Lemma. Let random vector $\mathbf{Z} \in \mathbb{R}^d$ be drawn from a distribution that has a probability density function p_Z, finite mean $\bar{\mathbf{Z}} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{S}^d_{++}$. Then, the Gaussian distribution that uniquely minimizes KL-divergence with respect to p_Z is given by $\mathcal{N}(\mathbf{z}; \bar{\mathbf{Z}}, \Sigma)$.

Let $\mathcal{N}(\mathbf{z}'|\mathbf{x}_i)$ be the projection of $p(\mathbf{z}'|\mathbf{x}_i)$ to the nearest Gaussian distribution using the lemma. Then, analogous to (12), we define the *projected RBF kernel* as

$$K_{\text{proj}}(\cdot, \mathbf{x}_j) \stackrel{\triangle}{=} \int \mathcal{N}(\mathbf{z}'|\cdot)\mathcal{N}(\mathbf{z}'|\mathbf{x}_j)\, d\mathbf{z}'. \tag{13}$$

When evaluating the kernel between a test sample and training sample $K_{\text{proj}}(\mathbf{z}, \mathbf{x}_j)$, we define the distribution $\mathcal{N}(\mathbf{z}'|\mathbf{z})$ needed for (13) to be the projection of a random variable $\mathbf{Z}'$ to the nearest Gaussian, where $\mathbf{Z}'$ results from propagating $\mathbf{X}' \sim \mathcal{N}(\bar{\mathbf{X}}', \frac{\gamma^{-1}}{2} I)$ through the stochastic channel, and $\bar{\mathbf{X}}'$ is chosen such that the mean $\mathrm{E}[\mathbf{Z}'] = \mathbf{z}$ is the observed test sample (see Sections VI-A and VI-B for examples). The kernel K_{proj} is a legitimate kernel because it is always an inner product of two distributions in $\mathbf{z}'$.

We next present the analytic forms of the projected RBF test and training kernels for the same two cases as the expected RBF kernel: discrete-time signal features and subband energy features.

A. Projected RBF Kernel for Discrete-Time Signals

Model the random vector $\mathbf{X}' \sim \mathcal{N}(\mathbf{x}, \frac{\gamma^{-1}}{2} I)$, then $\mathbf{Z}' = \mathbf{H} * \mathbf{X}' + \mathbf{W}$ has mean and covariance given by

$$\bar{\mathbf{Z}}' = \bar{\mathbf{H}} * \mathbf{x}, \quad \text{and}$$
$$\Sigma_{z'} = \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}}\bar{\mathbf{H}}^T\right) + \Sigma_h ** \mathbf{x}\mathbf{x}^T + \sigma_w^2 I. \tag{14}$$

Then by the lemma, the projection $p(\mathbf{z}'|\mathbf{x}_i)$ to the nearest Gaussian distribution $\mathcal{N}(\mathbf{z}'|\mathbf{x}_i)$ yields

$$\mathcal{N}(\mathbf{z}'; \bar{\mathbf{H}} * \mathbf{x}_i, \frac{\gamma^{-1}}{2} I ** \left(\Sigma_h + \bar{\mathbf{H}}\bar{\mathbf{H}}^T\right) + \Sigma_h ** \mathbf{x}_i\mathbf{x}_i^T + \sigma_w^2 I).$$

Substituting into (13), and simplifying with the product of Gaussians rule given in (25) yields

$$K_{\text{proj}}(\mathbf{x}_i, \mathbf{x}_j) = \mathcal{N}(\bar{\mathbf{H}} * \mathbf{x}_i; \bar{\mathbf{H}} * \mathbf{x}_j, \gamma^{-1} I ** \left(\Sigma_h + \bar{H}\bar{H}^T\right) + \Sigma_h ** \left(\mathbf{x}_i\mathbf{x}_i^T + \mathbf{x}_j\mathbf{x}_j^T\right) + 2\sigma_w^2 I).$$

To construct $\mathcal{N}(\mathbf{z}'|\mathbf{z})$, we assume that a test sample is the mean of the distribution, $\mathbf{z} = \mathrm{E}[\mathbf{Z}']$, where the random variable $\mathbf{Z}'$ is the projection through the stochastic channel of a random variable $\mathbf{X}'$ with covariance $\frac{\gamma^{-1}}{2} I$. Therefore, $\mathcal{N}(\mathbf{z}'|\mathbf{z})$ has covariance given by (14) with $\mathbf{x}$ substituted with Fourier deconvolution $\bar{H}^{-1} * \mathbf{z}$.

B. Projected RBF Kernel for Subband Energy Features

For subband energy features, let $\mathbf{U}'_x \sim \mathcal{N}(\mathbf{u}_{x_i}, \frac{\gamma^{-1}}{2} I)$. Then project the distribution of the random variable $\mathbf{U}'_{z_i} = \mathbf{U}_h \cdot \mathbf{U}'_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{X}^{f\prime} \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}$ to the nearest Gaussian $\mathcal{N}(\mathbf{u}'_z|\mathbf{u}_{x_i}) = \mathcal{N}(\mathbf{u}'_z; \bar{\mathbf{U}}'_{z_i}, R_{u_{z_i}})$, where

$$\bar{\mathbf{U}}'_{z_i} = \bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i} + \sigma_w^2 \mathbf{1}, \tag{15}$$

$$R_{u_{z_i}} = \frac{\gamma^{-1}}{2} \operatorname{diag}\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right) + \Sigma_{u_h} \cdot \mathbf{u}_{x_i}\mathbf{u}_{x_i}^T + \sigma_w^4 I + 2\sigma_w^2 \operatorname{diag}\left(\bar{\mathbf{U}}_h \cdot \mathbf{u}_{x_i}\right), \tag{16}$$

which follows from (48) in the appendix with $\Sigma_{u_x} = \frac{\gamma^{-1}}{2} I$ and $\bar{\mathbf{U}}_x = \mathbf{u}_{x_i}$.

Then, solving the integral in (13), the projected RBF training kernel takes the form

$$K_{\text{proj}}\left(\mathbf{u}_{x_i}, \mathbf{u}_{x_j}\right) = \mathcal{N}\left(\bar{\mathbf{U}}_{z_i}; \bar{\mathbf{U}}_{z_j}, R_{u_{z_i}} + R_{u_{z_j}}\right).$$

At test time, given an observation $\mathbf{u}_z$, the distribution $p(\mathbf{u}'_z|\mathbf{u}_z) = \mathcal{N}\left(\mathbf{u}_z, \tilde{R}_{u_z}\right)$, where $\tilde{R}_{u_z}$ is $R_{u_{z_i}}$ in (16) with $\hat{\mathbf{u}}_x$ substituted for $\mathbf{u}_{x_i}$; $\hat{\mathbf{u}}_x$ satisfies $\mathbf{u}_z = \hat{\mathbf{u}}_x \cdot \bar{\mathbf{U}}_h + \sigma_w^2 \mathbf{1}$:

$$\hat{\mathbf{u}}_x = \frac{\left[\mathbf{u}_z - \sigma_w^2 \mathbf{1}\right]}{\left[\bar{\mathbf{U}}_h\right]},$$

where $\frac{[\mathbf{a}]}{[\mathbf{b}]}$ denotes Hadamard (element-wise) division of $\mathbf{a}$ and $\mathbf{b}$. Then, solving the integral in (13), the projected RBF test kernel for subband energy features is

$$K_{\text{proj}}\left(\mathbf{u}_z, \mathbf{u}_{x_i}\right) = \mathcal{N}\left(\mathbf{u}_z; \bar{\mathbf{U}}_{z_i}, R_{u_{z_i}} + \tilde{R}_{u_z}\right).$$

C. Projected RBF vs. Expected RBF Kernels for Subband Energy Features

The projected RBF and expected RBF kernels differ in the way that the statistics of the channel features are incorporated, and in the way that the bandwidth parameter γ is used. Comparing the covariance terms in Table II, we observe that covariance of the expected RBF kernel is given by

$$\Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} I,$$

whereas the covariance of the projected RBF kernel is given by

$$R_{u_{z_i}} + R_{u_{z_j}} = \Sigma_{u_{z_i}} + \Sigma_{u_{z_j}} + \gamma^{-1} \operatorname{diag}\left(\Sigma_h + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right).$$

Therefore, the training kernels differ only by the diagonal matrix weighted by γ^{-1}; they are identical as $\gamma \to \infty$. Since for the projected RBF SVM, γ^{-1} acts as a weight on the channel statistics $\operatorname{diag}\left(\Sigma_h + \bar{\mathbf{U}}_h \bar{\mathbf{U}}_h^T\right)$, we expect that the projected SVM may be more sensitive to channel estimation errors when γ^{-1} is large.

Another key difference between the two kernels is that the projected RBF kernel, in order to be consistent with prior work, treats the test vector $\mathbf{u}_z$ as a realization of a random vector with nonzero covariance. Conversely, in defining the expected RBF kernel we treat the test point as deterministic.

D. Projected Kernel for Arbitrary Features

Analogous to Section V-D, projected kernels can be computed for arbitrary features, and in particular, it is straightforward to compute an RBF projected kernel by fitting a Gaussian distribution to the VE's for each training sample $\mathbf{x}_i$.

E. Unscaled Projected RBF Kernel

Similar to the discussion of the unscaled expected RBF kernel in V-E, we found that the unnormalized version of the projected RBF kernel was computationally more efficient and more robust to poor estimation of the channel statistics, and did not change the performance if the channel statistics were accurately estimated. Thus, we used the unscaled projected RBF kernel in our experiments in Section IX and unless specifically stated, when referring to the "projected RBF kernel" we mean specifically the unscaled version. Using the same arguments given in V-E, one can show that the unscaled projected RBF kernel is positive definite.

VII. Adapting SVMs Trained on Clean Data to Corrupted Test Features

The presented expected and projected RBF kernels require that statistics (e.g., sample mean and covariance) of the auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$ are available to train the SVM. For each new environment, the SVM must be re-trained using the statistics of the stochastic channel. While we believe that it is optimal to train the SVM for the particular environment, as a practical question we considered whether we could train an SVM without knowing the environment, and only adapt the SVM for the environment at test time.

When training an SVM, one solves for coefficients $\{\alpha_i\}_{i=1}^N$ which determine the contribution of each training sample as shown in (2). Notably, some α_i's are set to zero in the training process, removing certain training samples from influencing the classifier.

Suppose that a kernel function $K(\cdot,\cdot)$ is selected for SVM classification. For cases in which re-training the SVM for each new environment is undesirable, we propose the following approach:

1) Train an SVM on the dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with kernel K to obtain the weights $\{\alpha_i\}_{i=1}^N$ and bias b in Eq. (2);
2) For a new propagation environment, collect auxiliary channel samples $\{\mathbf{h}_i\}_{i=1}^M$ and compute relevant statistics;
3) Calculate a bias term for the new environment using the KKT conditions of the SVM [7, p. 374]:

$$b' = \frac{1}{N}\sum_{i=1}^{N} \frac{1-\xi_i}{y_i} - \sum_{j=1}^{N} \alpha_j y_j K_{te}(\mathbf{x}_i, \mathbf{x}_j),$$

where ξ_i are the SVM slack variables, and $K_{te}(\cdot,\cdot)$ is the channel-robust test kernel function. The new bias b' minimizes the average label prediction error over all support vectors.
4) Classify the test sample as the sign of

$$f(\mathbf{z}) = b' + \sum_{i=1}^{N} \alpha_i y_i K_{te}(\mathbf{z}, \mathbf{x}_i).$$

This approach is theoretically sub-optimal, since the weights $\{\alpha_i\}_{i=1}^N$ that minimized empirical risk using the kernel K are not optimally suited to the test kernel K_{te}. Note that the role of the $\{\alpha_i\}_{i=1}^N$ is to weight how important each training sample is in determining a decision boundary, and these relative importance may not change much for the test conditions. Further, re-calculation of the bias term b' grossly adjusts the decision boundary so that at least the label of the support vectors are, on average, predicted accurately.

VIII. Local Joint QDA

Anderson and Gupta proposed a *joint QDA* classifier that accounts for a convolution channel, and derived the classifier for use with subband energy features [3]. Joint QDA classifier is simple to train and was shown to yield good results. Joint QDA is a maximum a posteriori (MAP) classifier that solves $\arg\max_g p(\mathbf{z}|g)p(g)$. Traditional QDA assumes that $p(\mathbf{x}|g)$ is Gaussian; joint QDA assumes that $p(\mathbf{z}|g) = \int p(\mathbf{z}|\mathbf{x})p(\mathbf{x}|g)\, d\mathbf{x}$ is Gaussian. To compute $p(\mathbf{z}|g)$, one first estimates the mean and covariance of $p(\mathbf{x}|g)$ as in standard QDA. When the number of features is small relative to the number of training samples, maximum likelihood estimation may suffice. When the number of features is relatively large, one can use *Bayesian QDA* to compute the expected Gaussian [21] and then compute the best Gaussian approximation for $p(\mathbf{z}|g)$ using the lemma above.

As with the channel-robust kernels in Sec. IV, the Gaussian assumption for joint QDA is motivated by the maximum entropy principle and by mathematical convenience, but does admit model bias. To reduce the model bias of QDA, we relax the assumption that $p(\mathbf{z}|g)$ is globally Gaussian. Reducing the model bias of QDA is commonly done using a Gaussian mixture model (GMM) [7], but GMM's can have high estimation variance due to local maxima of the likelihood and the need to choose the number of Gaussian components. Instead, we use another approach that has recently been shown to work well, which is to apply the Gaussian model locally to the nearest-neighbors of the test sample, an approach aptly termed *local QDA* [22]. We propose local joint QDA where given an

observation $\mathbf{z}$, the Gaussian is learned only for the expected nearest neighbors *for each class*, defined as follows.

Definition. *Expected Nearest Neighbors.* Model random training vector $\mathbf{Z}_i$ as Gaussian with mean $\bar{\mathbf{Z}}_i$ and covariance Σ_{z_i}. Given a test feature vector $\mathbf{z}$, the expected nearest neighbor of $\mathbf{z}$ is the random vector $\mathbf{Z}_\ell$, where

$$\begin{aligned}\ell &= \arg\min_i \ \mathrm{E}\left[\|\mathbf{z}-\mathbf{Z}_i\|^2\right] \\ &= \arg\min_i \ \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\,\mathrm{E}\left[\mathbf{Z}_i\right] + \mathrm{E}\left[\mathbf{Z}_i^T\mathbf{Z}_i\right] \\ &= \arg\min_i \ \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\bar{\mathbf{Z}}_i + \mathrm{tr}\,\Sigma_{z_i} + \bar{\mathbf{Z}}_i^T\bar{\mathbf{Z}}_i \\ &= \arg\min_i \ \|\mathbf{z}-\bar{\mathbf{Z}}_i\|^2 + \mathrm{tr}\,\Sigma_{z_i}. \end{aligned} \tag{17}$$

Note that the nearest neighbor depends on both the mean $\bar{\mathbf{Z}}_i$ and covariance Σ_{z_i} of a random training vector $\mathbf{Z}_i$. The second nearest neighbor is found in similar fashion, after $\mathbf{Z}_\ell$ has been excluded from the set of candidate neighbors, and so on for the subsequent nearest neighbors.

For class g, let $\mathcal{X}_g = \{\mathbf{x}_i : y_i = g\}$. Given observation $\mathbf{z}$, let $\mathcal{K}_g$ be the set of k nearest neighbors of $\mathbf{z}$ in $\mathcal{X}_g$, using the expected nearest neighbors definition in (17). Then, the mean and covariance of Gaussian likelihood $p(\mathbf{z}|g)$ is calculated as the sample mean and covariance, respectively, of $\mathcal{K}_g$. In this manner, Gaussian likelihoods $p(\mathbf{z}|g)$ are computed for each class g.

For discrete-time signal features, parameters of the distribution $p(\mathbf{z}|g) = \mathcal{N}(\mathbf{z}; \bar{\mathbf{Z}}, \Sigma_z)$ are given by

$$\bar{\mathbf{Z}} = \bar{\mathbf{H}} * \bar{\mathbf{X}}, \qquad \Sigma_z = \Sigma_h ** \Sigma_x + \sigma_w^2 I,$$

where $\bar{\mathbf{X}}$ and Σ_x are taken as the sample mean and covariance of the elements of $\mathcal{K}_g$, and $\bar{\mathbf{H}}$ and Σ_h are the sample mean and covariance of $\{\mathbf{h}_j\}_{j=1}^M$.

When using subband energy features, the distribution $p(\mathbf{u}_z|g) = \mathcal{N}(\mathbf{u}_z; \bar{\mathbf{U}}_z, \Sigma_{u_z})$ is specified by parameters

$$\begin{aligned}\bar{\mathbf{U}}_z &= \bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x + \sigma_w^2\mathbf{1}, \\ \Sigma_{u_z} &= \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h\bar{\mathbf{U}}_h^T\right)\cdot\Sigma_{u_x} + \Sigma_{u_h}\cdot\bar{\mathbf{U}}_x\bar{\mathbf{U}}_x^T \\ &\quad + \sigma_w^4 I + 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h\cdot\bar{\mathbf{U}}_x\right) \quad \text{[from (48)]},\end{aligned}$$

where $\bar{\mathbf{U}}_x$ and Σ_{u_x} are estimated from $\mathcal{K}_g$, and $\bar{\mathbf{U}}_h$ and Σ_{u_h} are estimated from auxiliary set $\{\mathbf{u}_{h_i}\}_{i=1}^M$.

IX. Experiments

We compare the expected RBF SVM, the expected RBF SVM trained on uncorrupted training pairs, projected RBF SVM, projected RBF SVM trained on uncorrupted training pairs and local joint QDA to VE RBF SVM, VE k-NN and to a channel agnostic RBF SVM. All RBF kernels were unscaled. We report classification accuracy for three separate experiments in which subband energies were used as features: simulated narrowband signals in a simulated multipath environment, real Bowhead whale vocalizations in the same simulated multipath environment, and trumpet/cornet sounds recorded in an anechoic chamber and a reverberant room.

A. Algorithmic Experimental Details

Given N subband energy feature vectors $\{\mathbf{u}_{x_i}\}_{i=1}^N$ and M auxiliary samples $\{\mathbf{u}_{h_i}\}_{i=1}^M$, we generate VEs for VE RBF SVM and VE k-NN as follows. For each $\mathbf{u}_{x_i}$, we generate M VEs by taking $\mathbf{u}_{x_i}$ with every element of $\{\mathbf{u}_{h_j}\}_{j=1}^M$ to form

$$\mathbf{u}_{z_{ij}} = \mathbf{u}_{h_j}\cdot\mathbf{u}_{x_i} + \sigma_w^2\mathbf{1},\ j = 1\ldots M. \tag{18}$$

For VE RBF SVM, an SVM is then trained from the $M\times N$ VEs; for VE k-NN, k nearest neighbors are chosen among the VEs. We chose to use the noise power σ_w^2 in (18) instead of generating Gaussian noise draws, since to incorporate a noise draw $\mathbf{w}^f$, the VE method would also require the Fourier coefficients $\mathbf{x}_i^f$ and $\mathbf{h}_j^f$ as in (4), which are not assumed to be provided for any of the classifiers we compare.

The agnostic RBF SVM is trained on $\{\mathbf{u}_{x_i}\}_{i=1}^N$; auxiliary channel feature vectors $\{\mathbf{u}_{h_i}\}_{i=1}^M$ are ignored.

For the standard machine learning problem, the training and test data are normalized using the sample means and standard deviations of the training samples. However, in this research the training and test features are related by the expression in (4), so that normalizing would not properly center and scale the test data. If $\mathbf{m}$ and $\mathbf{s}$ are the mean and scale, respectively, of the training data, and $\tilde{\mathbf{U}}_x = \frac{[\mathbf{U}_x - \mathbf{m}]}{[\mathbf{s}]}$ is the random variable describing the normalized training data, then scaling the test data and taking the expectation over $\mathbf{W}^f$ and $\mathbf{U}_h$ yields

$$\begin{aligned}\tilde{\mathbf{U}}_z = \frac{\left[\mathbf{U}_z - \bar{\mathbf{U}}_h\cdot\mathbf{m}\right]}{[\mathbf{s}]} &\xrightarrow{\text{expectation}} \frac{\left[\mathbf{U}_x\cdot\bar{\mathbf{U}}_h + \sigma_w^2\mathbf{1} - \bar{\mathbf{U}}_h\cdot\mathbf{m}\right]}{[\mathbf{s}]} \\ &= \tilde{\mathbf{U}}_x\cdot\bar{\mathbf{U}}_h + \frac{\left[\sigma_w^2\mathbf{1}\right]}{[\mathbf{s}]} \\ &\neq \tilde{\mathbf{U}}_x\cdot\bar{\mathbf{U}}_h + \sigma_w^2\mathbf{1},\end{aligned}$$

so that data normalization distorts the relationship between the test and training data when the noise power is non-negligible.

Though we cannot normalize the data, we can achieve a similar effect by adjusting the RBF kernel bandwidth parameter. For each dataset, the RBF bandwidth parameter γ is cross-validated over a range of length-scales that are related to the inter-sample distances between points. As a heuristic to choosing reasonable values for γ, we introduce a parameter β that chooses γ^{-1} as a multiple of the minimum inter-neighbor distance according to a logarithmic scale:

$$\gamma^{-1} = \left(\frac{d_{\max}}{d_{\min}}\right)^\beta d_{\min},$$

where $d_{\min}$ and $d_{\max}$ are respectively the minimum and maximum inter-sample distances of the training set. Thus, for $\beta = 0$, $\gamma^{-1} = d_{\min}$, for $\beta = 1$, $\gamma^{-1} = d_{\max}$, and so on. For cross-validation, we cross-validate β over the set $\beta \in \{-1.5, -1.25, \ldots, 2.25, 2.5\}$. We allow γ^{-1} to be greater than the maximum inter-neighbor distance (for $\beta > 1$) or less than the minimum inter-neighbor distance (for $\beta < 0$) since γ^{-1} plays the role of a regularization parameter in (16), and these larger and smaller values are sometimes chosen. The SVM margin penalty C is cross-validated over the set $\{1, 10^1, 10^2, 10^3, 10^4\}$. The k-NN classifier cross-validates over a single parameter $k \in \{1, 3, 5, 9, 17, 33, N\}$ (goes like $2^n + 1$). Since local joint QDA estimates a class-conditional

mean and covariance from k_y training samples *per class*, the role of k_y differs from that of k. For local joint QDA, we cross-validated over the number of class-specific neighbors $k_y \in \{5, 9, 17, 33, |\mathcal{X}_y|\}$ and whether to use the maximum-likelihood estimate of the full covariance or assume a diagonal covariance.

For each of the classifiers, a tie for the parameter pairs (or single parameter for k-NN) that achieved the best cross-validation score was settled by choosing among the best performing parameter pairs randomly with equal probability.

B. Simulated Signals and Bathymetric Environment

This dataset simulates narrowband signals propagating in a shallow water sonar environment. Realistic sonar channel impulse responses were generated using the *CASS Eigenray* routine in the Sonar Simulation Toolset (SST) [23]. This dataset is the same as used in Anderson et al. [3], except that we partition the training and test data differently. There are $N = 200$ narrowband training signals from two classes (100 from each class), which were generated by varying the placement of poles in the z-transform domain. Each signal is a two zero, four pole (2 conjugate pairs) real signal model, where the poles are placed at fixed angles, but their distance from the origin is drawn from a multivariate Gaussian distribution. The class-conditional means and covariances of the pole placement model were selected to provide three binary classification problems of varying difficulty: `easy`, `medium` and `hard`. Subband energies at two frequencies (corresponding to the pole placement angles) were used as features. In addition, $M = 20$ channel impulse responses were generated by first randomly picking a source location in a simulated bathymetry, then simulating with SST the propagation of a pulse from that random source location to a fixed receiver location. We then computed $M = 20$ channel subband energy feature vectors $\{\mathbf{u}_{h_i}\}_{i=1}^{M}$, which were provided to each of the classifiers as training data. The 1800 test signals were formed by convolving an i.i.d. set of narrowband signals with 1800 i.i.d. randomly drawn channel impulse responses, generated i.i.d. as the training channel impulse responses. Then i.i.d. Gaussian noise was added in varying amounts to each test signal to achieve SNRs of -10dB, -5dB, 0dB, 5dB, and 10dB.

Results averaged over 10 runs of each of the experiments `hard`, `medium` and `easy` are shown in Fig. 1 (a), (b), and (c), respectively.

We note that the experimental setup for the simulated and Bowhead data differs in a few ways from previous publications with these datasets [6] and [3]. Jamieson et al. [6] compared performance of classifiers for a fixed training time, which necessitated utilizing a smaller number M of auxiliary samples for VE than for expected RBF SVM. In this paper, the experiment is set up to compare performance of algorithms when the same data is available to each, regardless of training time. In addition, in [6], leave-one-out-crossvalidation was performed for each combination of M auxiliary features and N training features. We employ ten-fold cross-validation to determine the values of both γ and C (for SVM) over a classifier-agnostic set of values. In [3], we note a mistake in the labeling of SNR in the results, so that the actual SNR is roughly 10 dB higher than the SNR as labeled.

C. Classifying Whales in Bathymetric Environment

The whale classification dataset is the same as used in Anderson and Gupta [3]. The data consists of recordings of song endnotes from two distinct Bowhead whales in deep water: 15 calls from one whale and 9 calls from another whale. The goal is to identify each whale from its vocalizations when located in a shallow-water environment. We employ the same bathymetry as in [3], so that the experimental setup follows that of Section IX-B. For each run, we randomly split the 24 whale calls into 10 and 14 signals: $N = 10$ training signals, and 14 signals from which we generate multipath-corrupted noisy test signals. The results shown in Fig. 2 were averaged over 1000 such i.i.d. training / test partitions.

D. Classifying Trumpeters

The third experiment uses real signals with real multipath corruption. The classifier must discriminate between two professional musicians, based on how they play the same note on either a trumpet or cornet in a reverberant environment. The training dataset consists of subband energy features extracted from recordings of two different professional trumpet players, Matthew Swihart (`Matt`) and Edward Castro (`Ed`), playing concert F in an anechoic chamber (Fig. 3) twenty times on both their own trumpet and cornet. Using the four datasets we constructed six different classification problems: `Ed Cornet` vs `Ed Trumpet`, `Matt Cornet` vs `Matt Trumpet`, `Matt Trumpet` vs `Ed Trumpet`, `Matt Cornet` vs `Ed Cornet`, `Matt Trumpet` vs `Ed Cornet`, and `Matt Cornet` vs `Ed Trumpet`. All six tasks can be classified with 97.4% accuracy or above by k-NN if anechoic signal features are used for both training and test.

Test signals were recorded in an outdoor semi-enclosed breezeway with an audible reverberation length of about 1 second. For a more controlled experiment, we played each of the anechoic signals in the breezeway through high-quality speakers in a fixed location, as well as quadratic chirps, which were used to estimate the reverberation impulse response. Signals were recorded at four locations in the breezeway with the same recorder. All recordings were stereo at a 48kHz sample rate and 16 bits per sample. When classifying a test signal $z[n]$ that corresponds to an anechoic signal $x[n]$, features of the stereo pair of $x[n]$ were excluded from the training set. An example training signal, test signal, impulse response and classification scatterplot are shown in Figure 4.

Features were taken to be the subband energies at the frequencies $\{349, 698, 1048, 1397, 1746, 347, 351\}$ (Hz), corresponding to the fundamental frequency $f_0 = 349$ (concert F), the first four harmonics, and f_0-2 and f_0+2 to capture the width of the fundamental. To correct for minor fluctuations in pitch, the fundamental and harmonic frequencies were adjusted to correspond to the highest peak within a 10 Hz window around the desired frequency, but the corrected fundamental frequency was typically within 2 Hz of 349 Hz. Noise energy

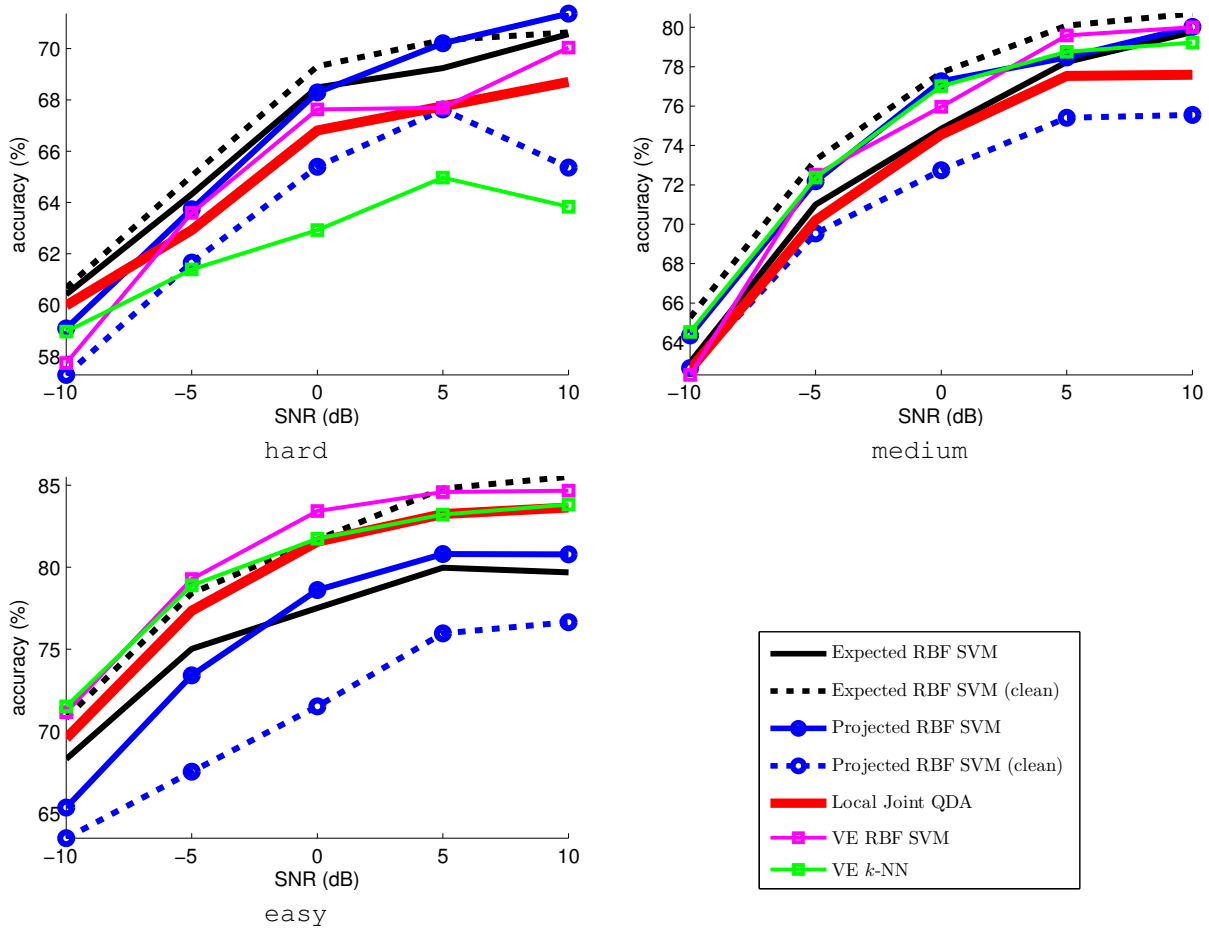


Fig. 1. Classification accuracy of simulated signals in simulated bathymetry using subband energy features. The datasets `easy`, `medium` and `hard` differ in how well the classes are separated in feature space. Note that the accuracy axis for each plot is on a different scale in order to highlight the relative performance of algorithms. RBF SVM (agnostic) achieves an accuracy of 50% for all SNR in each experiment, and is not shown.

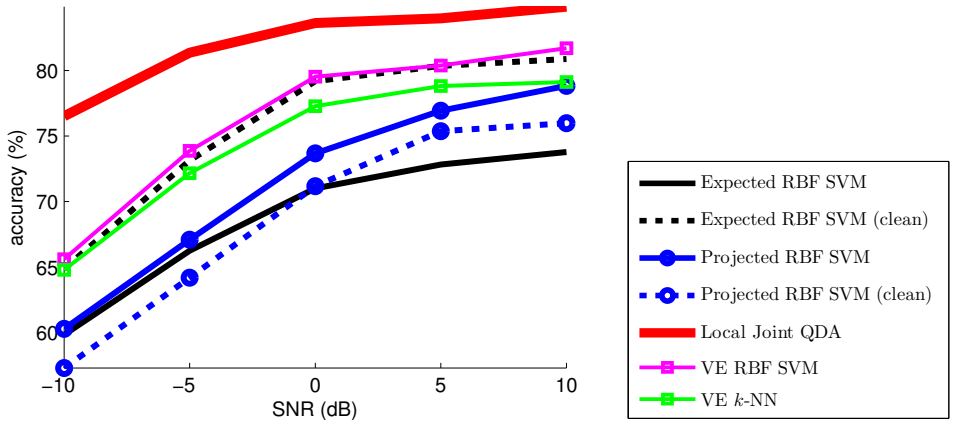
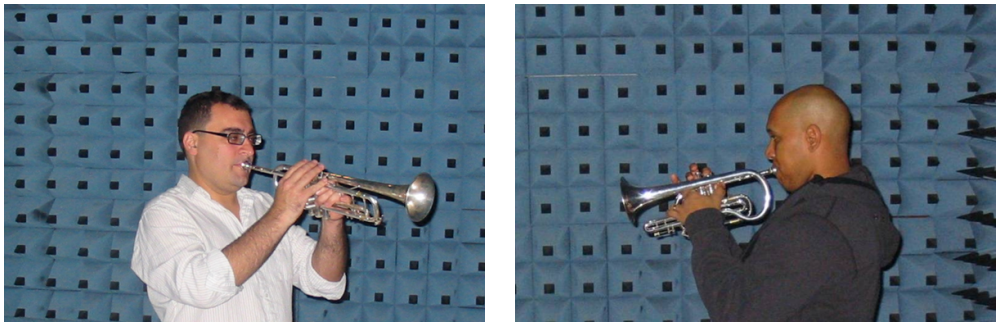


Fig. 2. Classification accuracy of Bowhead whale endnotes in simulated bathymetry using subband energy features. RBF SVM (agnostic) achieves an accuracy of $48\% \pm 1\%$ for all SNR, and is not shown.



(a) (b)

Fig. 3. (a) Matthew Swihart on trumpet and (b) Edward Castro on cornet in an anechoic chamber.

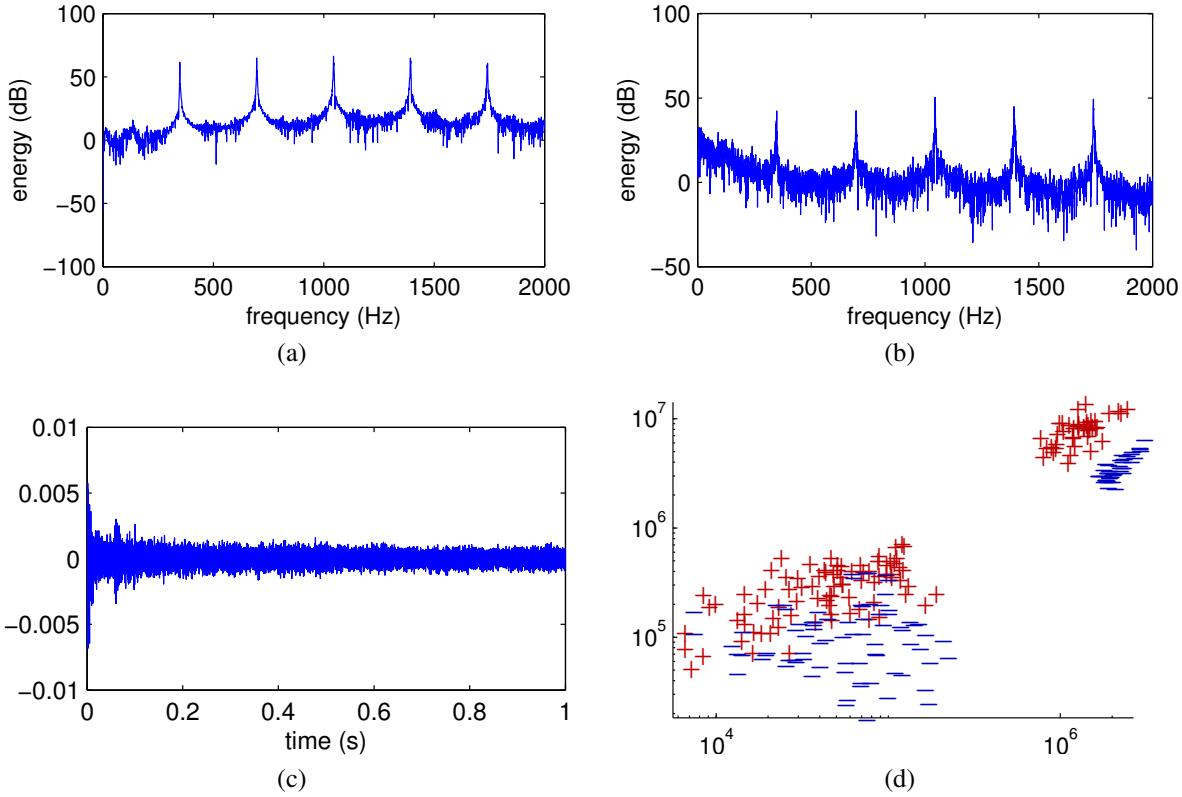


Fig. 4. (a) The energy spectrum of concert F played by Ed on the cornet in the anechoic chamber; (b) the energy spectrum of a test signal generated by playing back the recorded note in an echo chamber; (c) an impulse response estimated by probing the outdoor breezeway with a quadratic chirp; (d) training (upper right) and test features (lower left)—corresponding to subband energies at fundamental and first harmonic—plotted together on a log-log plot, where `Ed Cornet` is denoted by – and `Matt Trumpet` is denoted by +.

σ_w^2 was taken to be the median energy level across all frequency bins.

Results shown in Table III were averaged over the four locations.

X. Results

First, we consider experimental training times, and then we discuss the classification results experiment-by-experiment, and then make some overall notes comparing the classifiers.

A. Training Time

The expected and projected RBF SVM classifiers incur a training cost of $O\left(N^3\right)$. However, there is also a cost in populating the $N \times N$ kernel matrix, since each entry requires computing the inverse of a $d \times d$ non-diagonal matrix (see Table II). As noted previously, the VE RBF SVM incurs a training cost of $O\left(M^3N^3\right)$ since the dataset has been increased to a factor of M. A plot comparing the training times for the simulated signal experiment for the VE RBF SVM, expected RBF SVM, and agnostic RBF SVM using `libsvm` [24] on a 3.2 GHz Intel Core i7 CPU is shown in Fig. 5. (Total RAM was 12GB, so that the SVMs were not memory limited.)

B. Classification Results by Experiment

Classification results for simulated data in the synthetic bathymetry are shown in Fig. 1. For `hard`, clean-train expected SVM is the best overall performer with 95% con-

TABLE III
TRUMPET PLAYBACK RESULTS AVERAGES. BOLDED ITEMS IN EACH COLUMN ARE STATISTICALLY TIED WITH 95% CONFIDENCE ACCORDING TO A ONE-SIDED WILCOXON SIGN RANK TEST.

	Matt Cornet v. Ed Cornet	Matt Trumpet v. Ed Trumpet	Matt Cornet v. Matt Trumpet	Ed Cornet v. Ed Trumpet	Matt Cornet v. Ed Trumpet	Matt Trumpet v. Ed Cornet
Expected RBF SVM	**74.8**	**72.0**	**82.9**	**60.3**	**82.3**	60.4
Expected RBF SVM (clean)	68.6	52.0	57.9	52.4	57.3	60.4
Projected RBF SVM	57.6	56.4	73.7	57.4	68.0	59.1
Projected RBF SVM (clean)	55.2	63.9	76.9	57.1	69.8	58.1
Joint QDA	**73.2**	**72.0**	80.7	**61.2**	**81.1**	**65.4**
VE RBF SVM	**73.0**	68.0	72.5	51.0	66.9	61.1
VE k-NN	71.0	56.4	77.7	**60.4**	73.8	60.4
RBF SVM (agnostic)	51.2	49.7	57.7	51.3	52.5	**62.8**

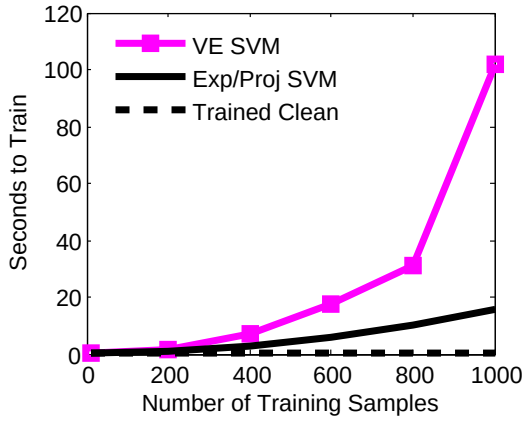


Fig. 5. SVM training time vs. training set size N for fixed $M = 20$ used in the simulation experiment. Timing results include the time required to populate the kernel matrix.

fidence, followed closely by expected SVM and projected SVM, which are statistically tied. VE SVM gives slightly better performance than local joint QDA. A similar trend holds for `medium`: clean-train expected SVM clean is the best overall performer, followed by projected SVM and VE k-NN (statistically tied overall). For `easy`, VE SVM is the best overall performer, followed closely by expected SVM clean, then VE k-NN and local joint QDA. In all of these experiments, the agnostic SVM, which treats the corrupted test data as though it were not corrupted, produces almost exactly a 50% classification rate, which is as poor as random guessing.

Classification results for the whale endnotes experiment are shown in Fig. 2. Local joint QDA is clearly the best performer. Though close, VE SVM is a slightly better overall performer for this case than clean-train expected SVM with statistical significance greater than 95%.

For the trumpet classification, Table III shows that in all datasets except `Matt Trumpet` v. `Ed Cornet`, expected SVM is the best performer or statistically tied with the best performer, and in `Matt Trumpet` v. `Ed Cornet`, it is the second best performer. Likewise, local joint QDA is the best (or tied for best) performer in all tests except `Matt Cornet` v. `Matt Trumpet`, for which it is the second best classifier. VE SVM is tied with expected SVM as the best performer in `Matt Cornet` v. `Ed Cornet`. Projected SVM and clean-train projected SVM—which yield similar results in most experiments—perform better than VE SVM in 3 experiments. The clean-train expected SVM classifier does not perform well on the trumpet/cornet experiments.

C. Classification Results by Classifier

First, we note that the agnostic RBF SVM, which ignores the channel, fails miserably for almost all of these experiments, and thus some form of channel-adaptation should be used. However, given how poor the channel estimates were for these experiments (especially for the trumpet classification), the classification gains produced by the adapted methods were pleasantly surprising.

The clean-train expected/projected SVM classifiers have the least channel adaptation. They use the SVM coefficients $\{\alpha_i\}$ trained on the clean training data, and only adapt the kernel at test time to attempt to better model similarity between the test sample and training sample. Both clean-train SVMs do significantly better than the agnostic over the datasets, suggesting that adapting only the kernel is worthwhile. The clean-train projected SVM generally performs worse than the clean-train expected SVM. We hypothesized that the expected SVM would always do better than its clean-train counterpart because its coefficients were trained for the test-environment. Surprisingly, for both experiments using the bathymetry to generate sonar impulse responses, the expected RBF SVM *clean* consistently does better than the expected RBF SVM. However, the expected RBF SVM clean does not do well at the trumpet identification. We suspect that this is because the channels were more corrupting for this experiment, and it was more important to take them into account at training time.

Overall, the expected and projected kernels performed similarly, with very comparable performance on the simulation results, a win for projected on the real whale data, and a win for expected on the real trumpet data. Based on comparing the mathematical formulas of the expected and projected RBF, we hypothesize that the projected RBF kernel will be more sensitive to the quality of the channel estimation errors. This hypothesis is supported by the experimental results showing that projected RBF SVM performs comparatively worse than other algorithms with the poor-quality estimates of the reverberation channels in the breezeway, whereas it was competitive in experiments with simulated multipath.

The VE methods performed poorly on the trumpet data relative to the expected SVM, but comparably when given

the simulated bathymetry channels. This may be because the regularization inherent in expected SVM by aggregating the example channels into a channel mean and covariance is more helpful when the channel examples are poor, as in the case of the trumpet data. Further, the VE methods do better relative to the proposed expected/projected kernels on problems where the classes are easier to separate: such as the `easy` simulation and the whale problem. But the VE methods do worse relative to the proposed expected/projected kernels on problems where the classes are harder to separate: such as the `hard` simulation and the trumpet problem. The clean-train expected SVM performs comparably to the VE SVM for all the simulated channel problems despite taking orders of magnitude less training time, but performs slightly worse on the trumpet datasets.

We found local joint QDA to be the most robust classifier. Its mean performance on the trumpet data is second-best, it is the clear winner in distinguishing the whales, and it performs fairly well with the narrowband signal experiment. Also, compared to the SVM classifiers, we found the local joint QDA method to be much more robust to the choice of its cross-validation parameters.

XI. Conclusions

We have presented and compared several classifiers to address the dataset shift problem that occurs in signal classification problems when the differences between training and test conditions can be modeled by a linear time-invariant channel and additive Gaussian white noise.

The contributions of this paper can be summarized as follows. Given a kernel function $K(\cdot,\cdot)$, the expected kernel is defined as the expectation of the similarity of random variables $\mathbf{Z}_i$ and $\mathbf{Z}_j$ according to the corruption model $p_{Z_i|x_i}$. Similarly, motivated by a recent interpretation of the RBF kernel in [20], we presented the projected RBF Kernel that accounts for a stochastic channel and additive noise. We derived the expected RBF and projected RBF kernels for discrete-time signal features and subband energy features. We noted that the kernels can be computed for arbitrary features using Monte Carlo approximations, and that RBF kernels for arbitrary features can be computed without Monte Carlo by fitting a Gaussian distribution to the VE's for each training sample. Unscaled expected and projected RBF kernel functions were defined and also shown to be positive definite functions. As a practical consideration, we considered a theoretically suboptimal procedure for the expected and projected RBF SVMs that do not require re-training the SVM for each new environment characterized by $\{\mathbf{h}_i\}_{i=1}^M$ and σ_w^2. In the *clean-train* expected and projected RBF SVMs, the coefficients $\{\alpha_i\}_{i=1}^N$ are learned from $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ once using a standard RBF kernel; at test time, only the SVM bias term is recalculated for a channel-robust kernel. To reduce model bias, the *local* joint QDA classifier was presented as a generalization of the joint QDA classifier [3].

Experiments with simulated and real data revealed the following trends. On easier problems, where the classes were well-separated and the channels were realistic but simulated, the VE methods performed well. On harder problems, where the classes are less well-separated and the channel estimation was poorer, the expected and projected kernel SVMs performed better. In particular, the expected kernel seemed most robust to non-idealistic conditions, but less able to take advantage of good conditions. In addition, not only are the expected/projected kernels theoretically much faster to train due to the $O(N^3)$ complexity of the SVM training procedure, we found that in practice with even relatively small sample sizes they were significantly faster to train.

Under cleaner conditions, we were surprised to see that the clean-train expected SVM outperformed the expected SVM, and was often the best performer of all the considered methods. Notably, clean-train SVMs are the fastest SVMs to train. Throughout, the local joint QDA method performed consistently well, was robust to parameter choices and estimates, and is trivial to train. Modifying local joint QDA for the problem of estimating Gaussian parameters in high dimensions with few training samples is straightforward by applying the results in [21].

While further experimental studies with a wider variety of channels and data are needed, our advice to practitioners based on the experimental evidence we have is to use the clean-train expected SVM or local joint QDA if the channels are not considered too severe, and to use the expected SVM if the channels are thought to be highly corrupting.

Because of their popularity, we have derived robust RBF kernels for discrete-time signal features and subband energy features. However, it would be beneficial to derive expected kernels for other popular kernels (e.g., polynomial, triangular, hyperbolic) and other standard features (e.g., cepstral or wavelet coefficients).

APPENDIX

Identities

We frequently use the following identities.

For any vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^d$ and scalar $\beta \in \mathbb{C}$,

$$(\mathbf{a}\cdot\mathbf{b})(\mathbf{a}\cdot\mathbf{b})^T = \left(\mathbf{a}\mathbf{a}^T\right)\cdot\left(\mathbf{b}\mathbf{b}^T\right) \tag{19}$$

$$\mathbf{a}\mathbf{b}^T\cdot(\beta I) = \beta\,\mathrm{diag}\,(\mathbf{a}\cdot\mathbf{b}), \tag{20}$$

which can be verified by writing the relationships in summation form.

Let $\mathbf{V} \in \mathbb{C}^d$ be a complex Gaussian vector that is zero mean, white, and proper. By definition of proper complex vectors [25],

$$\mathrm{E}\left[\mathbf{V}\mathbf{V}^H\right] = \mathrm{E}\left[\mathbf{V}^*\mathbf{V}^T\right] = \sigma_v^2 I \tag{21}$$

$$\mathrm{E}\left[\mathbf{V}\mathbf{V}^T\right] = 0. \tag{22}$$

Since $\mathbf{V}$ is zero-mean and Gaussian,

$$\mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,\mathbf{V}^T\right] = \mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,\mathbf{V}^H\right]$$
$$= \mathrm{E}\left[\mathbf{V}\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] = \mathrm{E}\left[\mathbf{V}^*\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] = 0. \tag{23}$$

From Isslerlis' Gaussian moment theorem [26],

$$\mathrm{E}\left[(\mathbf{V}\cdot\mathbf{V}^*)\,(\mathbf{V}\cdot\mathbf{V}^*)^T\right] = \sigma_v^4 I + \sigma_v^4 \mathbf{1}\mathbf{1}^T, \tag{24}$$

where $\mathbf{1}\mathbf{1}^T$ is a matrix of all ones.

Lastly, for $\mathbf{x}, \mathbf{a} \in \mathbb{R}^n$, $A \in \mathbb{S}^n_{++}$, $P \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $B \in \mathbb{S}^m_{++}$,

$$\mathcal{N}(\mathbf{x}; \mathbf{a}, A)\mathcal{N}(P\mathbf{x}; \mathbf{b}, B) = \mathcal{N}\left(\mathbf{b}; P\mathbf{a}, B + PAP^T\right)\mathcal{N}(\mathbf{x}; \mathbf{c}, C), \quad (25)$$

where $c = C\left(A^{-1}\mathbf{a} + PB^{-1}\mathbf{b}\right)$ and $C = \left(A^{-1} + P^TB^{-1}P\right)^{-1}$.

Derivation of Covariance of $\mathbf{U}_z$

Let $\mathbf{X}^f$, $\mathbf{H}^f$ and $\mathbf{W}^f$ be mutually independent random vectors in the subband energy relationship in (4), and let $\mathbf{W}^f$ be a complex Gaussian vector that is zero mean, white, and proper. Then,

$$\begin{aligned}
\mathrm{Cov}\left[\mathbf{U}_z\right] &= \mathrm{Cov}\left[\mathbf{U}_h \cdot \mathbf{U}_x + \mathbf{U}_w + 2\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right] \\
&= \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] + \mathrm{E}\left[\mathbf{U}_w\mathbf{U}_w^T\right] \\
&\quad + 4\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && (26)\\
&\quad + \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\mathbf{U}_w^T\right] + \mathrm{E}\left[\mathbf{U}_w\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] && (27)\\
&\quad + 2\,\mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && (28)\\
&\quad + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] && (29)\\
&\quad + 2\,\mathrm{E}\left[\mathbf{U}_w\,\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}^T\right] && (30)\\
&\quad + 2\,\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\mathbf{U}_w^T\right] && (31)\\
&\quad - \left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T - \bar{\mathbf{U}}_w\bar{\mathbf{U}}_w^T \\
&\quad - \bar{\mathbf{U}}_w\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T - \left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\bar{\mathbf{U}}_w^T, && (32)
\end{aligned}$$

where additional terms involving $\mathrm{E}\left[\mathrm{Re}\left\{\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right\}\right]$ are zero because $\mathbf{W}^f$ is zero mean and independent of $\mathbf{X}^f$ and $\mathbf{H}^f$. Line (27) cancels with (32). Lines (28) and (29) are zero since $\mathbf{W}^f$ is zero mean and uncorrelated with $\mathbf{X}^f$ and $\mathbf{H}^f$. Since $\mathbf{W}^f$ is proper and $\mathbf{U}_w = \mathbf{W}^f \cdot \mathbf{W}^{f*}$, lines (30) and (31) are zero by property (23). By expanding $\mathrm{Re}\{\mathbf{a}\} = \frac{1}{2}(\mathbf{a} + \mathbf{a}^*)$ and multiplying, line (26) can be rewritten as

$$\begin{aligned}
&\mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] && (33)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] && (34)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^T\right] && (35)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^*\left(\mathbf{X}^f \cdot \mathbf{H}^f \cdot \mathbf{W}^{f*}\right)^H\right] && (36)\\
&= \mathrm{E}\left[\left(\mathbf{X}^f\mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f\mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*}\mathbf{W}^{fT}\right)\right] && (37)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^{f*}\mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*}\mathbf{H}^{fT}\right) \cdot \left(\mathbf{W}^f\mathbf{W}^{fH}\right)\right]. && (38)
\end{aligned}$$

where properties (19) and (22) can be used to verify that lines (33) and (36) are zero. Using (19), lines (34) and (35) become (37) and (38), respectively. This yields

$$\begin{aligned}
\mathrm{Cov}\left[\ \mathbf{U}_z\right] &= \mathrm{E}\left[\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)\left(\mathbf{U}_h \cdot \mathbf{U}_x\right)^T\right] && (39)\\
&\quad - \left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)^T && (40)\\
&\quad + \mathrm{E}\left[\mathbf{U}_w\mathbf{U}_w^T\right] - \bar{\mathbf{U}}_w\bar{\mathbf{U}}_w^T && (41)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^f\mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f\mathbf{H}^{fH}\right) \cdot \left(\mathbf{W}^{f*}\mathbf{W}^{fT}\right)\right] && (42)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^{f*}\mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*}\mathbf{H}^{fT}\right) \cdot \left(\mathbf{W}^f\mathbf{W}^{fH}\right)\right] && (43)\\
&= \mathrm{E}\left[\mathbf{U}_h\mathbf{U}_h^T \cdot \mathbf{U}_x\mathbf{U}_x^T\right] - \bar{\mathbf{U}}_h\bar{\mathbf{U}}_h^T \cdot \bar{\mathbf{U}}_x\bar{\mathbf{U}}_x^T && (44)\\
&\quad + \mathrm{E}\left[\mathbf{U}_w\mathbf{U}_w^T\right] - \bar{\mathbf{U}}_w\bar{\mathbf{U}}_w^T && (45)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^f\mathbf{X}^{fH}\right) \cdot \left(\mathbf{H}^f\mathbf{H}^{fH}\right)\right] \cdot \sigma_w^2 I && (46)\\
&\quad + \mathrm{E}\left[\left(\mathbf{X}^{f*}\mathbf{X}^{fT}\right) \cdot \left(\mathbf{H}^{f*}\mathbf{H}^{fT}\right)\right] \cdot \sigma_w^2 I, && (47)
\end{aligned}$$

where property (19) was used to rewrite (39) and (40) as (44). Then, use (21) to simplify (42) and (43) as, respectively, (46) and (47). In (46) and (47), $\mathrm{E}\left[\left(\mathbf{X}^f\mathbf{X}^{fH}\right)\right] \cdot \sigma_w^2 I = \sigma_w^2\,\mathrm{diag}\left(\mathrm{E}\left[\mathbf{X}^f \cdot \mathbf{X}^{f*}\right]\right) = \mathrm{diag}\left(\bar{\mathbf{U}}_x\right)$ by (20) and by definition of $\mathbf{U}_x$ (similarly for terms involving $\mathbf{H}^f$). Thus, (46) and (47) simplify to $2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right)$. Applying (24) to $\mathrm{E}\left[\mathbf{U}_w\mathbf{U}_w^T\right]$, and recalling that $\bar{\mathbf{U}}_w = \sigma_w^2\mathbf{1}$, (45) reduces to $\sigma_w^4 I$. Finally, $\mathrm{E}\left[\mathbf{U}_h\mathbf{U}_h^T \cdot \mathbf{U}_x\mathbf{U}_x^T\right] - \bar{\mathbf{U}}_h\bar{\mathbf{U}}_h^T \cdot \bar{\mathbf{U}}_x\bar{\mathbf{U}}_x^T = \left(\Sigma_{u_h} + \bar{\mathbf{U}}_h\bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_x} + \Sigma_{u_h} \cdot \bar{\mathbf{U}}_x\bar{\mathbf{U}}_x^T$, so that be collecting terms, we have

$$\begin{aligned}
\mathrm{Cov}\left[\mathbf{U}_z\right] = &\left(\Sigma_{u_h} + \bar{\mathbf{U}}_h\bar{\mathbf{U}}_h^T\right) \cdot \Sigma_{u_x} + \Sigma_{u_h} \cdot \bar{\mathbf{U}}_x\bar{\mathbf{U}}_x^T + \sigma_w^4 I \\
&+ 2\sigma_w^2\,\mathrm{diag}\left(\bar{\mathbf{U}}_h \cdot \bar{\mathbf{U}}_x\right). \quad (48)
\end{aligned}$$

Proof of Lemma.

Let $f(\mathbf{m}, R) = \mathrm{KL}\left(p_Z || \mathcal{N}(\mathbf{m}, R)\right)$ for $\mathbf{m} \in \mathbb{R}^d$ and $R \in \mathbb{S}^d_{++}$. By definition, $\mathrm{KL}(p||q) = \mathrm{E}_{\mathbf{Z}}[\log p(\mathbf{Z})] - \mathrm{E}_{\mathbf{Z}}[\log q(\mathbf{Z})]$, and thus the mean $\mathbf{m}^*$ and covariance R^* we seek solve

$$\begin{aligned}
&\arg\min_{R\succ 0,\mathbf{m}} f(\mathbf{m}, R) = \arg\min_{R\succ 0,\mathbf{m}} -\mathrm{E}_{\mathbf{Z}}\left[\log\mathcal{N}(\mathbf{Z}; \mathbf{m}, R)\right] \\
&= \arg\min_{R\succ 0,\mathbf{m}} \log|R| + \mathrm{E}_{\mathbf{Z}}\left[(\mathbf{Z}-\mathbf{m})^T R^{-1}(\mathbf{Z}-\mathbf{m})\right] \\
&= \arg\min_{R\succ 0,\mathbf{m}} \log|R| + \mathrm{tr}\ \mathrm{E}_{\mathbf{Z}}\left[(\mathbf{Z}-\mathbf{m})(\mathbf{Z}-\mathbf{m})^T R^{-1}\right].
\end{aligned}$$

Since $f(\mathbf{m}, R)$ is convex in R, the minimizer $\mathbf{m}^*$ is found by solving

$$\nabla_m f(\mathbf{m}^*, R) = -2R^{-1}\,\mathrm{E}_{\mathbf{Z}}\left[(\mathbf{Z}-\mathbf{m}^*)\right] = 0,$$

and therefore $\mathbf{m}^* = \bar{\mathbf{Z}}$ is the unique global minimizer since $\mathbf{m}^*$ does not depend on R.

However, $f(\mathbf{m}, R)$ is not convex in $\mathbf{m}$, but for fixed $\mathbf{m} = \bar{\mathbf{Z}}$

$$\begin{aligned}
&\arg\min_{R\succ 0} \log|R| + \mathrm{tr}\ \mathrm{E}_{\mathbf{Z}}\left[(\mathbf{Z}-\mathbf{m})(\mathbf{Z}-\mathbf{m})^T R^{-1}\right] \\
&= \arg\min_{R\succ 0} -\log|R^{-1}| + \mathrm{tr}\ \Sigma R^{-1} \\
&= \arg\min_{Y\succ 0} -\log|Y| + \mathrm{tr}\ \Sigma Y,
\end{aligned}$$

since the change of variables $Y = R^{-1}$ is a bijection from $\mathbb{S}^d_{++}$ onto $\mathbb{S}^d_{++}$. The function $g(Y) = -\log|Y| + \mathrm{tr}\ \Sigma Y$ is

strictly convex [27], so that the unique global minimizer is found by solving

$$\nabla_Y g(Y^*) = -Y^{*-1} + \Sigma = 0,$$

so that $Y^* = \Sigma^{-1}$. We conclude that $\mathbf{m}^* = \bar{\mathbf{Z}}$ and $R^* = \Sigma$ uniquely minimize $f(\mathbf{m}, R)$. □

REFERENCES

[1] L. Buera, E. Lleida, A. Miguel, A. Ortega, and O. Saz, "Cepstral vector normalization based on stereo data for robust speech recognition," *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 15, no. 3, pp. 109–1113, March 2007.

[2] B. Raj and R. M. Stern, "Missing-feature approaches in speech recognition," *Sig. Proc. Magazine*, vol. 22, no. 5, pp. 101–116, 2005.

[3] H. S. Anderson and M. R. Gupta, "Joint deconvolution and classification with applications to passive acoustic underwater multipath," *J. Acous. Soc. Am.*, vol. 124, no. 5, pp. 2973–2983, November 2008.

[4] A. J. Llorens, T. L. Philip, I. W. Schurman, and C. R. Lorenz, "Enhancing passive automation performance using an acoustic propagation simulation," *J. Acoust. Soc. Am*, vol. 125, no. 4, pp. 2577, April 2009.

[5] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau, "Eigenface-domain super-resolution for face recognition," *IEEE Trans. Imag. Proc.*, vol. 12, no. 5, May 2003.

[6] K. Jamieson, M. R. Gupta, E. Swanson, and H. S. Anderson, "Training a support vector machine to classify signals in a real environment given clean training data," *Proc. IEEE ICASSP*, 2010.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2001.

[8] C. A. Cabrelli, "Minimum entropy deconvolution and simplicity: A noniterative algorithm," *Geophysics*, vol. 50, pp. 394–413, 1984.

[9] M. K. Broadhead and L. A. Pflug, "Performance of some sparseness criterion blind deconvolution methods in the presence of noise," *J. Acoust. Soc. Am.*, vol. 107, no. 2, pp. 885–893, Februrary 2000.

[10] M. J. Roan, M. R. Gramann, J. G. Erling, and L. H. Sibuld, "Blind deconvolution applied to acoustical systems identification with supporting experimental results," *J. Acoust. Soc. Am.*, vol. 114, no. 4, pp. 1988–1996, 2003.

[11] G. Okopal, P. Loughlin, and L. Cohen, "Dispersion-invariant features for classification," *J. Acoust. Soc. Am*, vol. 123, no. 2, pp. 832–841, February 2008.

[12] G. Okopal and P. J. Loughlin, "Propagation-invariant classification of signals in channels with dispersion and damping," *OCEANS*, 2007.

[13] P. Loughlin and L. Cohen, "Moment features invariant to dispersion," *Proc. SPIE*, vol. 5426, no. 235, 2004.

[14] Y. S. Abu-Mostafa, "Learning from hints in neural networks," *J. Complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[15] D. Decoste and B. Schölkopf, "Training invariant support machines," *Mach. Learn.*, vol. 46, pp. 161–190, 2002.

[16] B. Schölkopf, C. Burgess, and V. Vapnik, "Incorporating invariances in support vector learning machines," *Int'l Conf. Neural Networks*, pp. 47–52, 1996.

[17] B. Haasdonk and H. Burkhardt, "Invariant kernel functions for pattern analysis and machine learning," *Machine Learning*, vol. 68, pp. 35–61, 2007.

[18] J. N. Kapur, *Maximum-entropy models in science and engineering*, Wiley Eastern Limited, 1993.

[19] R. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[20] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, 2004.

[21] S. Srivastava, M. R. Gupta, and B. A. Frigyik, "Bayesian quadratic discriminant analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1277–1305, 2007.

[22] E. K. Garcia, S. Feldman, M. R. Gupta, and S. Srivastava, "Completely lazy learning," *IEEE Trans. Knowledge Data Engineering*, vol. 22, no. 9, pp. 1274–1285, 2010.

[23] R. P. Goddard, "The Sonar Simulation Toolset, Release 4.6: Science, Mathematics, and Algorithms," Tech. Rep. A352884, University of Washington Applied Physics Lab, 2008.

[24] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[25] F. D. Neeser and J. L. Massey, "Proper complex random processes with applications to information theory," *IEEE Trans. Info. Theory*, vol. 39, no. 4, July 1993.

[26] L. Isserlis, "On a formula for the product-mean coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, vol. 12, pp. 134–139, 1918.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

Hyrum S. Anderson is a senior member of technical staff at Sandia National Laboratories. He completed his PhD in Electrical Engineering at University of Washington in 2010, where he studied statistical signal processing and machine learning. He earned his BS EE and MS EE from Brigham Young University in 2003. He has also worked for Microsoft Research, MIT Lincoln Laboratory, and the green-tech start-up Acclima.

Maya R. Gupta is an Associate Professor in the Department of Electrical Engineering at the University of Washington. She completed the Ph.D. in EE at Stanford University in 2003, and the BS EE and BA Economics at Rice University in 1997. Her honors include the 2007 PECASE and 2007 ONR YIP awards. She has also worked for Ricoh, HP, Microsoft, AT&T, NATO, and founded and runs Artifact Puzzles.

Eric Swanson received his BS in Electrical Engineering from Oregon State University in 2008, and is a MS EE student at the University of Washington. He has worked for Datalogic Scanning Inc. and Artex Aircraft Supplies Inc.

Kevin Jamieson is a PhD student in Electrical and Computer Engineering at the University of Wisconsin, Madison. Kevin received his BS EE from the University of Washington in 2009 and his MS EE from Columbia University in 2010.